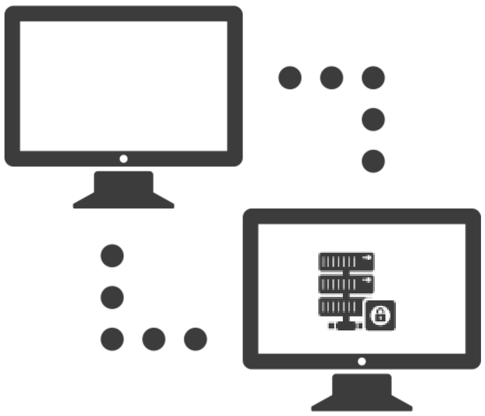
iMFAUT User Guide

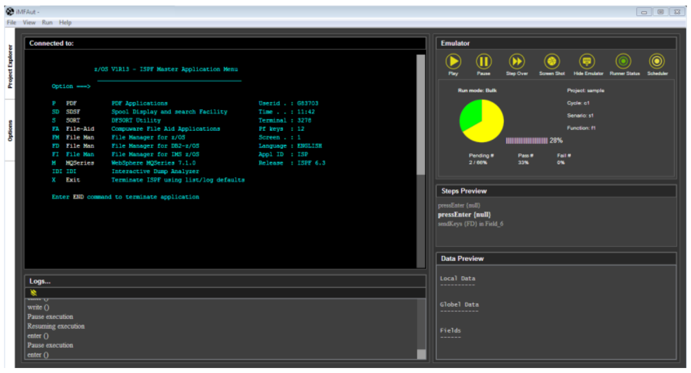
Everything you need to know about Infosys Mainframe Automation Tool



Contents

SAY HELLO TO INFOSYS MAINFRMAME AUTOMATION TOOL

iMFAUT



```
              z/OS V1R13 - ISPF Master Application Menu
              ________________________________________
   Option ===>

   P   PDF        PDF Applications                       Userid . : G83703
   SD  SDSF       Spool Display and search Facility      Time . . : 11:42
   S   SORT       DFSORT Utility                         Terminal : 3278
   FA  File-Aid   Compuware File Aid Applications       Pf keys  : 12
   FM  File Man   File Manager for z/OS                  Screen . : 1
   FD  File Man   File Manager for DB2-z/OS              Language : ENGLISH
   FI  File Man   File Manager for IMS z/OS              Appl ID  : ISP
   M   MQSeries   WebSphere MQSeries 7.1.0               Release  : ISPF 6.3
   IDI IDI        Interactive Dump Analyzer
   X   Exit       Terminate ISPF using list/log defaults

   Enter END command to terminate application
```

```
Run mode: Bulk          Project: sample
                        Cycle: c1
                        Senario: s1
                        Function: f1
              28%
Pending #     Pass #     Fail #
2 / 66%       33%        0%
```

```
pressEnter {null}
pressEnter {null}
sendKeys (FD) in Field_6
```

```
Local Data
----------

Globel Data
-----------

Fields
------
```

```
write ()
Pause execution
Resuming execution
enter ()
Pause execution
enter ()
```

1. Easy connection to Mainframe.
2. Steps recording and playback.
3. Built in report generation.
4. Execution scheduling.
5. Built in test data manager.
6. BDD-natural language support.
7. Built in library editing.

Pre-Launch Checks

System Requirements.

1) JDK 1.8+ installed and configured in system
2) JRE 1.7+ installed and configured in system
3) 1GB ram
4) 1 GB min Disk Space

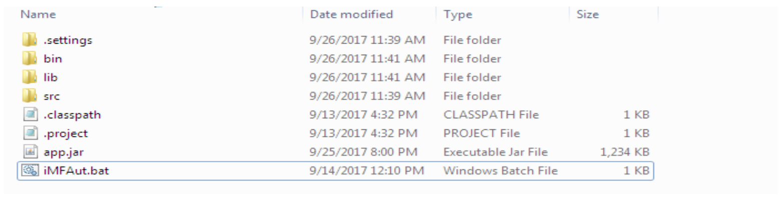
iMFAUT Requirements.

1) Should have a valid license key
2) iMFAUT is activated for the machine

Let's Get Started

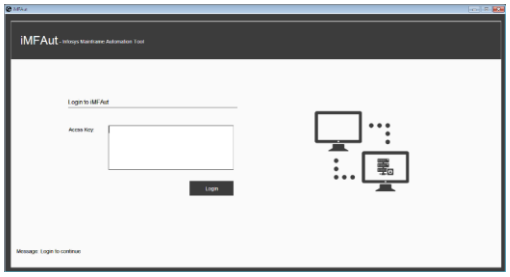
iMFAUT

Launch iMFAUT. Run iMFAut.bat file in tool directory.



Name	Date modified	Type	Size
.settings	9/26/2017 11:39 AM	File folder	
bin	9/26/2017 11:41 AM	File folder	
lib	9/26/2017 11:41 AM	File folder	
src	9/26/2017 11:39 AM	File folder	
.classpath	9/13/2017 4:32 PM	CLASSPATH File	1 KB
.project	9/13/2017 4:32 PM	PROJECT File	1 KB
app.jar	9/25/2017 8:00 PM	Executable Jar File	1,234 KB
iMFAut.bat	9/14/2017 12:10 PM	Windows Batch File	1 KB

Enter valid access key. Enter valid access key in access key field and click login button.

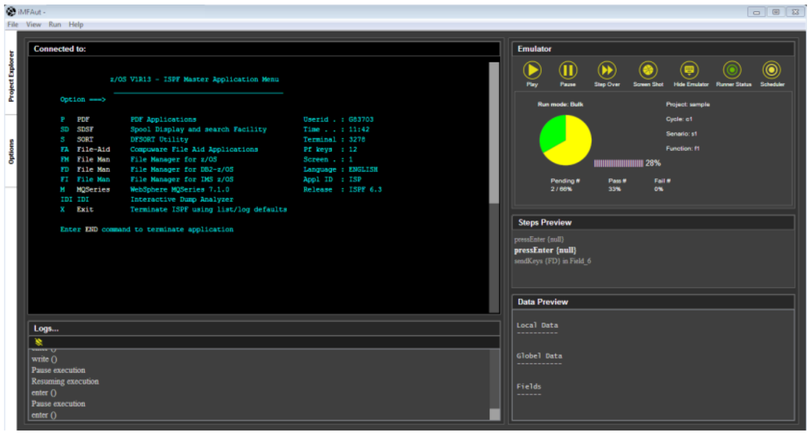


Message: Login to continue

Note: - On invalid 5th login attempt tool will automatically have deleted from system.

Close Scheduler. After login you will be in Scheduler screen. For time being you can close Scheduler window

iMFAUT Emulator. Now you are at iMFAUT Emulator



```
              z/OS V1R13 - ISPF Master Application Menu
               ____________________________________________
    Option ===>

    P   PDF        PDF Applications                  Userid . : G83703
    SD  SDSF       Spool Display and search Facility Time . . : 11:42
    S   SORT       DFSORT Utility                    Terminal : 3278
    FA  File-Aid   Compuware File Aid Applications   Pf keys  : 12
    FM  File Man   File Manager for z/OS             Screen . : 1
    FD  File Man   File Manager for DB2-z/OS         Language : ENGLISH
    FI  File Man   File Manager for IMS z/OS         Appl ID  : ISP
    M   MQSeries   WebSphere MQSeries 7.1.0          Release  : ISPF 6.3
    IDI IDI        Interactive Dump Analyzer
    X   Exit       Terminate ISPF using list/log defaults

    Enter END command to terminate application
```

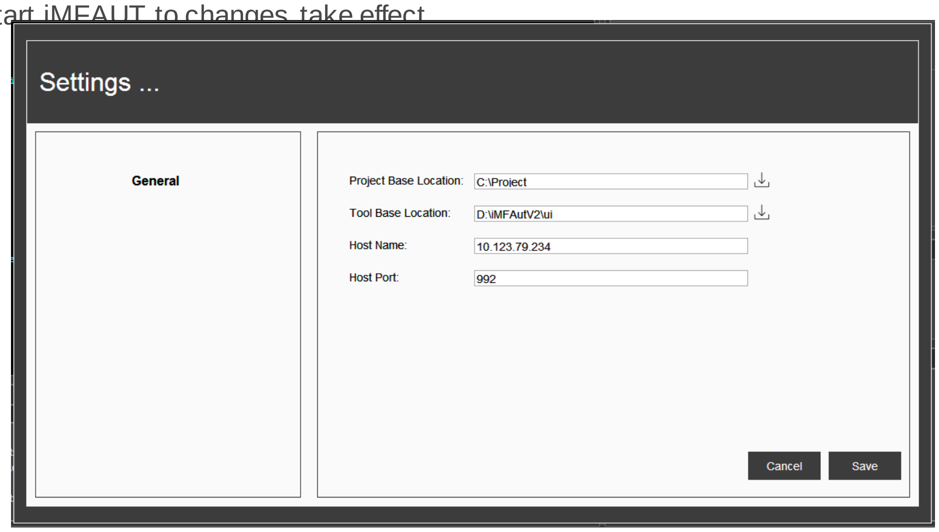
```
pressEnter {null}
pressEnter {null}
sendKeys (FD) in Field_6
```

```
Local Data
----------

Globel Data
-----------

Fields
------
```

```
write ()
Pause execution
Resuming execution
enter ()
Pause execution
enter ()
```

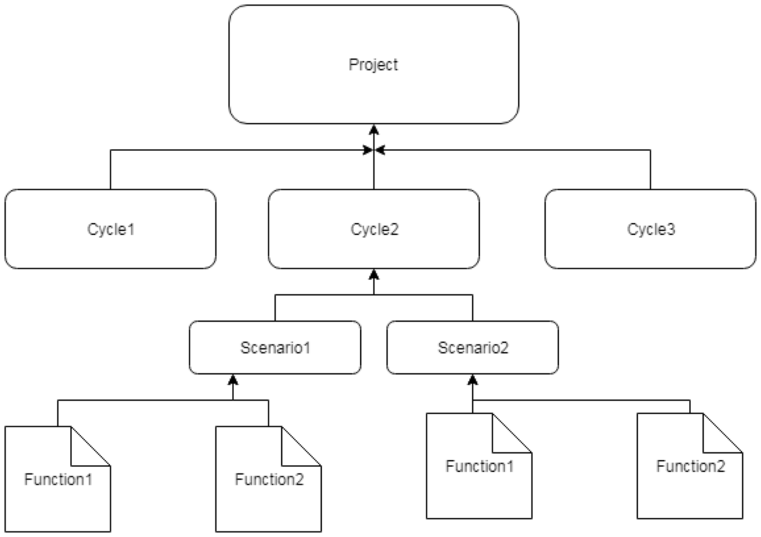
Set basic setting to get started. Perform following steps to configure tool for the first time

1) Navigate to File → Settings.
2) Set Project Base Location by navigating using Project Location Explorer.
3) Set Host Name. Note: - You can give mainframe host name or IP.
4) Set Host Port. Note: - Set port in which mainframe is assigned to external connections.
5) Click On Save Button.
6) Restart iMFAUT to changes take effect



Project Base Location:	C:\Project
Tool Base Location:	D:\iMFAutV2\ui
Host Name:	10.123.79.234
Host Port:	992

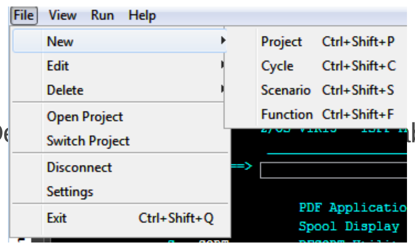
Create new Project. Everything you do in iMFAUT comes under Project. Project is having following structure.



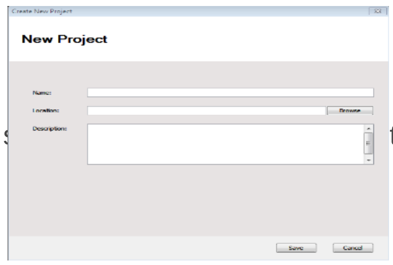
- To add new project, perform following steps.

  1) Navigate to File → New → Project
  2) Provide Name for the project
  3) If you want to change project base location, you can do it by clicking and selecting project location
  4) Provide optional description and click on save to create new project

To Edit or De



bove s



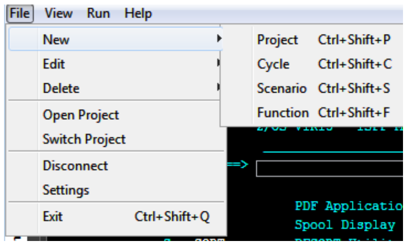
t/Delete → Project

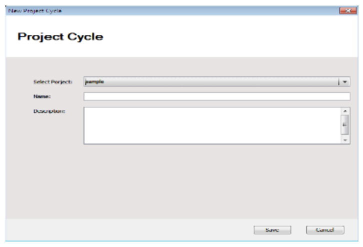
- To create new project cycle, perform following steps.

  1) Navigate to File → New → Cycle
  2) Select Project Name from project drop down.

3) Provide Cycle name.

4) Provide optional description and click on save to create new project cycle





Select Project: jsample
Name:
Description:

To Edit or Delete the project cycle you can follow above steps by navigating File → Edit/Delete → Cycle

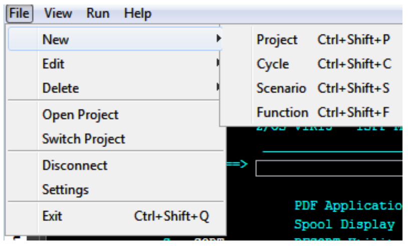
- To create new project scenario, perform following steps.

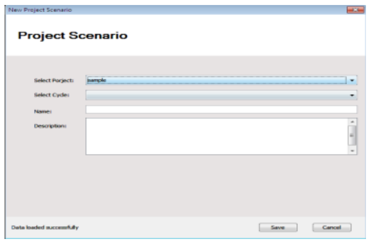
  1) Navigate to File → New → Scenario

  2) Select Project Name and Cycle Name.

  3) Provide Scenario name.

  4) Provide optional description and click on save to create new project scenario.





Select Project: jsample
Select Cycle:
Name:
Description:
Data loaded successfully

To Edit or Delete the project scenario you can follow above steps by navigating File → Edit/Delete → Scenario

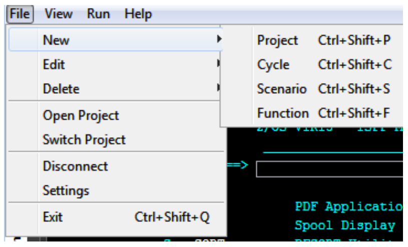
- To create new project function, perform following steps

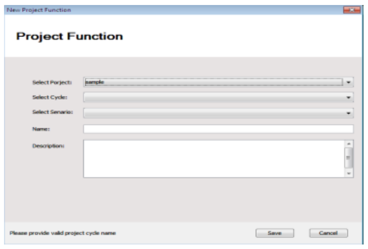
  1) Navigate to File → New → Function

  2) Select Project Name, Cycle Name and Scenario Name.

3) Provide Function name.

4) Provide optional description and click on save to create new project function.




New Project Function

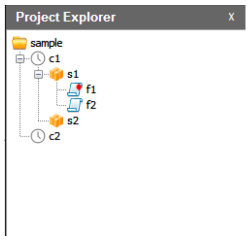
Project Function

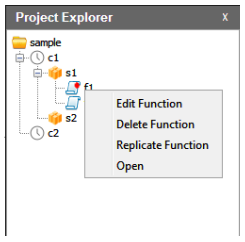
Select Porject: sample
Select Cycle:
Select Senario:
Name:
Description:

Please provide valid project cycle name

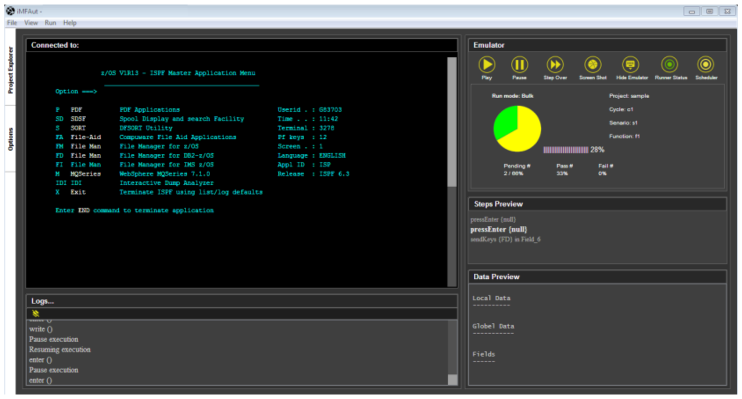
To Edit or Delete the project function you can follow above steps by navigating File → Edit/Delete → Function

Project Explorer. Using Project Explorer, you can easily manage project in iMFAUT. Other than new project functionality rest of all project activities you could done from Project Explorer itself. You could activate Project Explore by clicking on left side bar Project Explorer button.





iMFAUT Emulator. After login to tool you will have landed in iMFAUT Emulator. Or you could launch iMFAUT Emulator by navigating through View → View Emulator. Using Emulator, you could do following functionalities.



```
            z/OS V1R13 - ISPF Master Application Menu

   Option ===>

   P   PDF          PDF Applications                  Userid . . : G83703
   SD  SDSF         Spool Display and search Facility Time . . . : 11:42
   S   SORT         DFSORT Utility                    Terminal . : 3278
   FA  File-Aid     Compuware File Aid Applications   Pf keys  . : 12
   FM  File Man     File Manager for z/OS             Screen . . : 1
   FD  File Man     File Manager for DB2-z/OS         Language . : ENGLISH
   FI  File Man     File Manager for IMS z/OS         Appl ID  . : ISP
   M   MQSeries     WebSphere MQSeries 7.1.0          Release  . : ISPF 6.3
   IDI IDI          Interactive Dump Analyzer
   X   Exit         Terminate ISPF using list/log defaults

   Enter END command to terminate application
```

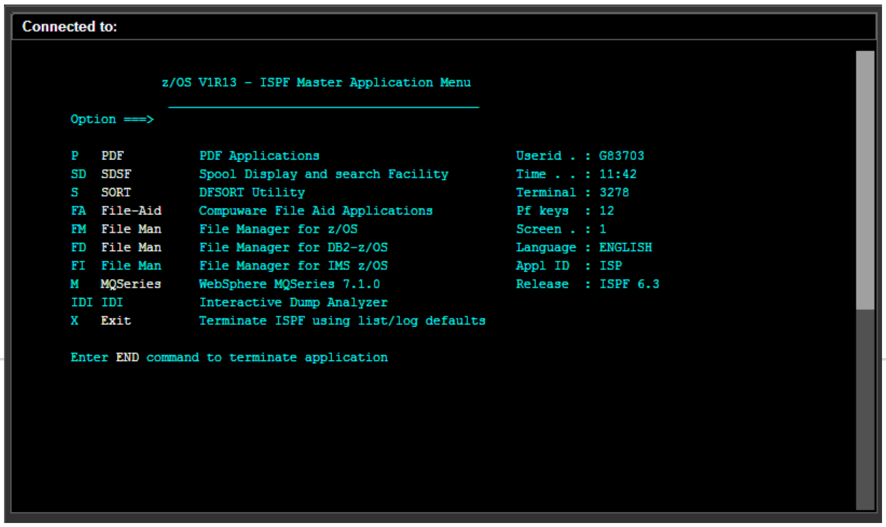
```
pressEnter {null}
pressEnter {null}
sendKeys (FD) in Field_6
```

```
Local Data
----------

Globel Data
-----------

Fields
------
```

```
write ()
Pause execution
Resuming execution
enter ()
Pause execution
enter ()
```

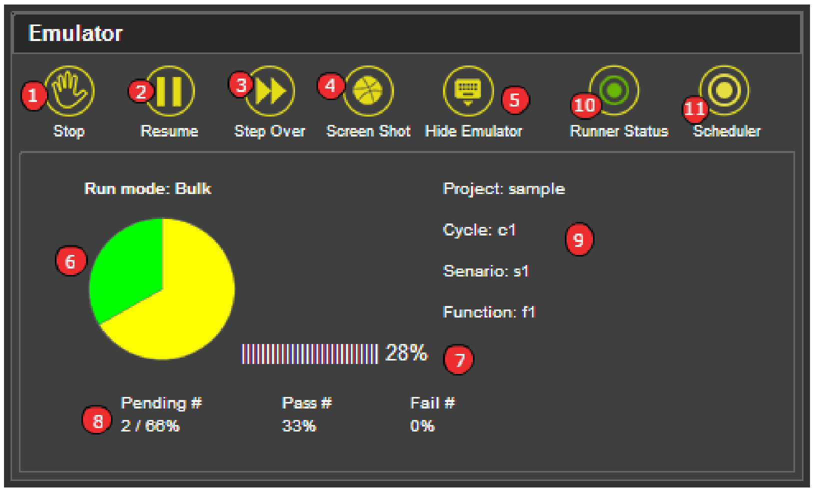
- Run project functionalities
- Control execution
- Provide Real-time summary of execution
- Provide Execution logs
- Provide Steps Preview
- Provide Data Preview of current step being executed

Emulator. Provide Real-time emulation of connected mainframe. You can connect to configured mainframe by navigating File → Connect.



```
            z/OS V1R13 - ISPF Master Application Menu

   Option ===>

   P   PDF          PDF Applications                  Userid . . : G83703
   SD  SDSF         Spool Display and search Facility Time . . . : 11:42
   S   SORT         DFSORT Utility                    Terminal . : 3278
   FA  File-Aid     Compuware File Aid Applications   Pf keys  . : 12
   FM  File Man     File Manager for z/OS             Screen . . : 1
   FD  File Man     File Manager for DB2-z/OS         Language . : ENGLISH
   FI  File Man     File Manager for IMS z/OS         Appl ID  . : ISP
   M   MQSeries     WebSphere MQSeries 7.1.0          Release  . : ISPF 6.3
   IDI IDI          Interactive Dump Analyzer
   X   Exit         Terminate ISPF using list/log defaults

   Enter END command to terminate application
```

Emulator Dashboard. Emulator Dashboard help you to control execution of functionalities.



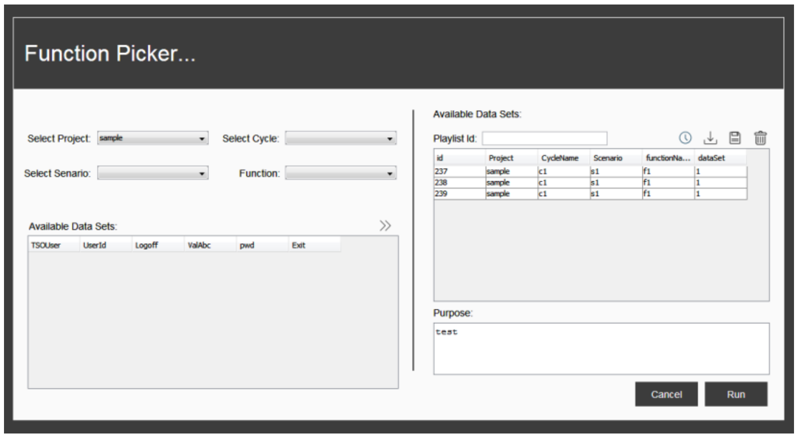
1) Start/Stop execution of functionalities. When you click play for first time (before execution or tool start it will launch following Function Picker window to select functions to run)

   1.1) Using Function Picker window, you can select different functionalities and map data set for execution and add to play list for execution by clicking >> button.

   1.2) You can give an optional purpose for execution (If you give a purpose for execution later you could use saved play list for scheduling execution. We will explain about later).

   1.3) Click Run. If mainframe is not in connected state in log it will show a message to connect. The playlist you have selected will start executing and you could observe its progress in Emulator Dashboard

1.4) After Executing HTML report gets generated automatically and opened in Web Browser.



Select Project: sample
Select Cycle:
Select Senario:
Function:

Available Data Sets:

TSOUser	UserId	Logoff	ValAbc	pwd	Exit

Available Data Sets:

Playlist Id:

id	Project	CycleName	Scenario	functionNa...	dataSet
237	sample	c1	s1	f1	1
238	sample	c1	s1	f1	1
239	sample	c1	s1	f1	1

Purpose:

test

2) **Pause/Resume** execution. You could pause/resume execution flow by clicking Pause/Resume button.

3) **Step Over**. After Pausing execution flow you could continue execution by step by step

4) **Screen Shot.** After Pausing execution, you could take multiple screen shots of mainframe screen by clicking Screen Shot button. Screen shots will be added to report's current step.

5) **Hide Emulator.** You can hide emulator by clicking on Hide Emulator button.

6) Graphical summary of overall progress

7) Progress of current functionality

8) Numerical summary of overall progress

9) Provides details of current functionality being executed.

10) Runner Status Indicator will show current status of **iMFAUT** Runner with 3 color codes

1.1) **Green** – runner is free and it is ready for execution

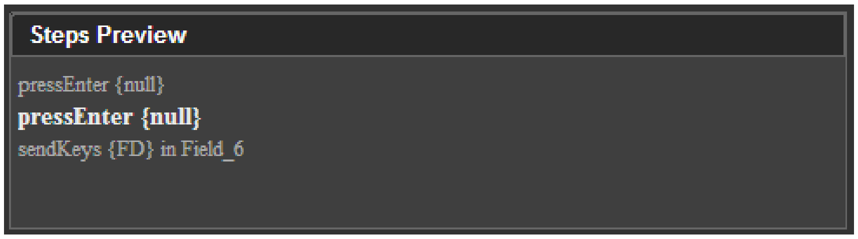
1.2) **Red** – runner is busy

1.3) **Orange** – runner is facing error due to last execution, at this point you need to re-initialize **iMFAUT** runner to continue exaction otherwise scheduler will stop.

To re-initialize **iMFAUT** runner you can click on flashing icon or View → Scheduler and toggle orange **iMFAUT** function runner state switch. **iMFAUT** function runner state switch should be green in color. Above mentioned color codes indications same for **iMFAUT** function runner state switch. Only in error you can reset **iMFAUT** function runner.

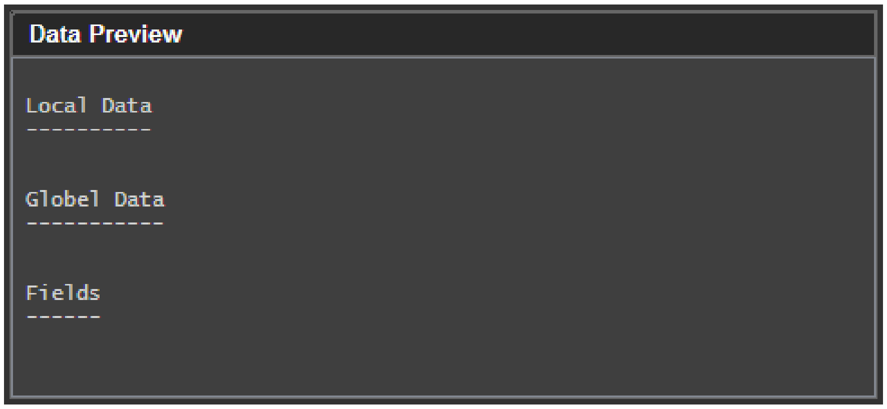
11) Indicate state of Scheduler is activated or not. On Scheduler is in activated stage icon will change to blue flashing state. We will explain more in Scheduler section.

Steps Preview. Provide preview of current step being executed



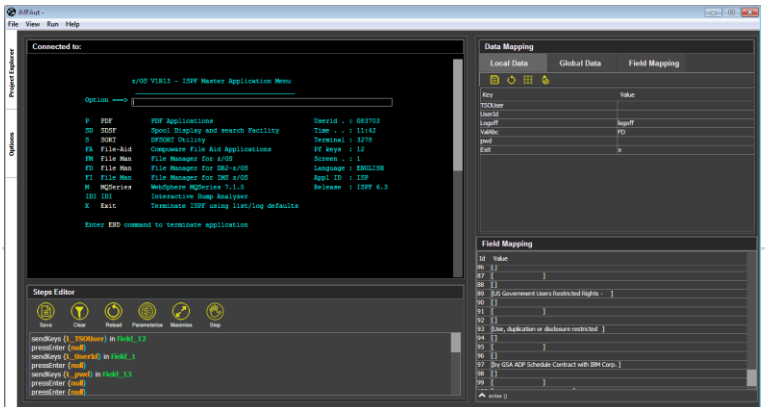
```
pressEnter {null}
pressEnter {null}
sendKeys {FD} in Field_6
```

Data Preview. Provide details of data being used by current steps



```
Local Data
----------

Globel Data
-----------

Fields
------
```

iMFAUT Script Editor. After clicking on Hide Emulator button in emulator window or navigating through View → Hide Emulator you could land in iMFAUT Script Editor. Using iMFAUT Script Editor, you could do following functionalities.



```
              z/OS V1R13 - ISPF Master Application Menu

   Option ===>

   P   PDF       PDF Applications                      Userid . : G83703
   SD  SDSF      Spool Display and search Facility     Time . . : 11:42
   S   SORT      DFSORT Utility                        Terminal : 3278
   FA  File-Aid  Compuware File Aid Applications       Pf keys  : 12
   FM  File Man  File Manager for z/OS                 Screen . : 1
   FD  File Man  File Manager for DB2-z/OS             Language : ENGLISH
   FI  File Man  File Manager for IMS z/OS             Appl ID  : ISP
   M   MQSeries  WebSphere MQSeries 7.1.0              Release  : ISPF 6.3
   IDI IDI       Interactive Dump Analyzer
   X   Exit      Terminate ISPF using list/log defaults

   Enter END command to terminate application
```

```
sendKeys {L_TSOUser} in Field_12
pressEnter {null}
sendKeys {L_UserId} in Field_1
pressEnter {null}
sendKeys {L_pwd} in Field_13
pressEnter {null}
pressEnter {null}
```

Key	Value
TSOUser	
UserId	
Logoff	logoff
ValAbc	FD
pwd	
Exit	x

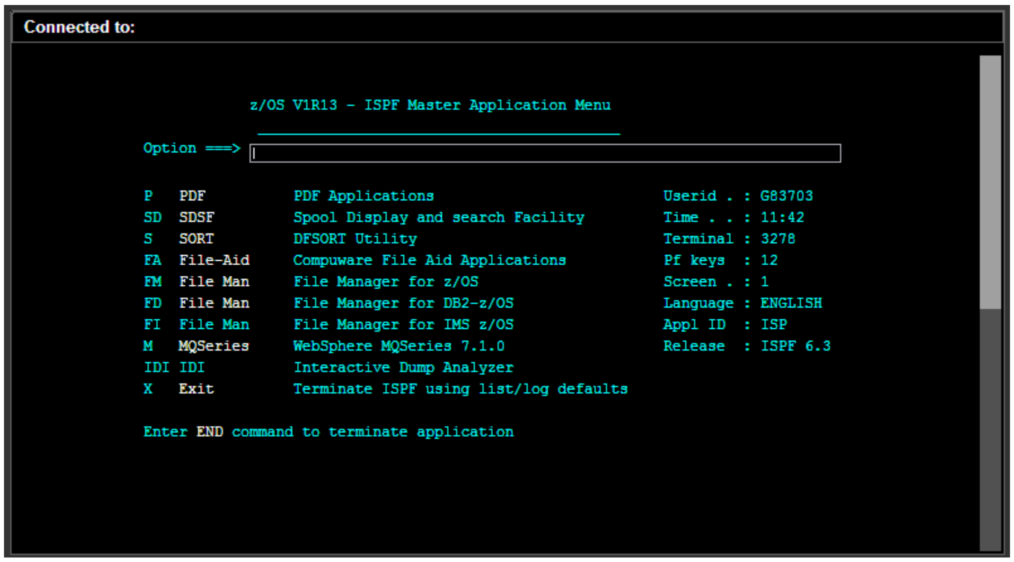
Id	Value
86	[]
87	[ ]
88	[]
89	[US Government Users Restricted Rights -]
90	[]
91	[ ]
92	[]
93	[Use, duplication or disclosure restricted]
94	[]
95	[ ]
96	[]
97	[by GSA ADP Schedule Contract with IBM Corp.]
98	[]
99	[ ]

enter ()

- ➢ Recode functionality steps
- ➢ Edit functionality steps using Steps Editor
- ➢ Perform Dry run of functionality steps

- Map Test Data and Mainframe Fields with steps
- Manage Globel Data Sets
- Provide information about mainframe fields in Field Mapping

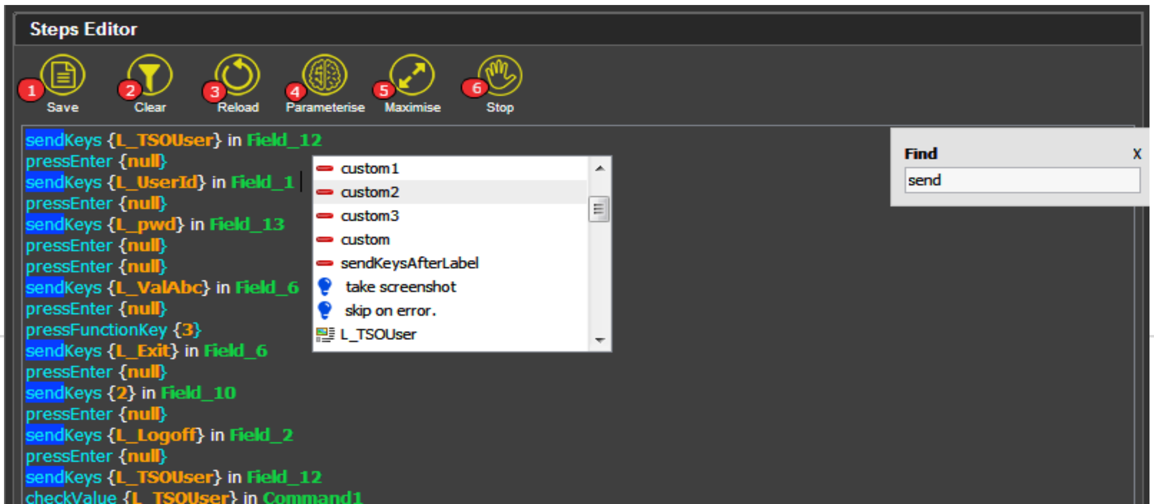
iMFAUT Steps Recorder. Provide easy to use interface to recode action. Which can be used to create functionality scripts. Recorder will highlight available input fields in which you could enter data. While doing action corresponding steps gets added to Steps Editor.



Connected to:

```
                 z/OS V1R13 - ISPF Master Application Menu

Option ===> 

P    PDF          PDF Applications                          Userid . . : G83703
SD   SDSF         Spool Display and search Facility         Time . . . : 11:42
S    SORT         DFSORT Utility                            Terminal : 3278
FA   File-Aid     Compuware File Aid Applications           Pf keys  : 12
FM   File Man     File Manager for z/OS                     Screen . : 1
FD   File Man     File Manager for DB2-z/OS                 Language : ENGLISH
FI   File Man     File Manager for IMS z/OS                 Appl ID  : ISP
M    MQSeries     WebSphere MQSeries 7.1.0                  Release  : ISPF 6.3
IDI  IDI          Interactive Dump Analyzer
X    Exit         Terminate ISPF using list/log defaults

Enter END command to terminate application
```

iMFAUT Steps Editor. Provide BDD - natural language based step editing functionality



```
sendKeys {L_TSOUser} in Field_12
pressEnter {null}
sendKeys {L_UserId} in Field_1
pressEnter {null}
sendKeys {L_pwd} in Field_13
pressEnter {null}
pressEnter {null}
sendKeys {L_ValAbc} in Field_6
pressEnter {null}
pressFunctionKey {3}
sendKeys {L_Exit} in Field_6
pressEnter {null}
sendKeys {2} in Field_10
pressEnter {null}
sendKeys {L_Logoff} in Field_2
pressEnter {null}
sendKeys {L_TSOUser} in Field_12
checkValue {L_TSOUser} in Command1
```

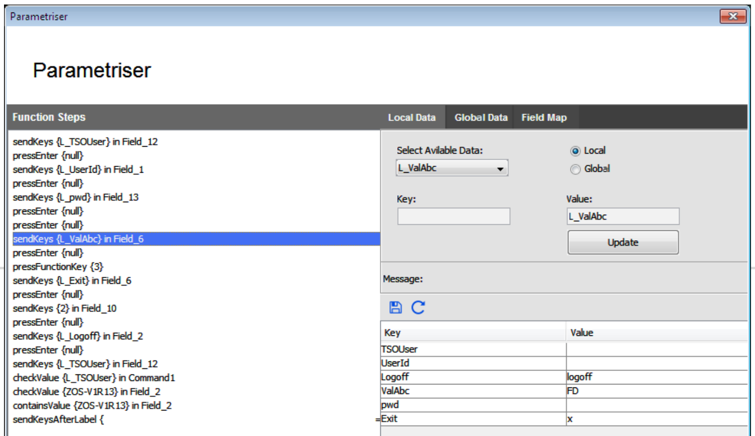
1) **Save** steps to current functionality. You can perform save action by pressing key combination Ctrl + S Keys

2) **Clear** steps.

3) **Reload** from last saved state.

4) **Parameterize** help you to easily map and manage Test Data and Field information with steps

   1.1) Using Parameterize window you can map Local, Globel Test Data and Field Information with steps



```
sendKeys {L_TSOUser} in Field_12
pressEnter {null}
sendKeys {L_UserId} in Field_1
pressEnter {null}
sendKeys {L_pwd} in Field_13
pressEnter {null}
pressEnter {null}
sendKeys {L_ValAbc} in Field_6
pressEnter {null}
pressFunctionKey {3}
sendKeys {L_Exit} in Field_6
pressEnter {null}
sendKeys {2} in Field_10
pressEnter {null}
sendKeys {L_Logoff} in Field_2
pressEnter {null}
sendKeys {L_TSOUser} in Field_12
checkValue {L_TSOUser} in Command1
checkValue {ZOS-V1R13} in Field_2
containsValue {ZOS-V1R13} in Field_2
sendKeysAfterLabel {
```

Select Avilable Data: L_ValAbc
[✓ Local] Global
Key:
Value: L_ValAbc

Key	Value
TSOUser	
UserId	
Logoff	logoff
ValAbc	FD
pwd	
Exit	x

1.1)1. In Parameterize window's steps related to current functionality will be shown

1.1)2. By selecting/clicking each steps in local data tab it will show available data prams which can be parametrize. In Field Map its will show Field information which can be parametrized.

1.1)3. You can select Data Parameter or Field Parameter and Provide Key

1.1)4. Click on Update. Selected steps will get parametrized automatically and it will get reflected in Steps Editor.

1.1)5. You could move parameter to Globel data set by selecting Globel radio button in Local Data tab

1.1)6. Data in Globel Data set will be available for all functionality in current project

5) Maximize/Minimize you can maximize/minimize Step Editor by Clicking Maximize/Minimize button

6) Play/Stop you can dry run current steps by clicking Play/Stop button. Note: - while dry running report will not get generated

IntelliSense Suggestions you can get IntelliSense Suggestions by pressing Ctrl + Space Key combination. Steps editor will show suggestions based on context

Find you can get find window by pressing Ctrl + F Key combination.

Steps Format Step editor follow advanced BDD flavored form of step interpreter. A valid step is having following format

Method {***parms1***, ***params2***, ***n***} ***in Field*** and ***take screenshot*** and ***skip on error***.

Eg: - containsValue {ZOS-V1R13} in Field_2

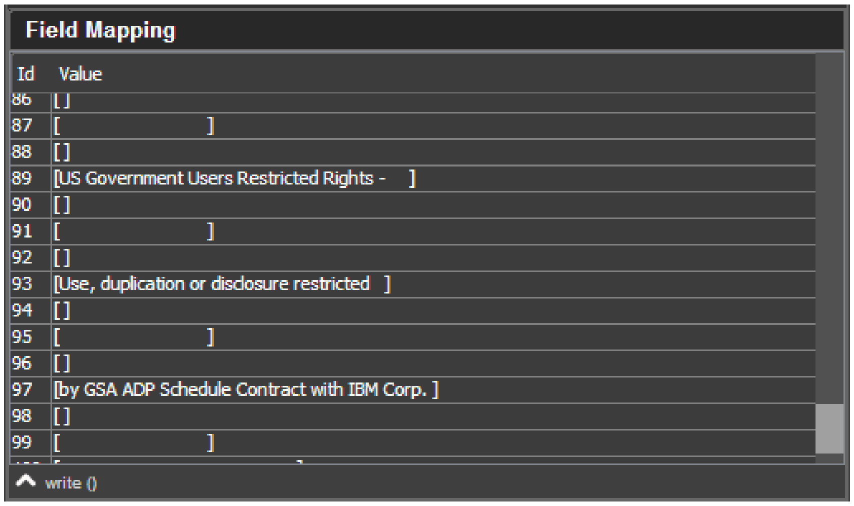
pressFunctionKey {3}

pressEnter {null}

sendKeys {L_Logoff} in Field_2

<u>containsValue {ZOS-V1R13} in Field_2 –</u>

- containsValue → Method/Keyword which is defined in user library
- {ZOS-V1R13} → Parameters to be passed, more number of parameter will be separated by comma and we will pass **null** if no parameter is not required for step.
- in Field_2 → Specify the main frame field in which action should be taken, you can Field number from Field Mapping table
- Let's say you want to interact with [US Government Users Restricted Rights -]. First get **Id** of Field – 89. So will specify ***Field_89***



Field Mapping

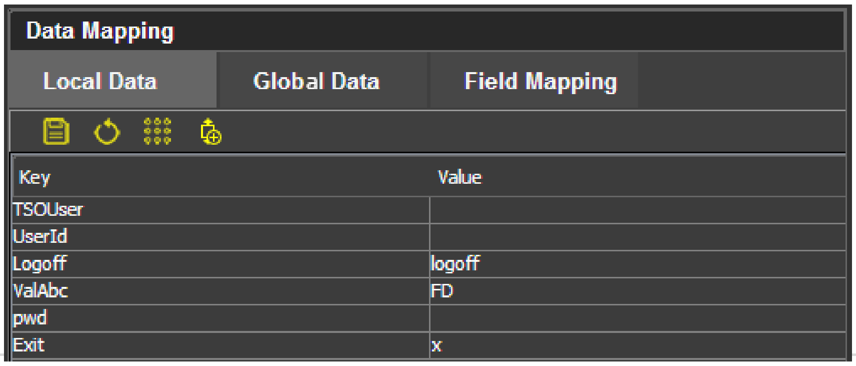
Id	Value
86	[ ]
87	[ ]
88	[ ]
89	[US Government Users Restricted Rights -]
90	[ ]
91	[ ]
92	[ ]
93	[Use, duplication or disclosure restricted]
94	[ ]
95	[ ]
96	[ ]
97	[by GSA ADP Schedule Contract with IBM Corp.]
98	[ ]
99	[ ]

write ()

- You can modify/Recode values from Field Mapping table. Edit value cell value and press **Enter** Key updated value will get reflected in Emulator if edited field is an input mainframe field.

  Note: - We will discuss more on parameters on **Data Mapping** session

iMFAUT Data Mapping. Provide inline Data and Field mapping functionality.



Data Mapping

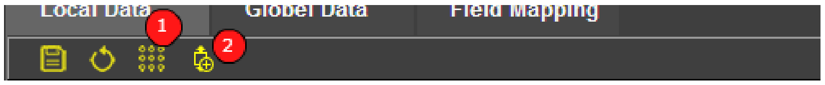
Key	Value
TSOUser	
UserId	
Logoff	logoff
ValAbc	FD
pwd	
Exit	x

1. **Parameter Data** is categories into Global and Local data sets.

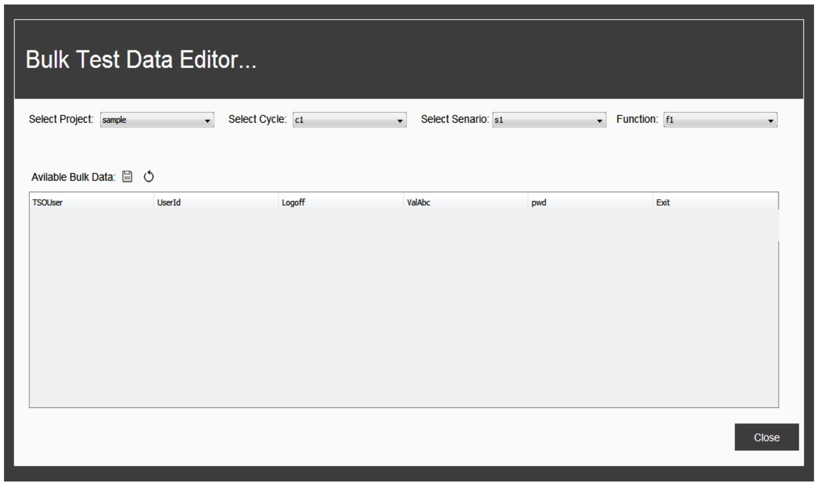
   a. **Local Data** which is applicable only for selected functionality. Data set is managed as Key Value Pair. Eg: - you want parameterize 'ZOS-V1R13' add parameter in Value new row cell similar to excel editing and provide Key (Variable Name) for the 'ZOS-V1R13' in Key field. Let's say **SysName**. Then you could access that Key in steps by Preceding **L_ → L_SysName**.

      L_ → Indicate it refer to Local Data Set.

      G_ → Indicate it refer to Global Data Set.



      i. **Bulk Data Editor** used to manage different set of data for selected functionality. In Bulk Data editor column names will be Key names from Local Data Sets



Bulk Test Data Editor...

Select Project: sample   Select Cycle: c1   Select Senario: s1   Function: f1

Avilable Bulk Data:

TSOUser	UserId	Logoff	ValAbc	pwd	Exit

Close

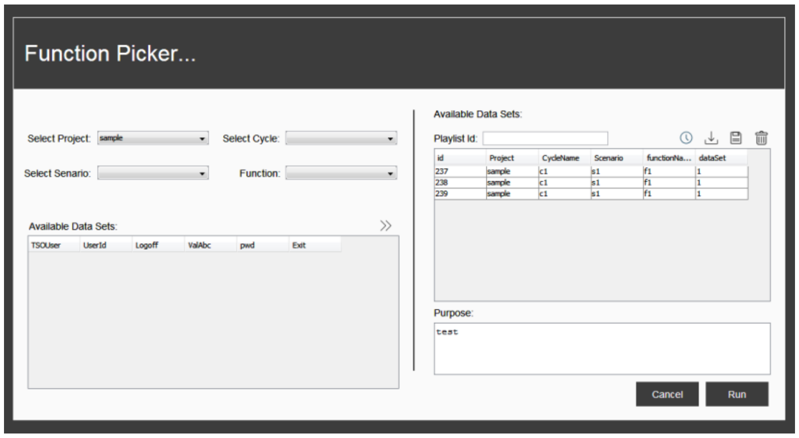
      ii. **Add to Global data** help you to easily copy selected Local Data rows to Global Data sets.

   b. **Global Data** which is applicable for current Project. Managing Global Data set is similar to Local Data sets. You can follow above instructions.

2. **Field Mapping** help you manage and map Mainframe field with user-friendly names. In iMFAUT mainframe field is referred as Field_1, Field_10 etc. Field Mapping help you to give more user-friendly names such as UserName, Command etc. which help to improve readability and understandability of function steps.

Inline Log Messages. Provide inline logs. You can view expand log view by clicking on Line Log message panel.

Function Picker. Help you to create Play list for execution. Which can be used for scheduling execution. Emulator Play Button refer to current play list. So if no play list is available currently iMFAUT automatically launch function picker to select Functionality to run. To create play list, follow below steps.


Function Picker...

Select Project: sample
Select Cycle:
Select Senario:
Function:

Available Data Sets:

TSOUser	UserId	Logoff	ValAbc	pwd	Exit

Available Data Sets:

Playlist Id:

id	Project	CycleName	Scenario	functionNa...	dataSet
237	sample	c1	s1	f1	1
238	sample	c1	s1	f1	1
239	sample	c1	s1	f1	1

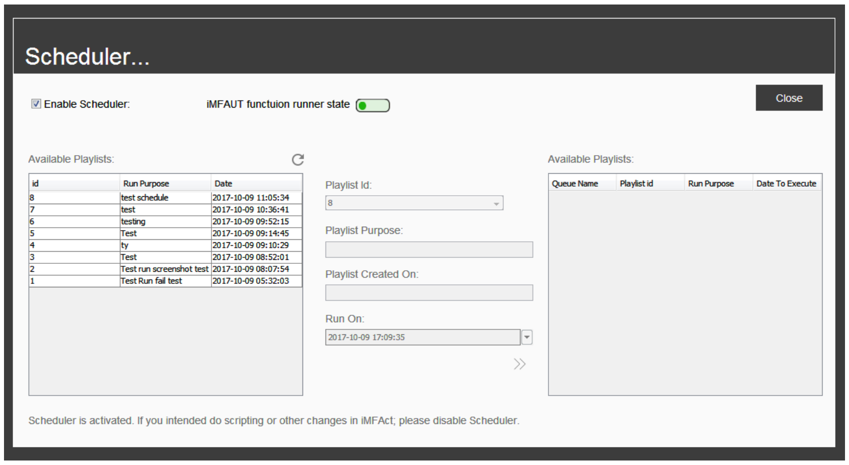
Purpose:
test

1) Navigate to Run → Function Selector
2) Select Project → Select Cycle → Select Scenario → Select Function To get available data sets for selected Function
3) Select sets of data applicable for run from Available Data Sets table
4) Click on >>
5) Selected Information will have added to right side data table with data set id
6) Provide optional purpose (if play list is intended for scheduling purpose is mandatory).
7) Click on Save
8) If you are creating new play list, right side data table id will be changed to non -1 after saving.
9) From here you can take two directions.
    1. You can click on Run; In Emulator it will started executing selected playlist functionalities if mainframe is connected else it will show message.

2. Or you can close Function Picker window and you can select play list for scheduling which we will explain in Scheduler selection.

Scheduler. Help you Schedule execution. To Schedule execution follow below steps.

1) Navigate to View → Function Scheduler
2) Select Play list from Available Playlist table
3) Set Date and Time by Clicking and Selecting from Date and Time picker
4) Click on >>
5) To enable scheduler, select Enable Scheduler check box. Note: - If you want do some changes in iMFAUT please disable scheduler. For schedule to work iMFAUT should be opened. During schedule running if mainframe is not connected it will connect automatically.



[✓] Enable Scheduler:

iMFAUT functuion runner state

Available Playlists:

id	Run Purpose	Date
8	test schedule	2017-10-09 11:05:34
7	test	2017-10-09 10:36:41
6	testing	2017-10-09 09:52:15
5	Test	2017-10-09 09:14:45
4	ty	2017-10-09 09:10:29
3	Test	2017-10-09 08:52:01
2	Test run screenshot test	2017-10-09 08:07:54
1	Test Run fail test	2017-10-09 05:32:03

Playlist Id: 8

Playlist Purpose:

Playlist Created On:

Run On: 2017-10-09 17:09:35

Available Playlists:

Queue Name	Playlist id	Run Purpose	Date To Execute

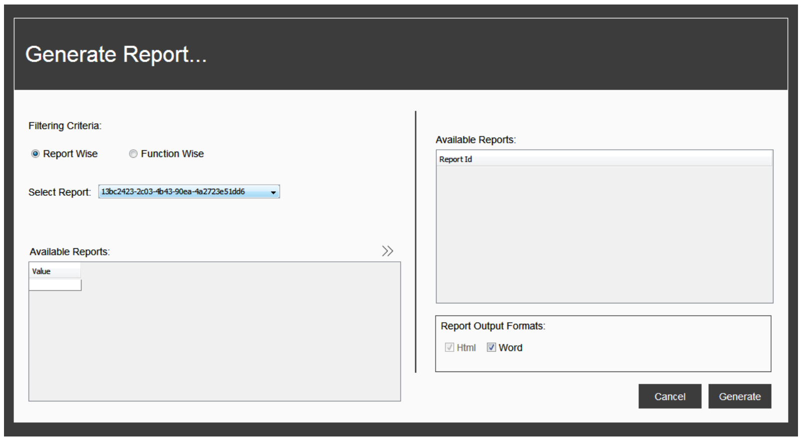
Scheduler is activated. If you intended do scripting or other changes in iMFAct; please disable Scheduler.

iMFAUT function runner state icon is used to indicate current state of runner and in case of error you can reset runner state. When a function executed and it ended in normal error exception scenario; scheduler will indicate state as **Orange** color and stop scheduling monitoring to prevent further error which could lead subsequent functions to fail. But in Business Exception with error skipping it will continue as pass functions. When reset runner state in error you need to click on runner state switch to re-reinitialize runner and re-enable scheduler. If you running directly from **iMFAUT** emulator, then you could avoid above steps since you are in charge of execution. But you can use above steps to re-initialize runner anytime.

Note: - If Scheduler is enabled then all activities of **iMFAUT** will be disabled.

Generate Report. Help you Schedule execution. To generate report of already executed playlist follow below steps.

1) Navigate to View → Generate Report
2) You can view Available Reports by Report Wise / Function Wise (Function Wise selection is similar to other windows).
3) Once selected it will show applicable reports
4) Select reports and click >> to add to right side Available Report table
5) By default, HTML formatted report get generated, you can opt for additional word format along with HTML report
6) Click on Generate button. Report will get generated and automatically launch in respective applications.


Generate Report...

Filtering Criteria:

[● Report Wise] ○ Function Wise

Select Report: 13bc2423-2c03-4b43-90ea-4a2723e51dd6

Available Reports:

Value

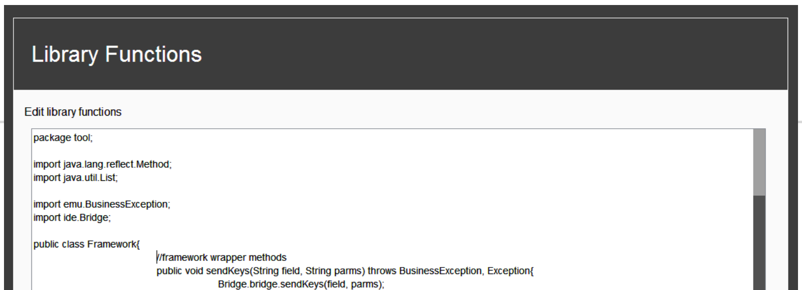
Available Reports:

Report Id

Report Output Formats:

[✓ Html] [✓ Word]

Library Functions. Help you to create your own library function using iMFAUT system defined core library and reference variable. **Library Functions** use java syntax. And you have to add all your own keyword/method to **Framework** class in **tool** package. By default, a few custom methods have been added to Framework class. **Bridge.bridge** is the reference point variable which will expose all available methods to create custom methods. All iMFAUT core library function required two parameters to be passed. First One for identifying Field information's and second one for other parameters as comma separated string which typically pass from steps as {parms1, parms2, n}. Once you have updated and saved library function please Restart iMFAUT to changes take effect.


Library Functions

Edit library functions

```
package tool;

import java.lang.reflect.Method;
import java.util.List;

import emu.BusinessException;
import ide.Bridge;

public class Framework{
                        //framework wrapper methods
                        public void sendKeys(String field, String parms) throws BusinessException, Exception{
                                    Bridge.bridge.sendKeys(field, parms);
```

iMFAUT Core Library Functions. List of core Library Function and its Usage

1) sendKeys
    a. **Input Prams:** Field_id,Value
    b. **Return Type:** void
    c. **Syntax:** Bridge.bridge.sendKeys(field, parms);
    d. **Special Case:**
    e. **Description:** Used to send keys to mainframe input fields by providing field id and value
    f. **Example:** sendKeys {User} in Field_12
    g. **Type:** Any Where

2) sendValuesAfterLabelText
    a. **Input Prams:** Mainframe Label Text, Number of Fields to Skip, Value}
    b. **Return Type:** void
    c. **Syntax:** Bridge.bridge.sendValuesAfterLabelText(field, parms);

d. **Special Case:**

e. **Description:** Used to send keys to mainframe input fields by providing nearby left label text, number field to skip to locate input field and value to be entered

f. **Example:** sendKeysAfterLabel {command,1, hello world}

g. **Type:** Any Where

3) pressEnter

a. **Input Prams:** null

b. **Return Type:** void

c. **Syntax:** Bridge.bridge.pressEnter(field, parms);

d. **Special Case:**

e. **Description:** Used to press enter key

f. **Example:** pressEnter {null}

g. **Type:** Any Where

4) pressTab

a. **Input Prams:** null

b. **Return Type:** void

c. **Syntax:** Bridge.bridge.pressTab(field, parms);

d. **Special Case:**

e. **Description:** Used to press tab key

f. **Example:** pressTab {null}

g. **Type:** Any Where

5) pressFunctionKey

a. **Input Prams:** Function Key Value in Number

b. **Return Type:** void

c. **Syntax:** Bridge.bridge.pressFunctionKey(field, parms);

d. **Special Case:**

e. **Description:** Used to press function keys i.e.: - F3, F5, F11 etc.

f. **Example:** pressFunctionKey {3}

g. **Type:** Any Where

6) checkValue

a. **Input Prams:** Field_id, Value

b. **Return Type:** void

c. **Syntax:** Bridge.bridge.checkValue(field, parms);

d. **Special Case:** On not matching cases or an error situation it will throw Exception

e. **Description:** Used to check value provided is matching in mainframe field. If not matching it will throw an Exception

f. **Example:** checkValue {ZOS-V1R13} in Field_2

g. **Type:** Any Where

7) containsValue

a. **Input Prams:** Field_id, Value

b. **Return Type:** void

c. **Syntax:** Bridge.bridge.containsValue(field, parms);

d. **Special Case:** On not matching cases or an error situation it will throw Exception

e. **Description:** Used to check value provided is contains in mainframe field. If not matching it will throw an Exception

f. **Example:** containsValue {ZOS-V1R13} in Field_2

g. **Type:** Any Where

8) getEmulatorFieldValue

a. **Input Prams:** Field_id

b. **Return Type:** String

c. **Syntax:** Bridge.bridge.getEmulatorFieldValue(field, parms)

d. **Special Case:**

e. **Description:** Get value of mainframe field

f. **Example:** Bridge.bridge.getEmulatorFieldValue(field, null)

g. **Type:** Only inside Framework class

9) isWritableField

a. **Input Prams:** Field_id

b. **Return Type:** boolean

c. **Syntax:** Bridge.bridge.isWritableField(field, parms)

d. **Special Case:**

e. **Description:** Check mainframe field is writable

f. **Example:** Bridge.bridge.isWritableField(field, null)

g. **Type:** Only inside Framework class

10) isInputField

a. **Input Prams:** Field_id

b. **Return Type:** boolean

c. **Syntax:** Bridge.bridge.isInputField(field, parms)

d. **Special Case:**

e. **Description:** Check mainframe field is input type

f. **Example:** Bridge.bridge.isInputField(field, null)

g. **Type:** Only inside Framework class

11) getScreenText

a. **Input Prams:** null

b. **Return Type:** String

c. **Syntax:** Bridge.bridge.getScreenText(field, parms)

d. **Special Case:**

e. **Description:** get full text of mainframe emulator

f. **Example:** Bridge.bridge.getScreenText(null, null)

g. **Type:** Only inside Framework class

12) waitUntil

a. **Input Prams:** Numeric String in Seconds

b. **Return Type:** void

c. **Syntax:** Bridge.bridge.waitUntil(field, parms);

d. **Special Case:** value should be a numeric type

e. **Description:** pause execution for given time in second

f. **Example:** wait {10}

g. **Type:** Any Where

iMFAUT Exceptions.

iMFAUT support normal Exception and BusinessException to handle error scenarios during run time.

1) Exception is normal exception which happen because of some un-expected condition occurred during execution. iMFAUT will terminate execution of playlist in case of **Exception**

2) BusinessException is a custom exception useful for business validation in which you will have ability to avoid termination of execution by specifying '**and skip on error.**' In function step and you can throw custom exception instead of exception so that iMFAUT will log Business Exception occurred in report and continue execution. If '**and skip on error.**' Is not provided in step then iMFAUT will treat Business Exception as normal Exception and terminate execution.

   Example: -

```
try{
        Bridge.bridge. containsValue(field, parms);
   }
catch (Exception exception) {
        throw new BusinessException ("Values not matching", 505)
  }
```