

# **Oracle Utilities Testing Accelerator**

Release Notes

Release 7.0.0.0

**F59685-01**

August 2022

Oracle Utilities Testing Accelerator Release Notes, Release 7.0.0.0

F59685-01

Copyright © 2019, 2022 Oracle and/or its affiliates.

---

---

# Contents

<b>Preface</b> .....	<b>i</b>
Abbreviations .....	ii
Related Documents .....	ii
Updates to the Documentation .....	ii
Documentation Accessibility .....	iii
Conventions.....	iii
<b>Chapter 1</b>	
<b>Release Notes</b> .....	<b>1-1</b>
Supported Oracle Utilities Applications.....	1-2
System Requirements .....	1-2
New Features.....	1-2
Test Planning Feature .....	1-3
Moving Data Between Components without Using Global Variables .....	1-3
Enhanced Test Data GUI.....	1-5
Java Script Support For Custom Functions.....	1-5
Redwood GUI .....	1-6
Dashboard Refresh .....	1-6
Deprecated Features.....	1-7
Planned Deprecation .....	1-7
Known Issues .....	1-7

---

---

# Preface

Welcome to the Oracle Utilities Testing Accelerator Release Notes.

The Release Notes describes the new features, enhancements, known issues, and other changes in Oracle Utilities Testing Accelerator V7.0.0.0.

The preface provides the following information:

- [Audience](#)
- [Abbreviations](#)
- [Related Documents](#)
- [Updates to the Documentation](#)
- [Documentation Accessibility](#)
- [Conventions](#)

## Audience

This document is intended for anyone installing or using Oracle Utilities Testing Accelerator V7.0.0.0.

## Abbreviations

The following terms are used in this document:

Term	Expanded Form
OUTA/UTA	Oracle Utilities Testing Accelerator

## Related Documents

For more information, refer to the following Oracle resources.

### Release Notes

- Oracle Utilities Testing Accelerator Release Notes

### Installation and Administration Guide

- Oracle Utilities Testing Accelerator Installation and Administration Guide

### User and Reference Guides

- Oracle Utilities Testing Accelerator Security Guide
- Oracle Utilities Testing Accelerator User's Guide
- Oracle Utilities Testing Accelerator Upgrade Guide
- Oracle Utilities Testing Accelerator Licensing Information User Manual

## Updates to the Documentation

The complete Oracle Utilities Testing Accelerator documentation set is available from Oracle Help Center at <https://docs.oracle.com/en/industries/utilities/index.html>.

Visit [My Oracle Support](#) for additional and updated information about the product.

---

# Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the [Oracle's Accessibility Program](#) website.

## Access to Oracle Support

Oracle customers have access to electronic support through [My Oracle Support](#). If you are hearing impaired, visit the [Oracle Accessibility Learning and Support](#) website for more information.

## Conventions

The following text conventions are used in this document:

Notation	Indicates
<b>boldface</b>	Graphical user interface elements associated with an action, terms defined in text, or terms defines in the glossary
<i>italic</i>	Book titles, emphasis, or placeholder variables for which you supply particular values
monospace	Commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter

# Chapter 1

---

## Release Notes

This release notes outlines general information about Oracle Utilities Testing Accelerator V7.0.0.0, including new functionality, known issues, and other important aspects of the product.

Prior to installing this release, it is important that you review all of the release notes and your business processes as they relate to the new enhancements. There may be steps to complete or information to review before accepting the release.

For information about supported platforms and installation, refer to the *Oracle Utilities Testing Accelerator Installation and Administration Guide* included in this release.

The document includes the following:

- [Release Overview](#)
- [Supported Oracle Utilities Applications](#)
- [System Requirements](#)
- [New Features](#)
- [Deprecated Features](#)
- [Known Issues](#)

---

# Release Overview

Oracle Utilities Testing Accelerator comprises test automation accelerators for automated testing of the Oracle Utilities applications.

## Supported Oracle Utilities Applications

Oracle Utilities Testing Accelerator 7.0.0.0 release supports respective versions that each of the below listed Oracle Utilities product currently supports:

- Oracle Utilities Customer Care and Billing
- Oracle Utilities Customer to Meter
- Oracle Utilities Work and Asset Management
- Oracle Utilities Operational Device Management
- Oracle Utilities Meter Data Management
- Oracle Utilities Smart Grid Gateway

Refer to the *Certification Matrix for Oracle Utilities Products (Document ID 1454143.1)* on [My Oracle Support](#) to determine if support for newer versions of the listed products have been added.

## System Requirements

For an updated list of system requirements and supported platforms, refer to the **System Requirements** section in *Oracle Utilities Testing Accelerator Installation and Administration Guide* included in this release. The documentation library is available on Oracle Help Center at: <https://docs.oracle.com/en/industries/utilities/index.html>

## New Features

This section focuses on the new features implemented in this release. While these notes provide high-level information, refer to the *Oracle Utilities Testing Accelerator Installation and Administration Guide* and *Oracle Utilities Testing Accelerator User's Guide* for comprehensive guidelines.

- [Test Planning Feature](#)
- [Moving Data Between Components without Using Global Variables](#)
- [Enhanced Test Data GUI](#)
- [Java Script Support For Custom Functions](#)
- [Redwood GUI](#)
- [Dashboard Refresh](#)



## Test Planning Feature

Test management is crucial to the success of any project. The test planning feature provides basic objects required to organize the automated tests as per your test plans and run them. Instead of running each flow/flow set and consolidating the results manually, this feature allows you to run a set of flow sets that belong to a specific test plan. A simple example would be a test plan for each iteration of regression/acceptance testing that is carried out as part of upgrade testing. Test plan provides a high-level view of the overall status of the automated test flows in the context of the health of the application being tested.

Basic test planning capability is now supported in Oracle Utilities Testing Accelerator which allows optimized organization of test flows through test sets, to better suite the testing life cycle of your project. The test plan object encompasses the flow set object. One or more flow sets can be grouped together to form a single test plan. The test plan can be executed as a whole, which provides consolidated run summary report of all the tests within the test plan.

A test plan is defined using flow sets, which in turn hold all the requisite flows. Run the specified set of flows with the prescribed environment using the credentials, on demand. Executing the tests to meet a given objective may take multiple iterations.

Example:

- A test plan is created to meet the objective of being able to successfully generate bills for any given customer.
- The test plan includes some automated test flows that verify the objective.
- Users can run this test plan 'n' number of times (during sprints in agile), logging bugs/issues during each iterative execution of the test plan.

A consolidated summary report is shown that provides an overview of the status of test plan run. This would be a direct indication of the status of the application being tested with respect to the test plan.

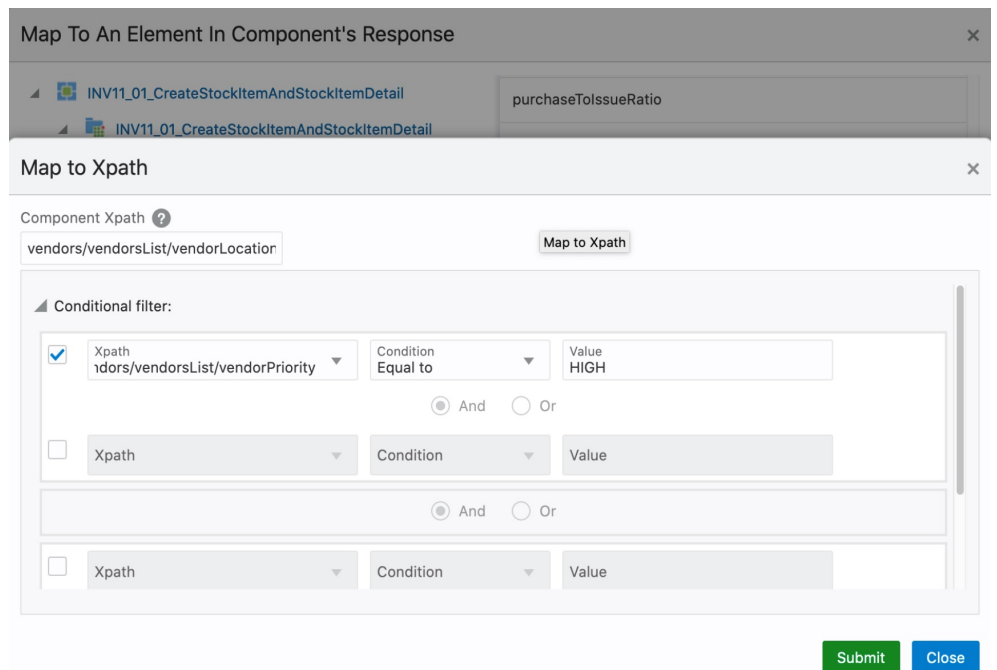
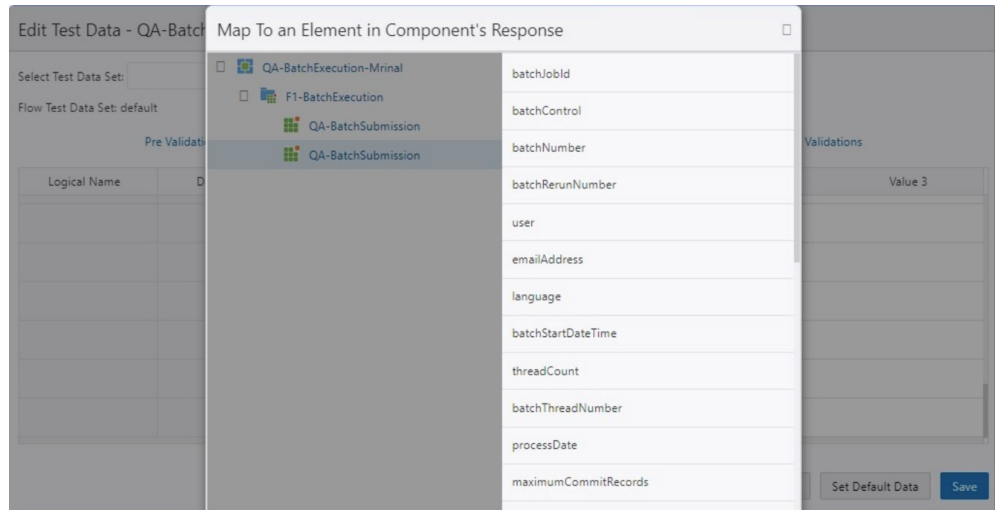
## Moving Data Between Components without Using Global Variables

Often, there is a need to move data from the response that a test component receives to another component's input data fields. This current method of capturing required values into variables, and then using these variables as inputs to another component's test data is conducive when the number of data elements is less. In cases where the number of elements to be passed between components is more, this feature provides an easier and quicker way to do the same without the need to declare multiple variables, thereby reducing the flow development time and effort.

To move test data from the response of one component to the input field of an another component, a new intuitive GUI has been provided. Through the GUI, you can directly select and map the component data fields so that the corresponding values are mapped. To invoke the GUI, Click **Search** next to the input test data fields on the Flow Test Data screen.

This feature reduces, or in some cases, removes the need to explicitly capture the response elements into global variables to be passed on to subsequent components in the flow. Flow development process benefits in the way to reduce steps and effort to map the test data elements.

With the newer ability to link test data based on conditions across components in the flow, by directly using the GUI instead of functions, you can now have faster flow development times and also reduce the upfront design time that involves planning out which test data elements need to be exposed through functions from a component. This can now be done on the fly, during the flow development.



## Enhanced Test Data GUI

Building test data is an important aspect of testing. Improving the test data experience yields greater reduction in costs and time to create effective tests. The test data user interface has been enhanced to reduce time to enter test data in a more logical way. Additionally, relationships expressing sharing of test data across components within a flow have been greatly improved. Definition of test data shared between components based on conditions can be configured using a simpler interface, rather than the raw metadata/functions, thereby making the test development process more streamlined and easier to understand. These enhancements are designed to reduce the test automation design and development times, so automation of more tests can be achieved in a shorter time span.

Test data user interface holds the test data for test automation flow and users spend significant amount of time on this screen, adding test data to a flow. New updates to the test data screen increase the overall usability, thereby reducing the test automation development time. The pre-validations, post-validations and the test data user interfaces have been enhanced to be more intuitive.

The screenshot displays the 'Test Data' tab in the GUI. It features three main sections: 'Pre Validations', 'Test Data', and 'Post Validations'. Under 'Pre Validations', there are three dropdown menus: 'Log Message' (set to 'Start'), 'Web Service Name' (set to 'ATW1SIDInventory'), and 'Web Transaction Type' (set to 'ADD'). The 'Test Data' section contains a 'Request XML' table with columns for 'Node Name', 'Value', and 'Action'. The table lists five nodes: 'stockItemDetailId', 'boStatus', 'statusReason', 'resourceTypeId', and 'nodeId'. The 'resourceTypeId' node has a value of 'gVar\_StockItemId1'. A tooltip 'Edit Test Data - W1-SIDInventory' is visible over the 'Action' column. At the bottom right, there are 'Close' and 'Save' buttons.

Node Name	Value	Action
stockItemDetailId		
boStatus		
statusReason		
resourceTypeId	gVar_StockItemId1	
nodeId	subNodeIdIn1	

## Java Script Support For Custom Functions

Custom functions can be used to implement custom test data generation or verification routines that are not available in the base set of functions. Java Script support ensures that you can build these custom functions using Java Script programming language that is widely known and used.

There may be scenarios in test flow development, where the test data is a result of some custom calculation or formatting of the web service response needs to be validated for specific expressions, which may not be available in the base set of libraries delivered with Oracle Utilities Testing Accelerator. To accommodate for such custom routines/processes, Oracle Utilities Testing Accelerator supports the creation of custom function libraries. Support for Java Script programming language has been added to allow you to create these custom function libraries. You can write your custom routine inside a custom function and use the function as a plugin into the component steps in a flow. A rich editor has been provided to allow the development of custom functions using Java Script.

New custom function libraries can only be created using JavaScript. Creation of new Groovy based custom function libraries is no longer supported, but your existing Groovy libraries will continue to work.

```

Edit Library ⓘ
NEWJSTESTTWO.js
1 var LinkedHashSet = Java.type('java.util.LinkedHashSet');
2 var OUTSCORELIB = Java.type('com.oracle.utilities.core.lib.OUTSCORELIB');
3
4 function testFunction(){
5     var linkedHS = new LinkedHashSet();
6     linkedHS.add("india".toUpperCase());
7     linkedHS.add("IPL".toUpperCase());
8     print(linkedHS);
9 }
10
11 function ouafFunction(s1,s2){
12     print("JS FUNCTION START")
13     var OUTSCORELIB_OBJECT = new OUTSCORELIB();
14     var res = OUTSCORELIB_OBJECT.compare2Strings(s1,s2);
15     print("JS FUNCTION END")
16     if(res == 'true'){
17         return 'F1-SYNRQ'
18     }else{
19         return 'FUNCTION-FAILED'
20     }
21 }

```

## Redwood GUI

Oracle Utilities Testing Accelerator now aligns with Oracle's universal design language, making it much more efficient to use, with richer user experience. The design flows across our application suite making it intuitive for users to work across multiple applications, seamlessly.

Oracle Utilities Testing Accelerator has been refreshed to align with Oracle's Redwood user experience. The user interface of the application has been updated to the latest Redwood specifications, providing refreshing new visuals with more intuitive user experience. The design language provides a cohesive and coherent experience across Oracle Utilities cloud services.

## Dashboard Refresh

Dashboards in Oracle Utilities Testing Accelerator now provides dynamic filters to view visualized reports and analyze the test run results giving holistic view of the automated testing status, pertaining to the corresponding Oracle Utilities cloud services. These dynamic reports help in determining the current state of health of the application being tested, while ensuring better planning and management of the test runs. The reports can be used to view the overall trend of flow runs, giving you the ability to ascertain application health, over time.

Oracle Utilities Testing Accelerator's dashboard has been enhanced to support dynamic filters that allow to create and view customizable reports to analyze the test automation run status. The filters support time based view of test runs that can be categorized into their corresponding product or flow modules, which can provide insights into determining the health of specific functional areas in the application. The reports support drill-down feature that displays the individual data points that contributed to the visualization being displayed, giving the flexibility to perform root cause analysis, directly through the visualization. The visualized reports can be used for sharing the status of the

---

test automation runs with concerned stakeholders of Oracle Utilities cloud services to aid in their decision making.

## Deprecated Features

The Groovy script based function library creation feature has been replaced with support for JavaScript based function library creation. New custom function libraries can only be created using JavaScript. Creation of new Groovy based custom function libraries is no longer supported, but your existing Groovy libraries will continue to work in this 7.0.0.0 release.

## Planned Deprecation

- Groovy language support for custom function libraries will be completely removed from the next release. Any custom function libraries in Groovy language will have to be re-written using Java Script language.
- Oracle Utilities Testing Accelerator Eclipse plugin for running Oracle Utilities Testing Accelerator flows on a local machine. Oracle Utilities Testing Accelerator Eclipse plugin will be deprecated in the next Oracle Utilities Testing Accelerator release. This deprecation in turn removes any support for GUI based test flows and components.

## Known Issues

This section lists the issues that were found but not resolved in Oracle Utilities Testing Accelerator at the time of release.

- **Edit Test Data** GUI does not show the **Remove List** button

In the **Flow > Component Test Data** GUI, after we add a new repetition of a list to add test data, the option to remove such a list repetition is not available through a button. The work around is to ensure that the list repetition values are blank. Leaving the values blank automatically removes the list repetition on saving the test data.

- **Creating a duplicate flow set throws 400 error**

When a user tries to create a flow set with the same name as the one that's already existing, then Oracle Utilities Testing Accelerator displays a 400 error, instead of showing an error dialogue with appropriate message informing user about the duplicate record error.