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





Table of Contents

1. Preface	3
1.1 Audience	
1.2 Related Documents	
2. Introduction	
2.1 How to use this Guide	
3. Overview of Online Form	
4. Screen Development	
4.1 Header Information	
4.2 Preferences	
4.3 Data Sources	
4.4 Data Blocks	
4.5 Screens	
4.6 Field Sets	
4.7 Actions	
4.8 Launch Forms	
4.9 Call Forms 4.9.1 Sub System Pickup/Processing	
4.10 Summary	
4.11 Preview	
5. Generated Units	
5.1 Front End Units	
5.1.1 Language xml	19
5.1.2 SYS JavaScript File	19
5.1.3 Release Type Specific JavaScript File	19
5.2 Database Units	19
5.2.1 Static Scripts	19
5.2.2 System Packages	19
5.2.3 Hook Packages	20
5.3 Other Units	20
5.3.1 Xsd	
6. Extensible Development	
6.1 Extensibility in JavaScript Coding	
6.2 Extensibility in Backend Coding	21

1. Preface

This document describes the features of Online Forms in FLEXCUBE and the process of designing an Online form screen using Oracle FLEXCUBE Development Workbench for Universal Banking

1.1 <u>Audience</u>

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

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

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

1.2 Related Documents

04-Development_WorkBench_Screen_Development-I.docx

05-Development_WorkBench_Screen_Development-II.docx

2. Introduction

2.1 How to use this Guide

The information in this document includes:

- Chapter 2, "Introduction"
- Chapter 3, "Overview of Online Form"
- Chapter 4 , "Screen Development"
- Chapter 5, "Generated Units"
- Chapter 5, "Extensible Development"

3. Overview of Online Form

Online Forms are function Id's (screens) which is used for creating Contracts for respective modules. Same contracts can be processed further for Payments, Availments, Amendments, Reassignments and Authorizations also using Online forms.

All the transaction processing in FLEXCUBE is carried out through Online screens Online form screens should be launched independently.

Example: Letter Of Credit (LC) contract

An LC contract is an instruction wherein a customer requests the bank to issue, advice or confirm a letter of credit, for a trade transaction. An LC substitutes a bank's name and credit for that of the parties involved. The bank thus undertakes to pay the seller/beneficiary even if the remitter fails to pay.

Thus for each module we should develop different function Id's for creating contracts and others online forms for other operations like Payments, Availments, Amendments, Reassignments and Authorizations.

LCDTRONL	- Contract Input
LCDAMEND	- Amend Confirmation Input
LCDAVMNT	- Availment Input
LCDTRPAY	- Payment Input
LCDTRANF	- Transfer Input
LCDEPMNT	- Manual Liquidation Input
LCDTREAS	- Contract Reassign
LCDTRAUT	- Amend Confirmation Input

On launching the Online form screen, user has to input the respective values to create the contract. Form may have the different user-defined actions like Product-Default, Enrich, and Subsystem-Pickup while creating contract. Once all the user-defined actions performed finally user has to save the contract.

4. Screen Development

Design and development of an Online Form function id is similar to any other function Ids. This section briefs the steps in designing an Online Form screen.

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

4.1 Header Information

Provide the header information as shown in the figure.

Function Generation		
		🖫 🗵 L C 🎸 🍕
Action New V	Function Type Parent	Function Category Transaction 🗸
Function Id UTDTXN01	Parent Function	Header Template None
Save XML Path	Parent Xml	Footer Template None
Search		
Preferences		
DataSource		
ListOfValues		
DataBlocks		
Screens		
FieldSets		
Actions		
CallForms		
LaunchForms		
Summary		

Fig: Online Form header Information

Note the following while providing header information.

- i) Name of the Online form :
 - Online Form name has to have the third character as 'D'. Ideally, the length of the name should be 8 characters.

Example: UTDTXN01 etc. are valid online form names

- ii) Online Form Category: Function Category has to be **Transaction**
- iii) Footer Template:

For Transaction screens, footer template has to be selected as **NONE**. System does not provide any default template for transaction screens; hence developer has to design the footer portion of the screen manually. Developer has to make sure that footer designed has generic fields like transaction status (TXNSTAT), authorization status (AUTHSTAT) etc.

For Online Process Flow Screens footer template should be selected as **PROCESS**.

iv) Function Type : Parent and Child functionality is supported for Online forms.

4.2 Preferences

Provide the menu details in the Preferences screen

ORACLE FLEXCUBE Development Workben	ch for Universal Banking — Mozilla Firefox				-	
) 👌 🗝 whf00iqw:8088/FCJRADTo	ol142GA/RADLoginServlet#nogo					☆
RACLE FLEXCUBE Developm	nent Workbench for Universal Banking					Adı
				Windows	Options	Sign O
ction Generation						
					×L	C 🖸
Action Load V	Function Type	arent v	Fu	unction Category Transaction V		
Function Id UTDTXN01	Parent Function		H	Header Template None V		
Save XML Path (N01_RAD.xml	Parent Xml			Footer Template None V		
ch	Preferences					
ferences aSource DNAtues aBlocks eens SSets SSets ons Forms nchForms nchForms nmary	 ☐ Head Office Functi ☐ logging Required ☐ Auto Authorization ☑ Module Auto Autohrization ☐ Field Log Required ☐ Mill Branch Accet ☐ Excel Export Requi ☐ Java Functions ☐ GateWay Screen Menu Detaile Parameter Value Mappings	55	Module UT Module Description Unit Module Group Branch Program Id Process Code SVN Repository URL	D Trust		
	Function Id	Module *		Module Description	String 🛨	
		UT	Dunit			
	UTSTXN01	UT	🔎 Unit	Trust		

Fig: Online Form Preferences

Note the following while providing Preferences for Online Forms.

i) Module name :

Module name is a mandatory field and has to be provided. It is recommended that the first two letters of the function id is kept as same as the module name. Naming of the generated package will be derived from the module code maintained

- *ii)* Script for the following tables will be generated by Workbench (menu details) which are essential for launching of an Online screen.
 - 1. SMTB_MENU
 - 2. SMTB_FCC_FCJ_MAPPING
 - 3. SMTB_FUNCTION_DESCRIPTION
 - 4. SMTB_ROLE_DETAILS

Type string of the Online screens will be generated as 'O' in *smtb_menu* table.

iii) Transaction specific action codes has to checked in the control string whichever applicable

Example: LIQUIDATE, ROLLOVER, REVERSAL etc.

4.3 Data Sources

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

😆 ORACLE FLEXCUBE Development Workbench	for Universal Banking — Mozi	lla Firefox						-		×
🛇 🖄 🖻 whf00iqw:8088/FCJRADTool142GA/RADLoginServlet#nogo									=	
	nt Workbench for Universal I	Banking							۸d	min
							Mr. dame	0-1		
Function Generation							Windows	Options	Sign O	_ ×
								X L	C 4	
									C V	9
Action Load V		Function Type Parent	~			Function Category Trans	action 🗸			
Function Id UTDTXN01		Parent Function				Header Template None	~			
Save XML Path (N01_RAD.xml BR	OWSE	Parent Xml				Footer Template None	~			
Search	Data Source Details							ę	þ =	🦻 ^
Preferences	Data Source	CONSOLIDATEDTXNTBL			Parent		×			
▲ DataSource		Yes 🗸			Relation					
► CONSOLIDATEDTXNTBL	Relation Type	One To One 🗸			Where Clause					
►vw_constxntbl	Multi Record	No 🗸			Default Order By					
►FCISDMSTXNIMAGETBL	PK Cols *	TRANSACTIONNUMBER	2]	,	Normal 🗸				
► TXNMATURITYTBL	PK Types *	VARCHAR2	2]	Type	Mandatory				
▶KYCTRANSACTIONTBL	Upload Table			1						
►vw_addinfo	Upload Where Clause		2							
► TXNRECEIPTSTBL	Fine Grained Service Detail	s Only								
▶ TRANSACTIONFEETBL										
► TXNDELIVERYTBL	Rest Data Block									
▶ TXNINTERESTDETAILSTBL	XSD Node									
▶ TXNDETAILSTBL	Rest Relation		\sim							
▶ LIMITORDERDETAILSTBL	Rest Relation Type	~								
► TXNINTERMEDIARYTBL	Rest PK Cols *		2							
► TXNSETTLEMENTTBL	Rest PK Types *									
▶ OVERRIDERETURNVALUESTBL										
DERIVEDOVERRIDERETVALTBL										
▶vw_overridereturnvaluestbl										
▶ VW_DERIVEDOVERRIDERETVALTBL										
▶ VWTXNSUMMARY										
▶VW_TXNINTERMEDIARYTBL										
VW TXNSETTLEMENT										

Fig: Adding data sources and maintaining properties

Note the following while creating data sources

- i) Master Data Source has to be a single entry data source.
- ii) Logical Relationships has to be maintained for all data sources except the parent
- Provide PK Cols and PK types for all data sources.
 If data source is a multi-record block, then make sure it has at least one more pk than its parent which helps to uniquely identify each record of multi record block
- iv) Minimize the use of views in the data sources. For transaction screens, system generated upload logic (fn_sys_upload_db) is not called within the system package. It is up to the developer to decide whether the system generated code can be used or not. If views are used in data sources, then this function should not be used by the developer.
- v) Usually for Online forms, a separate view can be used for summary purpose. This view will have all the fields required to be displayed in the summary.
 Example: UTDTXN01_SUMMARY

) 🖄 🗝 whf00iqw:8088/FCJRADTool142GA/R	IADLoginServlet#nogo	☆
RACLE' FLEXCUBE Development Workb	ench for Universal Banking	Admi
		Windows Options Sign Out
nction Generation		
		📓 🗷 🗉 C 🞸 🔮
Action Load V	Function Type Parent V	Function Category Transaction
Function Id UTDTXN01	Parent Function	Header Template None
Save XML Path 3N01_RAD.xml BROWSE	Parent Xml	Footer Template None
arch Data	Source Field Details	Refresh 📟 🦃
eferences JataSource COMPSCIDATEDTXNTBL DEALINGDATE COMPSCIDATE EUSDOPTION KYOCRECEIVED LOCKINFERIODFORGAN RECORD_STAT AUTH_STAT FXXALUEDATE FXXALUEDATE FXXALUEDATE FXXALUEDATE FXXALUEDATE FXXALUEDATE FXXALUEDATE FXXALUEDATE FXXALUEDATE FXXALUEDATE TRANSACTIONDATE FUNDID UNITHOLDERID MACER_ID MAKER_D MAKER_DT_STAMP REFERENCENUMBER TRANSACTIONCURRENCY	Column Name DEALINGDATE Block Name BLK_TRANSACTION_DETAILS Field Name DEALING_DATE	Data Type DATE Max.Length 7 Upload Table Column Nor Required in Upload Tables

Fig: Adding data sources fields and its properties

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

4.4 Data Blocks

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

ORACLE FLEXCUBE Development Workbench					-	ີ ເລ	3
Inction Generation	nt Workbench for Universal Ban	king		Window	ws Option	-	
				l	🛃 🗶 L		
Action Load Function Id UTDTXN01 Save XML Path 3001_RAD.xml Br		Function Type Parent v Parent Function Parent Xml		Function Category Transaction Header Template None Footer Template None			
earch	Block Properties				¢	- 🛛	9
Preferences DataSource ListOfValues DataBiocks ▶ BLK_TRANSACTION_DETAILS ▶ BLK_INTERNEDIARY DETAILS	Block Title Parent	L TRANSACTION_DETAILS	XSD Nod Comment IE Master Bloc Multi Recor Block Type	k Yes v d No v	P		
BLK_LOAD_OVERRIDE_DETALS BLK_DERWED_LOAD_DETALS BLK_XPC_DETALS BLK_DOCUMENT_DETALS BLK_DOCUMENT_DETALS BLK_ADOTIONAL_INFO BLK_SETILEMENTDTLS BLK_SETILEMENTDTLS BLK_VANOUNTDTLS BLK_VANOU		Datasource Available CSTB_UL_COLUMNS CONSOLIDATEDTXNITBL_A TXNADDMNCTBL CSTB_UL_COLUMNS_B CSTB_UL_COLUMNS_C	Datasourc Vwy_CONSTNUTBL FCISDMSTXNIMAGETBL TXMATURITYTBL TXMATURITYTBL TXMASACTIONFECTBL TXNDETALSTBL UNICTARETOETALSTB CONSOLIDATEDTXNTBL TXMISACTIONCERTTB	NL L			
+BIK_TOTALUNITS +BIK_FEEPAWMENTDTLS +BIK_DELVERVOTLS +BIK_CALCTYPE +BIK_CALCTYPE +BIK_SUMMARY							

Fig: Defining Data Blocks and maintaining its properties

Note the following while creating data blocks

- i) Master Data block has to be a single entry data source.
- ii) Provide Xsd node name if the block is normal and is required in gateway request
- Block order and block field order can be changed by re arranging blocks and block fields in the browser tree (drag and drop). Note that all units will have to be regenerated if block or block field order is changed (including xsd's)
- iv) Related currency fields should be placed above the amount field in the tree

Add block fields to the data block as required.

Search Carl Control of								
	2GA/RADLoginServlet#nogo			☆ ≡				
ORACLE FLEXCUBE Development V	ORACLE' FLEXCUBE Development Workbench for Universal Banking Admin							
=			Wind	dows Options Sign Out				
Function Generation				_ ×				
				🖫 🗶 L C 🎸 🧃				
Action Load V	Function Type Pa	rent 🗸	Function Category Transaction V					
Function Id UTDTXN01	Parent Function		Header Template None V					
Save XML Path (N01_RAD.xml BROW	/SE Parent Xml		Footer Template None					
Search	Block Field Properties			- 🛛 🗊 🤌 🔷				
Preferences	Field Name * FUNDID	XSD Tag	FUNDID	Required				
▶ DataSource	Field Label LBL_FUNDID	Comment ID	CMT_FUND_ID	Visible				
▶ ListOfValues	DataSource CONSOLIDATEDTXNT	BL Field Size	* 6	Read Only Calender Text				
DataBlocks	Column Name * FUNDID	Maximum Length	6	Popup Edit Required				
BLK_TRANSACTION_DETAILS	Data Type * Varchar2 v	Minimum Value		Uppercase Only				
FUNDID	Display Type Lov V	Maximum Value		LOV Validation Required				
UNITHOLDER_ID	Item Type Database Item V	Maximum Decimals		Input by LOV Only				
TRANSACTION_DATE	Parent Field	 TextArea Rows 		Not Required In Xsd				
TRANSACTIONNUMBER	Related Block Related Field	TextArea Columns		Format Required				
SUB_PAYMENTMODE	LOV Name LOV FUND	▼ Default Value		Hot Key Required				
STATUSFLAG	Off Line LOV Name	Preview Value		Focus Required				
MODULED_ID	Fieldset Name FST TRANSACTION I			Exact Fetch Joint Holder Hot Key				
PAYMENT_MODE	CLASSID			Required				
TRANSACTION_TYPE	00,0010							
ALLOTED_FLAG	ustom Attributes Events Bind Variables Return F	Related Field						
GROSSAMTIN_FBCCY NETAMTIN FBCCY	Return Fields Mapping		Default From Lov I	Definition				
	Query Column	Block Name	Return Field Name	^				
SETTLEMENTAMTIN_TANCC		BLK_TRANSACTION_DETAILS V	FUNDID ~					
REF_TYPE		BLK TRANSACTION DETAILS	FUNDIDENTIFICATIOI V					
SCRIPBASED								
TRANSACTIONCURRENCY	FUNDNAME	BLK_TRANSACTION_DETAILS V	FUNDNAME V					
	FUNDBASECURRENCY	BLK_TRANSACTION_DETAILS ~	FUNDBASECURREN V					

Fig: Attaching Block Fields and maintaining its properties

Note the following while attaching block fields to data blocks

- i) In case the field is not required in XSD, check not Required XSD
- ii) Ensure that Related Block and Field are given for Amount Fields
- iii) Minimize the use of query data sources by using DESC fields wherever possible. Note: Query data sources is rarely required for an Online Form screen; as launch form can be used for query only screens
- iv) Master block should contain reserved field names like AUTHSTAT, RECORDSTAT, ONCEAUTH, MODNO, MAKERID, CHECKERID, MAKERDTSTAMP and CHECKERDTSTAMP are added as part of the footer of the screen.

DRACLE FLEXCUBE Development Workbenc	th for Universal Banking — Mozilla Firefox					- 0
🛚 🔁 📽 whf00iqw:8088/FCJRADTo	ol142GA/RADLoginServlet#nogo					☆
RACLE FLEXCUBE Developm	nent Workbench for Universal Banking					Ad
					Wir	ndows Options Sign (
action Generation						
						🗄 🗙 L C 🞸
Action Load V	Function Type Parent	t v		Function Category Tra	nsaction v	
Function Id UTDTXN01	Parent Function			Header Template No		
Save XML Path (N01_RAD.xml	Parent Xml	_		Footer Template No		7
ch	Block Field Properties					- 🔍 🔯 💆
DINICELAIDIES	_					
BTN_VIEWBALANCE	Field Name * RECORDSTAT		XSD Tag RE	ECORDSTAT		Required
TOTALNOOFUNITS	Field Label LBL_RECORDSTAT	2	Comment ID CN	MT_RECORDSTAT		Visible Read Only
HIDFUNDID	DataSource CONSOLIDATEDTXNTBL		Field Size *			Calender Text
	Column Name * RECORD_STAT		Maximum Length			Popup Edit Required
	Data Type * Char 🗸		Minimum Value			Uppercase Only
	Display Type Select v		Maximum Value			LOV Validation Require
	Item Type Database Item V		Maximum Decimals			Input by LOV Only
SUBTYPEDESCRIPTION	Parent Field	~	TextArea Rows			Not Required In Xsd
HDNQUERYPARAM	Related Block	~	TextArea Columns			Format Required
RECORDSTAT	Related Field	~	Default Value			Hot Key Required
	LOV Name	~	Preview Value			Focus Required
MODNO	Fieldset Name FST FOOTER		Mask Id			Exact Fetch
MAKERID			Mask Iu			Joint Holder Hot Key Required
MAKERSTAMP	CLASSID					
	Custom Attributes Events Related Field					
CHECKERSTAMP						+ -
ONCEAUTH	Attribute Name		Attribute Value	Active	Move Up	Move D
POLICYNUMBER			Autoute funde	Yes v		
EXCHANGERATESTATUS	LBL_MNT_CLOSED	C		Yes 🗸		V
EXCHRATEENRICHER						
AMOUNTREJECTED						

4.5 Screens

Design the screen layout based on the requirement

ORACLE FLEXCUBE Development Workbe	-						-	٥	×
O A a= whf00iqw:8088/FCJRAD	DTool142GA/RADLoginServlet#nogo							☆	≡
ORACLE' FLEXCUBE Develop	opment Workbench for Universal Banking							Adr	nin
=					W	Vindows O	ptions	Sign Ou	
Function Generation						•	LC		. ×
Action Load V	-	action Type Parent V		Function Category	Transmitter				~
Action Load Function Id UTDTXN01		action Type Parent v		Function Category Header Template		,			
Save XML Path (N01_RAD.xml		Parent Xml		Footer Template		~			
Search	Screen Details						- R	-	
	Screen Details						- 2	Q →	
DataSource ListONAutes DataBlocks Screens CVS_IPOSUBMAIN HEADER BODY	Screen Title LBL_FC Screen Size Large Exit Button Type Default	~	O OBIEE Visible Query Query			Ŧ			
► TAB_UHFUNDDTLSTAB	Argument Name	Source Block Source Field	Argument Value	Target Block	Target Field	Active	^		
► TAB_TXNDTLSTAB	FBCCY	• •		BLK_VW_TXNSETTLEMENT V	FBCCY V	Yes v			
► TAB_OTHERDTLSTAB									
► TAB_INTERMEDIARY									
► TAB_LOADDTLS							-		
TAB_KYCDTLS									
TAB_ADDINFO									
► TAB_FEEDTLS ► TAB_CERTDTLS									
TAB_CERTDILS									
FOOTER									
► CVS SETTLEMENTDETAILS									
CVS TXNRECEIPTDTLS									
CVS FEEPAYMENTDTLS							~		
► CVS_DELIVERYDTLS							Ŧ		
CVS INTCALODTLS	\sim								~

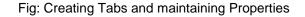
Fig: Designing Screens and providing Screen Properties

Note the following while creating screens

• One Screen should be identified as the main screen.

Add Tabs, sections and partitions as per the screen design

NRACLE FLEXCUBE Development Workb	ench for Univer	sal Banking — Mozill	la Firefox					-	۵	\times
○ & = whf00iqw:8088/FCJRAE	DTool142GA/R	RADLoginServlet#n	nogo						☆	≡
ORACLE FLEXCUBE Develo	opment Workb	ench for Universal I	Banking						Adr	nin
=							Windows	Options	Sign Or	ut
Function Generation										. ×
								×L	С 🞸	9
Action Load V			Function Type Parent Parent Function	~			Function Category Transaction Header Template None			
Save XML Path (N01_RAD.xml	BROWSE		Parent Xml				Footer Template None			
Search		Details					Dependent Field	ts 🕆 🖛	• 🗷 🧐	^
Preferences		Tab Name Tab Label	CVS_IPOSUBMAIN TAB_UHFUNDDTLSTAB [EB_UHANDFUND Data v]		Ø	ible				



Note the following when creating tabs and sections for the screen

- i) If the screen does not have multiple tabs, then only the TAB_MAIN needs to be used. TAB_HEADER should not contain any sections in this scenario
- ii) Normally Online forms are large screens with multiple tabs. In this case, all the tabs needs to be used .TAB_HEADER should contain the header information. TAB_MAIN should be the first tab in the body .Other tabs has to be added in the body portion as required
- iii) Footers are often designed by the developer for Online forms. Provide sections in TAB_FOOTER as required. Note that in large screens ,footer supports 4 partitions while other portions support 3 partitions

local CRACLE FLEXCUBE Development Workbench fo	or Universal Banking — Mozilla Fire	fox				— í	o ×
🛇 🗝 whf00iqw:8088/FCJRADTool14	42GA/RADLoginServlet#nogo						☆ =
ORACLE FLEXCUBE Development	t Workbench for Universal Banki	ing					Admin
=					Windows	Options Si	ign Out
Function Generation							_ ×
						× L C	77 🧃
Action Load V		Function Type Parent v			Function Category Transaction V		
Function Id UTDTXN01	Pa	arent Function			Header Template None V		
Save XML Path (N01_RAD.xml BRO	OWSE	Parent Xml			Footer Template None		
Search	Section Details					- 🔊	1 🦃 🔨
DataBlocks Screens CVS.JPOSUBMAIN HEADER	Section Name SEC Section Label	_UHFUND	٩	✓ Visible ☐ Collapse ☐ Multiple Section			
▲ BODY ▲ TAB_UHFUNDDTLSTAB SEC_UHFUND	Partition Details					+ F	
TAB TXNDTLSTAB	Partition SI No		Partition Name		Width Sub-partitions	^	
TAB OTHERDTLSTAB	1	TXN01_SC1_T2_S1_P1			100 🗸 🛛 2 🗸		
► TAB_INTERMEDIARY							
► TAB_LOADDTLS							
► TAB_KYCDTLS							j –
► TAB_ADDINFO							
► TAB_FEEDTLS							

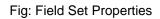
Fig: Section Properties

Multiple Screens can be designed if required.

4.6 Field Sets

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

CRACLE FLEXCUBE Development	t Workbench for	Universal Banking — Mozill	a Firefox							- 6	y ×
○ 🗞 🗝 whf00iqw:8088/F	FCJRADTool14	2GA/RADLoginServlet#n	ogo								☆ =
ORACLE' FLEXCUBE	Development	Workbench for Universal E	3anking								Admin
≡									W	/indows Options Si	gn Out
Function Generation											_ ×
										🗄 🗙 L C	V 🧐
Action Load	~		Function Type Parent	~			F	unction Category Tr	ansaction V		
Function Id UTDTX	(N01		Parent Function					Header Template No	ne 🗸		
Save XML Path (N01_R	RAD.xml BROV	WSE	Parent Xml					Footer Template	ne	~	
Search		Fieldset Properties									in 1997 (1997)
Preferences > DataSource > ListOfValues > DataBlocks > Screens A FieldSets FST_SET_NAV FST_TRANSACTION_INFO FST_TRANSACTION_VALUE		Fieldset Label Data Block Multi Record	FST_TRANSACTION_NFO LBL_TXNINFO BLK_TRANSACTION_DETAILS No ~ Single ~	2 5 ¥		Fieldset Typ Screen Nam Screen Portio Tab Nam Section Nam Partition Nam	e CVS_IPOSUBM n Header e TAB_TXNHEAD e SEC_TXN01_S	C1_T1_S1		 Horizontal Fieldset ReadOnly Navigation Button Navigation Button I Width ✓ Visible 	
FST_UHFUND FST_TXNDTLS_F1			Data Block Field	s		Fiel	dSet Fields	Subpartition Name	^		
FST_TXNDTLS_F2		TXNPAYM	ENTDETAILS	^		STATUSFLAG	3	1			
FST_TXNDTLS_F3		BTNPRO			0	HIDFUNDID		1	•		
FST_OTHER		BTNALLO				HIDUNITHOL	DERID	1	•		
FST_FEEDTLS FST TXNINTERMEDIARY		BTN_VIEV TOTALNO	VBALANCE		44	HIDTXNCUR	RENCY		•		
FST BTNLOADDET		BRANCHO				HIDTXNDATE		[<u>1</u>	·		
FST_LOADDTLS_F1					0	HIDEXRATE			•		
FST_LOADDTLS_F2					0	MODULED_I	D				
FST_KYCDTLS_F1					, [PAYMENT_M	ODE	<u>[1 · · · · · · · · · · · · · · · · · · ·</u>	· ·		
FST_KYCDTLS_F2											
FST_KYCDTLS_F3											
FST_ADDINFO	~										~



Note the following when attaching field to a field set

i) If a field is not required in the screen, but kept as hidden and value defaulted; then **The field has to be made invisible and attached to a field set**. If it is not attached to any fields set, the screen html won't contain the field and may result in script error while accessing the field.

4.7 <u>Actions</u>

Mention the web service and amendable information in Actions Screen

ORACLE FLEXCUBE Development Workbend	ich for Universal Banking	g — Mozilla Firefox					-
🔿 🖄 🗝 whf00iqw:8088/FCJRADToo	ol142GA/RADLogins	Servlet#nogo					
	nent Workbench for U	niversal Banking					
						Win	ndows Options S
Function Generation							
							🖁 🗙 L (
Action Load V		Function	Type Parent V		Function Category	Transaction	v
Function Id UTDTXN01		Parent Fun	ction		Header Template	None v	
Save XML Path (N01_RAD.xml B	ROWSE	Parent	Xml		Footer Template	None	v
Search	Form Actions						
Preferences							P
* DataSource	XSD Type Ic	dentifier IPOSubscriptio		Service Name FC XSD Module Folder	CISIPOSubscription	1	
ListOfValues	Oper	Rest Enabl			OSubscription		
DataBlocks		Fine Graine		Rest Service Name	Coubscription		
		Required					
FieldSets							
	Web Service	Action Code	Operation Code	Action Stage Type	Rest Enabled	Amendables	Comment Coc ^
CallForms							
		QUERY	QueryIPOSubscription			Amendables	Comments
		QUERY	QueryIPOSubscription CreateIPOSubscription			Amendables Amendables	
	_				-		Comments
		NEW	CreatelPOSubscription			Amendables	Comments Comments
		NEW MODIFY	CreatelPOSubscription ModifyIPOSubscription			Amendables Amendables	Comments Comments Comments
		NEW MODIFY AUTHORIZE	CreatelPOSubscription ModifyIPOSubscription AuthorizeIPOSubscription			Amendables Amendables Amendables	Comments Comments Comments
		NEW MODIFY AUTHORIZE DELETE	CreatelPOSubscription ModifyIPOSubscription AuthorizeIPOSubscription			Amendables Amendables Amendables Amendables	Comments Comments Comments Comments
		NEW MODIFY AUTHORIZE DELETE CLOSE	CreatelPOSubscription ModifyIPOSubscription AuthorizeIPOSubscription			Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments
LaunchForms		NEW MODIFY AUTHORIZE DELETE CLOSE REOPEN	CreatelPOSubscription ModifyIPOSubscription AuthorizeIPOSubscription DeleteIPOSubscription			Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments Comments
		NEW MODIFY AUTHORIZE DELETE CLOSE REOPEN REVERSE	CreatelPOSubscription ModifyIPOSubscription AuthorizeIPOSubscription DeleteIPOSubscription			Amendables Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments Comments
		NEW MODIFY AUTHORIZE DELETE CLOSE REOPEN REVERSE ROLLOVER	CreatelPOSubscription ModifyIPOSubscription AuthorizeIPOSubscription DeleteIPOSubscription			Amendables Amendables Amendables Amendables Amendables Amendables Amendables	Comments Comments Comments Comments Comments Comments Comments

Fig: Actions Screen

Note the following while maintaining web services and amendable information

- i) Online forms will generate Type XSD and Message XSD. Operation specific message xsd's will be generated.
 Example: for the example given in the figure, name of the xsd generated will be UT-FCISTransaction-Types.xsd (Type XSD for UT) UT-QueryTransaction-Res-Full-MSG.xsd (Create Message XSD for UT) UT-QueryTransaction-Req-IO-MSG.xsd (Create Message XSD for UT) UT-CreateTransaction -Res-Full-MSG.xsd (Create Message XSD for UT) UT-CreateTransaction -Res-Full-MSG.xsd (Create Message XSD for UT)
 i) Operation Id and Operation Code need be maintained for the above mentioned reason
- iii) Amendable information has to be maintained similar to any other function ids.

4.8 Launch Forms

Launch Forms can be attached to Online form screen.

🛯 🍅 ORACLE FLEXCUBE Development Workbench fo	or Universal Banking — Mozi	IIa Firefox					-		×
🛇 👌 🖻 whf00iqw:8088/FCJRADTool14	12GA/RADLoginServlet#r	nogo						5	? ≡
ORACLE FLEXCUBE Development	Workbench for Universal B	Banking						A	dmin
=						Windows	Options	Sign	Out
Function Generation									_ ×
							×L	С	V 🧐
Action Load 🗸		Function Type Parent	v	Functio	n Category	Transaction 🖌			
Function Id UTDTXN01		Parent Function		Heade	er Template	None 🗸			
Save XML Path N01_RAD.xml BRO	WSE	Parent Xml		Foote	er Template	None 🗸			
Search	Launch Form Details								🧐 ^
Preferences									
▶ DataSource						Screen Argumen	ts 🕂 🗖		
▶ ListOfValues			Function ID		Active	Туре	_		
▶ DataBlocks	UTDFNBAL		I unction ib		Yes v	Launch Form v	1		
Screens	OTDINDAL				103 +	Ladient on +	, 		
▶ FieldSets Actions									
CallForms									
LaunchForms									
Summary									

Screen Arguments should be maintained for the launch form to query the proper contract record from the main online functions.

6 ORACLE FLEXCUBE Development Workbench	n for Universal Banking — Mozilla Fi	refox				-		×
🗘 👌 🖻 whf00iqw:8088/FCJRADTool	142GA/RADLoginServlet#nogo						☆	≡
	nt Workbench for Universal Bank	ing					Adn	nin
=					Windows	Options	Sign Ou	at
Function Generation								×
						×L	C 7	 (g)
Action Load V Function Id UTDTXN01 Save XML Path N01_RAD.xml BR	P	Function Type Parent v arent Function Parent Xml			Function Category Transaction V Header Template None V Footer Template None V			
Search	Call Form Arguments			×				s ^
Preferences PolataSource ListOWalues PolataBlocks Screens FieldSets Actions CaliForms LaunchForms Summary	Argument Name UHID ACTION_CODE	Source Block	Source Field	Populate Reset Argument Va ^ EXECUTEQUEF	Active Type Yes Launch Form			

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

4.9 Call Forms

Call forms can be attached to Online form. Each call form should be mapped to Parent Data Block, Parent Data Source and proper relations should be maintained with parent data source of main online form.

Function Generation Image: Comparison of the comparison	
Image: Second	ŝ
Function Generation Image: Comparison of the Comparison	Admi
Action Load v Function Type Parent v Function Category Transaction v Function Id UTDTXN01 Parent Function Header Template None v Save XML Path (N01 RAD.xm) BROWSE Parent Xml Footer Template None v Search Call Form Details Preferences Screen Arguments Dependent Fields • • • * Licio/Values Parent Data Block Parent Data Source Relation Relation Type C • * Licio/Values • • • • • • * Consolid Consolid Consolid Ared Transaction © • • • • • • * Consolid	Sign Out
Action Load Function Type Parent Function Category Transaction Function Id UTDTXN01 Parent Function Header Template None Save XML Path (X01 RAD xm) BROWSE Parent Xml Footer Template None Search Call Form Details Footer Template None Preferences DataSource Screen Arguments Dependent Fields • • • ListOWalues Function ID Parent Data Block Parent DataSource Relation Relation Type C ^ Screens • I'UTCTXNPA BLK_TRANSACTION_DETAILS • CONSOLIDATEDTXNTBL * CONSOLIDATEDTXNTBL TRANSACTION © One To One • • PetaldStass • UTCPAYDT * BLK_TRANSACTION_DETAILS • CONSOLIDATEDTXNTBL * CONSOLIDATEDTXNTBL TRANSACTION © One * •	_
Function Id UTDTXN01 Parent Function Header Template None Save XML Path 2401_RAD.xml BROWSE Parent Xml Footer Template None Search Call Form Details Preferences DataSource Screen Arguments Dependent Fields • • DataSource Introduction ID Parent Data Block Parent DataSource VisioNalues Introduction ID Parent Data Block Parent DataSource VisioNalues Introduction ID Parent Data Block Parent DataSource VisioNalues Introduction ID Parent Data Block Parent DataSource Parent Screens Introduction ID Parent Data Block Parent DataSource VisioNalue Introduction ID Parent Data Block Parent DataSource VisioNalue Introduction ID Parent Data Block Parent DataSource Screens Introduction ID Parent Data Block Parent DataSource VisioNalue Introduction ID Extension ID ConsolidateDTXNTBL TRAINSACTION © fone To one Piekloses Introduction ID Extension ID ConsolidateDTXNTBL TRAINSACTION © fone © one To one VitCPAYDT B	С 🎸
Save XML Path [N01_RAD.xm] BROWSE Parent Xml Footer Template None Search Call Form Details Preferences DataSource * ListOValues Parent Dia Block Screen Arguments Dependent Fields • DataBlocks • ConsolidDatEDTXNTBL • UTCTXNPA • BLK_TRANSACTION_DETAILS • UTCTXNPA • BLK_TRANSACTION_DETAILS • UTCTXNPA • UTCTXNPA • BLK_TRANSACTION_DETAILS • ONSOLIDATEDTXNTBL • UTCTXNPA • BLK_TRANSACTION_DETAILS • ONSOLIDATEDTXNTBL • UTCPAYDT • UTCPAYDT	
Search Call Form Details Preferences PlatsOnure Screen Arguments Dependent Fields Parent Data Block	
Preferences PataSource PlataSource Relation Relation Type C^ PlataSource Relation Relation Type C^ PlataSource Relation Relation Type C^ PlataSource Relation Relation PlataSource C^ PlataSource Relation Relation PlataSource Relation Relation PlataSource Relation Relation PlataSource C^ PlataSource Relation Relation Relation PlataSource Relation Relation Relation PlataSource Relation Relation Relation Relation Relation R	
* DataSource Screen Arguments Dependent Fields * • * LIS/OValues Image: Screen Arguments Dependent Fields * • * DataBlocks Image: Screen Arguments Dependent Fields * • * Screen S Image: UTCTXNPA IBLK_TRANSACTION_DETAILS * CONSOLIDATEDTXNTBL TRANSACTION © • * FieldS4S Image: UTCTXNAD IBLK_TRANSACTION_DETAILS * CONSOLIDATEDTXNTBL TRANSACTION © • • Actions Image: UTCPAYDT IBLK_TRANSACTION_DETAILS * CONSOLIDATEDTXNTBL TRANSACTION © • •	1
* DataSource Screen Arguments Dependent Fields * • * LIS/OValues Image: Screen Arguments Dependent Fields * • * DataBlocks Image: Screen Arguments Dependent Fields * • * Screen S Image: UTCTXNPA IBLK_TRANSACTION_DETAILS * CONSOLIDATEDTXNTBL TRANSACTION © • * FieldS4S Image: UTCTXNAD IBLK_TRANSACTION_DETAILS * CONSOLIDATEDTXNTBL TRANSACTION © • • Actions Image: UTCPAYDT IBLK_TRANSACTION_DETAILS * CONSOLIDATEDTXNTBL TRANSACTION © • •	
ListOfValues ListOfValues Lation ID Function ID Func	
DataBlocks Function ID Parent Data Block Parent DataSource Relation Relation Relation C^ Screens UTCTXNPA BLK_TRANSACTION_DETAILS CONSOLIDATEDTXNTBL C	
Screens UTCTXNPA BLK_TRANSACTION_DETAILS CONSOLIDATEDTXNTBL	
Actions UTCPAYDT BLK_TRANSACTION_DETAILS V CONSOLIDATEDTXNTBL V CONSOLIDATEDTXNTBL TRANSACTION One To One V	
Gaironis	
LaunchForms	
Summary	

Sreen Arguments should be given to each callform. So that the call form will display the respective data of calling main function.

Dependent Fields are required to re default the call form values when the user changes input data in the main form.

Each of the subsytem pickup logic will have to be coded by the developer in release specific packages. Processing logic (sub system pickup) for the attached call forms has to be called from the main form package.

4.9.1 Sub System Pickup/Processing

Subsystem pickup refers to the process of picking up the values in sub systems. Normally values in sub systems will be defaulted based on the data given in the main screen of the online form.

1) Defaulting of sub system

After providing values in the main screen, user may click on any sub system to view or change the value.

On clicking the sub system for the first time, sub system values will be defaulted based on the values provided in the main screen. Action code passed will be **SUBSYSPKP**. The code for defaulting will have to written by the developer in corresponding hook packages in function

Fn_Post_Subsys_Pickup

In this case SUBSYSSTAT for all subsystems will go as 'D' and processing done based on this flag for each sub system (call form). Note that SUBSYSPKP action will default values for all subsystems and not only the sub system being launched Example:

MICTRMIS:D;ISCTRSTL:D;TACTRTAX:D;CSCTRUDF:D;CFCTROCH:D;CSCTRADV:D;FTC CGCLM:D;

If user saves the contract without visiting any call forms, then all the subs systems will be defaulted before saving

2) Uploading of sub system

If after launching the subsystem with defaulted values; User changes the value in subsystem; the new user input values has to be uploaded to the system. Hence while saving, *the*

subsystems which has been modified by user will be uploaded while others will be defaulted.

In this case SUBSYSSTAT for the subsystem which has been modified will go as 'U' .Developer has to write code for processing based on the flag

Example: if user changes MIS details (MICTRMIS) from what was defaulted; then SUBSYSSTAT will go as

MICTRMIS:U;ISCTRSTL:D;TACTRTAX:D;CSCTRUDF:D;CFCTROCH:D;CSCTRADV:D;FTC CGCLM:D;

3) Re defaulting of sub system

After launching and changing subsystem values; if user changes any values in main screen which are dependent field for the subsystem: subsystem values will have to be defaulted again based on the new main screen values. Hence the sub system will be re defaulted. In this case value entered by the user in subs system will be lost.

In this case SUBSYSSTAT for the subsystem whose dependent fields has been modified will go as 'R'. .Developer has to write code for processing based on the flag Example: In a Funds Transfer Contract Input Screen, assume that charge subsystem (CFCTROCH) is dependent on the values entered for debit and credit account. After launching the sub system and changing the charges manually; if user changes the account again the charges will have to re defaulted. The manually entered charges will not be considered. SUBSYSSTAT will go as

MICTRMIS:U;ISCTRSTL:D;TACTRTAX:D;CSCTRUDF:D;CFCTROCH:R;CSCTRADV:D;FTC CGCLM:D;

Values for other subsystems will depend on each of their dependencies.

4.10 Summary

Summary screens can be designed for Online Form if required

ORACLE FLEXCUBE Development Workbench	n for Universal Banking — Moz	illa Firefox									-		×
Q ≧ ≈ whf00igw:8088/FCJRADTool												5	2 ≡
• • white idw:suss/FCJRAD1001	142GA/RADLOgINServiet#	nogo										Z	- 1
ORACLE FLEXCUBE Development	nt Workbench for Universal	Banking										A	dmin
=										Windows	s Options	Sign	Out
Function Generation													_ >
											🗄 🗙 🛛	- C	V 🧐
Action Load V		Function Type Parent V					Fu	nction Category	Transacti	ion 🗸			
Function Id UTDTXN01		Parent Function					H	leader Template	None	~			
Save XML Path (N01_RAD.xml BR	OWSE	Parent Xml						Footer Template	None	~			
Search	Summary Details											C	3 🧐 🥤
Preferences	Title					Default Where Claus		ISACTIONTYPE	= '01'				
► DataSource	Data Blocks	BLK SUMMARY V				Default Order B		SACHONTTLE	- 01				
▶ ListOfValues	Data Source	VWTXNSUMMARY ~				Multi Branch When		SACTIONTYPE	= '01'				
▶ DataBlocks	Summary Type	Bulk Authorization 🗸				Claus	e						
▶ Screens	Summary Screen Size	Medium v				Main Summary Scree		/ebServices Reg					
▶ FieldSets		Criteria Based Search						vebbervices Req	uireu				
Actions	Data Block Fields Custom	Buttons Fields Ordering											
CallForms													
LaunchForms													
Summary		Data Block Fields				Fields Selected	Query	Properties	^				
		PAYMENTMODE	^			AUTHSTAT		Properties					
						RECORDSTAT	✓	Properties					
				44		STATUS		Properties					
				44		TRANSACTIONNUMBER		Properties					
						REFERENCENUMBER		Properties					
						FUNDID		Properties					
							_		~				
			~		1			,					

4.11 Preview

The figure shows the preview of the Online form Input screen developed

<u>_</u> ◆				×
New Enter Query				
Transaction Information		Transactio	on Value	^
Unit Holder ID *	Transaction Number		Transaction Mode * Amount V	
Find UH	Transaction Currency		Units Applied	
Fund ID *	Reference Number		Amount Applied	
Order Received Date	Order Received Time (HH24:MI)		Percent Applied	
Order Received Time Zone	Transaction Type		Enrich Transact	
Transaction Date *	Description		Reset Transact	ion
Payment Mode *	Transaction Category			
Payment Mode Description	Description			
UH And Fund Transaction Other Intermedian	y Load Override KYC And Document Add Ir	Fee Payment Certificate Limit Order		
CIF Number	ISIN No			
UH Deal	Fund Name			
Unit Holder Name	Fund Base Currency			
UH Signature	Price Currency			
	Currency of Expraccion No			✓
Settlement Details Receipt Details Pro	oject Allocation Details Allocation Details	View Balance		
Input By	DateTime	Mod No		
Authorized By	DateTime	Authorization Status Unauthorized v		Ok Exit
		Record Status Open 🗸		

Execute Query ⊄ Adva	inced Search 🗛 D						
Execute Query Q Auva	inced Search * 2 K						
Authorization S	tatus	~		Record Status	~		
Transaction Nu	mber	ρ		Reference Number		Q	
Fu	nd ID			Unit Holder ID		P	
Currency of Expres				Transaction Type		~	
Transa				Transaction Date		2	
Transaction I				Limit Order		ρ	
Dealing		-2	Tra	ansaction Currency			
Transaction S	tatus						
t Selected		~					
cords per page 15 🗸 🕅	1 of 1	Go to Page					
cords per page 15 v			saction Number	Reference Number	Fund ID	Unit Holder ID	Cur
			saction Number	Reference Number	Fund ID	Unit Holder ID	Cur
			saction Number	Reference Number	Fund ID	Unit Holder ID	Cu
			saction Number	Reference Number	Fund ID	Unit Holder ID	Cur
			saction Number	Reference Number	Fund ID	Unit Holder ID	Cur
			saction Number	Reference Number	Fund ID	Unit Holder ID	Cur
			saction Number	Reference Number	Fund ID	Unit Holder ID	Cur
			saction Number	Reference Number	Fund ID	Unit Holder ID	Cur
Authorization Status	Record Status	Status Tran	saction Number	Reference Number	Fund ID	Unit Holder ID	
	Record Status	Status Tran	saction Number	Reference Number	Fund ID	Unit Holder ID	
Authorization Status	Record Status	Status Tran	saction Number	Reference Number	Fund ID	Unit Holder ID	

The figure shows the preview of the Online form Summary screen developed

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

5. Generated Units

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

5.1 Front End Units

5.1.1 Language xml

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

Example – UTDTXN01.xml (UIXML for UT Screen)

5.1.2 SYS JavaScript File

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

Example – UTDTXN01_SYS.js (JS for UT Screen)

5.1.3 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 *Example* – UTDTXN01_KERNEL.js (JS for KERNEL Release) *Example* – UTDTXN01_CLUSTER.js (JS for CLUSTER Release) *Example* – UTDTXN01_CUSTOM.js (JS for CUSTOM Release)

5.2 Database Units

5.2.1 Static Scripts

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

5.2.2 System Packages

Main package would be generated by the Tool and should not be modified by the developer.

Example – utpks_utdtxn01_main.spc, utpks_utdtxn01_main.sql (Main Package for UT screen)

Main package contains functions for:

- Converting Ts to PL/SQL Composite Type
- Calling fn_main.

- Resolve Ref Numbers (fn_resolve_ref_numbers)
- Mandatory checks (fn_check_mandatory).
- Product Default (fn_product_default)
- Subsystem Pickup(fn_subsys_pickup)
- Enriching (fn_enrich)
- Default and validation(fn_default_and_validate)
- Uploading into DB tables(fn_upload_db)
- Processing the contract input values(fn_process)
- Querying(fn_query)
- Converting the Modified Composite Type again to TS

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

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

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

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

5.2.3 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. *Example* – utpks_utdtxn01_kernel.spc, utpks_utdtxn01_kernel.sql (Kernel Package) utpks_utdtxn01_cluster.spc, utpks_utdtxn01_cluster.sql (Cluster Package) utpks_utdtxn01_custom.spc, utpks_utdtxn01_custom.sql (Custom Package)

5.3 Other Units

5.3.1 <u>Xsd</u>

Only Type XSD and message XSD will be generated for an Online Form function Id. This type xsd will be used in the type xsd of any function which uses the particular online form. *Example* – UT-FCISTransaction-Types.xsd (Type XSD for UT) UT-CreateTransaction -Req-Full-MSG.xsd (Create Message XSD for UT) UT-CreateTransaction -Req-PK-MSG.xsd (Create Message XSD for UT) UT-CreateTransaction -Res-Full-MSG.xsd (Create Message XSD for UT) UT-CreateTransaction -Res-Full-MSG.xsd (Create Message XSD for UT) UT-CreateTransaction -Res-Full-MSG.xsd (Create Message XSD for UT)

6. Extensible Development

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

6.1 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 UTDTXN01_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 ()*

6.2 Extensibility in Backend Coding

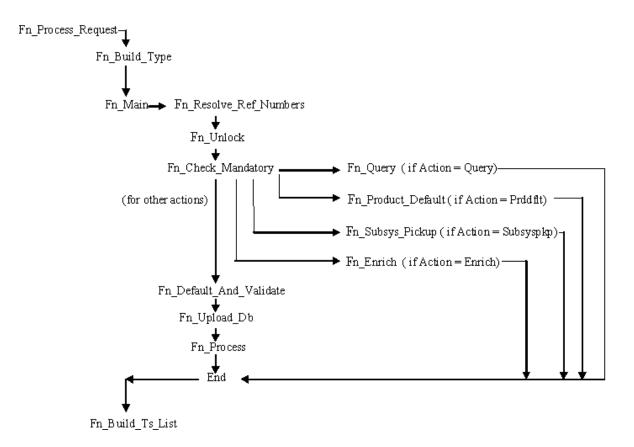
For online forms, generated code does not provide any business logic. Insert statements won't be present as part of generated code in online packages. Developer has to write the business logic in release specific packages (or make call I to server functions from release specific packages).

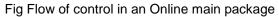
Hooks will be provided in the following stages Resolving reference numbers

- Checking mandatory fields
- Defaulting and validating
- Uploading to db
- Process
- Subsystem pickup
- Enrich
- Product Default
- Query

Note that the system generated code for uploading; defaulting etc. (*fn_sys_default_and_validate, fn_sys_upload_db etc.*) won't be called by the main package in online flow. If it is required, developer has to call it explicitly from release specific packages.

Note that in online flow, upload to base tables happens first and processing is done on the inserted data after uploading. After processing, the response type will be build





Release specific code has to be written in the Hook Packages generated. Different functions available in the Hook Package of an Online Form are:

1) Skip Handler : Pr_Skip_Handler

This can be used to skip the logic written in another release. Example: logic written in KERNEL release can be skipped in CLUSTER release

2) Fn Main

This is called form the fn_main in main package.

3) Fn_pre_resolve_ref_numbers

4) Fn_post_resolve_ref_numbers

This function validates the reference number. It is called from fn_ resolve_ref_numbers of the main package.

5) Fn_pre_unlock

6) Fn_post_unlock

This function holds the contract level validations and modification logic for existing contract. It is called from fn_unlock of main package.

7) Fn_pre_check_mandatory

8) Fn_post_check_mandatory

Any mandatory checks can be validated here. It is called from fn_chchk_mandatory of main package.

9) Fn_pre_query

10) Fn_post_query

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

11) Fn_pre_product_default

12) Fn_post_product_default

This function has the logic to default the values for the contract based on the product maintenance. It is called from fn_product_default of main package.

13) Fn_pre_subsys_pickup

14) Fn_post_subsys_pickup

This function does the subsystem pickup for the subsystem's (call form's) as per product maintenance for the contract. It is called from fn_subsys_pickup of main package.

15) Fn_pre_enrich

16) Fn_post_enrich

After product default, user can default others values. That logic can be put here. it is called from fn_enrich of main package.

17) Fn_pre_default_and_validate

18) 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.

19) Fn_pre_upload_db

20) Fn_post_upload_db

Any logic while uploading data to tables can be written here. It is called from fn_upload_db of main package.

21) Fn_pre_process

22) Fn_post_process

These hook functions are specific to transaction online form screens. This function should have the call to all the server functions which process the input data for the contract as per the functionality. These are called from fn_process of the main package.



Development of Online Forms [May] [2023] Version 14.7.2.0.0

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

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

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

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

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

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

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

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

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