Development of Maintenance Form Oracle FLEXCUBE Universal Banking Release 14.7.2.0.0 Part No. F87755-01 [Novemeber] [2023]

FINANCIAL SERVICES

Contents

1.1 Audience 1.2 Related Documents 2. Introduction 2.1 How to use this Guide 3. Overview of Maintenance Screen 4. Screen Development 4.1 Header Information 4.2 Preferences 4.3 Data Sources 4.4 Data Blocks	2
 Introduction	.3
2.1 How to use this Guide	.3
 3. Overview of Maintenance Screen	
4. Screen Development	
 4.1 Header Information	
 4.2 Preferences 4.3 Data Sources 	
4.3 Data Sources	.4
4.4 Data Blocks	
4.5 Screens	14
4.6 Field Sets	
4.7 LOV	
4.8 Attaching Call forms	
4.9 Adding Summary	
4.10 Amendable fields Maintenance	
5. Generation and Deployment of files	
6. Generated Units	
6.1 Front End Units	
6.1.1 Language xml	23
6.1.2 SYS JavaScript File	23
6.1.3 Release Type Specific JavaScript File	23
6.2 Data Base Units	23
6.2.1 Static Scripts	
6.2.2 System Packages	
6.2.3 Hook Packages	
6.3 Other Units	
6.3.1 Xsd	-
7. Extensible Development	25
7.1 Extensibility in JavaScript Coding	
7.2 Extensibility in Backend Coding	
7.2.1 Functions in Hook Packages	
7.2.2 Flow of control through Hook packages	26
7.2.3 By passing Base Release Functionality	27

1. Preface

This document describes Maintenance Screens in FLEXCUBE and the process of designing a simple Maintenance form 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	<u>Development Workbench - Screen</u>
	<u>Development I</u>
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

<u>Oracle FLEXCUBE Enterprise Limits and Collateral Management ODT Screen</u> Development Development Workbench - Screen Development II

2. Introduction

2.1 How to use this Guide

The information in this document includes:

- <u>Chapter 2 , "Introduction"</u>
- <u>Chapter 3</u>, "Overview of Call Form"
- <u>Chapter 4 , "Screen Development"</u>
- <u>Chapter 5 , "Generated Units"</u>
- <u>Chapter 5 , "Extensible Development"</u>

3. Overview of Maintenance Screen

Maintenance Function Id's are used for storing maintenance data which are required for processing of any contracts, batches or for any other maintenance which are dependent on this

Example: Customer maintenance screen

If any customer wants to use the service of a bank, details about the customer will have to be maintained in the system .This will be maintenance data which will be required for other maintenances (creating account for the customer) as well as for transaction processing (debiting of customer account)

Business logic for a maintenance function id would be provided by the Development Workbench generated files .Most of the cases, system provided logic would be sufficient .Extra validations can be coded in the hook packages by the developer.

4. Screen Development

Design and development of a Maintenance function id is similar to any other function Ids. This section briefs the steps in designing a Maintenance screen. STDCINF is sample function id used for demonstration in this document

For detailed explanation, refer the document: <u>Oracle FLEXCUBE Enterprise Limits and</u> <u>Collateral Management ODT Screen Development</u>

4.1 Header Information

Provide the header information as shown in the figure.

Function Generation		_ >
		II 🗙 L C 77 🧃
Action None V	Function Type Parent	Function Category Maintenance 🗸
Function Id	Parent Function	Header Template None 🗸
Save XML Path	Parent Xml	Footer Template None
Search		
Preferences		
DataSource		
istOfValues		
DataBlocks		
creens		
ïeldSets		
ctions		
allForms		
aunchForms		
ummary		

Fig 12.1: Providing Header Information for Maintenance Screen

- For new screen select action As New.
- Function Type 👁 Parent
- Function Category 👁 Maintenance
- Parent Function Id 👁 None
- Parent Xml 👁 None
- Header Template 👁 None (Only for Process flow screens)
- Footer Template 👁 Maint Audit

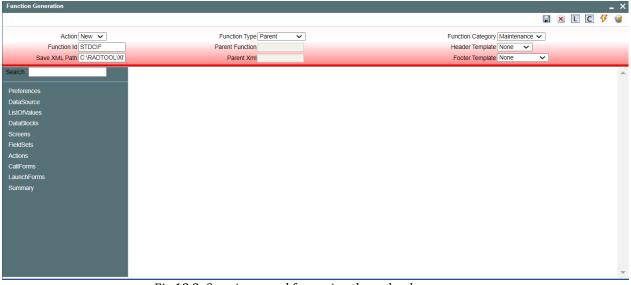


Fig 12.2: Save icon used for saving the radxml

User can save work at any point in time. Click the save icon on top right for the same .In order to work again with it select action as Load and load radxml from the hard disk path

Function Generation				XLC	×
Action New V Function Id STDCIF Save XML Path C:RADTOOLXI	Function Type[Parent Parent Function Parent Function Parent Xml		Function Category Maintenance Header Template None Footer Template None		
Search	Information				*
Preferences DataSource ListOVatues DataBlocks Screens FieldSets Actions CallinchForms LaunchForms Summary	Information Description	Information Code			
					*

Fig 12.3: Saved File Information page

Note the following while providing header information for Maintenance screen

i) Naming Convention:

The third letter of the function id has to be D. Ideally the function id name should have 8 characters.

ii) Footer Template

Make sure that the master data source has the audit columns if footer template is provided as Maint log.

Refer <u>Oracle FLEXCUBE Enterprise Limits and Collateral Management ODT Screen</u> <u>Development</u> for detailed explanation

4.2 Preferences

- Details entered in Preferences are used in generating INCS for SMTB_MENU, SMTB_FUNCTION_DESCRIPTION and SMTB_ROLE_DETAILS.
- **Control String** Developer needs to select the actions which should be available for this screen in FLEXCUBE.

Function Generation										_ ×
								×L	C	V 🧃
Action New Function Id STDCIF Save XML Path C: RADTOOLIXI		Function Type Parent Parent Function Parent Xml	~			Function Catego Header Templa Footer Templa				
Search	Preferences									🧐 🔺
Preferences DataSource ListOfValues DataBlocks	 Head Office Function Logging Required 									
Screens FieldSets Actions	Auto Authorization Module Auto									
CallForms LaunchForms Summary	Authorization									
	Field Log Required									
	Multi Branch Access Excel Export Required									
	Java Functions									
	GateWay Screen Module Description Module Group Branch Program Id Process Code SVN Repository URL	Static Maintenance	P							•
	Menu Details									
	Parameter Value Mappings	nction Id	ST	Mor	dule *	P	Control Static Maintenance	String 🕂 Mod		-

Fig 12.4: Providing Preferences for Maintenance Screen

Note the following points while providing details in Preferences screen

i) Control String

REVERSE, ROLLOVER, CONFIRM, LIQUIDATE, HOLD operations are not applicable for maintenance screens.

Defining Browser Menu Tree
 Browser menu tree will be defined in the script generated for smtb_function_description.
 The following labels has to be maintained for generation of proper script Main Menu: LBL_{function id}_MAIN_MENU
 Sub Menu 1: LBL_{function id}_SUB_MENU_1
 Sub Menu 2: LBL_{function id}_SUB_MENU_2
 Description: LBL_{function id}_DESC
 Example: For STDCIF, following labels has to be maintained
 LBL_STDCIF_MAIN_MENU, LBL_STDCIF_SUB_MENU_1, LBL_STDCIF_SUB_MENU_2, LBL_STDCIF_SUB_MENU_2, LBL_STDCIF_SUB_MENU_1,

Refer <u>Oracle FLEXCUBE Enterprise Limits and Collateral Management ODT Screen Development</u> for detailed explanation on preferences

4.3 Data Sources

- Right Click on Data Sources; click on Add. Add table window gets opened.
- If user knows the exact table name, he can enter name directly; else go to List Of values to get the list of tables available. Select the required table from the list.

Function Generation				
				7 🥥
Action New Function Id STDCIF Save XML Path C:RADTOOLXI	Function Type[Parent Parent Function Parent Xml		Function Category Maintenance Header Template None Footer Template None	
Search	AddTable		×	*
Preferences DataSource ListOrValues	Table Name STTM CUSTOMER	×		
DataBlocks	Table Name			-
Screens FieldSets	< < 1of1719 >	>> Search Reset		
Actions	Table Name			
CallForms	SMTBS_MSGS			
LaunchForms	SMTBS_PARAMETERS	_		
Summary	SMTBS_PASSWORD_HISTORY	-		
	SMTBS_RESTRICTIVE_PASSWD			
	SMTBS_SSO_PARAM SMTB_CURRENT_ENTITY	-		
	SMTB_CORRENT_ENTITY SMTB_ENTITY_DETAILS	-		
	SMTB MSGS			
	SMTB PARAMETERS			
	SMTB_PASSWORD_HISTORY			
	SMTB_RESTRICTIVE_PASSWD			
	SMTB SSO PARAM			* *

Fig 12.5: Adding Data Sources for the Function id

- Select Master as Yes if added data source is Master Data Source for the screen. Every function id should have one master data source.
- **Primary Key columns** (i.e. Pk Cols) and **Primary Types** (i.e. Pk Types) are mandatory. If it is already maintained in user schema in STTB_PK_COLS it will populated automatically otherwise user needs to enter values without fail. If user misses Pk cols and Pk Types package generation will fail. *Note: Master Data Source cannot have any parent.*

Development of Maintenance Form

Function Generation									- ×
						×	L	8	9
Action New 🗸		Function Type Parent		Function Category Maint	enance 🗸				
Function Id STDCIF		Parent Function		Header Template None	~				
Save XML Path C:\RADTOOLXI		Parent Xml		Footer Template None	~				
Search	Data Source Details						¢		õ 🔺
Preferences	Data Source	STTM_CUSTOMER							
DataSource	Master	Yes 🗸							
STTM_CUSTOMER		One To One 🗸							
ListOfValues	Multi Record	No 🗸							
DataBlocks	PK Cols	CUSTOMER_NO							
Screens	PK Types	VARCHAR2							
FieldSets	Upload Table								
Actions	Upload Where Clause								
CallForms	Fine Grained Service Detai	Is Only							
LaunchForms	Rest Data Block	-							
Summary	XSD Node		-						
	Rest Relation		~						
	Rest Relation Type	~							
	Rest PK Cols								
	Rest PK Types								
	Parent	~							
	Relation								-

Fig 12.6: Providing master Data Source Properties

• Right Click on Added Table (STTM_CUSTOMER) to add fields to the table. Popup window gets opened with available columns in data source. Select the required fields and click ok. Selected will get added to the Data Source Tree.

Function Generation				_ ×
				🖬 🗵 L C 🎸 🤤
Action New Function Id STDCIF Save XML Path C:RADTOOLXI		Function Type Parent ~ Parent Function Parent Xml		Function Category Maintenance 🗸 Header Template None 🗸 Footer Template None 🗸
Search	Data Source Details			e 📼 🦃 📥
Preferences	Master Relation Type Multi Record PK Cols	CUSTOMER_NO VARCHAR2		_
	Rest PK Cols Rest PK Types		2	
	Parent Relation	×		•

Fig 12.7: Including Data Source Fields for the Data Source

Function Generation			
Action New Function Id STDCIF Save XML Path C: RADTOOL'X	Function Type Paren Parent Function Parent Xml		Function Category Maintenance Header Template None Footer Template None
Search	Select Fields	×	
DataSource STTM_CUSTOMER ListOrValues DataBlocks Screens FieldSets Actions Califorms LaunchForms Summary	CLS_CCY_ALLOWED CLS_PARTICIPANT CONSOL_TAX_CERT_REQD COUNTRY CREDIT_RATING CRM_CUSTOMER CRS_TYPE CUSTOMER_CATEGORY CUSTOMER_NAME1 CUSTOMER_NO CUSTOMER_TYPE CUSTOMER_TYPE CUST_CLASSIFICATION CUST_CLG_GROUP	VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2	
	Relation	Ok Cancel	

Fig 12.7: Selecting Data Source Fields for the Data Source

Data Source Field Properties:

Only max length can be modified by the developer in data source field properties. Rest will be defaulted from table definition

Function Generation									-	×
							×L	С	V	9
Action New Function Id STDCIF Save XML Path C.VRADTOOLXI		Function Type Parent Parent Function Parent Xml	~		Function Category Ma Header Template No Footer Template No	ne 🗸				
Search	Data Source Field Deta	ils								-
	Block Name Field Name ADI	DRESS_LINE1 DR1 RCHAR2								

Fig 12.7: Providing properties for Data Source Fields

Data model of a single function id would include multiple tables .All the tables needs to added in the function id. Note the following while adding child data sources

Adding Child Data Source:

- Select Multi Record value as Yes if child data source is Multi record table.
- Child Data Source should always be associated with a parent.

• Relation is mandatory between parent and child. While giving relation, parent data source should come in left side of the relation.

Function Generation								- ×
						×L	C 7	9
Action New Function Id STDCIF Save XML Path C: RADTOOL XI		Function Type Parent Parent Function Parent Xml		Header	Category Maintena Template None Template None			
Search	Data Source Details					¢	- 4	🦻 🔶
Preferences * DataSource * STTM_CUSTOMER STTM_CUST_GROUP ListOMalees DataBlocks Screens FieldSets Actions CallForms LaunchForms Summary	Master Relation Type Multi Record PK Cols PK Types Upload Table Upload Where Clause Fine Grained Service Detai Rest Data Block XSD Node Rest Relation Type Rest PK Cols Rest PK Types	STTM_CUST_GROUP						•

Fig 12.7: Providing properties for Child Data Source

Note: A data source cannot be parent to itself.

Note the following while adding data sources:

- i) If the data source is designed with relation type as 1: N with its parent, then it should have at least one more Pk col than its parent (assuming relationship is based on Pk cols).
- ii) Master data source needs to have the audit columns if footer template is Maint audit; but those should not be added to data source fields as system will handle it

Refer <u>Oracle FLEXCUBE Enterprise Limits and Collateral Management ODT Screen Development</u> for detailed explanation on data sources.

4.4 Data Blocks

• Block Name should start with BLK_<short Name equivalent to data source but not exactly same as Data Source name>.

Function Generation			
			🗟 🗙 L C 🐬 🤮
Action New Function Id STDCIF Save XML Path C.VRADTOOLXI	Function Type[Parent Parent Function Parent Function Parent Xml	Function Category Maintenance · Header Template None · Footer Template None	 ✓
Search	Add Block	×	Ф 🔶
Preferences * DataSource * STTM_CUSTOMER STTM_CUST_GROUP LicIOValues DataBlocks Screens FieldSets Actions CaliForms LaunchForms Summary	Block Name BLK_CUSTOMER	Relation Type	

Fig 12.8: Creating a new Data Block

- Select Parent block if added block is not Master Block.
- Select Multi Record (Yes/No) based on this value, available data sources will displayed in data source available text area.

Function Generation										-	. ×
								×	LC	V	6
Action New Function Id STDCIF		Function Type Parent Parent Function	~			ategory Mainter					
Save XML Path C:\RADTOOL\XI		Parent Xml			Footer T	emplate None	~				
Search	Block Properties							¢		1 🦈	^
Preferences	Block Title Parent Relation Type Block PK Fields XSD Node Comment ID		~	0]] Ø							
LaunchForms Summany		Datasourc	e Available	•	Datasource Added						
				P 44							•

Fig 12.9: Providing properties for Data Block

• Select the required data source and click move button to attach Data Source to the block

Function Generation		-	×
			9
Action New Function Id STDCIF Save XIML Path C:\RADTOOL\XI	Function Type Parent Parent Function Parent Xml	Function Category Maintenance Header Template None Footer Template None	
Search Preferences + DataSource + STTM_CUSTOMER STTM_CUST_GROUP ListOfValues + DataBlocks BLX_CUSTOMER Screens	Block Title Parent Relation Type Block PK Fields XSD Node Comment ID Master Block Yes Multi Record No Block Type Normal		*
FieldSets Actions CallForms LaunchForms Summary	Datasource Available STTM_CUSTOMER	Datasource Added STTM_CUSTOMER	

Fig 12.10: Attaching Data Sources to Data Block

Adding multi record data source to data block:

User on selecting Multi record Yes in data block properties all the data sources with multi record Yes will be populated. *Multi Data Source once used to one block won't available for reuse where as single record data source can be used in multiple blocks*

Select Block Fields:

- Right click on added block. Select Fields window will get opened. Developer needs to check the right side check box to add the required fields.
- **Field Name**: It should not be the same as column name .Special characters are also not allowed in the field name (including underscore and space)
- **Label Code**: It will be automatically populated based on field name.

Action New Function Id STDCIF Save XML Path C \RADTOOL\XI		Function Type Parent Arent Function Parent Xml		Function Category Maintenance Header Template None Footer Template None
earch and an anna an anna anna anna Treferences DataSource	Select Fields & Add Ul Fields PataSource fields Ul Fields			×
	Datasource	Field Name	Label Code	
	2 ADDRESS_LINE1 2 COUNTRY	ADDR1 CNTY	LBL_ADDR1	
	2 CUSTOMER_NAME1 2 CUSTOMER_NO	CNAME	LBL_CNAME	ed
	2 CUSTOMER_TYPE 2 LANGUAGE	LANG	LBL_CUSTTYPE	
	2 NATIONALITY	NLTY	LBL_NLTY	

Fig 12.11: Adding Block Fields to Data Block

Refer <u>Oracle FLEXCUBE Enterprise Limits and Collateral Management ODT Screen Development</u> for detailed explanation on data blocks and block field properties

4.5 Screens

- Right click on Screens node to add a new screen
- Screen Name should start with CVS_<Name>...
- By default screen are divided into 3 parts.
- One Main Screen is Mandatory.
- Tabs can be defined on any of the screen portions as required
- User can add sections to tabs.
- Each section can be divided into partitions.

Function Generation										~
								× L C	8	9
Action New 🗸		Func	tion Type Parent	~		Function Category	Maintenance 🗸			
Function Id STDCIF	1	Parent	Function			Header Template	None 🗸			
Save XML Path C:\RADTOOL\XI	r I	Pa	arent Xml			Footer Template	None 🗸			
Search	Screen De	tails						- 🛛 [a 🦻	^
LataBlocks ADDR1 CNTY CNAME CUSTNO CUSTNO CUSTTYPE LANG NLTY Screens CVS MAIN	So	ired	· · · · · · · · · · · · · · · · · · ·	2 2						
 HEADER BODY 								+	-	
► FOOTER				1 CONTRACTOR		Landse Land				
FieldSets	U	Argument Name	Source Block	Source Field	Argument Value	Target Block	Target Field	Active	*	
Actions										
CallForms										
LaunchForms										
Summary										•

Fig 12.12: Providing properties to new Screen

Function Generation		_ ×
		🖬 🗶 L C 🎸 🤘
Action New	Function Type Parent Parent Function Parent Xml	Function Category (Maintenance Header Template None Footer Template None
Search	Add Section	🗙 Expansion Field 🔄 📼 🖾 🐬 🥎
* DataBlocks * BLK_CUSTOMER ADDRT CNTY CNAME CUSTNO CUSTTYPE LANG NLTY * Screens * CVS_MAIN * HEADER TAB_HEADER * DOBME	Section Name SEC_CUST	
BODY FOOTER FieldSets Actions CaliForms		
LaunchForms		· · · · · · · · · · · · · · · · · · ·

Fig 12.13: Creating new section in TAB_MAIN in the body of screen CVS_MAIN

Function Generation			_ ×
			🖫 🗵 L C 🎸 🧐
Action New 🗸	Function Type Parent	Function	Category Maintenance 🗸
Function Id STDCIF	Parent Function	Header	Template None V
Save XML Path C:\RADTOOL\XI	Parent Xmi	Footer	Template None
Search	Section Details		📼 🗷 🧐 🤶
DataBlocks BLK_CUSTOMER ADDR1 CNTY CNAME CUSTNO	Section Name SEC_CUST Section Label LBL_SECT		
CUSTTYPE LANG			
NLTY	Partition Details		+ -
Screens Cvs_MAIN VeADER FTAB_HEADER BODY ATAB_MAIN SEC_CUST FOOTER TAB_FOOTER	Partition SI No 1 PART 1 2 PART 2	Partition Name	Width Sub-partitions
FieldSets			-

Fig 12.14: Defining partitions for the Section

4.6 Field Sets

A group of fields can be grouped together in a Field set which can be placed together in the screen

- Field Set Name should start with FST_<>.
- Select the Block adding to field set.
- All fields available to the block will be displayed in to the data block fields text area. Move fields from data block fields to Field set fields.
- The order of fields in *field set fields* will reflect in the screen as well

Function Generation								-
							×L	C 7
Action New Function Id STDCIF Save XI/L Path C:RADTOOLW	Fieldset Properties	Function Type Parent Parent Function Parent Xml	~		н	nction Category Mainte eader Template None Footer Template None		
	Ticluset Tioperties							
Preferences	Fieldset Name	FST_CUST1						
DataSource	Fieldset Label		ρ					
ListOfValues			~					
DataBlocks	Multi Record							
Screens		Single 🗸						
 FieldSets 	Fieldset Height							
FST_CUST1	Number Of Rows							
Actions	Fieldset Type	Normal	~					
CallForms	Screen Name		~					
LaunchForms	Screen Portion		~					
Summary	Tab Name		~					
	Section Name		~					
	Partition Name		~					
	Horizontal Fieldset ReadOnly Navigation Button Navigation Button Full Width Visible							
		Data Blo	ck Fields	1	Field Set Fields	Subpartition Name		
		ADDR1 CNTY CNAME CUSTNO CUSTTYPE LANG NLTY	×	44				

Fig 12.14: Attaching Fields to a Field set

Function Generation												-	. ×
									×	L	С	V	9
Action New Function Id STDCIF Save XML Path C:\RADTOOLXI			Function Type Parent Parent Function Parent Xml	~			Header 1	Category Maintenance V Template None V Template None V					
Search Preferences PotatsSource LatOfValues DataBlocks Screens Screens	Horizontal F ReadOnly Navigation E Navigation B Width Visible	Button											•
 FieldSets FST_CUST1 			Data Block Fields				FieldSet Fields	Subpartition Name					
Actions CaliForms LaunchForms Summary		CNTY LANG NLTY			•	DD 44	CUSTTYPE CUSTNO CNAME ADDR1						

Fig 12.14: Order of fields in the field set highlighted

• Select the screen portion (Header/Body/Footer) where this field set has to be placed. Select remaining details like tab, section and partition.

Function Generation			_ ×
			🖬 🗶 L C 🞸 🤢
Action New V Function Id STDCIF	Function Type Parent Parent Function	~	Function Category Maintenance 🗸 Header Template None
Save XML Path C:\RADTOOL\XI	Parent Xml		Footer Template None
Search	Fieldset Properties		📼 🗷 🧖 🤷
Preferences PataSource ListOfValues PataBilocks Screens FieldSets FST_CUST1 Actions CallForms LaunchForms Summary	Fieldset Name FST_CUST1 Fieldset Label Data Block Mutt Record No ✓ View Type Fieldset Height Number Of Rows Fieldset Height Screen Portion Screen Portion Body Tab Name SEC_CUST Partition Name FART 1 Horizontal Fieldset ReadOnly Navigation Button Navigation Button Full Wdth		

Fig 12.15: Providing details where Field Set has to be placed

Once fields are added to field set, developer can check the preview of the designed screen. Right click on Screen Name and click on Preview.

Administration Function Generation	Customer					х
Screen Customizer Tracking Changes Notification Triggers	🖹 New 🏳 Enter Que	ry				
Notifications Bulk Generation Excel Template Gener	Туре	 Individual Corporate 	Full Name Short Name		Customer Category *	Q
Refresh Block Detail Upload	Special Customer No Generation	O Bank	Branch Code		Private Customer	
 Test Case Definition Runchart Definition Test Case Data Uploa 	Customer No	Р				
Create Request Test Case Execution	Personal	Corporate	Additional	Director Auxiliary	Check List	MFI Details
▶ Runchart Execution Test Case Data Bulk L	Basic Details		Address For Corre	spondence	Passport Details	
Execution Report Purge Generation	Prefix 1	Q	Name		Passport Number	
Services Service Refresh	Prefix 2	Q	Address Code	Q 🗖	Issue Date	
	Prefix 3	Q	Address 1 *		Expiry Date	
	First		Address 2		Additional Details	

Fig 12.16: Preview of the designed Screen

Adding Multi entry block to field set.

- On selecting a multiple block, Multi Record Property will be defaulted to Yes..
- In case of Multi record, View type can be either Single or Multiple (By Default).

Below image shows a multiple view multi record field set

Fig 12.17: Multiple View Multi Record Field set

• For multi record single view navigation button should be checked.

Fieldset Properties				-	Aï
Fieldset Name	FST_REMARKS	Ĩ.			
Fieldset Label	LBL_REMRKS	Q			
Data Block	BLK_DOCTYPE_REMARKS	~			
Multi Record	No 🗸				
View Type	Single 🗸				
Fieldset Height					
Number Of Rows					
Fieldset Type	Normal	~			
Screen Name	CVS_MAIN	~			
Screen Portion	Body	~			
Tab Name	TAB_CHECKLIST	~			
Section Name	SEC_1	~			
Partition Name	PART2	~			
Horizontal Fieldset					
ReadOnly					
Navigation Button					
Navigation Button Full					
	Fig 12.18: P	roperti	es for Single View Multi Record Field set		

4.7 LOV

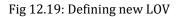
List Of values can be defined for the function id using LOV node

• To add LOV right click on List of Values Node. LOV Name should start with LOV_<name>.

Example: LOV_COUNTRY.

• Enter valid query and click on populate button

cat, cust_cat_desc from sttms_customer_cat where auth_stat = 'A' and record_stat = 'O' Populate Data Type Visible Reduction Field Reduction/Column Label Is Mandatory Min No. of * ARCHAR2 Yes Yes TEXT LBL_CUSTCGY No Image: Control of the state in the stat	ist O	f Values Details							Refr	esh 💻 🛛
Data Type Visible Reduction Field Reduction Field Text ARCHAR2 Yes Yes TEXT LBL_CUSTCGY No		LOVINAINO		lesc from	a sttms_customer_	cat where auth	_stat = 'A	' and record_stat = 'O'		
ARCHAR2 V Yes V Yes V TEXT V LBL_CUSTCGY		Query Columns	Data Type	Visible	Reduction Field	Reduction Fi	eld Type	Reduction/Column Label	Is Mandatory	•
ARCHAR2 V Yes V Yes V TEXT V LBL_DESC D No V		CUST_CAT	VARCHAR2 V							
	2	CUST_CAT_DESC	VARCHAR2 V	Yes 🗸	Yes 🗸	TEXT	~	LBL_DESC	▶ No ▼	
							_	-		



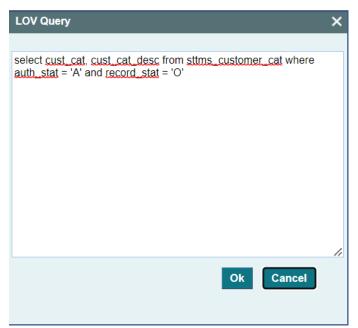
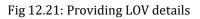


Fig 12.20: Providing LOV query

List Of Values Details Re								Refres	sh 📟	Aï.	9			
	LOV Name LOV Query LOV Type	LOV_CUST select cust_ Internal	cat, cust_cat_d	esc from	sttms_customer_c	cat where auth_s	tat = 'A'	' and record_stat = 'O'						2
												Popula	te	
	Query Colum	nns	Data Type	Visible	Reduction Field	Reduction Field	d Type	Reduction/Column Label		Is Mand	latory I	Min No. d	f 🔶	
	CUST_CAT	V	ARCHAR2 🗸	Yes 🗸	Yes 🗸	TEXT 🗸		LBL_CUSTCGY	ρ	No 🗸				
~	CUST_CAT_DESC) V/	ARCHAR2 🗸	Yes 🗸	Yes 🗸	TEXT 🗸		LBL_DESC	Q	No 🗸				
													~	



- Redn/Col Labels are mandatory. If user won't provide will get error on click of LOV button after deployment in FLEXCUBE
- After defining LOV go to block and corresponding field where the LOV has to be attached.

Block Field Properties to attach LOV to the field

- **Display Type:** Select display type as Lov.
- Lov Name: Select the required Lov name from the list of all defined LOV's.
- Click on return fields tab. The result fields maintained in the LOV query will be populated on click of *Default from Lov Definition* button
- Select the desired field (and its block)to which the result of the LOV query should be defaulted

• If return field is not required to be defaulted to any field in the screen, return field value can be left blank

Use of Bind Variable

If the list of values should be based on any other field value from the screen, bind variables can be used.

Example:

Define lov as shown in below query; where clause should contain condition with '?'.

SELECT cust_ac_no, branch_code, ccy from sttms_cust_account where cust_no = ? and record_stat = 'O' and once_auth = 'Y' and ac_stat_de_post = 'Y'

In the block field, after selecting return fields, click on bind variables tab. Click on **Default from Lov Definition** button. New rows will be created depending on the number of bind variable provided in the LOV query. Select the bind filed in the screen (and its block) for the LOV. Data type of the field has also to be selected.

4.8 Attaching Call forms

Maintenance Call forms can be attached to a maintenance screen. Refer the document <u>Development of Online Forms</u> for developing call forms

Attaching Call forms

- Add button to block to launch call form on button click.
 - Right click on Block
 - Select Add fields. Select fields and Add UI field's window will be launched
 - Select UI Fields tab. Click add row button. Enter button name and click ok.
 - Select display type as button and enter field label.

Select Fields & Add Ul Fields									
		rce fields							
UI F	UI Fields								
		Column Name	Field Name	Label Code	^				
		JOINT_VENTURE							
					Ok Cancel				
					ouncer				

Fig 12.24: Defining Button field

• Add Call form details to Call form node

Action Load ~		Function Type Parent		Function Category Maintenan	ce 🗸			
Function Id STDCIF		Parent Function	Header Template Maint Audit					
Save XML Path STDCIF_RAD.x	BROWSE	Parent Xml						
Search	Call Form Details							
Preferences								
DataSource				Screen Arguments	Dependent Field			
ListOfValues	Function ID	Parent Data Block	Parent DataSource	Relation	Relation Type			
DataBlocks					One To One V			
Screens	0		_					
FieldSets	STCCIFJ				One To One 🗸			
Actions	SICDIARC				One To One 🗸			
CallForms		BLK_CUSTOMER	STTMS_CUSTOMER V	STTMS_CUSTOMER.CUSTOMER_NO = S	One To One 🗸			
LaunchForms	STCCIFTX	BLK_CUSTOMER	STTMS_CUSTOMER V	STTMS_CUSTOMER.CUSTOMER_NO = S	One To One 🗸			
Summary	CSCFNUDF	BLK_CUSTOMER	STTMS_CUSTOMER V]	One To Many 🗸			
	STCCRDSA	BLK_CUSTOMER	STTMS_CUSTOMER V	STTMS_CUSTOMER.CUSTOMER_NO = S	One To One 🐱			
	CSCOFACM	BLK_CUSTOMER	STTMS_CUSTOMER V	STTMS_CUSTOMER.CUSTOMER_NO=CS 🔁	One To One 🗸			
	STCCUACC	BLK_CUSTOMER	STTMS_CUSTOMER -	STTMS_CUSTOMER.CUSTOMER_NO = S	One To One 🗸			
	STCNSCK	BLK_CUSTOMER	STTMS_CUSTOMER V	STTMS_CUSTOMER.CUSTOMER_NO =ST	One To One 🗸			
				STTMS CUSTOMER.CUSTOMER NO= CS				

Fig 12.25: Defining details of the Call form to be attached in call form node

- Add event to button.
 - On selecting event type as call form or launch form or sub screen button will be displayed on bottom of the screen.
 - If user needs to place button position in desired place on the screen, event type should be Normal .User has to write code in release specific JavaScript file to launch the screen
- Check the preview.

4.9 Adding Summary

- 1) Add entry in Preferences node for Summary screen
- 2) Click on Summary Node.
 - Enter Summary title .Select label code from lov.
 - Select Data Block master block and summary blocks will be displayed. Select required block from drop down list.
 - Select Data Source for summary.
 - Select Summary Type.
 - Select Summary Screen size.
 - Enter if any where clause is required.
 - Enter Default order by if required.
 - Enter Multi Branch where clause if required.
 - Attach the fields required in the summary result grid
 - If the field is required as part of filtering, query has to be checked for the particular field
 - Provide the position of fields in Result grid and Summary Query set .

Summary Preview

Right click on summary node and click on preview.

4.10 Amendable fields Maintenance

Amendable Fields

If user needs to modify data of a particular field on unlock, in Workbench developer has to maintain fields as amendable.

- Click ACTIONS node.
- Click on Amendables button next to the action for which the field has to be made amendable
- Select the fields in each block which user can modify for the selected action.

5. Generation and Deployment of files

Generate Files

• Click on generate button select the required files to generate and click on Generate button.

Deploy files

• Click on deploy button select the required files to deployed to server and click on deploy. On successful deployment status will be displayed as Deployed.

Testing

- Launch the screen from FLEXCUBE
- Try sample operations on the screen (NEW, MODIFY, QUERY etc)

6. Generated Units

The following units will be generated for a Maintenance screen.

Refer document <u>Development Workbench - Screen Development II</u> for 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, SMTB_ROLE_DETAIL, SMTB_FCC_GCJ_MAPPING are required for the functioning of Maintenance screen
- ii) Lov Details
- iii) Amendable Details
- iv) Label details
- v) Screen Details
- vi) Block details
- vii) Data Source Details
- viii) Call form details

ix) Summary Details

6.2.2 System Packages

The Main Package contains the basic validations and backend logic for the Maintenance function id. The Main package contains the mandatory checks required. It will also contain function calls to the other packages generated by Workbench. The main package has the below stages for a maintenance form:

The main package has the below stages for a maintenance

- Converting Ts to PL/SQL Composite Type
- Checking for mandatory fields
- Defaulting and validating the data
- Writing into Database
- Querying the Data from database
- Converting the Modified Composite Type again to TS

Each of these stages has a 'Pre' and 'Post' hooks in the Kernel, Cluster and Custom Packages. And these Hooks are called from the Main Package itself

Main Package has the system-generated code and should not be modified by the developer Kernel, Cluster and Custom Packages are the packages where the respective team can add business logic in appropriate functions using the Pre and Post hooks available

6.2.3 Hook Packages

Release specific packages will be generated based on the release type (KERNEL.CLUSTER or CUSTOM). Developer can add his code in the release specific hook package.

The Main Package has designated calls to these Hook Packages for executing any functional checks and Business validations added by the user. The structure for all the Hook Packages are the same, like:

Fn_Post_Build_Type_Structure Fn_Pre_Check_Mandatory

Fn_Post_Check_Mandatory

Fn_Pre_Default_and_Validate

Fn_Post_Default_and_Validate

Fn_Pre_Upload_Db

Fn_Post_Upload_Db

Fn_Pre_Query

Fn_Post_Query

These Functions are called from the Main package using the Pre and Post Hooks available in the Main Package. The 3 Hook Packages namely Kernel, Cluster and Custom Packages have similar structure and are for the respective teams to work on.

6.3 Other Units

6.3.1 Xsd

Xsd 's will be generated if gateway operations are required for the particular function id. Maintenance for the same has to be done in *Actions* node

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 STDCIF_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.

7.2.1 Functions in Hook Packages

Different functions available in the Hook Package of a Maintenance 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_post_bulid_type_structure If any change has to be made in the field values obtained from the form befor start of processing, code can be written here
- 3) Fn_pre_check_mandatory
- 4) Fn_post_check_mandatory

Any extra mandatory checks on the field values from the screen can be written here.

5) Fn_pre_query

6) Fn_post_query

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

- 7) Fn_pre_upload_db
- 8) Fn_post_upload_db

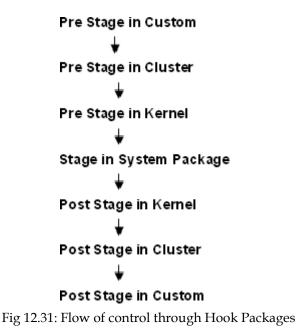
Any logic while uploading data to tables can be written here .

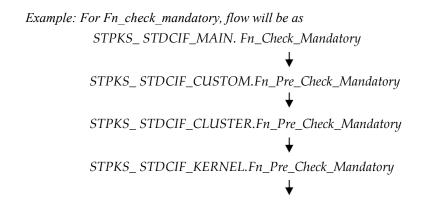
- 9) Fn_pre_default_and_validate
- 10) Fn_post_default_and_validate

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

7.2.2 Flow of control through Hook packages

The flow of control through the Hook Packages for a particular stage is as explained in the figure below





STPKS_STDCIF_MAIN .Fn_Sys_Check_Mandatory

STPKS_STDCIF_KERNEL.Fn_Check_Mandatory

STPKS_STDCIF_CLUSTER.Fn_Check_Mandatory

STPKS_STDCIF_CUSTOM.Fn_Check_Mandatory

7.2.3 By passing Base Release Functionality

There are auto generated functions like FN_SKIP_<RELEAE_TYPE> which would determine whether or not a particular hooks needs to be called.

Developer also has an option to bypass the base release hook if need be. For example if the validations written in *STPKS_STDCINF_KERNEL.FN_PRE_CHECK_MANDATORY* are not required or not suitable for the Cluster release, system provides an option to bypass the code written by Kernel team. Similarly a Custom release can also bypass the code written by Kernel and Custom Releases. This can be achieved by calling procedures

PR_SET_SKIP_<RELEASE_TYPE> and *PR_SET_ACTIVATE_<RELEASETYPE>*. These procedures will be made available in the main package and the development teams of Customization teams can use these procedures to skip and re-activate the hooks of parent release.

The Developer should avoid adding validations or Checks in the Pre Stage of any function, like Fn_Pre_Check_Mandatory, etc and should aim to add all the validations in the Fn_Post_Default_and_Validate.

For Example let us see the flow for the Mandatory Stage for STDCIF:

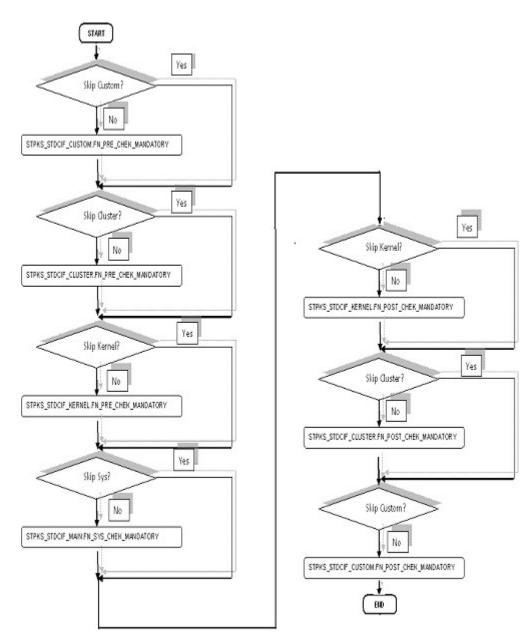


Fig 12.31: Flow of control explaining skip logic in pacakges



Development of Maintenance Form [Novemeber] [2023] Version 14.7.2.0.0 racle 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. All rights reserved.

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.