## Development of Maintenance Form

Oracle Banking Enterprise Limits and Collateral Management

Release 14.5.0.0.0 Part No. F42399-01

[May] [2021]



## Contents

1.1       Audience	1. Prefa		3
2.       Introduction       4         2.1       How to use this Guide       4         3.       Overview of Maintenance Screen       4         4.       Screen Development       4         4.1       Header Information       4         4.2       Preferences       6         4.3       Data Sources       7         4.4       Data Blocks       12         4.5       Screens       14         4.6       Field Sets       16         4.7       LOV       20         4.8       Attaching Call forms       23         4.9       Adding Summary       23         4.9       Adding Summary       27         4.10       Amendable fields Maintenance       29         5.       Generation and Deployment of files       30         6.1       Front End Units       33         6.1.1       Language xml       33         6.1.2       SYS JavaScript File       33         6.2       Data Base Units       33         6.2.1       Static Scripts       33         6.2.2       System Packages       33         6.3       Other Units       34         <	1.1	Audience	3
2.1       How to use this Guide       4         3.       Overview of Maintenance Screen       4         4.       Screen Development       4         4.1       Header Information       4         4.2       Preferences       6         4.3       Data Sources       7         4.4       Data Blocks       12         4.5       Screens       14         4.6       Field Sets       16         4.7       LOV       20         4.8       Attaching Call forms       23         4.9       Adding Summary       27         4.10       Amendable fields Maintenance       29         5.       Generation and Deployment of files       30         6.1       Front End Units       33         6.1.1       Language xml       33         6.1.2       SYS JavaScript File       33         6.2       Data Base Units       33         6.2.1       Static Scripts       33         6.2.2       System Packages       33         6.3       Other Units       34         6.3       Other Units       34         6.3.1       Xsd       34         6.3.1	1.2	Related Documents	3
3. Overview of Maintenance Screen       4         4. Screen Development       4         4.1 Header Information       4         4.2 Preferences       66         4.3 Data Sources       7         4.4 Data Blocks       12         4.5 Screens       14         4.6 Field Sets       16         4.7 LOV       20         4.8 Attaching Call forms       23         4.9 Adding Summary       227         4.10 Amendable fields Maintenance       29         5. Generation and Deployment of files       30         6.1 Front End Units       33         6.1.1 Language xml       33         6.1.2 SYS JavaScript File       33         6.1.3 Release Type Specific JavaScript File       33         6.2 Data Base Units       33         6.2.1 Static Scripts       33         6.2.2 System Packages       33         6.3 Other Units       34         6.3 Other Units       34         7.1 Extensible Development       34         7.1 Extensible Development       34         7.2 Extensiblity in JavaScript Coding       34         7.2 Extensibility in JavaScript Coding       35         7.2.1 Functions in Hook Packages       35 </td <td>2. Intro</td> <td></td> <td></td>	2. Intro		
4. Screen Development.       4         4.1 Header Information       4         4.2 Preferences.       6         4.3 Data Sources       7         4.4 Data Blocks.       12         4.5 Screens.       14         4.6 Field Sets.       16         4.7 LOV       200         4.8 Attaching Call forms.       23         4.9 Adding Summary       27         4.10 Amendable fields Maintenance.       29         5 Generation and Deployment of files.       30         6.1 Front End Units.       33         6.1.1 Language xml       33         6.1.2 SYS JavaScript File       33         6.1.3 Release Type Specific JavaScript File       33         6.2 Data Base Units.       33         6.2.1 Static Scripts       33         6.2.2 System Packages.       33         6.3.1 Xsd.       34         6.3.1 Xsd.       34         7.4.2 Extensible Development       34         7.4.3 Cher Units       34         7.4.4 Data Bace Units       34         6.2.7 System Packages       33         6.2.8 Hook Packages       34         6.3 Other Units       34         6.3.1 Xsd.       34     <			
4.1       Header Information       4         4.2       Preferences       6         4.3       Data Bources       7         4.4       Data Blocks       12         4.5       Screens       14         4.6       Field Sets       16         4.7       LOV       20         4.8       Attaching Call forms       23         4.9       Adding Summary       27         4.10       Amendable fields Maintenance       29         5.       Generated Units       33         6.1       Front End Units       33         6.1.1       Language xml       33         6.1.2       SYS JavaScript File       33         6.1.3       Release Type Specific JavaScript File       33         6.2       Data Base Units       33         6.2.1       Static Scripts       33         6.2.2       System Packages       33         6.3       Other Units       34         6.3       Ot			
4.2       Preferences	4. Scree		
4.3       Data Sources       7         4.4       Data Blocks       12         4.5       Screens       14         4.6       Field Sets       14         4.6       Field Sets       16         4.7       LOV       20         4.8       Attaching Call forms       23         4.9       Adding Summary       27         4.10       Amendable fields Maintenance       29         5.       Generation and Deployment of files       30         6.1       Front End Units       33         6.1.1       Language xml       33         6.1.2       SYS JavaScript File       33         6.1.3       Release Type Specific JavaScript File       33         6.2       Data Base Units       33         6.2.1       Static Scripts       33         6.2.2       System Packages       33         6.3.1       Xsd       34         7.4       Extensible Development       34         7.1       Extensibility in JavaScript Coding       34         7.2       Extensibility in Backend Coding       35         7.2.1       Functions in Hook Packages       35			
4.4       Data Blocks       12         4.5       Screens       14         4.6       Field Sets       16         4.7       LOV       20         4.8       Attaching Call forms       23         4.9       Adding Summary       27         4.10       Amendable fields Maintenance       29         5.       Generation and Deployment of files       30         6.1       Front End Units       33         6.1.1       Language xml       33         6.1.2       SYS JavaScript File       33         6.1.3       Release Type Specific JavaScript File       33         6.2.1       Static Scripts       33         6.2.2       System Packages       33         6.2.3       Hook Packages       34         6.3       Other Units       34         6.3       Other Units       34         6.4       Atta Script Coding       34         7.1       Extensible Development       34         7.2       Extensibility in JavaScript Coding       34         7.2       Extensibility in Backend Coding       35         7.2.1       Functions in Hook Packages       35			
4.5       Screens       14         4.6       Field Sets       16         4.7       LOV       20         4.8       Attaching Call forms       23         4.9       Adding Summary       27         4.10       Amendable fields Maintenance       29         5.       Generation and Deployment of files       30         6.       Generated Units       33         6.1       Front End Units       33         6.1.1       Language xml       33         6.1.2       SYS JavaScript File       33         6.1.3       Release Type Specific JavaScript File       33         6.2       Data Base Units       33         6.2.1       Static Scripts       33         6.2.2       System Packages       33         6.2.3       Hook Packages       34         6.3       Other Units       34         6.3       Other Units       34         7.1       Extensible Development       34         7.2       Extensibility in JavaScript Coding       35         7.2.1       Functions in Hook Packages       35			
4.6       Field Sets       16         4.7       LOV       20         4.8       Attaching Call forms       23         4.9       Adding Summary       27         4.10       Amendable fields Maintenance       29         5.       Generation and Deployment of files       30         6.       Generated Units       33         6.1       Front End Units       33         6.1.1       Language xml       33         6.1.2       SYS JavaScript File       33         6.1.3       Release Type Specific JavaScript File       33         6.2       Data Base Units       33         6.2.1       Static Scripts       33         6.2.3       Hook Packages       33         6.3       Other Units       34         6.3       Other Units       34         6.3.1       Xsd       34         7.1       Extensible Development       34         7.2       Extensibility in JavaScript Coding       35         7.2.1       Functions in Hook Packages       35			
4.7LOV204.8Attaching Call forms234.9Adding Summary274.10Amendable fields Maintenance295.Generation and Deployment of files306.Generated Units336.1Front End Units336.1.1Language xml336.1.2SYS JavaScript File336.1.3Release Type Specific JavaScript File336.2Data Base Units336.2.1Static Scripts336.2.3Hook Packages336.3Other Units346.3Other Units347.Extensible Development347.1Extensibli ty in JavaScript Coding347.2Extensibility in Backend Coding357.2.1Functions in Hook Packages35			
4.8Attaching Call forms234.9Adding Summary274.10Amendable fields Maintenance295.Generation and Deployment of files306.Generated Units336.1Front End Units336.1.1Language xml336.1.2SYS JavaScript File336.1.3Release Type Specific JavaScript File336.2Data Base Units336.2.1Static Scripts336.2.2System Packages336.2.3Hook Packages346.3Other Units347.Extensible Development347.1Extensiblity in JavaScript Coding347.2Extensibility in Backend Coding357.2.1Functions in Hook Packages35			
4.9Adding Summary274.10Amendable fields Maintenance295.Generation and Deployment of files306.Generated Units336.1Front End Units336.1.1Language xml336.1.2SYS JavaScript File336.1.3Release Type Specific JavaScript File336.2Data Base Units336.2.1Static Scripts336.2.2System Packages336.2.3Hook Packages346.3Other Units346.3.1Xsd347.Extensible Development347.1Extensibility in JavaScript Coding357.2.1Functions in Hook Packages35			
4.10Amendable fields Maintenance295.Generation and Deployment of files306.Generated Units336.1Front End Units336.1.1Language xml336.1.2SYS JavaScript File336.1.3Release Type Specific JavaScript File336.2Data Base Units336.2.1Static Scripts336.2.2System Packages336.2.3Hook Packages346.3Other Units346.3.1Xsd347.Extensible Development347.1Extensibility in JavaScript Coding357.2.1Functions in Hook Packages35			
5.       Generation and Deployment of files       30         6.       Generated Units       33         6.1       Front End Units       33         6.1.1       Language xml       33         6.1.2       SYS JavaScript File       33         6.1.3       Release Type Specific JavaScript File       33         6.2       Data Base Units       33         6.2.1       Static Scripts       33         6.2.2       System Packages       33         6.2.3       Hook Packages       33         6.3       Other Units       34         6.3.1       Xsd       34         7.1       Extensibility in JavaScript Coding       34         7.2       Fxtensibility in Backend Coding       35         7.2.1       Functions in Hook Packages       35			
6. Generated Units336.1 Front End Units336.1 I Language xml336.1.2 SYS JavaScript File336.1.3 Release Type Specific JavaScript File336.2 Data Base Units336.2.1 Static Scripts336.2.2 System Packages336.2.3 Hook Packages346.3 Other Units346.3.1 Xsd347. Extensible Development347.1 Extensibility in JavaScript Coding347.2 Extensibility in Backend Coding357.2.1 Functions in Hook Packages35			
6.1Front End Units336.1.1Language xml336.1.2SYS JavaScript File336.1.3Release Type Specific JavaScript File336.2Data Base Units336.2.1Static Scripts336.2.2System Packages336.2.3Hook Packages346.3Other Units346.3.1Xsd347.Extensible Development347.1Extensibility in JavaScript Coding347.2Extensibility in Backend Coding357.2.1Functions in Hook Packages35		1 2	
6.1.1Language xml336.1.2SYS JavaScript File336.1.3Release Type Specific JavaScript File336.2Data Base Units336.2.1Static Scripts336.2.2System Packages336.2.3Hook Packages346.3Other Units346.3.1Xsd347.Extensible Development347.1Extensibility in JavaScript Coding347.2Extensibility in Backend Coding357.2.1Functions in Hook Packages35			
6.1.2SYS JavaScript File336.1.3Release Type Specific JavaScript File336.2Data Base Units336.2.1Static Scripts336.2.2System Packages336.2.3Hook Packages346.3Other Units346.3.1Xsd347.Extensible Development347.1Extensibility in JavaScript Coding347.2Extensibility in Backend Coding357.2.1Functions in Hook Packages35	011		
6.1.3Release Type Specific JavaScript File336.2Data Base Units336.2.1Static Scripts336.2.2System Packages336.2.3Hook Packages346.3Other Units346.3.1Xsd347.Extensible Development347.1Extensibility in JavaScript Coding347.2Extensibility in Backend Coding357.2.1Functions in Hook Packages35	0.1.1		
6.2Data Base Units.336.2.1Static Scripts336.2.2System Packages.336.2.3Hook Packages.346.3Other Units.346.3.1Xsd.347.Extensible Development.347.1Extensibility in JavaScript Coding.347.2Extensibility in Backend Coding.357.2.1Functions in Hook Packages35	6.1.2	SYS JavaScript File	33
6.2.1Static Scripts	6.1.3	Release Type Specific JavaScript File	33
6.2.2System Packages336.2.3Hook Packages346.3Other Units346.3.1Xsd347.Extensible Development347.1Extensibility in JavaScript Coding347.2Extensibility in Backend Coding357.2.1Functions in Hook Packages35	6.2	Data Base Units	
6.2.3Hook Packages	6.2.1		
6.2.3Hook Packages	6.2.2	System Packages	
6.3       Other Units	623		
6.3.1Xsd	0	C C	
7. Extensible Development	0.0		
7.1Extensibility 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	7.2.1	Functions in Hook Packages	35
0 1 0	7.2.2	Flow of control through Hook packages	35
7.2.3 By passing Base Release Functionality	7.2.3	By passing Base Release Functionality	36

## 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.

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

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

### **1.2 Related Documents**

<u>04-Development\_WorkBench\_Screen\_Development-I.docx</u> <u>05-Development\_WorkBench\_Screen\_Development-II.docx</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"
- <u>Chapter 4 , "Screen Development"</u>
- Chapter 5 , "Generated Units"
- <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>4-Development\_WorkBench</u>

<u>\_Screen\_Development-I.docx</u>

### 4.1 Header Information

Provide the header information as shown in the figure.

DRACLE' FLEXCUBE Development Workbi	nch for Universal Banking	DEMOUSER
Browser .		Windows Options Sign Out
Function Generation		
		[] 2   2   2   2   2   2   2   2   2   2
Action None -	Function Type Parent -	Function Category Maintenance -
Function Id	Parent Function	Header Template None +
Save XML Path	Parent Xml	Footer Template None ~
DataSource DataSource DataSlocks Screens FieldSets Califorms Califorms Summary	12.1: Providing Header Information	

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

rowser .		DEMOUSER Windows Options Sign Out
action Generation		
200 Marca 1		Eurotice Category Maintenance - Serve (CTRL - S.)
Action New	Function Type Parent   Parent Function	Function Category Maintenance  Header Template None
Save XML Path D'RADTOOL	Parent Xml	Footer Template None
DataSource		

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

Browser -	nent Workbench for Universa	I Banking		Wind	Sign Out
Function Generation					-
Adion New + Function Id STDCIFD Save XML Path D:RADTOXL		Function Type Parent  Parent Function Parent Xml		Function Category Maintenance Header Template Hone - Fooler Template None	 0 17 Wa 4
Barch Preferances DalaSource ListOvlakes DalaBilocis Screens FriedSels Actions Califorms LaunchForms Summary	File Saved	Tor Description Dis of DownLoadFile from 10.184.132.100 Compl File Download Do you want to open or save this file? Name: RAD.21P Type: WinRAR ZIP archive From: 10.184.132.100 Open Si While files from the Internet can be useful harm your computer. If you do not hust the save this file. <u>What sites mk2</u>	we Cancel		

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>04-Development\_WorkBench\_Screen\_Development-I.docx</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.

Browser -	ment Workbench for Universal Banking					Windows	Options	MOUS Sign O	
unction Generation						F	XE	7 0	-
Action New ·	Fundi	on Type Parent			Function Category				
Function Id STDCIFD	Parent F				Header Template				
Save XML Path DIRADTOOL	A. 2010 (1996)	ent Xml			Footer Template	La Avio			
sarch	Preferences							a d	9
Preferences DataSource DataSource DataSource DataBlocks DataBlocks Screens FieldSets Actions CallForms LaunchForms Summary	Head Offic     Logging R     Auto Autho     Tank Modil     Field Log F     Multi Branc     Excel Expo	equired rization fications Required ch Access		Module Module Description Branch Program Id Process Code SVN Repository URL Transaction Block Name Transaction Field Name	ST Static Maintenance Choose Block Choose Field	•			
	Encoder 14						introl String	+-	
	Function Id     STDCIFD	βī	Module *	[4]	Static Maintenance	le Description			
	Fig 12.4: Provid								

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.

## ii) 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 STDCIFD, following labels has to be maintained* LBL\_STDCIFD\_MAIN\_MENU, LBL\_STDCIFD\_SUB\_MENU\_1, LBL\_STDCIFD\_SUB\_MENU\_2, LBL\_STDCIFD\_DESC

Refer <u>Development\_WorkBench\_Screen\_Development-I.docx</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.

Browser -	ment Workbench for Universal Banking	DEMOUSER Windows Options Sign Out
Action New + Function Id STDCIFD Save XML Path D/RADTOVL	Function Type Parent - Parent Function Parent Xma	Function Category Maintenance + Header Template Hone + Footer Template None +
Search Preferences DataSource ListoValues DataBlocks Screins FredSels Actions Califorms Califorms Summary	AddTable Table Katter Data Source  Table Name STTM_CUSTOMER%  Search Reset  Tof Tof	arent Relation Type
	Table Name       STTM_CUSTOMER_ALTERNATE_BRANCH       STTM_CUSTOMER_CAT       STTM_CUSTOMER_NAM_DETAIL       STTM_CUSTOMER_NAM_MASTER       STTM_CUSTOMER_PARAM       STTM_CUSTOMER_PRE_IMAGE       STTM_CUSTOMER_SOURCE_DETAILS       STTM_CUSTOMER_SRNO       STTM_CUSTOMER_SRNO       STTM_CUSTOMER_SINO       STTM_CUSTOMER_VINUSED       STTM_CUSTOMER_VINUSED	

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

CRACLE FLEXCUBE Development Workber	nch for Universal Banking - Windows Internet Explorer	And And Manual State					x
ORACLE FLEXCUBE Develop	ment Workbench for Universal Banking			[	DEMO	USE	R
Browser +			Windows	Optic	ns Si	gn Out	
Function Generation						-	. ×
				2	<b>I</b> 7	(1)	\$
Action New -	Function Type Parent		Function Category Maintenance +				
Function Id STDCIFD	Parent Function		Header Template None -				
Save XML Path D/RADTOOL	Parent Xml		Footer Template None	•			
Search	Data Source Details					- 🦃	*
<ul> <li>Preferences</li> <li>DataSource</li> <li>STTM_CUSTOMER</li> <li>ListOVAlues</li> <li>DataBlocks</li> <li>Screens</li> <li>FieldSets</li> <li>Actions</li> <li>CallForms</li> <li>LaunchForms</li> <li>Summary</li> </ul>	Data Source STTM_CUSTOMER Master Yes • Relation Type One To One • Multi Record No • PK Cols • CUSTOMER_NO PK Types • Upload Table		Normal  Mandatory	000			

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.

CRACLE FLEXCUBE Development Workbenc	ch for Universal Banking - Windows Internet Explorer	1 Instant, Chevrolet, Sec. in 1	and that			۰	23
ORACLE FLEXCUBE Developm	nent Workbench for Universal Banking			C	EMO	USE	R
Browser -			Windows	Option	ns Si	gn Ou	t
Function Generation							- ×
			6	<b>X</b>	E 7	(1)	4
Action New -	Function Type Parent		Function Category Maintenance -				
Function Id STDCIFD	Parent Function		Header Template None 💌				
Save XML Path D:RADTOOL	Parent Xml		Footer Template None -				
Search	Data Source Details				4	- 9	) ^
Preferences DataSource The CuSTOMER ListONalues DataBlocks DataBlocks Delete Delete Add CaliForms CaliForms Summary	Data Source Master Relation Type Multi Record PK Colss CUSTOMER PK Types VARCHAR2 Upload Table	Parent Relation Where Clause Default Order By Type	Mandatory	00			

Fig 12.7: Including Data Source Fields for the Data Source

Browser -	ment Workbench for Universal Banking				Windows		EMOUSER Sign Out
Function Generation				_			-
Action New -	Function Type Parent	*			Function Category Maintenance		
Function Id STDCIFD	Parent Function				Header Templals None •		
Save XML Path DORADTOOL	Parent Xmi				Foolar Template None	•	
learch	Select Fields		×				+ - 9)
Preferances				Parent			
	U CUSTOMER_NO	VARCHAR2		Relation		2	
Function 1d STDCIFD Save XML Path DIRADTOOL	CUSTOMER_TYPE	CHAR		Where Clause Delaul Order By		000	
	V CUSTOMER_NAME1	VARCHAR2		Type	Normal 👻		
Ca FieldSals	ADDRESS_LINE1	VARCHAR2			F Mandalory		
	ADDRESS_LINE3	VARCHAR2					
CaunchForms	ADDRESS_LINE2	VARCHAR2					
Summary	ADDRESS_LINE4	VARCHAR2					
	COUNTRY COUNTRY	VARCHAR2					
	SHORT_NAME	VARCHAR2					
	V NATIONALITY	VARCHAR2	E				
	U LANGUAGE	VARCHAR2	-				
		10 10 1					
		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

Browser -	ment Workbench for Unive	rsal Banking		Win	dows	DEN	NOUS Sign C	
unction Generation								-
Action New   Function Id STDCIFD Save XML Path D1RADTOOL		Function Type Parent  Parent Function Parent Xml		Function Category Maintenand Header Template None Footer Template None		XE	¥ (	
earch Preferences DataSource CUSTOMER_NO CUSTOMER_NO CUSTOMER_NO CUSTOMER_NHE1 ADDRESS_LINE1 ADDRESS_LINE1 COUNTRY NATIONALITY LANGUAGE LISTOVAILES DataBilocids Screens FieldSets Califorms Califorms Califorms Summary	Data Source Field Column Name Block Name Field Name		Data Type MaxLength Upload Table Column	VARCHAR2  Not Required in Upload Tables		Refr	esh — (	9

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.

RACLE FLEXCUBE Developm	ent Workbench for Universal Banking		DE	MOUSER
rowser .		Wi	ndows Options	Sign Out
nction Generation				- 77 🕤 (
Action New  Function Id STDCIFD Save XML Path D/RADTOOL	Function Type Parent  Parent Function Parent Xml	Function Category Maintenar Header Template None Footer Template None		
arch  Preferences  Graduate Structure  Graduation  Gra	Data Source Details       Data Source     STTM_CUST_GROUP       Master     No •       Relation Type     One To Many •       Multi Record     Yes •       PK Colls •     GROUP_JD-CUSTOMER_NO       PK Types •     VACHAR2-VARCHAR2       Upload Table     Image: Coll of the second sec	Parent STTM_CUSTOMER Relation STTM_CUSTOMER CUSTOMER Where Clause Default Order By Type Normal • Mandatory	▼ Q Q Q Q	+ - 9

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>Development\_WorkBench\_Screen\_Development-I.docx</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>.

Add Block		×
Block Name	BLK_CUSTOMER	
	Ok Cancel	
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.

RACLE FLEXCUBE Developm	nent Workbench for Univers	al Banking			D	EMC	DUSE	ΞR
rowser .				Windows	Optior	ns Si	ign Ou	ıt
nction Generation								_ >
					×	∎ 1⁄2	÷ 🤤	4
Action New -		Function Type Parent -	Function Category N	aintenance 👻				
Function Id STDCIFD		Parent Function	Header Template	one 🔻				
Save XML Path D:\RADTOOL		Parent Xml	Footer Template	one 🔻				
arch	Block Properties					<b>4</b> -	R 🧳	)
Preferences DataSource STM_CUSTOMER DataSource STM_CUST_GROUP ListOf/alues DataBlocks BLK_CUSTOMER Screens FieldSets Addons CaliForms LaunchForms Summary	Block Title Parent	SLK_CUSTOMER	XSD Node XSD Node Annotation Master Block Mutti Record Block Type Normal Datasource Added					

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

ORACLE FLEXCUBE Development Workbench for Universal Banking - Windows Internet Explorer											
DRACLE FLEXCUBE Development Workbench for Universal Banking DEMOUSER							ER				
Browser -							Windows	Opti	ons	Sign Ou	Jt
Function Generation											_ ×
							6	<b>X</b>		V 🧐	4
Action New 👻		Function Type Parent	-			Function Category	Maintenance 👻				
Function Id STDCIFD		Parent Function				Header Template	None 👻				
Save XML Path D:\RADTOOL		Parent Xml				Footer Template	None -				
Search	Block Properties								4 -		> ^
<ul> <li>Preferences</li> <li>DataSource</li> <li>STTM_CUSTOMER</li> <li>STTM_CUST_GROUP</li> <li>ListOfValues</li> <li>DataBlocks</li> <li>BLK_CUSTOMER</li> <li>Screens</li> <li>FieldSets</li> <li>Actions</li> <li>CallForms</li> <li>LaunchForms</li> <li>Summary</li> </ul>	Block Title	One To One  Datasource ISTIM CUSTOMER	e Available		XSD Node XSD Node Annotation Master Block Multi Record Block Type Datasourc STTM_CUSTOMER	Customer Yes • No • Normal •					

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.

ACLE FLEXCUBE Develo	pment V	Workbench for Universal B	anking					DE	MOUSE
wser 🗸							Window	s Options	Sign Out
tion Generation									
									77 🌖
Action New 👻			Function Type Parent	*			Function Category Maintenance	*	
Function Id STDCIFD			Parent Function				Header Templale None 👻		
Save XML Path DNRADTOOL			ParentXml				Fooler Template None	*	
2	^ Se	lect Fields & Add Ul Fields					×	4	- 📰 🕻
Preferences		ataSource fields UI Field							
DalaSource							omer		
STTM_CUSTOMER	Δ	Datasource STT	M_CUSTOMER -				•		
STTM_CUST_GROUP     ListOfValues	✓	Column Name	Field Name	Label Code	*		¥		
DataBlocks		CUSTOMER_NO	CUSTNO	LBL_CUSTNO			nal 💌		
BLK_CUSTOMER		CUSTOMER_TYPE	CUSTTYPE	LBL_CUSTTYPE					
Screens FieldSols	<b>V</b>	CUSTOMER_NAME1	CNAME	LBL_CNAME			led		
Actions	<b>V</b>	ADDRESS_LINE1	ADDR1	LBL_ADDR1					
CaliForms		COUNTRY	CNTY	LBL_CNTY					
LaunchForms Summary		NATIONALITY	NLTY	LBL_NLTY					
		LANGUAGE	LANG	LBL_LANG					
	V								
	£								
					-				
						Ok Canc	al		

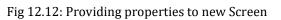
Fig 12.11: Adding Block Fields to Data Block

Refer <u>Development\_WorkBench\_Screen\_Development-I.docx</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.

ORACLE FLEXCUBE Development Workber	ench for Universal Banking - Windows Internet Explorer					х
ORACLE' FLEXCUBE Develop	opment Workbench for Universal Banking		D	ЕМО	JSE	R
Browser .	Wind	ows	Options	s Sig	n Out	(
Function Generation					-	- ×
			×	14	4	4
Action New -	Function Type Parent   Function Category Maintenance	е 🔻				
Function Id STDCIFD	Parent Function Header Template None					
Save XML Path D:\RADTOOL	Parent Xml Footer Template None	•				
Search	Screen Details		-	- Aï	<b>J</b> 🌖	^
<ul> <li>Preferences</li> <li>DataSource</li> <li>ListOfValues</li> <li>DataBlocks</li> <li>Screens</li> <li>CVS_MAIN</li> <li>HEADER</li> <li>BODY</li> </ul>	Screen Name CVS_MAIN  Screen Screen Titte LBL_CUST Screen Size Small Exit Button Type Default Cancel			+	-	
POTER     FOOTER     FoldSets     CallForms     LaunchForms     Summary	Argument Name         Source Block         Source Field         Argument Value         Target Block         Target Field	eld	Acti	ve	~	



ORACLE FLEXCUBE Develop	ent Workbench for Universal Banking		Windows	DEN Options	IOUSE Sian Out	
Function Generation					_	
Action New  Function Id STDCIFD Save XML Path DXRADTOOL	Function Type Parent  Parent  Parent  Parent Xml	Function Category Mainle Header Template None Fooler Template None				
Search	Tab Details		Dependent	Fields 👍	- 🛛 🧐	
Preferances DataSource ListOfValues DataBlocks CVS_MAIN HEADER	Screen Name CVS_MAIN  Tab Name TAB_MAIN Tab Label Tab Type  Add Section	×				
TAB_HEADER TAB_HEADER TAB_MANN FOOTER TAB_FOOTER FieldSels Actions	Section Name SEC_CUST Ok Cancel					
Califorms LaunchForms Summary						

Fig 12.13: Creating new section in TAB\_MAIN in the body of screen CVS\_MAIN

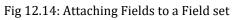
ORACLE® FLEXCUBE Development Workbench for Universal Banking Browser  Function Generation	DEMOUSER Windows Options Sign Out -> III IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
	 ■ X ■ 17 9 →
Function Generation	🖬 🗙 🗏 🔗 🇐 🔿
	a n a l Helshanna
Action New	Function Category Maintenance -
Function Id STDCIFD Parent Function	Header Template None 💌
Save XML Path D:RADTOOL Parent Xml	Footer Template None -
Search Section Details	- R 🦃
Preferences     Section Name     Sec_CUST     ✓ Visible     DataSource     ListOfValues     DataBlocks     Screens	
CVS_MAIN     Partition Details     HEADER	+ -
TAB_HEADER     Partition SI No     Partition Name     DOY     PART1	Width     Sub-partitions       50     •
BODY     1     PART1     PART2	50 -
	50 •
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

RACLE' FLEXCUBE Developm	ent Workbench for Universal Banking					DEMOUS
rowser 🗸				Wi	ndows O	ptions Sign O
nction Generation						
						🗙 🗏 🎸 🧃
Action New 👻	Function Type Parent		Function Ca	egory Maintenar	nce 🔻	
Function Id STDCIFD	Parent Function		Header Ten	nplate None	-	
Save XML Path D:\RADTOOL	Parent Xml		Footer Ten	nplate None	-	
rch	Fieldset Properties					- A( C
Preferences DetaSource ListOValues DataBlocks DataBlocks FFICSets FFICSets CUST1 FFICSets CallForms LaunchForms Summary	Fieldset Name Fieldset Label Data Block Mutil Record View Type Fieldset Height CUSTNO CUSTTYPE CNAME CNTY LANG	Screen Name Screen Portion Tab Name Section Name Partition Name Number Of Rows	CVS_MAIN	v v v		-Jorizontal Fieldset ReadOnly Navigation Button /isible



Browser -	ment Workbench for Universal Banking	DEMOUSE Windows Options Sign Out
Action New  Function Id STDCIFD Save XML Path D:RADTOOL Search Preferences DataSource ListOValues	Function Type Parent  Parent Function Parent Xml Fieldset Properties Fieldset Name FST_CUST1 Fieldset Label	Function Category Maintenance  Header Template None Footer Template None Screen Name CVS_MAIN CVS_MAIN ReadOnly
<ul> <li>DataBlocks</li> <li>Screens</li> <li>FFI_CUST1</li> <li>FST_CUST2</li> <li>Actions</li> <li>CallForms</li> <li>LaunchForms</li> <li>Summary</li> </ul>	Data Block BLK_CUSTOMER   Multi Record View Type Single   Fieldset Height  CNTY LANG  Data Block Fields	Tab Name Section Name Partition Name Number Of Rows

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

		on Acc. Manual Real.		
	ent Workbench for Universal Banking			DEMOUSER
Browser -			Window	1 3
Function Generation				_
Action New  Function Id STDCIFD Save XML Path D:IRADTOOL Search Preferences DataSource ListOfValues DataBlocks DataBlocks Creens	Function Type Parent  Parent  Parent Function Parent Xml  Fieldset Properties  Fieldset Label Data Block BLK_CUSTOMER Multi Record No	Heade	n Category Maintenance r Template None	Horizontal Fieldset     ReadOnly     Navigation Button     Visible
FieldSets     FST_CUST1     FST_CUST2     Actions     CallForms     LaunchForms     Summary	View Type Single + Fieldset Height	Partition Name Number Of Rows  Field Set Fields  CUSTNO CUSTNO CUSTYPE ADDR1  44	Subpartition Name	

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.

🔶 Main		
🗗 New 🖾 Enter Query		
Customer No		
Name		
Туре		
Address		
Maker	Date Time:	
Checker		
	Date Time:	Exit
Mod No	Record Status	
mourto	Authorization Status	
l		

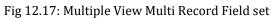
Fig 12.16: Preview of the designed Screen

### 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).

🔶 Main		×
🖹 New 🦻 Enter Query		
Customer No Name Type Address		
I≪ 1 of 1 ▶ ▶I	Go to Page	+-
Group Id	Customer No Relation	
•		4
Maker Checker	Date Time: Date Time:	Exit
Mod No	Record Status Authorization Status	

### Below image shows a multiple view multi record field set



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

	mont Morthborgh for I	niversal Banking					DEMOUR
	ment Workbench for U	niversai Bañking					DEMOUS
rowser -						Windo	ws Options Sign O
nction Generation							
							🔚 🗶 🗏 7 🔮
Action Load -		Function Type Parent 👻			Fur	ction Category Maintenance	-
Function Id STDCIFD		Parent Function			He	ader Template None 👻	
Save XML Path STDCIFD_RAI	BROWSE	Parent Xml			F	ooter Template Maint Audit	•
arch	Fieldset Proper	ties					- AI C
늘 Preferences	Fieldset Nar	ne FST_CUST2	S	creen	Name CVS_MAIN	•	🔲 Horizontal Fieldse
DataSource	Fieldset La	el 🔀	Sc	reen F	Portion Body	•	ReadOnly
STTM_CUSTOMER CUSTOMER NO	Data Blo			Tab	Name TAB_MAIN	•	Navigation Button
CUSTOMER_TYPE	Multi Reco	rd Yes 👻	Se	ection	Name SEC_GROUP	•	Visible
CUSTOMER_NAME1	View Tv	Single 🔹			Name PART1	•	1.
DDRESS_LINE1	Fieldset Hei				Rows		
COUNTRY							
LANGUAGE		Data Block Fields			Field Set Fields	Subpartition Name	
GROUP_ID						Casparation name	
CUSTOMER_NO					GROUP_ID		
Carl RELATIONSHIP					CUST_NO	•	
ListOfValues					RELATION	-	
DataBlocks			44				
BLK_CUSTOMER     DELK_GROUP			44				
Screens							
🖃 🧰 CVS_MAIN							
🗉 🚞 HEADER							
BODY							
EC_CUST							
B DOTER							
i DieldSets							
FST_CUST1							
EST_CUST2							
🗁 Astiana							

#### Fig 12.18: Properties for Single View Multi Record Field set

🔶 Main		×
🗗 New 🔄 Enter Query		
Customer No Name Type Address		
Group Id Customer No Relation		
Maker Checker	Date Time: Date Time:	Exit
Mod No	Record Status Authorization Status	

Below figure shows the preview of a single view multi record field set

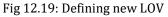
Fig 12.18: Preview for Single View Multi Record Field set

## 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

					<b></b>	× = 1	چ
Action Load 👻	Function Ty	pe Parent 👻		Function Cate	gory Maintenance 👻		
Function Id STDCIFD	Parent Functi	on		Header Temp	late None 👻		
Save XML Path STDCIFD_RAI	BROWSE Parent X	ml		Footer Temp	late Maint Audit 👻		
irch	List Of Values Details					-	· 🔊 🗳
DataSource  STTM_CUSTOMER  STTM_CUST_GROUP  ListOfValues	LOV Name * LOV_OCUNTRY LOV Query select country_cod	e,description from sttm_	country where auth_stat = 'A	A' and record_stat = 'O'			
						Pop	ulate
🧰 DataBlocks   🛅 Screens	Query Columns Data	Type Visible	Reduction Field	Reduction Field Type	Reduction/Column	Label	^
FieldSets							-
CallForms							



LOV	Query	×
	select country_code,description from sttm_country where auth_stat = 'A' and record_stat = 'O'	
	Ok Cancel	

Fig 12.20: Providing LOV query

Function Generation											-	. ×
									×	<b>I</b> 7	<ul> <li>§</li> </ul>	4
Action Load  Function Id STDCIFD Save XML Path STDCIFD_RAI	Pare	nction Type Parent nt Function Parent Xml	<b>•</b>			Head	tion Category Maintenance der Template None - ter Template Maint Audit	▼ ▼				
Search	List Of Values Details									-	ai 🌍	*
<ul> <li>Preferences</li> <li>DataSource</li> <li>Titt_cUSTOMER</li> <li>STTM_CUST_GROUP</li> <li>LIStONalues</li> <li>LIStONalues</li> </ul>	LOV Name * LOV_OCU LOV Query select cou		from sttm_co	untry where auth_stat :	= 'A' and recor	'd_stat = 'O'			(	Рори	late	
DataBlocks	Query Columns	Data Type	Visible	Reduction Field	Reducti	on Field Type	Reduction/Colu	mn La	bel		^	
Greens     FieldSets	COUNTRY_CODE	VARCHAR2 -	Yes 🔻	Yes 👻	TEXT	-	LBL_CNTRY		×E			
CallForms CallForms CallForms CallForms CallSummary	DESCRIPTION	VARCHAR2 -	Yes 🕶	Yes -	TEXT	-	LBL_COUNTRYCD		M		- - -	

Fig 12.21: Providing LOV details

- 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

Function Generation					_ ×
					🖫 🗵 🗏 7 🧐 🔿
Action Load - Function Id STDCIFD Save XML Path STDCIFD_RAI	Function Type Parent Function BROWSE Parent Xm	n		Function Category Maintenance - Header Template None - Footer Template Maint Audit	] •
Search	Block Field Properties				- 🛛 🗔 🧐 🤺
Preferences DataSource ListOWalues ListOWalues ListOWalues ListOWalues ListOWalues Custower Custow	Field Name * CNTY Field Label LBL_CNTY DataSource STIM_CUSTOMER Column Name * COUNTRY Data Type * Varchar? ~ Display Type * Lov • Item Type Database Item • Parent Field Related Field LOV Name COUNTY Off Line LOV Name	Ma M Maxin Text	XSD Tag CNTY SD Annotation Field Stze immum Length aximum Value aximum Value cationum Pocinais catArea Rows Area Columns Default Value Preview Value Mask Id		Required  Visible  Read Only  Calender Text  Oppercase Only  LOV Validation  Required  Input by LOV Only  Not Required In Xsd  Report Parameter
	Custom Attributes Events Bind Variables	Return Fields Related Field			L D.C.W.
CallForms	Query Column	Block	Namo	Return Field Nan	m Lov Definition
Summary		BLK_CUSTOMER BLK_CUSTOMER			
I	Fig 12.22: A	Attaching LOV to a l	block Field		

### 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.

Action Land	Function Type Parent	14	Function Category II	antanance -			
Function Id STOCHE	Parent Function		Header Template 14	cne 💌			
Save XML Path D1R4DTOOL1	Parent Xml		Footer Template	aint Audit 💌			
learch	Block Field Properties					- A) 9	
Preferences Ustofrailues Lustofrailues LUS(_COUNTRY LUS(_COUNTRY CUSTONER CUSTONER CUSTONER CUSTONER CUSTONER CUSTNO CUTPE FIAME ADDRLN1 COUNTRY FIATS FIATS CUSTNO CUSTN	Field Name CUSTNO     Field Label (LBL_CUSTNO     XSD Tag CUSTNO     Display Type Text     Mem Type Database frem      Parent Field     Related Field     Related Field     Max Decimals     LOV Name LOV_ACCOUNT Fieldset Name FST_GROUP		Data Type Vanchar2 M DataSource STTME_CUST Max Length B     Field Size     Column Name CUSTOMER_ Default Value     Preven Value     Accessive Cols     Max Val     Mass Id     Off Line LOV Name     Image Source	r_GROUP	Popup Edit Regit Reguired Valible Input by LOY Only Calender Text Select Multiple Uppercase Only LOY Validation Regit Report Parameter Read Only		
a FieldSets Actions a Califorms a Califorms a Califorms a Califorms a Summary	Custom Attributes Events Bind Variables Bind Variables Mapping Biock Name Biock Name BLK_CUSTONER	Katurn Fields	Bind Vanabie	Catault toon LoV definitio Detatype STRING			

Fig 12.23: Defining bind variable for the LOV

## 4.8 Attaching Call forms

Maintenance Call forms can be attached to a maintenance screen. Refer the document <u>14-</u> <u>Development of Call Form.docx</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.

S	elect F	ields & Add	Ul Fields						×
ſ	DataS	ource fields	UI Fields						
							 	+-	
				Field Name	;	-	 Data Type		<b>^</b>
	1	BTM_MIS					•		
									<b>*</b>
							Γ	Ok	Cancel
							L		

Fig 12.24: Defining Button field

• Add Call form details to Call form node

							E 7	9
Action Load 👻		Function Type Parent	*	Funct	ion Category Mair	ntenance 👻		
Function Id STDCIFD		Parent Function		Head	ler Template Non	e 🔻		
Save XML Path STDCIFD_RAI	BROWSE	Parent Xml		Foo	ter Template Mair	nt Audit 👻		
rch	Call Form Details							9
DataSource								
STTM_CUSTOMER     Description			1	1	creen Arguments	-	Fields <mark>+</mark> -	]
ListOfValues	Function ID	Parent Data Block	Parent Data Source	Relation	Relation Type	Callform Screen	Display 1 4	÷
LOV_COUNRTY     DataBlocks	MICCUSTM	BLK_CUSTOMER -	STTM_CUSTOMER -	TTM_CUSTOMER.COSTOMER_NO =	One To One 🔻	•	Button	
ADDR1 ADDR1 CNTY ADNLTY ANK BTM_MIS BTM_MIS BL_CROUP Screens FieldSets Actions CallForms LaunchForms Summary								
	٠		III				- F	r.

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

Function Generation									-
							ŀ	× =	V 🧐
Action Load 👻		Function Type	Parent	•		Function Category	Maintenance 👻		
Function Id STDCIFD		Parent Function				Header Template	None 🔻		
Save XML Path STDCIFD_RAI	BROWSE	Parent XmI				Footer Template	Maint Audit 🛛 🔫		
Search	Block Field Prope	ties						- 2	. 😋 🦻
<ul> <li>Preferences</li> <li>DataSource</li> <li>STM_CUSTOMER</li> <li>STTM_CUST_GROUP</li> <li>ListOfValues</li> </ul>	Field Name Field Label DataSource Column Name	LBL_MIS	×E	XSD Anr	eld Size *			<ul> <li>Required</li> <li>Visible</li> <li>Read Only</li> <li>Calender 1</li> </ul>	
LOV_COUNRTY  DataBlocks  BLK_CUSTOMER  CUSTNO  CUSTNO  CUSTYPE	Data Type Display Type Item Type Parent Field	Text  Control		Minimur Maximur Maximum De Textee	n Value cimals a Rows			<ul> <li>Popup Edit</li> <li>Uppercase</li> <li>LOV Valida Required</li> </ul>	only tion
CNAME ADDR1 CNTY NLTY LANG BTM_MIS	Related Block Related Field LOV Name Off Line LOV Name Fieldset Name		-	Preview	v Value	)     		Input by LC Not Requir Report Par	ed In Xsd
BLK_GROUP     Screens	Custom Attributes	vents Related Field							
CVS_MAIN HEADER	Event Nam	e Fun	ction Name	Event Type	Button Screen	CallForm Nam	ne Scree	+ - n Name	- -
🖃 🧰 BODY	onunload	• Tui		Callform -	CVS_MAIN -				
TAB_MAIN  TAB_MAIN  TAB_MAIN  SEC_CROUP  TSC_GROUP  TS									Ŧ

• Check the preview.

🔶 Main				×
🖹 New 🦻 <u>Enter Query</u>				
Customer No Name Type Address				
I≪ 1 of 1 ▶ ▶	Go to Page		+-==	
Group Id	Customer No	Relation	×	
			~	
•		III	▲	
MIS   Change Log				<b></b>
Maker	I	Date Time:		
Checker		Date Time:	Exi	it
Mod No	Rec	ord Status		
	Authoriza	tion Status		

Fig 12.27: Preview of the screen with the Call Form button

## 4.9 Adding Summary

1) Add entry in Preferences node for Summary screen

			🖪 🗵 🗏 🖗	· 🧐
	Function Ty Parent Functi BROWSE Parent X Preferences	ion	Function Category Maintenance 👻 Header Template None 👻 Footer Template Maint Audit 👻	ý
Search  Preferences DataSource  STM_CUSTOMER  STM_CUSTGROUP  ListONalues DataBlocks  BLK_CUSTOMER  CUSTNO CUSTTVPE CNAME CNAME ADDR1 CNTY		red Module Descripti ion Branch Program ons Process Cou irred SVN Repository UF ccess Transaction Bloc Nam	Distatic Maintenance	~
🚞 NLTY 🛅 LANG			Control String +	Ξ
BTM_MIS     BLK_GROUP	Function Id	Module *	Module Description	^
Screens	STDCIFD		Static Maintenance	
a) CVS_MAIN  b) TeldSets  c) Actions  c) CallForms  c) LaunchForms  Summary	STSCIFD	mmary screen details in Prefer	E Static Maintenance	Ŧ

- 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 .

										×	<b>I 1</b>	9	
Action Load 🔻			Function Type Parent	-			Function	Category Maintenance	-				
Function Id STDCIFD			Parent Function				Header	Template None 🔻					
Save XML Path STDCIFD_RAI	BROWSE		Parent Xml				Footer	Template Maint Audit	•				
h	Summary De	etails										<b>0</b>	)
Preferences		Title		×E		Default Where Clause			-				
DataSource     Data Blocks     BLK_CUSTOMER						Default Order By			1				
Action Load  Function Type Parent Function Id STDC/FD Function Id STDC/FD Parent Function Save XML Path STDC/FD_RAI BROWSE Parent Xml Footer Template Maint Audit  Search Summary Details  File Default Where Clause File Default Where Clause File File File File File File File Fil				•									
				•		Clause							
				•		Main Summary Screen							
	-	1					require						
	Data Block Fiel	Ids Custo	m Buttons   Fields Ordering										
			Data Block Fields			Fields Selected	Query	LOV Name					
					Γ	CUSTNO	<b>v</b>		•				
					Г	CNAME			Ŧ				
						CUSTTYPE			Ŧ				
						ADDR1			Ŧ				
a STTM_CUSTOMER     Data Block Fields     Data Block Fields       a STTM_CUSTOMER     Data Source     Multi Branch Where       b ListOvAues     Summary v     Multi Branch Where       a DataBlocks     Summary Screen Size     Medium       a Data Block Fields     Custom Buttons     Fields Ordering	Ŧ												
					Ε	NLTY			Ŧ	İ.			
					Г	LANG			Ŧ				

#### **Summary Preview**

Right click on summary node and click on preview.

🔷 😂 Exe	ecute Query 👔 Advance	ed Search 🏼 🏟 Res	et 🖓 Clear All							×
	Authorization Status Customer No	<b>•</b>	2		Rec	ord Status	-			
Reco	rds per page 15 👻 🚺	🚽 1 of 1 🕨 🔰		9						
	Authorization Status	Record Status	Customer No	Name	Туре	Address	Country	Nationality	Language	
										E
										-
									1	F
									Exit	

Fig 12.29: Summary Screen 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.

ndable DetailsQUERY		
Data Blocks	DataBlock Fields	;
JSTOMER ROUP	New Allowed Delete Allowed All I	Records 🔲 Mandatory
	Field Name	Amendable
	CUSTNO	
	CUSTTYPE	
	CNAME	
	ADDR1	
	CNTY	
	NLTY	
	LANG	
	BTM_MIS	
		Ok Cance

Fig 12.30: Maintaining amendable fields

## 5. Generation and Deployment of files

### **Generate Files**

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

ation		×		Vela Data	Others
Error Description Request successfully Processed		Image: Second		Xsds     Xsd With Annotations     Screen Html     Upload Table Trigger     Upload Tables Definition     Archive Table Definition	
		File Download		rge Details	
		Do you want to open or sa		ile Type	Status
		Type: WinRAR ZIP archive From: 10.184.132.100			Generated * Generaled *
			Open Save Cancel		Generaled *
STDCIFDCVS_MANTAB_F0OTER html Sipks_stdcild_main spc While files from the inte sipks_stdcild_main spc		ternet can be useful, some files can potent f you do not trust the source, do not open of	ially or	Generaled *	
	stpks_sldcifd_kernel.spc		une hisk <u>y</u>		Generated 👻
	stplos_stdold_main.sql		SQI		Generaled *
	stpks_sldclfd_kernel.sql		SO		Generated 👻
	CST0_FIELD_LABELSSTOCIFD.INC		INC		Generaled *
	CSTB_OTHER_LABELSSTDCIFD INC		INC		Generated 👻
	OSTB_FID_CALLFORMSSTOCIFD.INC		INC		Generaled *

Fig 12.30: Generation of Files

### **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.

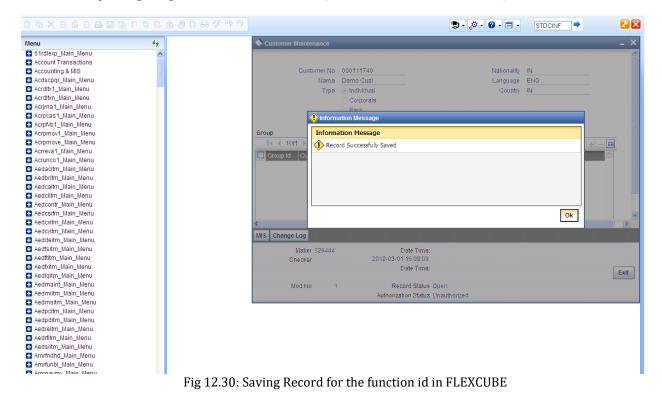
Front-End Files	System Packages	Hook Packages	м	eta Data	Others	*
RadXML ℤ Screen Xml ℤ System JS	Main Package Spec     Main Package Body     Notification Triggers     Upload Package Spec     Upload Package Body	Kernel Package Spec Kernel Package Body Cluster Package Spec Cluster Package Body Custom Package Spec Custom Package Body	Menu Details     Datasource Details     LOV Details     Slock Details     Sreen Details     Amendable Details     Call form Details     Summary Details	Label Details     Diock PK Columns     Function Call Forms     Gateway Details     Notification Details     Prunction Parameters     Purge Details	Xsds Xsd With Annotations Screen Html Upload Table Trigger Upload Tables Definition Archive Table Definition	
C	STB_FIELD_LABELSSTDCIFD.INC		INC		Deployed -	*
	STB_OTHER_LABELSSTDCIFD.INC		INC		Deployed -	
C	STB_SUMMARY_INFOSTDCIFD.INC		INC		Deployed -	
i s	TTB_AUDIT_PK_COLSSTDCIFD.INC		INC		Deployed -	
C	STB_FID_DATA_BLOCKSSTDCIFD.INC		INC		Deployed -	
C	STB_FID_DATA_SOURCESSTDCIFD.INC		INC		Deployed -	
C	STB_FID_SCR_TABSSTDCIFD.INC		INC		Deployed -	=
0 0	STB_FID_SCREENSSTDCIFD.INC		INC		Deployed -	
1 5	MTB_MENUSTDCIFD.INC		INC		Deployed -	
2 5	MTB_ROLE_DETAILSTDCIFD.INC		INC		Deployed v	
3 5	MTB_FUNCTION_DESCRIPTION_STDCIFD.INC	;	INC		Deployed -	
4 S	MTB_FCC_FCJ_MAPPINGSTDCIFD.INC		INC		Deployed -	
5 5	TDCIFD RAD.xml		RAD	XMI	Generated -	-

Fig 12.30: Deployment of Files

#### 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.docx</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:

- 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 STDCIFD\_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\_dbAny 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

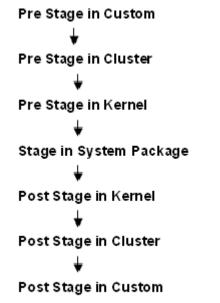
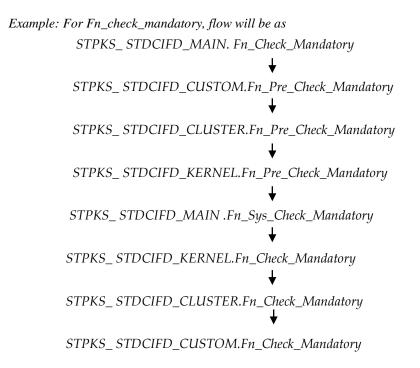


Fig 12.31: Flow of control through Hook Packages



### 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 STDCIFD:

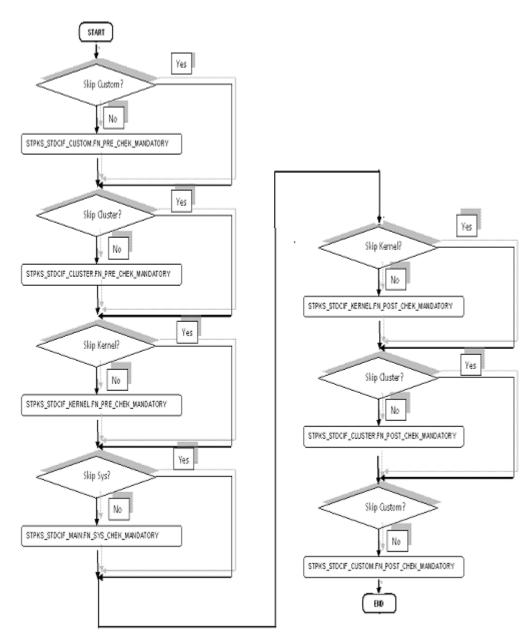


Fig 12.31: Flow of control explaining skip logic in pacakges



Development of Maintenance Form [May] [2021] Version 14.5.0.0.0

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, 2021, 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.