Prepared By: Rajendra Biradar

Version 1.0 | date: 26 oct 2023

iDataSyncr

USER MANUAL

**Table of Contents**

[1. INTRODUCTION 2](#_Toc149257993)

[2. PRE REQUESITS 3](#_Toc149257994)

[3. DETAILS 5](#_Toc149257995)

[3.1 ACTION BUTTONS 5](#_Toc149257996)

[3.2 OBJECT\_REPO 6](#_Toc149257997)

[3.3 FUNC\_MASTER 8](#_Toc149257998)

[3.4 SUBFUNC\_MASTER 9](#_Toc149257999)

[3.5 SUBFUNC\_STEPS 10](#_Toc149258000)

[3.6 FUNC\_SUBFUNC\_MAPPING 13](#_Toc149258001)

[3.7 SCN\_FUNC\_BATCHING 16](#_Toc149258002)

[3.8 API 19](#_Toc149258003)

[3.9 SCN\_STEPS\_GENERATOR 21](#_Toc149258004)

[3.10 RUN\_TIME\_DATA 24](#_Toc149258005)

[3.11 COMMON\_DATA 25](#_Toc149258006)

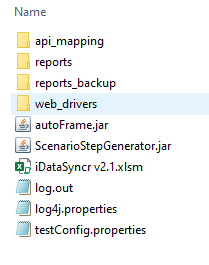
[3.12 RUNCHART 26](#_Toc149258007)

[3.13 LIST\_ENUMS 27](#_Toc149258008)

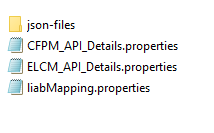
# INTRODUCTION

**iDataSyncr** is Excel based GUI tool designed for interacting with database and perform actions required for maintaining automation data.

Below is the folder content of GUI tool.



1. **iDataSyncr v2.1.xlsm** - This is a macro-enabled Excel workbook that facilitates interaction with the database.
2. **testConfig.properties** - This configuration file contains crucial information, including Automation and ELCM [ connected to CFPM handoff ] database schema details. It also includes settings for controlling execution, such as implicit wait times, page load timeouts, preferred web browser, and the screenshot level.
3. **autoFrame.jar** - Automation Framework code converted to JAR file. This is required for execution.
4. **ScenarioStepGenerator.Jar** - Code to generate the scenario steps.
5. **Api\_mapping** - This folder is holds configuration required for ELCM handoff API and data match.



* 1. **CFPM\_API\_Details.properties** - This file holds CFPM API configuration details like hostname, endpoints, header details etc.
  2. **ELCM\_API\_Details.properties** - This file holds ELCM API configuration details like hostname, endpoints, header details etc.
  3. **liabMapping.properties** - This file contains mapping of liability details which are verified as a part of ELCM handoff. Similar mapping files will be made available for other ELCM handoff details.
  4. **json-files** - This folder holds CFPM and ELCM API response json files received during ELCM handoff API calls.

1. **reports / reports\_backup** folders -
   1. The "reports" folder contains the most recent execution reports, including an overall dashboard, scenario-specific reports, and screenshots associated with each scenario.
   2. The "reports\_backup" folder preserves the details of the last five reports. The tool automatically moves reports from the "reports" folder to the "reports\_backup" folder during each execution.
2. **Web\_drivers folder** - This folder houses the latest chrome browser driver required for execution. Users must manually download and place the appropriate drivers in this directory.

Refer <https://googlechromelabs.github.io/chrome-for-testing/> to download latest chrome drivers.

1. **log4j.properties / log.out**
   1. log4j.properties - This configuration file specifies the settings required for log generation. Users generally do not need to modify any properties within this file.
   2. log.out - This file is generated when users create scenario steps. It serves as a reference for accessing detailed logs.

# PRE REQUESITS

User needs to install below two software’s before using excel GUI.

1. Oracle Client Database: Use this link to download the same

<https://www.oracle.com/database/technologies/oracle19c-windows-downloads.html>

A screenshot of a computer

Description automatically generated

Please unzip once downloaded.

Kindly follow the training video for complete installation process.

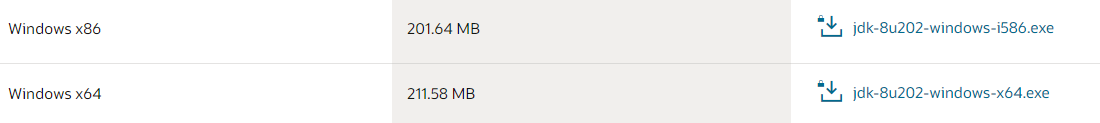
1. Java 1.8

**Installing Java 1.8:**

1. Download Java 1.8:

* Visit below link to download Java 1.8

<https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html>



* You may need to create or log in to an Oracle account to download Java from their website.

1. Accept License Agreement:

* After downloading, run the installer.
* You will be asked to accept the Oracle Binary Code License Agreement.

1. Choose Installation Location:

* Follow the installation wizard to choose the installation location. The default location is usually fine.

1. Complete Installation:

* Continue with the installation process by following the on-screen instructions.

**Setting Up Environment Path:**

1. Find Java Installation Path:

* Locate the folder where Java 1.8 was installed. This is typically in the "Program Files" directory on Windows.
* Set Environment Variable (Windows):
* Right-click on "My Computer" or "This PC" and select "Properties."
* Click on "Advanced system settings" or "Advanced" on the left.
* In the System Properties window, click the "Environment Variables" button.
* Under "System Variables," scroll down and find the "Path" variable. Select it and click "Edit."
* In the "Edit Environment Variable" window, click "New" and add the path to the "bin" directory of your Java installation. For example, it might be "C:\Program Files\Java\jdk1.8.0\_xx\bin" for a 64-bit Windows installation.
* Click "OK" to close the windows.

**Verify Installation:**

* Open a command prompt or terminal window.
* Type java -version and press Enter. You should see information about the installed Java version, indicating that Java 1.8 has been successfully installed.

1. Users need to enable macro in excel. Follow below steps to enable.
2. Open Excel: Launch Microsoft Excel on your computer.
3. Open the Workbook: Open the Excel workbook that contains the macro you want to enable. If you're creating a new macro, you can do that within Excel as well.
4. Access the Trust Center:
   1. In Excel 2010 and later versions, click on the "File" tab in the top left corner.
   2. In older versions of Excel, click on "Tools" and then "Options."
5. Access Excel Options:
   1. In Excel 2010 and later, click on "Options."
   2. In older versions, click on the "Security" tab.
6. Open Trust Center Settings:
   1. In Excel 2010 and later, click on "Trust Center" on the left sidebar and then click "Trust Center Settings" on the right.
   2. In older versions, you should see the "Security Options" button. Click that.
7. Enable Macros: In the Trust Center dialog box, click on "Macro Settings" on the left.
8. Choose Macro Settings: You'll see different options for enabling macros. Select "Enable all macros"
9. Click OK: After selecting the appropriate macro setting, click "OK" in the Trust Center dialog box.
10. Save and Reopen: If you made changes to an existing workbook, save it and then close and reopen it for the changes to take effect. If you're creating a new macro, you can start creating it without any issues.

# DETAILS

Excel GUI will contain below worksheets for each database table.

|  |  |
| --- | --- |
| **Name** | **Description** |
| OBJECT\_REPO | Stores information related to web elements. |
| FUNC\_MASTER | Lists main functions [stages] across CFPM application. |
| SUBFUNC\_MASTER | Lists sub functions as per data segments and actions performed across CFPM application. |
| FUNC\_SUBFUNC\_MAPPING | Used to map main functions to their corresponding sub functions for each scenario. |
| FUNC\_STEPS | Contains the steps or actions to be performed within each sub function. |
| API | Detailed steps to perform ELCM handoff. |
| SCN\_FUNC\_BATCHING | Create stage wise end to end flow for each scenario. |
| SCN\_STEPS\_GENERATOR | To generate scenario steps based on mapping and batching table. Also, to maintain data for each scenario. |
| RUN\_TIME\_DATA | User can access scenario specific data that is retrieved during execution. |
| COMMON\_DATA | COMMON\_DATA holds common data elements like URLs, user IDs, passwords, currencies, and more, ensuring consistency and reusability. |
| RUNCHART | To control the execution. Here user can define which scenarios to be executed, tagging etc. |
| LIST\_ENUMS | This sheet manages lists and enumerations used within the tool, providing a reference for structured data. |

Refer section 3.2 onwards for details about each worksheet.

## ACTION BUTTONS

The following action buttons are utilized throughout the worksheets, and their functionality remains consistent across all worksheets.

|  |  |
| --- | --- |
| **#** | **Description** |
|  | Saves any changes or data input to the worksheet.    User will be prompted if any mandatory data is not entered while adding or editing the data. Same will be highlighted in red color. |
|  | Allows users to add new records to the worksheet. A new row with a green background will be added to distinguish between existing and new entries. Users can add multiple rows by repeating the same process. To add a new row after a specific row in the existing data, first select that row and then click the Add button.  In FUNC\_MASTER and SUBFUNC\_MASTER worksheet, row will be added at end by default. |
| Badge Cross with solid fill | To cancel add action when user no longer wants to add record. This will remove added row. |
| Pencil with solid fill | Allows users to make changes to existing record. The selected row will be enabled for editing and highlighted in yellow background.  The edit icon will be displayed when the user selects a row that contains data. |
| Garbage with solid fill | Removes selected record from the database. Users can delete unwanted or obsolete data. The delete icon will be displayed when the user selects a row that contains data.    Click ‘Yes’ to delete the record.  Click ‘No’ to cancel delete operation. |
|  | Updates the content of the worksheet, ensuring that users are working with the most current data. |
|  | This indicates the status of the database connection. Users can click the button to initiate the connection. Status will be across the worksheets.  Green: The connection has been successfully established, and there are no issues.  Red: The database connection is not established. |

## OBJECT\_REPO

This worksheet stores information related to web elements. Here, users capture the XPath of fields present on specific screens and define the field type, such as button, textbox, dropdown, etc.

Whenever a new screen is added or a new element is introduced on an existing screen, users refer to this OBJECT\_REPO worksheet to document the information.

**Fields**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Mandatory** |
| SCREEN\_NAME | Name of the screen or data segment. This should be unique name.  Non editable. | Yes |
| FIELD\_NAME | The name of the field present in the screen. This should be unique name. | Yes |
| TYPE | This defines the type of the field, such as textbox, dropdown, text, radio button, etc.  Ex:  organization\_name : textbox  currency : dropdown  This is a dropdown field, and the user needs to select from the available options. Dropdown values are derived from the "LIST\_ENUMS" sheet. | Yes |
| LOCATOR | Locators are one of the essential components of automation, which help automation scripts in uniquely identifying the WebElements (such as text box, button, etc.) present of the web page. Users need to capture the xpath from the application. | Yes |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
|  | The 'ADD' button within the OBJECT REPOSITORY section facilitates the capture of element details for a new screen. When the 'ADD' button is clicked, a popup window will appear, prompting the user to enter the name of the screen for which details needs to be captured.   * Screen Name is a mandatory field. * Save - System will remove any data already present and shows screen name in below data label      * Cancel – Closes the popup. |
|  | If a user wishes to view information about an existing screen name, they can click the 'SELECT' button within the OBJECT REPOSITORY section. This action opens a popup where the user can select the desired screen name from a dropdown menu.    The dropdown menu will display screen names based on the text entered, facilitating a smart selection process. |
|  | If user wants to rename the existing screen name in one click, then click ‘EDIT’ button under OBJECT REPOSITORY section.  Below popup will open with existing screen name. User can provide new unique name.    Note: To perform the 'EDIT' action, the user should have already selected a screen name using the 'SELECT' option. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to add, edit, delete and save the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Screen name already exists. Please enter a different screen name. | This message appears when a user enters a screen name that already exists in the database. |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |
| Screen Name and Field Name combination Is already exists!!! Kindly recheck data. | This message appears when a user enters the same field name for a new or selected screen name. Screen name and Field name combination must be unique. |

## FUNC\_MASTER

This worksheet serves as a central repository for listing and managing the main functions or stages across the CFPM application. Users should populate this worksheet when introducing new feature or stage. Each function will be assigned a unique identifier for reference.

**Fields**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Mandatory** |
| FUNC\_ID | System generated unique ID. System will generate next available ID.  Non editable.  Format: “FN\_<number>.  Ex: FN\_001 | Yes |
| FUNC\_NAME | Name of the functionality or stage. | Yes |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
|  | To view the list of existing functions, click the 'VIEW' button. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to add, edit, delete and save the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |
| Record cannot be deleted. please ensure that there are no related records depending on it. You may need to delete child records first. | This message appears when a user attempts to delete a record that is already being used in another table. The system prevents deletion in such cases, as there are dependent records. To delete the record, the user may need to delete the related child records first. |

## SUBFUNC\_MASTER

This worksheet serves as a central repository for listing and managing the sub functions as per data segments and actions performed across CFPM application. Users should populate this worksheet when introducing new data segment or action. Each sub function will be assigned a unique identifier for reference.

**Fields**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Mandatory** |
| SUBFUNC\_ID | System generated unique ID. System will generate next available ID.  Non editable.  Format: “SFN\_<number>.  Ex: SFN\_001 | Yes |
| SUBFUNC\_NAME | Name of the data segment or action. | Yes |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
| A logo of an eye  Description automatically generated | To view the list of existing sub functions, click the 'VIEW' button. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to add, edit, delete and save the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |
| Record cannot be deleted. please ensure that there are no related records depending on it. You may need to delete child records first. | This message appears when a user attempts to delete a record that is already being used in another table. The system prevents deletion in such cases, as there are dependent records. To delete the record, the user may need to delete the related child records first. |

## SUBFUNC\_STEPS

Test steps are specific tasks or actions that testers perform to verify the behavior of a software application. This worksheet helps to capture the same. Test steps are typically organized in a specific sequence to ensure the flow.

Use this worksheet to create steps for all sub functions defined in SUBFUNC\_MASTER.

**Fields**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Mandatory** |
| SUBFUNC\_NAME | This field is automatically filled based on the selected SUBFUNC\_NAME for which you are creating steps. You cannot edit this. | Yes |
| STEP\_ID | Use this column to specify the order of your steps. You can assign a unique number to each step to define the sequence. If you need to insert a step between two existing ones, you can do so in two ways.  Example: Let's say you have three steps for SUBFUNC\_MASTER, and you want to add a step between the second and third steps.  Method 1:   1. Insert a new row after the 2nd step and specify STEP\_ID as 3. 2. Edit the steps that come after the new one (step a), and renumber their STEP\_ID, e.g., 4. Repeat this if you have more steps.   Method 2:   1. Insert a new row after the 2nd step and give it a STEP\_ID of 2.1. | Yes |
| STEP\_DESCRIPTION | This is where you describe the step in detail. | Yes |
| SCREEN\_NAME | Select the screen name from the dropdown menu based on the SUBFUNC\_NAME for which you are creating steps. SCREEN\_NAME options are derived from OBJECT\_REPO. | Yes |
| FIELD\_NAME | The options in the dropdown menu depend on the SCREEN\_NAME you've chosen. Simply pick the FIELD\_NAME that matches the element you're working with in your step. | Yes |
| KEYWORD | A "KEYWORD" is a term used to define the type of action being performed on a field within the context of software testing. It helps testers and test automation scripts specify how they should interact with the software under test. KEYWORDS can be broadly categorized into three types:   1. input: This KEYWORD represents actions performed on elements such as entering data, making selections, or clicking buttons. 2. retrieve: When users need to fetch specific values from the application, whether they are system-generated or user input, and reuse these values in subsequent steps, use "retrieve" keyword. For example, retrieving a system-generated Application Number. 3. verify: This keyword is used when users want to check whether the values they entered are properly displayed on other screens or validate calculated values. For example, ensuring that Customer Name and Demographic type are correctly displayed on a summary screen.   Additionally, there are custom KEYWORDS created for specific situations:   1. insertCust: This keyword is used to insert a customer into ELCM using an SQL query before the manual-retry stage. 2. api: This keyword is used to send API requests to retrieve details of liabilities, facilities, covenants, etc., from CFPM and ELCM databases, which are part of the ELCM handoff process. 3. matchdata: This keyword is used to compare API responses to check whether the details match as part of the ELCM handoff process. | Yes |
| SKIP\_STEP | If you want to skip a particular step for all scenarios, you can define it here. Choose 'Y' to skip the step for all scenarios, or 'N' to execute the step for all scenarios. By default, it's set to 'N', but you can change it if needed. | Yes |
| SKIP\_SS | If you don't want to capture a screenshot for a specific field across all scenarios, specify it here. Select 'Y' to skip taking screenshots for all scenarios, or 'N' to capture screenshots for all scenarios. The default value is 'N', but you can customize it. | Yes |
| HARD\_ASSERT | A "HARD ASSERT" is a way to validate if something in the test is as expected. It is used for crucial checks. If a HARD ASSERT fails, the test immediately stops, indicating a critical problem that requires attention. For example, if the "SUBMIT" button doesn't work, there's no point in proceeding further. Not allow to enter/select mandatory field, Insert customer query fails, if issues in api response etc.  When defining a HARD ASSERT, you choose 'Y' to stop the test if the step fails.  There are times when a step may fail but doesn't necessarily require stopping the test, like verifying data or dealing with non-mandatory fields. In such cases, you can set HARD ASSERT to 'N'.  The default is 'N', but you can modify it based on your requirements. | Yes |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
|  | To view the steps defined for SUBFUNC\_NAME then use ‘SELECT FUNC’ button. This action opens a popup where the user can select the desired sub function name from a dropdown menu.    The dropdown menu will display sub function names based on the text entered, facilitating a smart selection process.  Click ‘View’ button to load existing steps if any.  Click ‘Cancel’ to close the popup.  Use this button, if user needs to add any new steps, edit and delete existing steps of particular sub function. |
|  | Use ‘NEW’ button to create steps for new sub function. This action opens a popup where the user can select the new sub function name from a dropdown menu.    New : System will remove any data already present and shows sub function name in below data label    Cancel – Closes the popup.  System will show below message if steps are already available in selected sub function.    Click ‘Yes’ to view the steps.  Click ‘No’ to close the popup.  Note: The dropdown menu will display sub function names based on the text entered, facilitating a smart selection process. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to add, edit, delete and save the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |
| SUBFUNC\_ID and STEP\_ID combination is already exists. Kindly recheck the data entered !!! | This message appears when a user enters same STEP\_ID for more than one step. Only unique STEP\_ID is allowed for selected subfunction name. |

## FUNC\_SUBFUNC\_MAPPING

The purpose of this worksheet is to map functions and sub-functions for specific scenarios. This mapping is a critical step, as any incorrect mapping can have a significant impact on the final scenario flow. Here, you define the sequence of functions and sub-functions, along with the number of iterations, which determines the repetition of steps. In essence, this mapping helps structure the logical flow of functions and sub-functions within a scenario, ensuring that the execution follows the intended sequence and repetition.

**Fields**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Mandatory** |
| SCN\_ID | This field is automatically filled based on the selected SCN\_ID for which you are creating mapping. You cannot edit this. | Yes |
| FUNC\_NAME | Select the function name from the dropdown menu which are part of scenario. FUNC\_NAME options are derived from FUNC\_MASTER. | Yes |
| FUNC\_SEQUENCE | Use this column to specify the order of function/ stages in scenario flow. You can assign a unique number to each function/stage to define the sequence. If you need to insert a function/stage between two existing ones, you can do so in two ways.  Example: Let's say you have three stages for scenario, and you want to add a new stage between the second and third.  Method 1:   1. Insert a new row after the 2nd stage/function and specify SEQUENCE as 3 2. Edit the functions/stages that come after the new one (step a), and renumber their SEQUENCE, e.g., 4. Repeat this if you have more functions/stages.   Method 2:  Insert a new row after the 2nd step and specify SEQUENCE as 2.1. | Yes |
| SUBFUNC\_NAME | Select the sub function name from the dropdown menu which are part of function/stage. SUBFUNC\_NAME options are derived from SUBFUNC\_MASTER. | Yes |
| SUFUNC\_SEQUENCE | Use this column to specify the order of sub function in function/stage. You can assign a unique number to each sub function to define the sequence. If you need to insert a sub function between two existing ones, you can do so in two ways.  Example: Let's say you have three sub functions, and you want to add a new sub function between the second and third.  Method 1:   1. Insert a new row after the 2nd sub function and specify SEQUENCE as 3 2. Edit the sub functions that come after the new one (step a), and renumber their SEQUENCE, e.g., 4. Repeat this if you have more sub functions.   Method 2:  Insert a new row after the 2nd step and specify SEQUENCE as 2.1. | Yes |
| ITERATION | Iteration is a concept in which you specify the number of times a set of sub-functions should be repeated in a scenario.  Ex: if user wants to add two facility schedules then mention ITERATION as 2. This will repeat the facility schedule steps twice in scenario so that they capture data accordingly in scenario steps. | Yes |
| ITERATION\_SEQUENCE | Iteration Sequence is a related concept used when you need to repeat a set of subfunctions multiple times, but the order of repetition is important.  For example, if you want to repeat subfunc\_1, subfunc\_2, and subfunc\_3 for 2 times in a specific sequence, it will be structured as follows:    This means that subfunc\_1, subfunc\_2, and subfunc\_3 are repeated together in the specified order for the first iteration. The same sequence is repeated for the second iteration, maintaining the order. | Yes |
| RUNMODE | If you want to exclude a function or stage from the scenario, you define it here. Choose 'Y' to skip it, or 'N' to include it. By default, it's set to 'N', but you can change it as required. | Yes |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
|  | If a user wishes to view mapping of an existing scenario, they can click the 'SELECT' button. This action opens a popup where the user can select the desired scenario id from a dropdown menu.    The dropdown menu will display scenario ids based on the text entered, facilitating a smart selection process.  Click ‘Select’ to view the mapping.  Click ‘Cancel’ to close. |
|  | Use ‘NEW’ button to create mapping for new scenario id. This action opens a popup where the user enter new scenario id.    New : System will remove any data already present and shows scenario id in below data label    Cancel – Closes the popup.  System will show below message if flow is already defined for entered scenario id. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to add, edit, delete and save the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |

## SCN\_FUNC\_BATCHING

You can outline the sequence of stages that a scenario or process should follow. If any new stages need to be added or existing ones need to be removed in the future, you can make those changes here. This section provides the flexibility to determine whether a specific stage should be executed as part of the scenario. You can mark a stage to be included or excluded as needed.

**Fields**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Mandatory** |
| SCN\_ID | This field is automatically filled based on the selected SCN\_ID for which you are outlining the sequence of stages. You cannot edit this. | Yes |
| FUNC\_NAME | Select the function name from the dropdown menu which are part of scenario. FUNC\_NAME options are derived from FUNC\_MASTER. | Yes |
| SEQUENCE | Use this column to specify the order of your scenario flow. You can assign a unique number to each function/stage to define the sequence. If you need to insert a function/stage between two existing ones, you can do so in two ways.  Example: Let's say you have three stages for scenario, and you want to add a new stage between the second and third.  Method 1:   1. Insert a new row after the 2nd stage/function and specify SEQUENCE as 3 2. Edit the functions/stages that come after the new one (step a), and renumber their SEQUENCE, e.g., 4. Repeat this if you have more functions/stages.   Method 2:  Insert a new row after the 2nd step and specify SEQUENCE as 2.1. | Yes |
| RUNMODE | If you want to exclude a function or stage from the scenario, you define it here. Choose 'Y' to skip it, or 'N' to include it. By default, it's set to 'N', but you can change it as required. | Yes |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
|  | If a user wishes to view end to end flow of about an existing scenario, they can click the 'SELECT' button. This action opens a popup where the user can select the desired scenario id from a dropdown menu.    The dropdown menu will display scenario ids based on the text entered, facilitating a smart selection process. |
|  | Use ‘NEW’ button to create end to end flow for new scenario id. This action opens a popup where the user enter new scenario id.    New : System will remove any data already present and shows scenario id in below data label    Cancel – Closes the popup.  System will show below message if flow is already defined for entered scenario id. |
|  | If you want to duplicate the end-to-end flow from an existing scenario and create a new one, click the 'COPY' button. This opens a pop-up where you select the scenario ID you want to copy from and then enter the new scenario ID.    Click 'Copy' to replicate the flow, or 'Cancel' to close the pop-up.  The 'COPY' action is a convenient way to create and duplicate scenario flows, especially when the new scenario is similar to an existing one or requires only minor changes. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to add, edit, delete and save the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |

## API

This worksheet used to prepare steps required for API calls to fetch data from CFPM and ELCM as a part of ELCM handoff. Here we mention details like Request Type, URI, header data and response verifications required for API call.

**Fields**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Mandatory** |
| SCN\_ID | This field is automatically filled based on the selected SCN\_ID for which you want to mention API step details. You cannot edit this. | Yes |
| SUBFUNCTION\_NAME | Name of sub function for which API calls are made. Normally its elcm\_cfpm\_verification. | Yes |
| STEP\_ID | STEP\_ID mentioned here should match with the STEP\_ID mentioned in scenario under elcm\_cfpm\_verification sub function.  Use this column to specify the order of your steps. You can assign a unique number to each step to define the sequence. If you need to insert a step between two existing ones, you can do so in two ways.  Example: Let's say you have three steps and you want to add a step between the second and third steps.  Method 1:   1. Insert a new row after the 2nd step and specify STEP\_ID as 3. 2. Edit the steps that come after the new one (step a), and renumber their STEP\_ID, e.g., 4. Repeat this if you have more steps.   Method 2:  Insert a new row after the 2nd step and give it a STEP\_ID of 2.1. | Yes |
| REQUEST\_TYPE | Please select appropriate REQUEST\_TYPE from the dropdown. Currently, only the GET method is applicable. | Yes |
| URI | In this field, provide the hostname and endpoint for the API call. The format is parameterized and depends on whether it's for CFPM or ELCM. Hostnames and endpoints can be found in respective configuration files in the api\_mapping folder.  For CFPM, use this format {{CFPM\_HostName}}/{{CFPM\_Query\_Liability\_Endpoint}}.  For ELCM, use this format  {{ELCM\_HostName}}/{{ELCM\_Query\_Liability\_Endpoint}} | Yes |
| HEADER\_DATA | Specify the header data required for API calls. The format is parameterized as well and depends on whether it's for CFPM or ELCM. Header details are available in respective configuration files in the api\_mapping folder.  For CFPM, use this format - {{CFPM\_Header}}  For ELCM, use this format - {{ECLM\_Header}} | Yes |
| RESPONSE\_VERIFICATION | Use this field to define the verifications needed to confirm that the API call receives the correct response. Identify and list the necessary validations within curly braces, separating them with commas if there's more than one. | Yes |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
|  |  |
|  | If a user wishes to view API steps of an existing scenario, they can click the 'SELECT' button. This action opens a popup where the user can select the desired scenario id from a dropdown menu.    The dropdown menu will display scenario ids based on the text entered, facilitating a smart selection process. |
|  | Use ‘NEW’ button to create API steps for new scenario id. This action opens a popup where the user enter new scenario id.    New : System will remove any data already present and shows scenario id in below data label    Cancel – Closes the popup. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to refresh the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |

## SCN\_STEPS\_GENERATOR

This worksheet is used to automatically generate the end-to-end steps for scenarios based on the SCN\_FUNC\_BATCHING table using ScenarioStepGenerator.jar file. After the steps are generated, users can specify the step-wise data required to execute the scenarios.

Before generating scenario steps, please ensure that the following activities and data are in place:

1. Make sure that SCN\_FUNC\_BATCHING is defined for the scenario. This table outlines the sequence of stages for the scenario.
2. Ensure that FUNC\_SUBFUNC\_MAPPING is defined for all the functions and subfunctions that are part of the scenario.
3. SUBFUNC\_STEPS - Define the steps for all subfunctions that are part of the scenario.
4. Ensure that the OBJECT\_REPO contains the necessary locators and types for all screens and fields that are part of the scenario.

Having these elements in place is essential to enable the automatic generation of scenario steps and to ensure that the end-to-end flow is correctly structured and ready for data input.

In the "scn\_steps\_generator" worksheet, users can conveniently access and view all the columns that are part of the other tables. This eliminates the need to navigate back to individual worksheets to reference the data, streamlining the process and making it more efficient. This centralized view provides a comprehensive overview of the relevant information, making it easier to work with scenario steps and associated data.

**Fields**

The following fields are part of the scenario steps and are non-editable. Any changes required in these columns then user needs to visit corresponding worksheet for modifications.

* SCN\_ID
* FUNC\_ID
* FUNC\_NAME
* SUB\_FUNC\_ID
* SUBFUNC\_NAME
* STEP\_ID
* STEP\_DESCRIPTION
* OR\_ID
* SCREEN\_NAME
* FIELD\_NAME
* KEYWORD
* TYPE
* LOCATOR
* FUNC\_SEQUENCE
* SUBFUNC\_SEQUENCE
* ITERATION
* ITERATION\_SEQUENCE

|  |  |
| --- | --- |
| **Name** | **Description** |
| SKIP\_STEP | Data will be populated from the SUBFUNC\_STEPS by default. If user still wants to skip a particular step for the scenario, you can override it here. Choose 'Y' to skip the step for scenario, or 'N' to execute the step for scenario. |
| SKIP\_SS | Data will be populated from the SUBFUNC\_STEPS by default. If user still don't want to capture a screenshot for a specific field for scenario, specify it here. Select 'Y' to skip taking screenshots, or 'N' to capture screenshots. |
| HARD\_ASSERT | Data will be populated from the SUBFUNC\_STEPS by default. If user still wants to change the value then specify it here.  Refer [2.5 SUBFUNC\_STEPS](#_SUBFUNC_STEPS) - Fields section on HARD\_ASSERT definition and use. |
| DATA | By default, EMPTY will be displayed under DATA column. However, users have the flexibility to replace this default value with the actual data required for each step.  For specific guidelines on how to input the data for each step, you can refer to the "Points to Consider" section for detailed instructions. This customization allows users to provide the relevant and specific data needed for each scenario step. |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
|  | If a user wishes to view detailed end to end steps of the scenario, they can click the 'SELECT SCN' button. This action opens a popup where the user can select the desired scenario id from a dropdown menu.    The dropdown menu will display scenario ids based on the text entered, facilitating a smart selection process.  Click ‘Select’ to view the data.  Click ‘Cancel’ to close |
|  | Use GENERATE button to generate steps for the scenario. This action opens a popup where user selects the scenario id for which steps needs to generate.    There's a checkbox labelled 'Retain SKIP\_STEP, SKIP\_SS, and HARD ASSERT VALUES of SCN\_STEPS.' If checked, this option will preserve the existing values of SKIP\_STEP, SKIP\_SS, and HARD ASSERT from SCN\_STEPS instead of overriding them with values from the SUBFUNC\_STEPS table.  This is useful when you're regenerating steps due to changes in iteration or subfunction deletion. In such cases, retaining SKIP\_STEP, SKIP\_SS, and HARD ASSERT values as specified in SCN\_STEPS is necessary.  Click ‘Cancel’ button to close popup.  Clicking the 'Generate' button initiates the step generation process. A message will prompt you, as the system may take some time to generate the steps, depending on network speed.  You can click 'Yes' to proceed with step generation or 'No' to cancel it.  During the process, a command prompt will be opened. Do not close it.  After step generation, the system will display a pop-up message indicating whether the steps were generated successfully or if there were any error messages.  The system will also generate a log.out file for each step generation. You can refer to this file for any error messages.  Below pop-up will appear, allowing you to view the steps immediately upon clicking 'Yes.' If you choose 'No,' the steps won't be displayed, but you can use the 'SELECT SCN' button to view them later. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to refresh the records.

Scenario steps should be regenerated in the following situations:

* Changes in FUNC\_SUBFUNC\_MAPPING: If there are any additions or deletions of records in FUNC\_SUBFUNC\_MAPPING for a specific scenario, it's necessary to regenerate the scenario steps.
* Changes in ITERATION Numbers:
  + Increase ITERATION: For instance, if the ITERATION column originally had 2 facility schedules and this repetition needs to be increased (e.g., from 2 to 3), you should update the ITERATION column and regenerate SCN\_STEPS. This addition will create steps for the third schedule with empty data, and you can then enter the data for the newly added steps.
  + Decrease ITERATION: Conversely, if the ITERATION needs to be decreased (e.g., from 2 to 1), make sure to update the ITERATION column and regenerate SCN\_STEPS. This action will remove the steps associated with the second schedule from the scenario steps.
* Addition of New Steps: If new steps are added to any subfunction in the SUBFUNC\_STEPS worksheet, it's important to regenerate the scenario steps to incorporate these new steps into the end-to-end flow.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| No records available for selected Scenario ID!! | This message appears when no data is available for selected scenario id. |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |

## RUN\_TIME\_DATA

In this worksheet user can access scenario specific data that is retrieved during execution. This helps whether to cross check whether correct data is retrieved during the execution.

There is only view and refresh option. User can not add, edit or delete the data.

**Fields**

|  |  |
| --- | --- |
| **Name** | **Description** |
| SCN\_ID | This field is automatically filled based on the selected SCN\_ID for which you want to see data. You cannot edit this. |
| VARIABLE\_NAME | Unique name to store the data. This helps whenever user to refer in further step. |
| DATA | Actual data retrieved during execution. |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
|  | If a user wishes to view data retrieved during execution for the scenario, they can click the 'SELECT SCN' button. This action opens a popup where the user can select the desired scenario id from a dropdown menu.    The dropdown menu will display scenario ids based on the text entered, facilitating a smart selection process.  Click ‘Select’ to view the data.  Click ‘Cancel’ to close |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to refresh the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| No records available for selected Scenario ID!! | This message appears when no data is available for selected scenario id. |

## COMMON\_DATA

COMMON\_DATA holds common data elements like URLs, user IDs, passwords, currencies, and more, ensuring consistency and reusability.

**Fields**

|  |  |
| --- | --- |
| **Name** | **Description** |
| VARIABLE\_NAME | Unique name to store the data. This helps whenever user to refer in further step. |
| DATA | Actual data used during execution. |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
| A logo of an eye  Description automatically generated | To view the already defined common data, click the 'VIEW' button. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to add, edit and delete the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |

## RUNCHART

This worksheet allows user to execute specific scenarios based on tags and view the execution logs and reports conveniently.

Runchart will be specific to user. This provision is provided avoid conflict between two users execution.

**Fields**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Mandatory** |
| SCN\_ID | Mention scenario id | Yes |
| SCN\_DESCRIPTION | You can provide a brief description of the scenario if needed, | No |
| TAG | Users can define tags to categorize scenarios. For example, you can use tags like "regression" or "sanity." If you want to assign multiple tags, separate them with commas. It's required for creating and managing test suites, but only one suite can be executed at a time based on your input. | Yes |
| USERID | This field is automatically filled with your system login ID. You can't edit it. | Yes |
| DATASET\_ID | Here, you define which set of data should be used for the execution. Currently, only one set of data is allowed. | Yes |
| EXECUTE | This field allows you to decide whether a particular scenario should be executed or skipped during the test run. Select 'Y' to execute the scenario or 'N' to skip it. The system will only execute scenarios for which you specify 'Y.' | Yes |

**Actions**

|  |  |
| --- | --- |
| **#** | **Description** |
| A logo of an eye  Description automatically generated | To view the already defined run chart, click the 'VIEW' button. |
|  | You can initiate the automation execution of scenarios that are marked as 'EXECUTE: Yes.'  When you start the execution, a pop-up will appear. In this pop-up, you need to select a tag from the dropdown menu.  The system will execute only those scenarios that are marked with the selected tag and have 'EXECUTE' set to 'Y.'    Click ‘Run’ to start execution.  Click ‘Cancel’ to close.  The system will open a command prompt and display the execution logs. |
|  | After the execution is complete, if you want to view the report, you can click the 'VIEW REPORT' button. This will open the Test Execution dashboard HTML page in your system's default web browser. |

Kindly refer [section 2.1 ACTION BUTTONS](#_ACTION_BUTTONS) to add, edit and delete the records.

**Validations:**

|  |  |
| --- | --- |
| **Error Message** | **Description** |
| Please enter data highlighted in red color. | This message appears when mandatory data is not entered. The system highlights the missing information in red to prompt the user to enter it. |

## LIST\_ENUMS

In this worksheet, you can find a list of ENUM that are used in different parts of the GUI. Users have the ability to define Enum names, values, and set default values if needed.

Currently, this worksheet is locked to prevent accidental changes. If you need to edit the data, you can unlock the worksheet. After making modifications, you can click the "SAVE" button to save the changes you've made.

