Development of Online Forms Oracle Banking Trade Finance Release 14.6.1.0.0 [August][2022]



Contents

1. Prefa	ace	3
1.1	Audience	3
1.2	Related Documents	3
2. Intro	oduction	3
2.1	How to use this Guide	3
3. Over	rview of Online Form	4
4. Scre	en Development	4
4.1	Header Information	5
4.2	Preferences	5
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	15
4.10	Summary	16
4.11	Preview	18
5. Gene	erated Units	19
5.1	Front End Units	19
5.1.1	Language xml	19
5.1.2	SYS JavaScript File	20
5.1.3	Release Type Specific JavaScript File	20
5.2	Data Base Units	
5.2.1		
5.2.2	System Packages	20
5.2.3	Hook Packages	21
5.3	Other Units	21
5.3.1		
	nsible Development	
6.1	Extensibility in JavaScript Coding	
6.2	Extensibility in Backend Coding	22

1. Preface

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

1.1 Audience

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

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

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

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

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 a Online Form function id is similar to any other function Ids. This section briefs the steps in designing a 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.

			🖫 🗷 🗏 🖗 🌾
Action New Function Id LCDTRONL Save XML Path	Function Type Parent Parent Function Parent Xml	Function Category Transaction	
Preferences DataSource ListOrValues DataBlocks Screens FieldSets Actions LaunchForms LaunchForms Summary			

Fig 4.1 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: LCDTRONL, BCDTRONL 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

Action Load -		Functio	n Type Parent 👻		Function Category Transaction			
Function Id LCDTRONL		Parent Fu			Header Template None 🔻			
Save XML Path LCDTRONL_F	BROWSE	Pare	ent Xml		Footer Template None	•		
earch	Preferences							5
🗀 Preferences		Head Office	• Function	Module	LC			
DataSource		🗖 Logging Re	equired	Module Description	A CONTRACTOR OF			
∃ 🛅 ListOfValues ∃ 🛅 DataBlocks		Auto Author	ization	Branch Program Id				
Screens		Tank Modifi	cations	Process Code				
🗄 🚞 FieldSets		Field Log R	tequired	SVN Repository URL				
CallForms		Multi Branci	h Access	Transaction Block Name	BLK_CONTRACT_DETAILS -			
LaunchForms		Excel Expor	t Required	Transaction Field	BRANCH			
Summary					Name			
	-				C	Control St	ring 🕂	
		Function Id		Module *	Module Description			*
	LCDTRONL LC		LC	×=	Letters Of Credit			
	LCSTRONL LC			E Letters Of Credit				

Fig 4.2 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 Onlne 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.

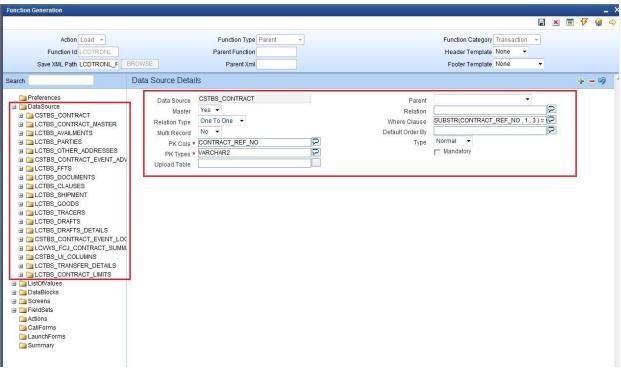


Fig 4.3 Adding data sources and maintaining properties

Note the following while creating data sources

- i) Master Data Source has to be a single entry data source.
- ii) Logical Relationships has to be maintained for all data sources except the parent
- iii) Provide PK Cols and PK types for all data sources.
 If data source is a multi record block, then make sure it has at least one more pk than its parent which helps to uniquely identify each record of multi record block
- *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: LCVWS_FCJ_CONTRACT_SUMMARY*

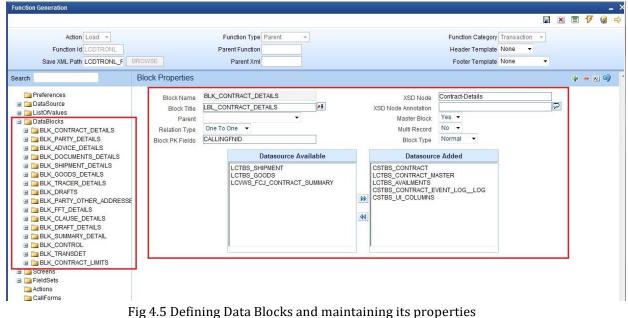
Function Generation				-
				🗶 🗏 🖗 🍕 🛛
Action Load 👻	Function Type Parent		Function Category Transaction 👻	
Function Id LCDTRONL	Parent Function		Header Template None 👻	
Save XML Path LCDTRONL_F BROWS	Parent Xml		Footer Template None 🔹	
Search Data	a Source Field Details			Refresh 🗕 🌍
Preferences	Column Name AUTH_STATUS	Data Type	CHAR	
	Block Name	MaxLength	1	
CSTBS_CONTRACT AUTH_STATUS	Field Name	Upload Table Column		
BOOK_DATE			Not Required in Upload Tables	
BRANCH				
CONTRACT_REF_NO				
EXTERNAL_REF_NO				
LATEST_EVENT_SEQ_NO				
LATEST_VERSION_NO				
MODULE_CODE				
SOURCE				
STOP_DATE				
USER_DEFINED_STATUS				
DSER_REF_NO				
CURR_EVENT_CODE				
LCTBS_CONTRACT_MASTER				
CTBS_AVAILMENTS Delta LCTBS_PARTIES				
LCTBS_PARTIES LCTBS_OTHER_ADDRESSES				
CSTBS_CONTRACT_EVENT_#				

Fig 4.4 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



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

- iii) 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

Function Generation			-					
						<u>,</u>	I 🗙 🗉 7	9
Action Load		Function Type	Parent -		Function Categ	ory Transaction 👻	1	
Function Id LCDTRONL		Parent Function			Header Templ	ate None 🔻		
Save XML Path LCDTRONL_F	BROWSE	Parent Xml			Footer Templ		-	
Search	Block Field Propert	ies					- 81	a 9
Preferences	Field Name *	PRTYTYP		VOD T-	PRTYTYP		Required	
DataSource		LBL_PRTYTYP	* E	XSD Tag				
ListOfValues	Tierd Edger			XSD Annotation	P		Visible	
🖃 🛅 DataBlocks	Dataovarou	LCTBS_PARTIES		Field Size *			F Read Only	
BLK_CONTRACT_DETAILS	Column Name *	PARTY_TYPE		Maximum Length	3		Calender Text	1
BLK_PARTY_DETAILS	Data Type *	Varchar2 👻		Minimum Value			Popup Edit Re	aquired
CONTREFN05	Display Type	Lov		Maximum Value			Uppercase Or	
PRTYTYP	Item Type	Database Item 👻		Maximum Decimals			Real resources and the states of	
DARTYDESC		-		TextArea Rows			C LOV Validation Required	1
PARTYCIFID	Parent Field			TextArea Columns			Input by LOV C	July
CUSTNAME	Related Block		*				MAN PERSONAL AND CONTRACTOR	
CUSTADDLIN1	Related Field	•		Default Value	e	-	Not Required I	In Xsd
CUSTADDLIN2	LOV Name	LOV_PART_TYPE	-	Preview Value			E Report Param	/eter
CUSTADDLIN3	Off Line LOV Name		-	Maskld	2	•		
CUSTADDLIN4		FST CONTRACT PAR						
COUNTRYCD	Fieldset Name		(TIES					
CUSTREFNO	Custom Attributes Eve	ents Bind Variables	Return Fields Rela	ated Field				
CUSTREFDATE	Return Fields Map	ping				Default From	m Lov Definition	
ISSBANK	0000	Column	1	Block Name		Return Field Nam	aa (Å	
ESN		Column					ile ile	
TEMPLATE_ID	PARTY_TYPE		BLK_PARTY_DET/	AILS 🔻	PRTY	TYP 🔻		
BLK ADVICE DETAILS	ITEM_VAL_DESC		BLK PARTY DET	AILS -	PARTY	DESC -		
BLK_DOCUMENTS_DETAILS			Land and the second second					
BLK_SHIPMENT_DETAILS								
BLK_GOODS_DETAILS								
BLK_TRACER_DETAILS								
BLK_DRAFTS								
BLK_PARTY_OTHER_ADDRES								
BLK_FFT_DETAILS								
BLK_CLAUSE_DETAILS								
BLK_DRAFT_DETAILS								
							-	

Add block fields to the data block as required.

Fig 4.6 Attaching Block Fields and maintaining its properties

Note the following while attaching block fields to data blocks

- i) In case the field is not required in XSD, check not Required XSD
- ii) Ensure that Related Block and Field are given for Amount Fields

iii) Minimize the use of query data sources by using DESC fields wherever possible. *Note: Query data sources is rarely required for a Online Form screen; as launch form can be used for query only screens*

 iv) Master block should contain reserved field names like TXNSTAT, AUTHSTAT and SUBSYSSTAT(this is not shown) as shown in the figure .These are reserved field names which are essential for an online form. These will be used by FLEXCUBE Infra while processing. Normally TXNSTAT and AUTHSTAT are added as part of the footer of the screen

COLUMN NAME	BLOCK FIELD NAME
CONTRACT_STATUS	TXNSTAT
AUTH_STATUS	AUTHSTAT
SUBSYSTEM_STAT	SUBSYSSTAT

Function	1 Generation								<u> </u>
									(🗏 🐼 🎯
	Action Load -		Function Type P	arent 👻		Fur	nction Category Transact	ion 👻	
				arent					
	Function Id LCDTRONL		Parent Function				ader Template None	T	
	Save XML Path LCDTRONL_F	BROWSE	Parent Xml			F	ooter Template None	-	
Search		Block Field Pro	perties						- R 🗖 🗐
		Field Nat	ne * CONSTAT		XSD Tag	CONSTAT			equired
	CUMMULATIVE	Field La	F	×=	XSD Annotation		2		isible
	DINITS	DataSou			Field Size *	1		Part Contract	ead Only
	FREQ		ne * CONTRACT_STATUS		Maximum Length	1		Read 10000	
	ALLOWREPAY				Minimum Value			All and a second	alender Text
	CLOSTYP		pe * Char 👻		Maximum Value				opup Edit Required
	TRANSBLE	Display Ty			Maximum Decimals			ΓU	ppercase Only
	MAYCONFIRM	Item Ty	pe Database Item 👻		TextArea Rows				OV Validation
	C REMARK	Parent Fi	bld	-					equired
	C RELLCREF	Related Blo	ick	•	TextArea Columns		*=		put by LOV Only
	BTN_DEFAULT	Related Fi	eld 🔻		Default Value			and the second se	ot Required In Xsd
	CONREFNOLOG	LOV Na	ne	-	Preview Value		25	L B	eport Parameter
	MAKER	Off Line LOV Nar	ne	•	Mask Id	12	<u></u>		
		Fieldset Na	ne FST_PRODDET						
		-	Events Related Field						
	TXNSTAT	Custom Attributes	Events Related Field						
	CONSTAT								+ -
	C AUTHSTAT		Attribute Name		Attribute Value	Active	Pos	ition	*
	BTN_NEXT	-	Attribute nume		Attribute Value	Acure			
	BTN_PREVIOUS	-							
	DAVAILESN BTN_CHARGES								
	BTN_CHARGES								
	BTN_TAX								
	BTN_COLLATERAL								
	BTN_EVENTS								
	BTN_LINKAGES								
	BTN_UDF								
	BTN_MIS								
	OF								-
	DERSIONLEL								
	ACKNREVCD								
	ACKDT								
	AMDDT								

4.5 Screens

Design the screen layout based on the requirement

			ent v				× = 1	3
Action Load -		Function Type Par	ent 👻		Function Category			
Function Id LCDTRONL	BROWSE	Parent Function			Header Template			
Save XML Path LCDTRONL_F	BROWSE	Parent Xml			Footer Template	None -		
rch	Screen Details						- <u>A</u>	
Preferences	Screen Name	CVS_MAIN		🔽 Main Sc	creen			
DataSource	Screen Title		s 🗾	Visible				
DataBlocks	Screen Size	Large	•					
Screens	Exit Button Type	Default Cancel						
CVS_MAIN								
BODY TAB_MAIN							+	
SEC_MAIN	Argument	Name Source Block	Source Field	Argument Value	Target Block	Target Field	Active	^
SEC_TOL	CONTREF		×]	BLK_CONTRACT_DETAILS *	CONREFNO -	Yes 🔻	
SEC_CUST	ESN ESN	· ·	*		BLK_CONTRACT_DETAILS -	LATEVNSEQNO -	Yes 🔻	
B TAB PREFERENCES								
TAB_PARTIES_LIMIT								
∃ □ TAB_SHIPMENT								
TAB_DOCUMENTS								
TAB_TRACERS TAB ADVICES								
CVS_DRAFT								
CVS_PRE_CLOSE								
CVS_TRANSDET								
FieldSets								*
Actions								
CallForms								
CallForms								

_ ×

Fig 4.7 Designing Screens and providing Screen Properties

Note the following while creating screens

• One Screen should be identified as the main screen.

nction Generation			
			🖪 🗵 🗐 🐼 🧐
Action Load 👻	Function Type Parent		Function Category Transaction
Function Id LCDTRONL	Parent Function		Header Template None 👻
Save XML Path LCDTRONL_F BROW	SE Parent Xm		Footer Template None -
arch Tab	Details		Dependent Fields 🐥 🗕 🗷 🗳
Preferences DataSource ListOfValues DataBlocks Screens C. S.MAIN	Screen Name CVS_MAIN Tab Name TAB_MAIN Tab Label LBL_MAIN A Tab Type Data ↓	r⊽ Visible	
BODY TAB_MAIN SEC_MAIN SEC_TOL SEC_STAT SEC_STAT TAB_PREFERENCES TAB_PARTIES TAB_PARTIES TAB_SHIPMENT TAB_DOCUMENTS			
TAB_TRACERS TAB_ADVICES TAB_ADVICES FOOTER CVS_DRAFT			
CVS_PRE_CLOSE CVS_TRANSDET FieldSets			

Add Tabs, sections and partitions as per the screen design

Fig 4.8 Creating Tabs and maintaining Properties

Note the following when creating tabs and sections for the screen

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

Function Generation				
				🗏 🖗 🗐 🖨
Action Load Function Id CDTRONL Save XML Path LCDTRONL_F BROW	P	Function Type Parent v arent Function Parent Xml	Function Category Transaction * Header Template None * Footer Template None *	
Search Sec	ction Details			- 🛯 🌍
Preferences DataSource ListOfValues DataBlocks DataBlocks	Section Name SEC_1 Section Label	roL Visible		
🗉 🧰 CVS_MAIN	Partition Details			+ -
HEADER BODY	Partition SI No	Partition Name	Width Sub-partition	s ^
🖃 🧰 TAB_MAIN	1	PART_TOL1	66 👻 2 👻	
SEC_MAIN	2	PART_TOL2	33 🗸 🗸	
■ SEC_STAT ■ TAB_PREFERENCES ■ TAB_PARTIES ■ TAB_PARTIES_LIMIT ■ TAB_SHIPMENT ■ TAB_SHIPMENT ■ TAB_DOCUMENTS ■ TAB_DOVICES ■ TAB_ADVICES ■ CVS_DRAFT ■ CVS_DVS_DRAFT ■ CVS_DVS_				
	Fig	g 4.8 Section Properties		

Multiple Screens can be designed if required.

4.6 Field Sets

Eur

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

Action Load -	Function Type Parent					Funct	ion Category Transaction	*
Function Id LCDTRONL	Parent Function						ler Template None 👻	
Save XML Path LCDTRONL_F	ROWSE Parent Xml	Footer Template None					-	
rch	Fieldset Properties							- <u>A</u>
Screens ^	Fieldset Name FST_PROD		9	Scre	en Name	CVS_MAIN	•	🔽 Horizontal Fields
FieldSets	Fieldset Label		S	cree	n Portion	Header	+	ReadOnly
ST_PRODDET	Data Block BLK_CONTRACT_DETAILS ~			Т	ab Name	TAB_HEADER	+	Navigation Butto
FST_REF	Multi Record No -		S	ecti	on Name	SEC_HEADER2	-	Visible
FST_USER_REF	View Type Single 👻			artiti	on Name	PART1		
FST_AMEND	Fieldset Height		Num	nber	Of Rows			
FST_CREDIT								
Data Block Fields						ield Set Fields	Subpartition Name	
FST_REV_DET	BTN_DRAFTS	Â	-			ielu set Fielus		
FST_TERMS	BTN_COMMISSION			Г			-	
FST_PREADVDESC	PRTYTYP			Г	BTN_P		-	
FST_BUTTONS	CRDLIN LINECID	E	D	V	CALLING	GENID	-	
FST CUSTOMER	INCAMDNO		PP					
ST_DRAWEE_DETAILS	EVENTCD BTN_CHARGES		44					
FST_INSURANCE_DETAILS	BTN_SETT							
FST_GOODS	BTN_TAX BTN_COLLATERAL							
FST_SHIPMENT	BTN_EVENTS							
ST FOOTER 1	BTN_LINKAGES	*						
FST_CONTRACT_PARTIES								
FST_CONTRACT_ADVICES								
FST_CONTRACT_DOCUMENT				_				
FST_CONTRACT_TRACERS								
ST_CONTRACT_FFTS								
FST_CONTRACT_CLAUSES								
FST_CONTRAT_DRAFTS_DET								
FST_VER								
FST_CONTROL	1							
FST_GAURENTEE								
FST_GUAR								
FST REIMUNDERTKNG								

Fig 4.9 Field Set Properties

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 Actions

Mention the web service and amendable information in Actions Screen

Action Load -		Function Type	Parent -		Function Category Transacti	on 👻	
Function Id LCDTRONL		Parent Function				 ▼ 	
Save XML Path LCDTRONL	F BROWSE	Parent Xml			Footer Template None	•	
rch	Form Actions						G
Preferences	XSD Type Iden	tifier Contract		Service Name	CUBSLCService	*1	
DataSource ListOfValues DataBlocks	Operatio						
🚞 Screens 🛅 FieldSets							+ -
CallForms	Web Service	Action Code	Ope	ration Code	Action Stage Type	Amendables	*
aunchForms		QUERY	QueryContract			Amendables	
🚞 Summary		NEW	CreateContract			Amendables	
		MODIFY	ModifyContract			Amendables	
		AUTHORIZE	AuthorizeContract			Amendables	
		DELETE	DeleteContract			Amendables	
		CLOSE	CloseContract			Amendables	
		REOPEN	ReopenContract			Amendables	
		REVERSE	ReverseContract			Amendables	
	V	ROLLOVER	RolloverContract			Amendables	
		CONFIRM				Amendables	
		LIQUIDATE				Amendables	
		SUMMARYQUERY			V		

Fig 4.10 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 LC-Contract-Types.xsd (Type XSD for LC Contract) LC-CreateContract-Req-Full-MSG.xsd (Create Message XSD for LC Contract) LC-CreateContract-Req-IO-MSG.xsd (Create Message XSD for LC Contract) LC-CreateContract-Res-Full-MSG.xsd (Create Message XSD for LC Contract) LC-CreateContract-Res-Full-MSG.xsd (Create Message XSD for LC Contract)

ii) 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.

Function Id Parent Function Header Template None Save XML Path LCDTRONL_F BROWSE Parent Xml Footer Template None Search Launch Form Details		
earch Launch Form Details		
Preferences DataSource DataSource Screen Argument Screen Argument		
DataSource DataSource Screen Argument Screen Argument		
La La Division Screen Argument		
DataBlocks	s + -	ients -
Screens		
EFIEldSets CSDEVENT Yes •		
Actions MSDALMSG Yes V		
Launchforms WSDMSPRV Yes -		

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

Function Generation								_ ×
							× ×	■ 梦 🥥 🌩
Action Load *		Functio	n Type Parent 👻		Functio	n Calegory Transa	ction -	
Function Id LCDTRONU		Parent Fu	unction		Heade	r Templale None	*	
Save XML Pain LCDTRONL	JE BROWS	Pare	ant Xmi		Foole	rTemplate	*	
Search	Laun	ch Form Details						16g - 1
Preferences								
DalaSource ListOf/alues							Screen Argun	norte
DalaBlocks				Function ID			Active	
Screens FieldSels		CSDEVENT					Yes -	
Actions							Yes -	
CaliForms							Yes 🔻	
Summary		Call Form Arguments						
		Call Form Arguments			· · · · · · · · · · · · · · · · · · ·			
					Populate Reset			
		Argument Name	Source Block	Source Field	Argument Value			
		CONTREF	BLK_CONTRACT_DETAILS					
		ACTION_CODE	↓ 	•	EXECUTEQUERY			
		L KONON_CODE			EXECCIEdOEKT			
					· · · · · · · · · · · · · · · · · · ·			
					Ok Cancel			
								*

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.

Action Load -		Function Type Parent		Function Category Transaction	-	
Function Id LCDTRONL		Parent Function		Header Template None 🔻		
Save XML Path LCDTRONL	FBROWSE	Parent Xml		Footer Template	•	
rch	Call Form Details					
Preferences						
ListOfValues				Screen Arguments Dep	pendent Fields 🕂 🔸	-
) 🧰 DataBlocks) 🧰 Screens	Function ID	Parent Data Block	Parent DataSource	Relation	Relation Type	^
i 🧰 FieldSets	CFCTRCOM	BLK_CONTRACT_DETAILS	✓ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
CallForms	CFCTRCHG	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
LaunchForms	ISCTRSTL	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
Carl Summary	LCCTRCLT	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	TACTRTAX	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	CSCTRLNK	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	CSCTRUDF	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To Many 💌	
	MICTRMIS	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	LCCBCLNK	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	CSCTRSPT	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	BCCTRPRF	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	CSCOFACT	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	BCCBRDET	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	CSCDOCTR	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	
	LCCILUTL	BLK_CONTRACT_DETAILS	✓ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻	

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

Dependant 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 .

i) 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; FTCCGCLM:D;

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

ii) 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;CSCTR ADV:D;FTCCGCLM:D;

iii) 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 susb 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 manully entered charges will not be considered. SUBSYSSTAT will go as

MICTRMIS:U;ISCTRSTL:D;TACTRTAX:D;CSCTRUDF:D;CFCTROCH:R;CSCTR ADV:D;FTCCGCLM: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

Function Generation											-
									×	3	🧐 d
Action Load -		Function Type Parent				Function Category Transaction					
Function Id LCDTRONL		Parent Function				Header 1	emplate None 👻				
Save XML Path LCDTRONL_F	BROWSE	Parent Xml				Footer 1	Template	•			
Search	Summary Details									[a 9
Preferences DataSource ListOfValues ListOfValues CataBlocks Careens FieldSets Catifons CaliForms LaunchForms LaunchForms LaunchForms	Title Data Blocks Data Source Summary Type Summary Screen Size	LBL_SUMMARY BLK_SUMMARY_DETAIL LCWWS_FCJ_CONTRACT_SUM Summary Medium stom Buttons Fields Ordering	1		Default Where Clause Default Order By Multi Branch Where Clause Main Summary Screen	MODULE_CO MODULE_CO MODULE_CO Required	DDE = 'LC' AND EXISTS	2			
		Data Block Fields			Fields Selected	Query	LOV Name		*		
	CLOSDT				AUTHSTAT			-			
	SETLMTH				CONSTAT	v		•	-		
CIFID		117			CONREFNO	N		•			
	MAXLIABAN	MAXCONTAMT MAXLIABAMT			PRDCD	V		-			
	CURRAVAI	L			CONTCCY	1		-			
	0.00000				CONTAMT	v		-			
					USEREFNO			-			
					EXTREFNO			÷.	-		
									_		1

4.11 Preview

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

🖹 New 📴 Enter Query	,	
Product Code *	P Contract Reference	Operation Code *
Product Description	User Reference	Source Code FLEXCUBE
i reduct Decemption	Source Reference	Version Number 4 Of
Product Type	•	
Main Preferences Part	ies Parties Limits Shipment Documents Tracers Adv	ices
LC Details		
Currency *	Customer	* Issue Date
Contract Amount *	Customer Name	Effective Date
Positive Tolerance	Party Type	* Tenor
Negative Tolerance	Dated	Expiry Date
Max Amount	Customer Reference	Expiry Place
Liability Tolerance	License Expiry Date	Auto Closure
Liability Amount	Amount	Closure Date
Tolerance Text	- Liability	Stop Date
		Pre-Advice Date
		Reference To Pre-
		advice
Credit		- Guarantee Details
Туре	Credit Available With	Type of guarantee
	Credit Available With	Guarantee
Mode	Details	Guarantee
- Revolving Detail		
Revolves in		Automatic Reinstatement Remarks
Units		Cumulative
	Next Reinstatement	Loan for Collateral
Frequency	Date	Partial Closure
- Reimbursement Undert	aking	
Undertaking Expiry	Availed Undertaking	
Date Undertaking Amount	Amount	
Undertaking Amount		
- Status		•
Drafts Commission	Charges Settlement Tax Collateral Ev	ents Linkage Details Fields MIS Transfer Details BC Linkages 📃 🔺
Split Settlement Loar	n Preference Brokerage All Messages Docum	ents Message Preview Import License
Maker	Date Time	Status
Checker		Authorization Status
	Date Time	EXIL

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

	Authorization Status			Contract S	tatus		
	Contract Reference	-	-	Product (
	Contract Reference Currency	x =	7	Contract Arr			
	Branch		E	Operation (▼	
	Dialici			operation	June	•	
leco	rds per page 15 👻 🚺	1 of 1	Go to Page				
	Authorization Status	Contract Status	Contract Reference	Product Code	Currency	Contract Amount	User Refer
							•

Fig 4.10 Online Form Summary Screen Preview

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 a 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 - LCDTRONL.xml (uixml for LC Contract 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 - LCDTRONL_SYS.js (JS for LC Contract 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 – LCDTRONL_KERNEL.js (JS for KERNEL Release) *Example* – LCDTRONL_CLUSTER.js (JS for CLUSTER Release) *Example* – LCDTRONL_CUSTOM.js (JS for CUSTOM Release)

5.2 Data Base Units

5.2.1 Static Scripts

The following static scripts generated are required for the proper functioning of a 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 – Lcpks_Lcdtronl_Main.spc, Lcpks_Lcdtronl_Main.sql (Main Package for LC Contract)

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.

<i>Example</i> – Lcpks_Lcdtronl_Kernel.spc,	Lcpks_Lcdtronl_Kernel.sql (Kernel Package)
Lcpks_Lcdtronl_Cluster.spc,	Lcpks_Lcdtronl_Cluster.sql (Cluster Package)
Lcpks_Lcdtronl_Custom.spc,	Lcpks_Lcdtronl_Custom.sql (Custom Package)

5.3 Other Units

5.3.1 Xsd

Only Type XSD and message XSD will be generated for a Online Form function Id. This type xsd will be used in the type xsd of any function which uses the particular Online form.

Example - LC-Contract-Types.xsd (Type XSD for LC Contract)

LC-CreateContract-Req-Full-MSG.xsd (Create Message XSD for LC Contract) LC-CreateContract-Req-IO-MSG.xsd (Create Message XSD for LC Contract) LC-CreateContract-Res-Full-MSG.xsd (Create Message XSD for LC Contract) LC-CreateContract-Res-PK-MSG.xsd (Create Message XSD for LC Contract)

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 LCDTRONL_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 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

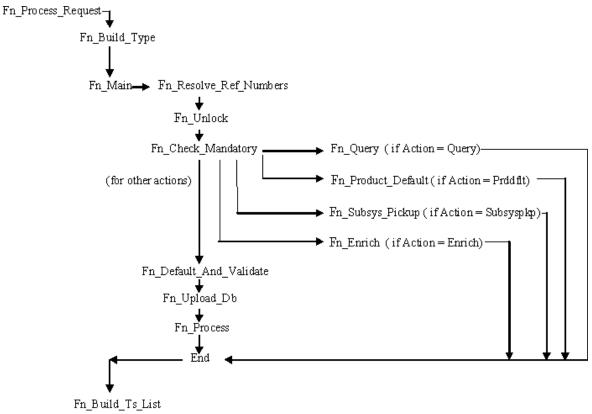


Fig 4.10 Flow of control in an Online main package

Release specific code has to be written in the Hook Packages generated. Different functions available in the Hook Package of a 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 [August] [2022]

Version 14.6.1.0.0

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 © 2022, 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.