

Development of Call Forms
Oracle FLEXCUBE Investor Servicing
Release 14.7.0.1.0
Part No. F71715-01
[May] [2023]



Table of Contents

1. Preface	3
1.1 Audience	3
1.2 Related Documents	3
2. Introduction	4
2.1 How to use this Guide	4
3. Overview of Call Form	5
4. Screen Development	6
4.1 Header Information	6
4.2 Preferences	7
4.3 Data Sources	8
4.4 Data Blocks	9
4.5 Screens	11
4.6 Field Sets	13
4.7 Actions	14
4.8 Launch Forms	15
4.9 Call Forms	15
4.10 Summary	15
4.11 Preview	15
5. Attaching Call Form to Main Function Id	16
6. Generated Units	17
6.1 Front End Units	17
6.1.1 Language XML	17
6.1.2 SYS JavaScript File	17
6.1.3 Release Type Specific JavaScript File	17
6.2 Data Base Units	17
6.2.1 Static Scripts	17
6.2.2 System Packages	18
6.2.3 Hook Packages	18
6.3 Other Units	18
6.3.1 Xsd	18
7. Extensible Development	19
7.1 Extensibility in JavaScript Coding	19
7.2 Extensibility in Backend Coding	19

1. Preface

This document describes the features of a Call Form screen in FLEXCUBE and the process of designing a Call form screen using Oracle FLEXCUBE Development Workbench for Universal Banking.

1.1 Audience

This document is intended for FLEXCUBE Application developers/users that use Development Workbench to develop various FLEXCUBE components.

To Use this manual, you need conceptual and working knowledge of the below:

<i>Proficiency</i>	<i>Resources</i>
FLEXCUBE Functional Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Technical Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Screen Development	<i>04-Development_WorkBench_Screen_Development-I.docx</i>
Working knowledge of Web based applications	Self-Acquired
Working knowledge of Oracle Database	Oracle Documentations
Working knowledge of PLSQL & SQL Language	Self-Acquired
Working knowledge of XML files	Self-Acquired

1.2 Related Documents

[*04-Development_WorkBench_Screen_Development-I.pdf*](#)

[*05-Development_WorkBench_Screen_Development-II.pdf*](#)

[*14-Development_of_Online_Forms.pdf*](#)

2. Introduction

2.1 How to use this Guide

The information in this document includes:

- [Chapter 2 , "Introduction"](#)
- [Chapter 3 , "Overview of Call Form"](#)
- [Chapter 4 , "Screen Development"](#)
- [Chapter 5 , "Generated Units"](#)
- [Chapter 5 , "Extensible Development"](#)

3. Overview of Call Form

Call Forms are function Id's (screens) which can be used for processing of a feature which is common across multiple function Ids.

Call Forms can be attached to the main function Id for processing the common functionality. Call form screens cannot be launched independently.

*Example: **Tax Processing for a Contract***

Tax Processing depends on common tax rules attached for the product/contract. Same processing can be used for various contract screens like Funds Transfer Input Screen, Letters Of Credit etc.

Thus a common function id can be developed which can be attached to all the contract screens requiring tax processing

On launching the call form screen from the main screen, the values will be picked up based on the data input in main screen. User will have the option to change the data in call form screen if desires so.

There are two types of Call forms

- 1) Maintenance Call Forms
- 2) Transaction Call Forms

Maintenance Call forms can be attached to only maintenance function id's while transaction call forms can be attached to transaction screens only.

4. Screen Development

Design and development of a Call Form function id is similar to any other function ids. This section briefs the steps in designing a Call Form screen.

For detailed explanation, refer the document: **04-Development_WorkBench_Screen_Development-1.docx**

4.1 Header Information

Provide the header information as shown in the figure.

ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox

whf001qw:8088/FCJRADTool142GA/RADLoginServlet#nogo

ORACLE FLEXCUBE Development Workbench for Universal Banking

Admin

Windows Options Sign Out

Function Generation

Action: Load

Function Id: UTCCIFND

Function Type: Parent

Function Category: Maintenance

Header Template: None

Footer Template: None

Save XML Path: FND_RAD.xml

BROWSE

Parent Function

Parent Xml

Search

Preferences

Head Office Function

Logging Required

Auto Authorization

Module Auto Authorization

Tank Modifications

Field Log Required

Multi Branch Access

Excel Export Required

Java Functions

GateWay Screen

Module: UT

Module Description: UT

Module Group

Branch Program Id

Process Code

SVN Repository URL

Fig: Call Form header Information

Note the following while providing header information.

- i) Name of the Call form :
Call Form name has to have the third character as 'C'. This is how system differentiates a call form from other screens. Ideally, the length of the name should be 8 characters.
Example: UTCCIFND, UTCAPPUH etc. are valid call form names
- ii) Call Form Category:
It has to be either Maintenance or Transaction depending on the functionality and the screens from which it will be invoked
- iii) Footer Template:
Footer template can be provided as required. Note for Transaction screens, footer template has to be selected as NONE unless it is a process screen
- iv) Function Type :
Parent and child functionality is supported for call forms

4.2 Preferences

Provide the menu details in the Preferences screen

ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox

whf00iqw:8088/FCIRADTool142GA/RADLoginServlet#nogo

☆

☰

ORACLE FLEXCUBE

Development Workbench for Universal Banking

Admin

Windows

Options

Sign Out

Function Generation

Action

Load

Function Id

UTCCIFND

Save XML Path

FND_RAD.xml

BROWSE

Function Type

Parent

Parent Function

Parent Xml

Function Category

Maintenance

Header Template

None

Footer Template

None

Search

Preferences

Data Source

List Of Values

Data Blocks

Screens

Field Sets

Actions

Call Forms

Launch Forms

Summary

Menu Details

Parameter Value Mappings

☐ Head Office Function
☐ Logging Required
☐ Auto Authorization
☐ Module Auto Authorization
☐ Tank Modifications
☐ Field Log Required
☐ Multi Branch Access
☐ Excel Export Required
☐ Java Functions
☐ GateWay Screen

Module

UT

Module Description

Unit Trust

Module Group

Branch Program Id

Process Code

SVN Repository URL

Control String

+

-

	Function Id	Module *	Module Description
<input checked="" type="checkbox"/>	UTCCIFND	UT	Unit Trust

Fig: Call Form Preferences

Note the following while providing Preferences for Call Forms.

- i) Module name :
Module name is a mandatory field and has to be provided. It is recommended that the first two letters of the function id is kept as same as the module name. Naming of the generated package will be derived from the module code maintained
- ii) Of the menu details INC generated, only script for SMTB_MENU and SMTB_FCC_FCJ_MAPPING is required for Call Forms
- iii) Browser menu options :
Call Forms cannot be launched independently .Hence browser menu labels need not be maintained. **Script for smtb_function_description is not required for call forms.**

4.3 Data Sources

Identify the tables/views for the call form. Define data sources and add data source fields as required.

The screenshot shows the Oracle Flexcube Development Workbench interface. The 'Data Source Details' tab is active, displaying configuration for the 'STTM_CUSTOMER' data source. The 'Data Source' is set to 'STTM_CUSTOMER', 'Master' is 'Yes', and 'Relation Type' is 'One To Many'. 'Multi Record' is set to 'No'. The 'PK Cols' field contains 'CUSTOMER_NO', and 'PK Types' is set to 'VARCHAR2'. The 'Parent' field is empty, and 'Relation' is set to 'Query'. The 'Where Clause' and 'Default Order By' fields are empty. The 'Type' is set to 'Query', and 'Mandatory' is unchecked. The 'Upload Table' and 'Upload Where Clause' fields are empty. The 'Fine Grained Service Details Only' section is also visible, showing 'Rest Data Block', 'XSD Node', 'Rest Relation', 'Rest Relation Type', 'Rest PK Cols', and 'Rest PK Types'.

Fig: Adding data sources and maintaining properties

Note the following while creating data sources

- Master Data Source has to be a single entry data source.
- Logical Relationships has to be maintained for all data sources except the parent
- Provide PK Cols and PK types for all data sources.

If data source is a multi-record block, then make sure it has at least one more pk than its parent which helps to uniquely identify each record of multi record block.

The screenshot shows the Oracle Flexcube Development Workbench interface. The 'Data Source Field Details' tab is active, displaying configuration for the 'STTM_CUSTOMER' data source. The 'Column Name' is 'CUSTOMER_NO', 'Block Name' is 'BLK_DETAILS', and 'Field Name' is 'CUSTOMER_NO'. The 'Data Type' is 'VARCHAR2', and 'Max Length' is '9'. The 'Upload Table Column' field is empty, and 'Not Required in Upload Tables' is unchecked. The 'Preferences' pane on the left shows the 'Data Source' tree with 'STTM_CUSTOMER' expanded, and 'CUSTOMER_NO' selected.

Fig: Adding data sources fields and its properties

Max length of the data source field can be modified as per requirement.

4.4 Data Blocks

Determine the block structure for the function id .Define Data Blocks as per the design

The screenshot shows the Oracle Flexcube Development Workbench for Universal Banking. The 'Function Generation' tab is active, and the 'Block Properties' window is open. The 'Block Name' is 'BLK_DETAILS'. The 'Function Type' is 'Parent'. The 'Function Category' is 'Maintenance'. The 'Header Template' is 'None'. The 'Footer Template' is 'None'. The 'Action' is 'Load'. The 'Function Id' is 'UTCCIFND'. The 'Save XML Path' is 'FND_RAD.xml'. The 'Block Properties' window shows the following details:

- Block Name: BLK_DETAILS
- Block Title: (empty)
- Parent: (empty)
- Relation Type: One To One
- Block PK Fields: (empty)
- XSD Node: Details
- Comment ID: (empty)
- Master Block: Yes
- Multi Record: No
- Block Type: Normal

The 'Datasource Available' and 'Datasource Added' lists are shown below the properties. The 'Datasource Added' list contains the following entries:

- STTM_CUSTOMER
- STTM_CUST_CORPORATE
- STTM_CUST_PERSONAL
- STTM_CUST_FCISINFO
- STTM_CUSTFATCA_DETAILS

Fig: Defining Data Blocks and maintaining its properties

Note the following while creating data blocks

- Master Data Source has to be a single entry data source.
- Logical Relationships with the parent has to be maintained for all data sources.
If data source is a multi-record block, then make sure it has at least one more pk than its parent which helps to uniquely identify each record of multi record block
- Provide PK Cols and PK types for all data sources.
- Provide Xsd node name if the block is normal and is required in gateway request

Add block fields to the data block as required.

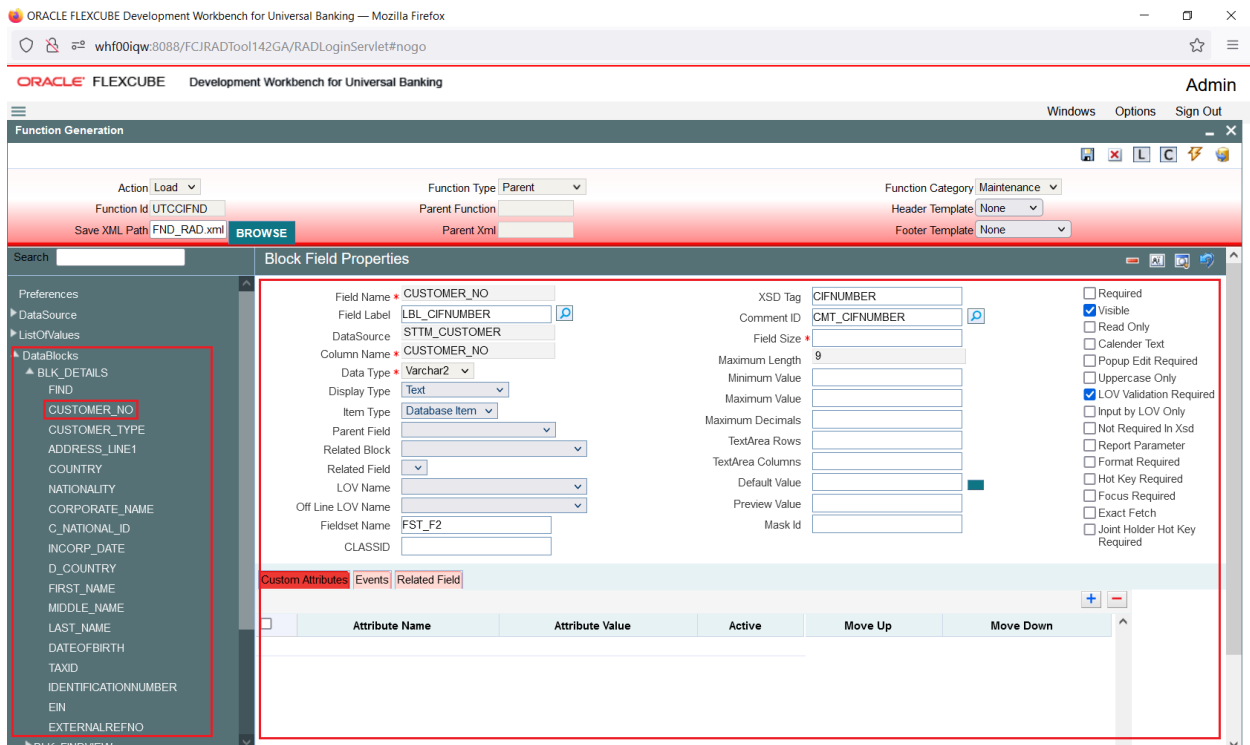


Fig: Attaching Block Fields and maintaining its properties

Note the following while attaching block fields to data blocks

- i) In case the field is not required in XSD, check not Required XSD
- ii) Ensure that Related Block and Field are given for Amount Fields
- iii) Minimize the use of query data sources by using DESC fields wherever possible.

Note: Query data sources is rarely required for a Call Form screen; as launch form can be used for query only screens

4.5 Screens

Design the screen layout based on the requirement

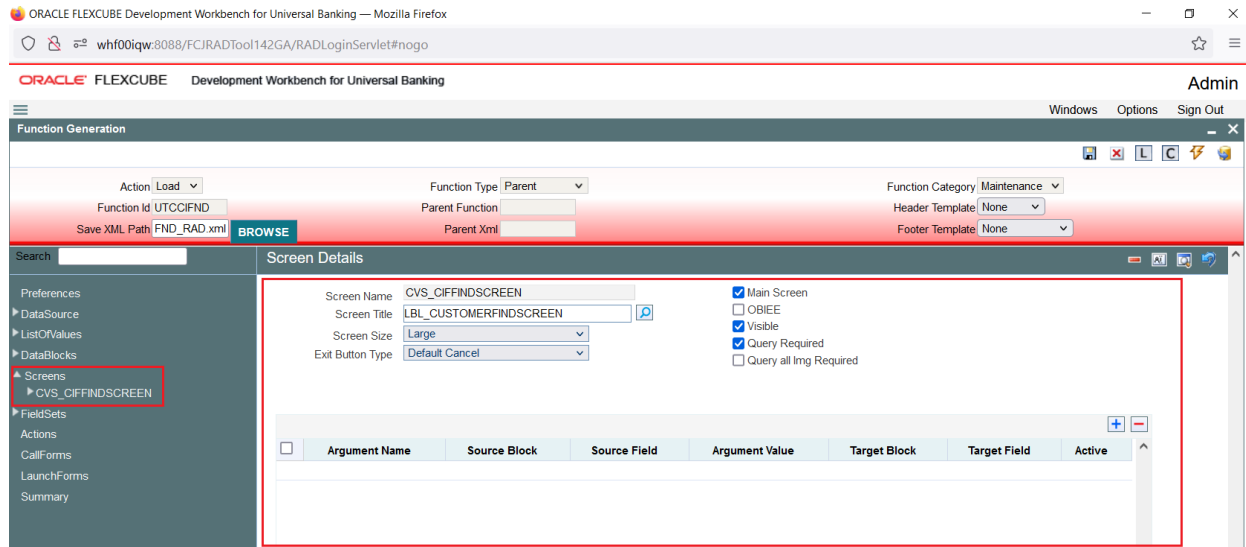


Fig: Designing Screens and providing Screen Properties

Note the following while creating screens

- i) One Screen should be identified as the main screen; if multiple screens present
- ii) In the function id ,where the call form is called :
For the button (which launches call form) events, the main screen of the call form has to be mentioned
- iii) **Screen Arguments :**
Screen Arguments has to be provided for the main screen. Any field which has to be populated based on the data from the calling Function id can be provided as the target block and target field.
Normally values for the pk fields of the master data source can be retrieved from the screen arguments .Relationship between the calling function id and the call form will also be based on the pk columns of master data source.

Add Tabs, sections and partitions as per the screen design

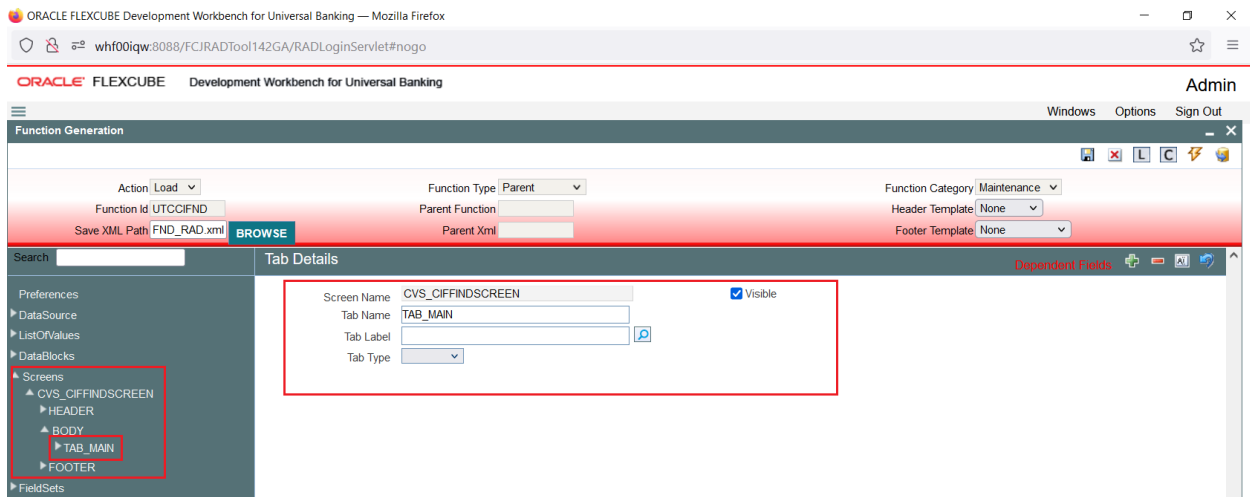


Fig: Creating Tabs and maintaining Properties

Note the following when creating tabs and sections for the screen

- i) If the screen does not have multiple tabs, then only the TAB_MAIN needs to be used. TAB_HEADER should not contain any sections in this scenario

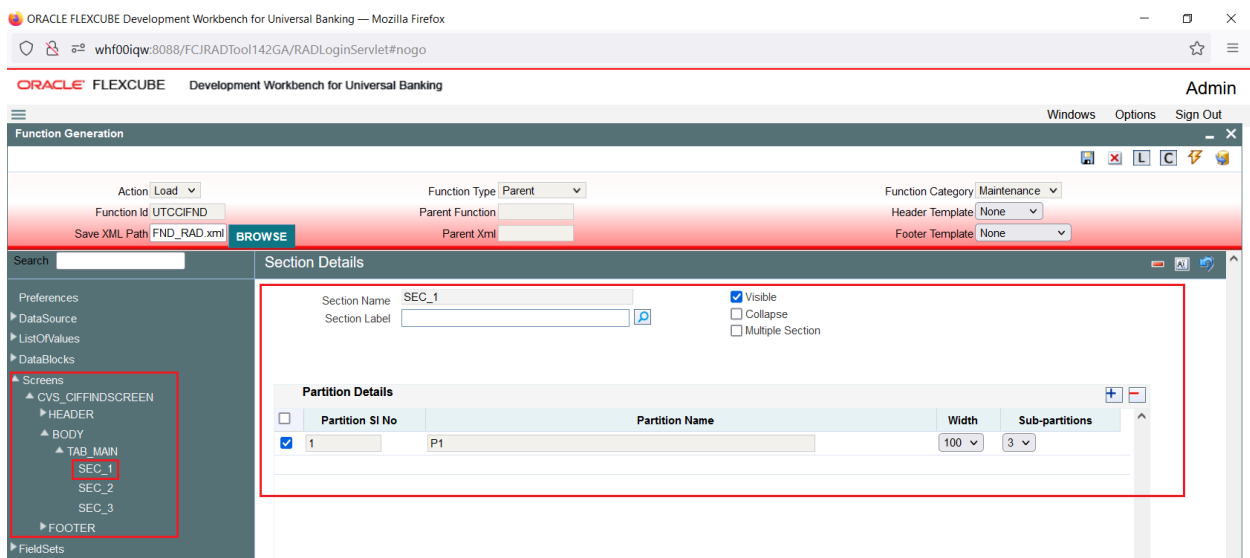


Fig: Section Properties

Multiple Screens can be designed if required.

4.6 Field Sets

Create Field sets and attach the fields to the field sets as required

ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox

whf00iqw:8088/FCJRADTool142GA/RADLoginServlet#nogo

ORACLE FLEXCUBE Development Workbench for Universal Banking

Admin

Windows Options Sign Out

Function Generation

Action Load Function Type Parent Function Category Maintenance

Function Id UTCCIFND Header Template None

Save XML Path FND_RAD.xml BROWSE Parent Xml Footer Template None

Search

Fieldset Properties

Fieldset Name FST_F2 Fieldset Type Normal

Fieldset Label LBL_QUICK_FIND_OPTIONS Screen Name CYS_CIFFINDSCREEN

Data Block BLK_DETAILS Screen Portion Body

Multi Record Yes Tab Name TAB_MAIN

View Type Single Section Name SEC_1

Fieldset Height Partition Name P1

Number Of Rows

Horizontal Fieldset

ReadOnly

Navigation Button

Navigation Button Full Width

Visible

Data Block Fields

Fieldset Fields	Subpartition Name
CUSTOMER_NO	1
EXTERNALREFNO	1
DATEOFBIRTH	1
INCORP_DATE	1
D_COUNTRY	1
FIRST_NAME	2
LAST_NAME	2
MIDDLE_NAME	2

Fig: Field Set Properties

Note the following when attaching field to a field set

- If a field value is passed as screen argument ,but is not required to be shown in the screen, The field has to be made invisible and attached to a field set.**

If it is not attached to any fields set, the screen html won't contain the field and may result in script error while loading.

4.7 Actions

Mention the web service and amendable information in Actions Screen.

ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox

whf001qw:8088/FCJRADTool142GA/RADLoginServlet#nogo

ORACLE FLEXCUBE Development Workbench for Universal Banking Admin

Windows Options Sign Out

Function Generation

Action Load Function Id UTCCIFND Save XML Path FND_RAD.xml BROWSE Function Type Parent Parent Function Parent Xml Function Category Maintenance Header Template None Footer Template None

Search

Form Actions

XSD Type Identifier Operation Id Rest Enabled Fine Grained Service Required Service Name UTCCIFND XSD Module Folder Rest Service Name

Web Service	Action Code	Operation Code	Action Stage Type	Rest Enabled	Amendables	Comment Coc
<input type="checkbox"/>	QUERY		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	NEW		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	MODIFY		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	AUTHORIZE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	DELETE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	CLOSE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	REOPEN		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	REVERSE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	ROLLOVER		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	CONFIRM		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	LIQUIDATE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments
<input type="checkbox"/>	SUMMARYQUERY		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amendables	Comments

Fig: Actions Screen

Note the following while maintaining web services and amendable information

- Call forms will generate only Type XSD.
Operation specific message xsd's will not be generated. Call form Type will be part of the main function Id xsd; hence separate message xsd is not required for call form
'Subsys' will be added to the name of call form type xsd.
Example: for the example given in the figure, name of the xsd generated will be SubSys-TxnChgDtls-Types.xsd
- Operation Id and Operation Code need not be maintained for the above mentioned reason
- Amendable information has to be maintained similar to any other function id's.

4.8 Launch Forms

Launch Forms can be attached to a Call form screen. Though it is technically supported, practical scenarios where launch form is part of a call form is very rare.

Process to attach launch forms is similar to any other function Id's.

4.9 Call Forms

Call forms can themselves be attached to a call form. This scenario also is practically very rarely used. Processing logic (sub system pickup) for the attached call forms has to be called from the main call form.

4.10 Summary

Summary screens are not required for Call Form screens. Since a Call Form screen cannot be launched independently in FLEXCUBE, it doesn't require a summary screen.

4.11 Preview

The figure shows the preview of the call form screen developed.

Customer Find Screen

Quick find Options

CIF Number First name Investor Type

External Reference Last Name Tax ID

Number Middle Name National ID

Date Of Birth Company Name IdentificationNumber

Incorporation Date Nationality

Country Of Domicile

Advanced Find Options

Find Results

1 of 1

<input type="checkbox"/>	Investor Type	First name	Last Name	Middle Name	Company Name	Customer No.	Tax I
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Fig: Call Form Screen Preview

Generate the units for call form and deploy them in the FLEXCUBE server for unit testing

5. Attaching Call Form to Main Function Id

Call Forms cannot be launched independently. It has to be called from a main function id.

Refer *Call Forms* section in *04-Development_WorkBench_Screen_Development-I.docx* for detailed explanation.

Note: *Scripts for CSTB_CALL_FORM_NODES and SMTB_MENU tables generated by Call Form screen has to be deployed in FLEXCUBE schema before attaching Call form to the main function Id.*

6. Generated Units

The following units will be generated for a Call Form screen. Refer document on generated units on detailed explanation on the same.

6.1 Front End Units

6.1.1 Language XML

This file is an XML markup of presentation details, for the designed Call Form specific to a language.

6.1.2 SYS JavaScript File

This JavaScript file mainly contains a list of declared variables required for the functioning of the screen.

6.1.3 Release Type Specific JavaScript File

This file won't be generated by the Tool. It has to be manually written by the developer if he has to write any code specific in that release.

6.2 Data Base Units

6.2.1 Static Scripts

The following static scripts generated are required for the proper functioning of a Call Form screen. Refer document on generated units for detailed explanation

- i) **Menu Details**
Scripts for SMTB_MENU and SMTB_FCC_FCJ_MAPPING are required for the functioning of Call Form screen
- ii) **Call Form details**
Script for CSTB_CALL_FORM_NODES is required for attaching the call forms to the main function id. This has to be compiled in the schema before attaching the Call form to the main function Id
- iii) **Lov Details**
- iv) **Amendable Details**
- v) **Label details**
- vi) **Screen Details**
- vii) **Block details**
- viii) **Data Source Details**

6.2.2 System Packages

Main package would be generated by the Tool and should not be modified by the developer. There is small change in the structure of the package depending on the type of the call form (Maintenance or Transaction).

Unlike normal maintenance function ids, call form packages does not have any call to the business logic within itself (similar to transaction function id). If developer wishes to uses any functions within the main package, call has to be made from the release specific package.

Main package contains functions for:

- Converting Ts to PL/SQL Composite Type
- Calling fn_main.
- Mandatory checks (fn_check_mandatory).
- Default and validation(fn_default_and_validate)
- Querying(fn_query)
- Converting the Modified Composite Type again to TS

Except the functions for type conversions, others functions calls the respective hook functions in hook packages of the call forms. Thus no processing logic within the main package is used

It is to be noted that each of these functions are called from the main package of the main function id (where this call form is used) during respective stages.

But the package contains many other system generated functions for operations like

- Mandatory checks(fn_sys_check_mandatory)
- Default and validation(fn_sys_default_and_validate)
- Uploading to DB(fn_sys_upload_db)
- Query operation (fn_sys_query) etc.

These functions are not called anywhere in the package. These functions if required can be called by the developer from the release specific package. Otherwise developer can write his own logic for the same in the Hook Packages.

6.2.3 Hook Packages

Release specific packages will be generated based on the release type (KERNEL, CLUSTER or CUSTOM). The structure of the package depends on the type of call form (Maintenance or Transaction). Developer can add his code in the release specific hook package.

6.3 Other Units

6.3.1 Xsd

Only Type XSD will be generated for a Call Form function Id. Subscript *Subys* will be added before XSD Type identifier in the name of the generated xsd.

This type xsd will be used in the type xsd of any function which uses the particular call form.

7. Extensible Development

Developer can add his code in hook packages and release specific JavaScript file.

7.1 Extensibility in JavaScript Coding

For release specific JavaScript coding, code has to be written in release specific JavaScript file. It follows the naming convention as: (Function Id)_(Release Type).js

Example: Code in *UTCCIFND_CLUSTER.js* is exclusive to cluster release

This JavaScript file allows developer to add functional code and is specific to release. The functions in this file are generally triggered by screen events. A developer working in cluster release would add functions based on two categories:

- Functions triggered by screen loading events
Example: fnPreLoad_CLUSTER(), fnPostLoad_CLUSTER()
- Functions triggered by screen action events
Example: fnPreNew_CLUSTER (), fnPostNew_CLUSTER ()

7.2 Extensibility in Backend Coding

Release specific code has to be written in the Hook Packages generated. Structure of a Maintenance and Transaction Call Form hook packages are almost the same

Note: Though structure is almost the same, arguments differ in transaction and maintenance call forms .Hence Transaction Call Form can be attached only with Transaction screen and similarly for Maintenance screens

Different functions available in the Hook Package of a Call Form are:

1) Skip Handler : Pr_Skip_Handler

This can be used to skip the logic written in another release.

Example: logic written in *KERNEL* release can be skipped in *CLUSTER* release

2) Fn Main

This is called from the fn_main in main package.

3) Fn_pre_query

4) Fn_post_query

Any specific logic while querying can be written in these functions. It is called from fn_query of the main package

5) Fn_pre_upload_db

6) Fn_post_upload_db

Any logic while uploading data to tables can be written here.

7) Fn_pre_default_and_validate

8) Fn_post_default_and_validate

Any release specific logic for defaulting and validation can be written here. It is called from the fn_default_and_validate in the main package

9) Fn_pre_check_mandatory

10) Fn_post_check_mandatory

Any mandatory checks can be validated here

11) Fn_pre_process

12) Fn_post_process

These hook functions are specific to transaction call form screens. These are called from fn_process of the main package which in turn is called from fn_process of the calling function id

Refer maintenance and Transaction Screen development document for further explanation



Development of Call Forms
[May] [2023]
Version 14.7.0.1.0

Oracle Financial Services Software Limited
Oracle Park
Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India

Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax: +91 22 6718 3001
www.oracle.com/financialservices/

Copyright © [2007], [2023], Oracle and/or its affiliates.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.