Oracle Functional Testing Advanced Pack for Oracle Utilities
Reference Guide for Oracle Utilities Application Framework
Release 5.0.0
E67844-01

October 2015
Preface

This guide describes the Oracle Utilities Application Framework (OUAF) components and the function libraries used to create those components for Oracle Functional Testing Advanced Pack for Oracle Utilities (OFTAPOU) v5.0.0. These components are used to build test flows in Oracle Flow Builder (OFB).

This preface includes the following sections:

• Audience
• Related Documents
• Notational Conventions

Audience

This guide is intended for QA/Test Engineers and Automation Developers to understand the various components and libraries available for them to automate the business test flows for Oracle Utilities Application Framework using Oracle Functional Testing Advanced Pack for Oracle Utilities (OFTAPOU) for Oracle Utilities Application Framework.

Related Documents

For more information, see the following documents:

• Oracle Functional Testing Advanced Pack for Oracle Utilities Release Notes
• Oracle Functional Testing Advanced Pack for Oracle Utilities Installation and Administration Guide

See also:

• Oracle Utilities Application Framework Documentation Library
Notational Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
This chapter lists the Oracle Utilities Application Framework starter components available to create flows in Oracle Flow Builder.

The chapter includes the following sections:

- Overview
- Components

Overview


Oracle Functional Testing Advanced Pack for Oracle Utilities for Oracle Utilities Application Framework contains out-of-the-box product-specific components used to build new test flows in Oracle Flow Builder to test the Oracle Utilities Application Framework based applications. These out-of-the-box components correspond to specific business entities, such as business objects, service scripts, or business services used for interfacing with the application. Users can use these components as available or can extend them. Users can also create new components to be used to create flows. This starter pack also contains a set of function libraries that can be used for creating custom components.

**Note:** See Chapter 2: Function Library Reference for detailed information about using these function libraries.

Consider this pack to be a starter kit which can be expanded and built upon. A few sample flows are included as an example.

**Note:** See the Oracle Functional Testing Advanced Pack for Oracle Utilities User's Guide for information about creating components and flows.

The components are categorized under the following functional areas:

- Admin
- Batch
- ToDo
- AdminUI
- ToDo UI
Components

The following table lists the starter components available in Oracle Utilities Application Framework.

**Pre-requisites:** The Inbound Web Service using the respective business object should be available in the application.

**Additional Notes:** Failure while creating, reading, or updating the component is logged in the test execution report, thus facilitating debugging/analysis of the problems.

### Inbound Web Service Components

The following table lists out the Inbound Web Service based components.

<table>
<thead>
<tr>
<th>Component</th>
<th>Functional Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1-Algorithm</td>
<td>Admin</td>
<td>Used to create, read, update, and delete Algorithm via a Web service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>After creation, the CRUD operations can be performed through these components against the F1-AlgorithmPhysicalBO business object.</td>
</tr>
<tr>
<td>F1-BatchControl</td>
<td>Admin</td>
<td>Used to create, read, or update a ‘Batch Control’ via a Web service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>After creation, the CRUD operations can be performed through these components against the F1-BatchContolMO business object.</td>
</tr>
<tr>
<td>F1-Country</td>
<td>Admin</td>
<td>Used to create, read, or update a ‘Country’ via a Web service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>After creation, the CRUD operations can be performed through these components against the F1-COUNTRY business object.</td>
</tr>
<tr>
<td>F1-DisplayProfile</td>
<td>Admin</td>
<td>Used to create, read, or update a 'DisplayProfile via a Web service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>After creation, the CRUD operations can be performed through these components against the F1-DisplayProfilePhysicalBO business object.</td>
</tr>
<tr>
<td>Component</td>
<td>Functional Area</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>F1-ExternalSystem</td>
<td>Admin</td>
<td>Used to create, read, or update an 'ExternalSystem' via a Web service. After creation, the CRUD operations can be performed through these components against the F1-ExternalSystemPhysicalBO business object.</td>
</tr>
<tr>
<td>F1-FeatureConfig</td>
<td>Admin</td>
<td>Used to create, read, or update a 'FeatureConfig' via a Web service. After creation, the CRUD operations can be performed through these components against the F1-FeatureConfigPhysicalBO business object.</td>
</tr>
<tr>
<td>F1-TimeZone</td>
<td>Admin</td>
<td>Used to create, read, or update a 'TimeZone' via a Web service. After creation, the CRUD operations can be performed through these components against the F1-TimeZonePhysicalBO business object.</td>
</tr>
<tr>
<td>F1-User</td>
<td>Admin</td>
<td>Used to create, read, or update a 'User' via a Web service. After creation, the CRUD operations can be performed through these components against the F1-UserPhysicalBO business object.</td>
</tr>
<tr>
<td>F1-UserGroup</td>
<td>Admin</td>
<td>Used to create, read, or update a 'UserGroup' via a Web service. After creation, the CRUD operations can be performed through these components against the F1-UserGroupPhysicalBO business object.</td>
</tr>
<tr>
<td>F1-WorkCalendar</td>
<td>Admin</td>
<td>Used to create, read, or update a 'WorkCalendar' via a Web service. After the creation, the CRUD operations can be performed through these components against the F1-WorkCalendarPhysicalBO business object.</td>
</tr>
<tr>
<td>Component</td>
<td>Functional Area</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>F1-ToDoRole</td>
<td>ToDo</td>
<td>Used to create, read, or update a ‘ToDoRole’ via a Web service. After creation, the CRUD operations can be performed through these components against the F1-ToDoRolePhysical business object.</td>
</tr>
<tr>
<td>F1-ToDoType</td>
<td>ToDo</td>
<td>Used to create, read, or update a ‘ToDoType’ via a Web service. After creation, the CRUD operations can be performed through these components against the F1-ToDoTypePhysicalBO business object.</td>
</tr>
<tr>
<td>F1-BatchSubmission</td>
<td>Batch</td>
<td>Used to run or retrieve the Batch submission status via a Web service. After creation, execute the batch job and retrieve the batch job running status against the F1-BatchJob business object.</td>
</tr>
<tr>
<td>F1-ToDoEntryAdd</td>
<td>ToDo</td>
<td>Used to create a To Do Entry via a Web service. After creation, add To Do entries through these component against the F1-AddToDoEntry business service.</td>
</tr>
<tr>
<td>F1-ToDoEntryRead</td>
<td>ToDo</td>
<td>Used to retrieve a To Do Entry via a Web service. After creation, retrieve the To Do Entry through this component against the F1-MaintainToDoEntry business service.</td>
</tr>
<tr>
<td>F1-ToDoEntryUpdate</td>
<td>ToDo</td>
<td>Used to update a To Do Entry via a Web service. After creation, retrieve the To Do Entry through this component against the F1-UpdateToDoEntry business service.</td>
</tr>
</tbody>
</table>
**Component Functional Area Description**

<table>
<thead>
<tr>
<th>Component</th>
<th>Functional Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1-ToDoEntryComplete</td>
<td>ToDo</td>
<td>Used to complete a To Do Entry via a Web service. After creation, complete the To Do Entry through this component against the F1-CompleteToDoEntry business service.</td>
</tr>
<tr>
<td>F1-SendResultsMail</td>
<td>Admin</td>
<td>Parses the results and generates the email. This component sends the test executions results to the configured email ID.</td>
</tr>
</tbody>
</table>

**UI Components**

The following table lists out the UI based components. These components only work for the Oracle Utilities’ products which are based on Oracle Utilities Application Framework (OUAF) v4.3 SP1.

<table>
<thead>
<tr>
<th>Component</th>
<th>Functional Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1-Login</td>
<td>Admin</td>
<td>Used to login into Oracle Utilities Application Framework through UI.</td>
</tr>
<tr>
<td>F1-LoginSsl</td>
<td>Admin</td>
<td>Used to login into Oracle Utilities Application Framework, ignoring the SSL certificate error.</td>
</tr>
<tr>
<td>F1-Logout</td>
<td>Admin</td>
<td>Used to logout from Oracle Utilities Application Framework on UI.</td>
</tr>
<tr>
<td>F1-ToDoRoleAdd</td>
<td>ToDo</td>
<td>Used to create the ToDo Role from UI. After creating, this component can be used to create To Do types and used for CRUD operations.</td>
</tr>
<tr>
<td>F1-ToDoRoleRead</td>
<td>ToDo</td>
<td>Used to read the To Do Role from UI.</td>
</tr>
<tr>
<td>F1-ToDoRoleUpdate</td>
<td>ToDo</td>
<td>Used to update the To Do Role from UI.</td>
</tr>
<tr>
<td>F1-ToDoRoleDelete</td>
<td>ToDo</td>
<td>Used to delete the To Do Role object from Oracle Utilities Application Framework application.</td>
</tr>
<tr>
<td>Component</td>
<td>Functional Area</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>F1-ToDoTypeAdd</td>
<td>ToDo</td>
<td>Used to create the ToDo Type through UI in Oracle Utilities Application Framework application.</td>
</tr>
<tr>
<td>F1-ToDoTypeRead</td>
<td>ToDo</td>
<td>Used to read the To Do Type existing in the application.</td>
</tr>
<tr>
<td>F1-ToDoTypeUpdate</td>
<td>ToDo</td>
<td>Used to update the To Do Type existing in the application.</td>
</tr>
<tr>
<td>F1-ToDoTypeDelete</td>
<td>ToDo</td>
<td>Used to delete the To Do Type existing in the application.</td>
</tr>
<tr>
<td>F1-ToDoEntryAdd</td>
<td>ToDo</td>
<td>Used to create an To Do Entry for the specific To Do Type.</td>
</tr>
<tr>
<td>F1-ToDoEntryRead</td>
<td>ToDo</td>
<td>Used to retrieve the To Do entry existing in Oracle Utilities Application Framework, using the ToDo ID.</td>
</tr>
<tr>
<td>F1-ToDoEntryAssign</td>
<td>ToDo</td>
<td>Used to read the ToDo entry and assign the To Do Entry to user.</td>
</tr>
<tr>
<td>F1-ToDoEntryComplete</td>
<td>ToDo</td>
<td>Used to read the To Do Entry and complete it.</td>
</tr>
</tbody>
</table>
This chapter lists the Oracle Utilities Application Framework library OUAFLIB and functions available to create components and flows in Oracle Flow Builder to test Oracle Utilities Application Framework.

**OUAFLIB**

The OUAFLIB library comprises functions that work on the UI components. These functions are typically used to click the text area before setting values, and also Alert and Confirm windows. This section provides a list of functions included in the library, along with their usage details.

**confirmButton**

Clicks OK on the Confirm window displayed while deleting an object.

Example:

```java
confirmButton();
```

Input Parameters: NULL
Return Type: Boolean

**alertButton**

Identifies the Alert window and clicks OK. Use this function in negative test scenarios where an Alert window is displayed.

Example:

```java
alertButton();
```

Input Parameters: NULL
Return Type: Boolean

**ToDoEntryAdd_textAreaClick**

Clicks the text area called before setting the value in that text area. This function is used while creating a To Do Entry.

Example:

```java
toDoEntryAdd_textAreaClick();
```

Input Parameters: NULL
Return Type: Boolean
**toDoEntryAssign_textAreaClick()**

Clicks the text area to be called before setting the value in that text area. This function is used while assigning a To Do Entry.

Example:

```java
toDoEntryAssign_textAreaClick()
```

Input Parameters: NULL
Return Type: Boolean

**getToDoId()**

Gets the To Do ID from the To Do Entry created. This To Do ID is used further to assign or complete the ToDo Entry.

Example:

```java
getToDoId()
```

Input Parameters: NULL
Return Type: String
Chapter 3
Sample Work Flows

This chapter describes the Oracle Utilities Application Framework sample flows that illustrate common use cases for Oracle Functional Testing Advanced Pack for Oracle Utilities. It also explains the procedure to execute these sample flows.

The chapter includes the following sections:

- Sample Flows
- Executing Sample Flows

Sample Flows

The sample flows delivered as part of Oracle Functional Testing Advanced Pack for Oracle Utilities for Oracle Utilities Application Framework demonstrate how flows can be created for Web services based testing and for a combination of Web services and UI based testing using the same framework.

These flows are designed to run using the demo data, giving the user the ability to deploy Oracle Functional Testing Advanced Pack for Oracle Utilities for Oracle Utilities Application Framework and execute the sanity flows immediately. The flows perform a part of the basic sanity testing required to certify that the Oracle Utilities Application Framework environment has been setup appropriately.

This section includes the following sample work flows for both UI and Web Services:

- F1-ToDoFlow
- ToDoRoleFlow
- ToDoFlow

F1-ToDoFlow

F1-ToDoFlow is a Web service flow comprising the creation for To Do Role, To Do Type, To Do Entry, Assign To Do Entry, and Complete To Do Entry, and also the completion life cycle of an To Do.

The following table lists the tasks that are created and their respective components in Oracle Utilities Application Framework.

<table>
<thead>
<tr>
<th>Task</th>
<th>OUAF Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Do Role</td>
<td>F1-ToDoRole</td>
</tr>
<tr>
<td>To Do Type</td>
<td>F1-ToDoType</td>
</tr>
</tbody>
</table>
Sample Flows

The F1-ToDoFlow flow includes the complete ToDo flow:
1. Creates ToDoRole.
2. Creates ToDoType using the ToDoRole already created.
3. Adds ToDo Entry.
4. Reads the created ToDoEntry.
5. Updates the ToDoEntry entry status and verifies it.
6. Completes the ToDo Entry.
7. Sends the result email.

ToDoRoleFlow

ToDoRoleFlow is a UI flow that includes the CRUD operations of ToDo Role.

The following table lists the tasks that are created and their respective components in Oracle Utilities Application Framework.

<table>
<thead>
<tr>
<th>Task</th>
<th>OUAF Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToDo Role Add</td>
<td>F1-ToDoRoleAdd</td>
</tr>
<tr>
<td>ToDo Role Read</td>
<td>F1-ToDoRoleRead</td>
</tr>
<tr>
<td>ToDo Role Update</td>
<td>F1-ToDoRoleUpdate</td>
</tr>
<tr>
<td>ToDo Role Delete</td>
<td>F1-ToDoRoleDelete</td>
</tr>
</tbody>
</table>

The ToDoRoleFlow flow includes the complete ToDo flow:
1. Logs in and then creates a ToDoRole.
2. Retrieves the existing ToDoRole.
3. Updates ToDoRole.
4. Deletes ToDoRole.
**ToDoFlow**

ToDoFlow includes the creation of ToDo Role, ToDo Type, ToDo Entry, and also to complete the ToDo Entry.

The following table lists the tasks that are created and their respective components in Oracle Utilities Application Framework.

<table>
<thead>
<tr>
<th>Task</th>
<th>OUAF Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Do Role Add</td>
<td>F1-ToDoRoleAdd</td>
</tr>
<tr>
<td>To Do Type Read</td>
<td>F1-ToDoTypeAdd</td>
</tr>
<tr>
<td>To Do Entry Add</td>
<td>F1-ToDoEntryAdd</td>
</tr>
<tr>
<td>To Do Entry Complete</td>
<td>F1-ToDoEntryComplete</td>
</tr>
</tbody>
</table>

The ToDoRoleFlow flow includes the complete ToDo flow:

1. Logs in and then creates a ToDoRole.
2. Creates ToDoType.
3. Creates a ToDoEntry for the ToDoType already created.
4. Completes the ToDoEntry.

**Executing Sample Flows**

This section describes the procedure to setup sample flows and execute them.

- Pre-requisites
- Setting Up Inbound Web Service Sample Flows
- Setting Up UI Sample Flows

**Pre-requisites**

To execute a sample flow, ensure the following pre-requisites are met:

- Oracle Utilities Application Framework is up and running.
- OpenScript is installed in the local machine. See the *Oracle Functional Testing Advanced Pack for Oracle Utilities Installation and Administration Guide* for the version details.
- Oracle Functional Testing Advanced Pack for Oracle Utilities is installed and repository/directory is setup in the local machine appropriately. See the *Oracle Functional Testing Advanced Pack for Oracle Utilities Installation and Administration Guide* for more details.

**Setting Up Inbound Web Service Sample Flows**

To setup an Inbound Web Service based sample flow, follow these steps:

1. Login to Oracle Utilities Application Framework.
2. Import the Inbound Web services into the Oracle Utilities Application Framework where the scenarios need to be executed.

   See the *Importing Inbound Web Services* section in the *Oracle Functional Testing Advanced Pack for Oracle Utilities User’s Guide* for steps to import the Inbound Web services.

3. Navigate to **Admin > B > Bundle Import > Add**.
4. Enter the **External Reference, Detailed Description, and Bundle Details** from the IWS Bundle Export Dump.

5. Click **Save**, and then click **Apply bundle**.

6. Launch OpenScript in the local machine and perform the following steps:
   a. Navigate to **View > OpenScript Preferences**.
   b. In the left tree, select **OpenScript**. In the sub tree, select **Playback**, and then select **Error Recovery**.
   c. Click **SetAll** and select **Report Error and Continue**.
   d. Click **Apply**, and then click **Close**.

7. Configure the **configuration.properties** file as follows:
   a. Provide the application URL for the parameter:
      
      ```
gStrApplicationURL = http:\/\/ <%serverName%>\:<%portNumber%>/ ouaf
      ```
   b. Provide the additional path required for Inbound Web service URL:
      
      ```
gStrApplicationXAIServerPath=/<%webservices/<%AppendThisToAbove gStrApplicationURL%>/
      ```
   c. Provide an environment name for display in the results email:
      
      ```
gStrEnvironmentName= <%testEnvironmentName%>
      ```
   d. Provide the application login user ID:
      
      ```
gStrApplicationUserName= <%UserName%>
      ```
   e. Provide the application login password:
      
      ```
gStrApplicationUserPassword= <%password%>
      ```
   f. Provide the SMTP email server and e-mail ID:
      
      ```
gStrSMTP_HOST_NAME=<%SMTP ServerName%>
gStrSMTP_PORT=<%PortNumber%>
gStrTO_EMAIL_RECIPIENTS=<%e-mail Id%
      ```
   g. Provide the application database details as below:
      
      ```
gStrApplicationDBConnectionString =<%jdbc Connectionstring%>
eg: jdbc\:oracle\:thin:\@<%DBserverName%>:\:<%port%>:\:<%DBSID%>
gStrApplicationDBUsername=<%DBUserID%>
gStrApplicationDBPassword=<%DBPassword%>
      ```
   h. Provide the full directory path of Oracle Application Testing Suite repository directories in the local machine.
      
      ```
gStrOutputFilePath=<%LogFilePath%>
Example: C:\OUAF_DEMO\OUTSP\Logs\
gStrXSDFiles=<%XSD Folder path%>
Example: C:\OUAF_DEMO\OUTSP\Logs\
      ```

9. Create a folder (Core) in the outsp-function-libs folder in the Oracle Functional Testing Advanced Pack for Oracle Utilities repository directory. Copy the function libraries to the respective folders.
   - Core
     - OUTSPCORELIB
• WSCOMMONLIB
• WSVALIDATELIB

10. Copy all the .jar files provided in the installer into the **genericJars** folder in the Oracle Functional Testing Advanced Pack for Oracle Utilities repository directory.

### Setting Up UI Sample Flows

To setup a UI based sample flow, follow these steps:

1. Create a folder ‘Core’ in the `outsp-function-libs` folder in the Oracle Functional Testing Advanced Pack for Oracle Utilities repository directory. Copy the function libraries to the respective folders.
   - Core
     - OUAFULIB

2. Copy all the .jar files provided in the installer into the **genericJars** folder in the Oracle Functional Testing Advanced Pack for Oracle Utilities repository directory.

3. Ensure the **ebs-function-libs** folder exists in the Oracle Functional Testing Advanced Pack for Oracle Utilities repository directory, and then copy all the latest function libraries.

   **Note:** To execute the UI based sample flows, provide the Application URL, User Name, and Password in the F1-LoginSsl component test data before downloading/generating the Oracle Functional Tester script.
Appendix A

Inbound Web Services

The Oracle Utilities Application Framework components are developed using Web services method, and these components require Inbound Web Services to be defined in the application.

For instructions to create, import, or search an Inbound Web Service, see the Setting Up Inbound Web Services appendix in Oracle Functional Testing Advanced Pack for Oracle Utilities User’s Guide.

List of Inbound Web Services

The list of Inbound Web Services provided to use with the delivered components and flows is as follows:

- ATF1DisplayProfile
- ATF1Algorithm
- ATF1ExternalSystem
- ATF1FeatureConfig
- ATF1BatchContol
- ATF1User
- ATF1WorkCalendar
- ATF1UserGroup
- ATF1TimeZone
- ATF1ToDoRole
- ATF1ToDoType
- ATF1ToDoEntryADD
- ATF1ToDoEntryREAD
- ATF1ToDoEntryUPDATE
- ATF1ToDoEntryComplete
- ATF1Country
- ATF1BatchSubmission