Development of Call Forms Oracle FLEXCUBE Investor Servicing Release 14.7.2.0.0 Part No. F75930-01 [May] [2023]





Table of Contents

1. Prefa	асе	.3
1.1	Audience	
1.2	Related Documents	
	duction	
2.1	How to use this Guide	
	view of Call Form	
	en Development	
4.1	Header Information	
4.2	Preferences	
4.3	Data Sources	
4.4	Data Blocks	
4.5	Screens1	
4.6	Field Sets	-
4.7	Actions	
4.8	Launch Forms	
4.9	Call Forms1	
4.10	Summary1	
4.11	Preview	
	ching Call Form to Main Function Id1	
6. Gene	erated Units	
6.1	Front End Units	
0.1.1		
6.1.2	2 SYS JavaScript File1	7
6.1.3	B Release Type Specific JavaScript File	7
6.2	Data Base Units1	7
6.2.1		
6.2.2	2 System Packages1	8
6.2.3	B Hook Packages	8
6.3	Other Units	8
6.3.1		
	nsible Development	
7.1	Extensibility in JavaScript Coding	
7.2	Extensibility in Backend Coding	.9

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:

Proficiency	Resources
FLEXCUBE Functional Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Technical Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Screen Development	04-Development_WorkBench _Screen_Development-I.docx
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

<u>14-Development_of_Online_Forms.pdf</u>

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-I.docx**

4.1 Header Information

Provide the header information as shown in the figure.

i ORAG	CLE FLEXCUBE Development Workbench	for Universal Banking — Mozilla Firefox			-	٥	×
0 8	or whf00iqw:8088/FCJRADTool	142GA/RADLoginServlet#nogo				☆	≡
ORA		nt Workbench for Universal Banking					min
Functio	on Generation			Windows		Sign O - C 7	ut X
Search	Action Load V Function Id UTCCIFND Save XML Path FND_RAD.xml BR	Function Type Parent v Parent Function OWSE Parent Xml Preferences		Function Category Maintenance Header Template None Footer Template None			
Prefere DataSo ListOfV DataBl Screen FieldSe Actions CallFor Launct	ource /alues locks 18 ets 5	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 Module Description Module Group Branch Program Id Process Code SVN Repository URL	עד [0] עד [0] [0]			

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 Footer Template: iii) 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 Workbence	-	- • ×
O A ≈ whf00iqw:8088/FCJRADTo	ol142GA/RADLoginServlet#nogo	☆ =
ORACLE FLEXCUBE Developm	ent Workbench for Universal Banking	Admin
=		Windows Options Sign Out
Function Generation		_ ×
		🖫 🗵 C 7 🤤
Action Load 🗸	Function Type Parent V	Function Category Maintenance V
Function Id UTCCIFND	Parent Function	Header Template None 🗸
Save XML Path FND_RAD.xml	Parent Xml	Footer Template None 🗸
Search	Preferences	🔊 🖆
Preferences > DataSource > LisiON/alues > DataBlocks > Screens > FrieldSets Actions CallForms LaunchForms Summary	Head Office Function Logging Required Auto Authorization Module Auto Auto functions Field Log Required Multi Branch Access Excel Export Required Java Functions GateWay Screen	Module UT Module Description Unit Trust Module Group Image: Constraint of the second s
	Function Id UTCCIFND UT	Module * Module Description Image: Control String Im

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
- 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.

CRACLE FLEXCUBE Development Workbench f	for Universal Banking — Mozil	la Firefox					-	
O 🖄 ≅ whf00iqw:8088/FCJRADTool	142GA/RADLoginServlet#	nogo						\$
ORACLE FLEXCUBE Development	nt Workbench for Universal	Banking						Admir
=						Windows	Options	Sign Out
Function Generation								_ >
						-	×L	C 7 🧃
Action Load ~		Function Type Parent Parent Function	×		Function Category Mainter Header Template None	nance V		
Save XML Path FND_RAD.xml BR	OWEE	Parent Xml	-		Footer Template None			
Search	Data Source Details	T GIOR / GIN			r ootor rompiato			
Search	Data Source Details						ť	- 🤊
Preferences	Master Relation Type Multi Record PK Cols * PK Types *	STTM_CUSTOMER Yes v One To Many v No v CUSTOMER_NO VARCHAR2		Parent Relation Where Clause Default Order By Type		v	777	
	Upload Table Upload Where Clause							
ListOfValues DataBlocks Screens FredSets Actions CallForms LaunchForms Summary	Fine Grained Service Detail Rest Data Block XSD Node Rest Relation Rest Relation Type Rest PK Cols • Rest PK Types •	s Only						J

Fig: Adding data sources and maintaining properties

Note the following while creating data sources

- i) Master Data Source has to be a single entry data source.
- ii) Logical Relationships has to be maintained for all data sources except the parent
- iii) 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.

🔿 🗞 ᄙ whf00igw:8088/FCJRADTool142GA/RADLo	ginServlet#pogo				☆
	gillserviet#illogo				2
DRACLE FLEXCUBE Development Workbench for	or Universal Banking				Adr
			Windows	Options	Sign O
unction Generation					
			3	×L	С 🎸
Action Load 🗸	Function Type Parent V		Function Category Maintenance V		
Function Id UTCCIFND	Parent Function		Header Template None V		
Save XML Path FND_RAD.xml BROWSE	Parent Xml		Footer Template None		
earch Data Sour	ce Field Details				= 🧐
Preferences	lumn Name CUSTOMER_NO	Data Type	VARCHAR2		
	Block Name BLK_DETAILS	Max.Length	9		
	Field Name CUSTOMER_NO	Upload Table Column			
CUSTOMER_NO CUSTOMER TYPE			Not Required in Upload Tables		
ADDRESS LINE1					
COUNTRY					
NATIONALITY					
EXTERNALREFNO					

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

CRACLE FLEXCUBE Development Workbench for University	ersal Banking — Mozilla Firefox	-	
O 🖄 ≅ whf00iqw:8088/FCJRADTool142GA	/RADLoginServlet#nogo		☆ =
ORACLE' FLEXCUBE Development Work	bench for Universal Banking		Admir
=		Windows Options	s Sign Out
Function Generation			_ >
		📰 🗶 L	C 7 🧐
Action Load V	Function Type Parent V	Function Category Maintenance V	
Function Id UTCCIFND	Parent Function	Header Template None V	
Save XML Path FND_RAD.xml BROWSE	Parent Xml	Footer Template None	
Search Bloc	x Properties	ep	- 🛛 🦃
Preferences > DataSource > ListONalues > BLK_DETALS > BLK_FINDVIEW > Screens > FiedSets Actions CaliForms LaunchForms Summary	Block Name BLK_DETAILS Block Title Parent Relation Type Block PK Fields Datasource Available	XSD Node Details Comment ID Master Block Yes V Mult Record No V Block Type Normal V Datasource Added STTM_CUST_ORPORATE STTM_CUST_CORPORATE STTM_C	

Fig: Defining Data Blocks and maintaining its properties

Note the following while creating data blocks

- i) Master Data Source has to be a single entry data source.
- ii) Logical Relationships with the parent has to be maintained for all data sources.
- iii) 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
- iv) Provide Xsd node name if the block is normal and is required in gateway request

Add block fields to the data block as required.

🍯 ORACLE FLEXCUBE Development Workbend	ch for Universal Banking — Mozilla Firefox					-	٥	×
O A ≅ whf00iqw:8088/FCJRADTo	ol142GA/RADLoginServlet#nogo						☆	≡
	nent Workbench for Universal Banking				Win	dows Options	Adr Sign O	
Function Generation						🖬 🗙 L		- ×
Action Load V	Function Type Pare	ent 🗸		Function Ca	tegory Maintenance v			
Function Id UTCCIFND	Parent Function			Header Ter	mplate None V			
Save XML Path FND_RAD.xml	BROWSE Parent Xml			Footer Ter	mplate None ~			
Search	Block Field Properties					- 2	I 🗔 🦻	^
Preferences DataBlocks blickOrValues blickOrValues blickOrValues blickOrValues blickOrValues blickOrValues customEr_NO customEr_TYPE ADDRESS_LINE1 cOUNTRY NATIONALITY cORPORATE_NAME c_NATIONAL_ID INCORP_DATE	Field Name ← CUSTOMER_NO Field Label DataSource STIM_CUSTOMER Column Name ← CUSTOMER NO Data Type ← Varchar2 ↓ Display Type ← Varchar2 ↓ Text ↓ Database tem ↓ Parent Field Related Field LOV Name Off Line LOV Name Fieldset Name FST_F2 CLASSID	× × ×	XSD Tag Comment ID Field Size ¹ Maximum Length Mainmum Value Maximum Decimals TextArea Columns TextArea Columns Default Value Preview Value Mask Id	CIFNUMBER CMT_CIFNUMBER 9 1		Required Visible Read Only Calender Tex Popup Edit R Uppercase C UV Validatio Input by LOV Not Required Report Parar Format Requ Hot Key Required focus Required	equired Inly In Required Only In Xsd In Xsd neter iired uired red	đ
D_COUNTRY FIRST_NAME MIDDLE_NAME	Custom Attributes Events Related Field					+ -		
LAST_NAME DATEOFBIRTH TAXID IDENTIFICATIONNUMBER EIN EXTERNALREFNO	Attribute Name	Attribute Value	Active	Move Up	Move Down			

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 <u>Screens</u>

Design the screen layout based on the requirement

NRACLE FLEXCUBE Development Workbench for	Universal Banking — Mozilla Firefox						-	٥	\times
O A and whf00iqw:8088/FCJRADTool14	2GA/RADLoginServlet#nogo							☆	≡
ORACLE' FLEXCUBE Development	Workbench for Universal Banking							Adn	nin
=					1	Windows (Options	Sign Ou	ıt
Function Generation								-	×
						;	LC	17	9
Action Load V Function Id UTCCIFND Save XML Path FND_RAD xml BROV	Function Type Paren Parent Function WSE Parent Xml	t v		Header Te	ategory Maintenance mplate None v mplate None	v			
Search	Screen Details						- 🗵	🗔 🦃	^
Preferences DataSource ListOfValues DataBlocks Screens Screens CS_OFFINDSCREEN FieldSets Actions CallForms LaunchForms Summary	Screen Name CVS_CIFFINDSCREEN Screen Title LBL_CUSTOMERFINDSC Composition Carge Exit Button Type Argument Name Source Bloc	~	Main Screen OBIEE Visible Query Required Query Required Query all Img Re Argument Value	quired Target Block	Target Field	Active			

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

CRACLE FLEXCUBE Development Workbench	ı for Universal Banking — Mozilla Firefox		– o ×
🔿 👌 🖻 whf00iqw:8088/FCJRADToo	ol142GA/RADLoginServlet#nogo		☆ ≡
ORACLE' FLEXCUBE Development	ent Workbench for Universal Banking		Admin
=			Windows Options Sign Out
Function Generation			_ ×
			🖫 🗵 C 🎸 🤤
Action Load V Function Id UTCCIFND Save XML Path FND_RAD.xml B	Function Type Parent Parent Function ROWSE Parent Xml		Function Category Maintenance V Header Template None V Footer Template None V
Search	Tab Details		Dependent Fields 🖶 📼 🔊 🔷
Preferences > DataSource > ListOfValues > DataSlocks > Screens > CVS_OFFINDSCREEN > HEADER > BODY > TAB_MAIN > FOOTER > FieldSets	Screen Name Tab Name Tab Label Tab Type	Visible	

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

📦 ORACLE FLEXCUBE Development Workbench fo	or Universal Banking — Mozilla Firefox		– 🗆 ×
O A ≈ whf00iqw:8088/FCJRADTool1	42GA/RADLoginServlet#nogo		☆ ≡
ORACLE FLEXCUBE Developmen	t Workbench for Universal Banking		Admin
≡			Windows Options Sign Out
Function Generation			_ ×
			🖬 🗵 🕻 🎸 🍕
Action Load V Function Id UTCCIFND Save XML Path FND_RAD.xml BR0	Function Type Parent Parent Function Parent Xml Parent Xml		Function Category Maintenance Header Template None Footer Template None
Search	Section Details		🗕 🗷 🦃 <mark>^</mark>
Preferences ▶ DataSource ▶ ListOfValues ▶ DataBlocks	Section Name SEC_1 Section Label	Visible Collapse Multiple Section	
▲ Screens ▲ CVS_CIFFINDSCREEN ▶ HEADER ▲ BODY ▲ TAB_MAIN ▲ SEC_1 SEC_2 SEC_3 ▶ FOOTER ▶ FieldSts	Partition Details Partition SI No 1 P1	Partition Name	Width Sub-partitions

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

	ent Workbench for Universal	Denking								
PRACE FLEXCOBE Developme	ent workbench for Universal	Banking								Ad
Inction Generation								Wind	lows Options	Sign (
									🔚 🗙 L	
A C Lord of		E C E Devet								
Action Load V Function Id UTCCIFND		Function Type Parent	·				Function Category Mainte Header Template None			
Save XML Path FND_RAD.xml		Parent Xml					Footer Template None			
		T arent Arni					Tooler template toole			_
rch	Fieldset Properties								E	- 🔊 🕯
eferences	r icidoct Hume	FST_F2			Fieldset	Type Normal	~		Horizontal F	ieldset
	Fieldset Label	LBL_QUICK_FIND_OPTIONS			Screen N	ame CVS_CIFFIND	SCREEN V		ReadOnly	
tOfValues	Data Block		~		Screen Po		~		Navigation E	
aBlocks	Multi Record	Yes ¥			Tab N		~		Width	
	View Type	Single V			Section N		~		Visible	
eldSets	Fieldset Height				Partition N	ame P1	*			
	Number Of Rows									
tions		Data Block Fields		(_ F	ieldSet Fields	Subpartition Name	^		
IForms inchForms			^	ſ		R NO	1 ~			
nmary						-	1 *			
					DATEOFE		1 ~			
					INCORP_	DATE	1 ~			
				44			1 ~			
				(ME	2 ~			
				(LAST NA	ME	2 ~			
							2 ~	~		
			~							



Note the following when attaching field to a field set

i) 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.

> √) −0 1 (221) −2 −2 −2 −2 −2 −2 −2 −2 −2 −2 −2 −2 −2	 ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox O 경 후르 whf00iqw.8088/FCJRADTool142GA/RADLoqinServlet#noqo 								
) 🖄 📬 whf00iqw:8088/FCJR	RADTool142GA/RADLoginS	Servlet#nogo						☆	
DRACLE FLEXCUBE Dev	velopment Workbench for U	niversal Banking						Ad	
=						٧	Vindows Options	Sign C	
Function Generation									
							🖁 🗙 L	C 🞸	
Action Load V		Function Type Pa	arent v		Function Category	Maintenance	1		
Function Id UTCCIFND		Parent Function			Header Template None V				
Save XML Path FND_RAD.	xml BROWSE	Parent Xml			Footer Template	None	~		
Search	Form Actions							2	
Preferences	XSD Type Id	dentifier		Service Name UT	TCCIFND		2		
DataSource	Oper	ration Id		XSD Module Folder					
istOfValues		Rest Enabled		Rest Service Name					
			-						
		Fine Grained Servic Required	e						
			e						
Screens FieldSets			e						
Screens FieldSets Actions	Web Service		e Operation Code	Action Stage Type	Rest Enabled	Amendables	Comment Coc ^		
Screens FieldSets Kotions CallForms	Web Service	Required			Rest Enabled	Amendables Amendables	Comment Coc ^ Comments		
Screens FieldSets Actions CallForms LaunchForms		Required Action Code		Action Stage Type					
Screens TieldSets Kations CallForms aunchForms		Required Action Code QUERY		Action Stage Type		Amendables	Comments Comments		
Screens FieldSets Kotions CallForms .aunchForms		Action Code QUERY NEW		Action Stage Type		Amendables Amendables	Comments Comments		
Screens FieldSets Kotions CallForms .aunchForms		Action Code QUERY NEW MODIFY		Action Stage Type		Amendables Amendables Amendables	Comments Comments Comments		
Screens TieldSets Kations CallForms aunchForms		Action Code QUERY NEW MODIFY AUTHORIZE		Action Stage Type		Amendables Amendables Amendables Amendables	Comments Comments Comments Comments		
Screens FieldSets CallForms aunchForms		Action Code QUERY NEW MODIFY AUTHORIZE DELETE		Action Stage Type		Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments		
icreens iieldSets ctions JallForms aunchForms		Action Code QUERY NEW MODIFY AUTHORIZE DELETE CLOSE		Action Stage Type		Amendables Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments		
Screens FieldSets Kotions CallForms .aunchForms		Action Code I QUERY I NEW I MODIFY I AUTHORIZE I DELETE I CLOSE I REOPEN I		Action Stage Type		Amendables Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments Comments Comments		
Screens FieldSets Kotions CallForms .aunchForms		Action Code I QUERY I NEW I MODIFY I AUTHORIZE I DELETE I CLOSE I REOPEN I REVERSE I		Action Stage Type		Amendables Amendables Amendables Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments Comments Comments		
DataBlocks Screens FieldSets CallForms LaunchForms Summary		Action Code QUERY NEW MODIFY AUTHORIZE DELETE CLOSE REOPEN REVERSE ROLLOVER		Action Stage Type		Amendables Amendables Amendables Amendables Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments Comments Comments Comments		

Fig: Actions Screen

Note the following while maintaining web services and amendable information

- i) 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
 ii) Operation Id and Operation Code need not be maintained for the above mentioned
- ii) Operation Id and Operation Code need not be maintained for the above mentioned reason
- iii) 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 External Reference Number Date Of Birth Incorporation Date Country Of Domicile	So to Page 🕂 🗕 📰	First name Last Name Middle Name Company Name Nationality			nvestor Type Tax ID National ID ationNumber	Both Find	~	
Investor Type	First name	Last Name	Middle Name	Company Name	Custom	er No.	Tax	^
								~
								Exit

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 <u>Xsd</u>

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 r

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 form 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 14.7.2.0.0

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.