Development of Online Forms Oracle Banking Treasury Management Release 14.5.1.0.0 [June] [2021]



# Contents

1. Prefa	ace	3
1.1	Audience	3
1.2	Related Documents	3
2. Intro	oduction	
2.1	How to use this Guide	3
3. Over	rview of Online Form	4
4. Scree	en Development	4
4.1	Header Information	
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	20
5.2.1		
5.2.2	System Packages	20
5.2.3	Hook Packages	21
5.3	Other Units	
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
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</u>, "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 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.

Function Generation						2	×	I 7	- > >
	on New ▼ Id LCDTRONL th	Function Type F Parent Function Parent Xml	Parent	Function Category Tran Header Template None Footer Template None	• •				
Search Preferences DataSource ListOrValues CataBlocks Greens FieldSets Actions CaliForms 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

						2			<b>9</b>
Action Load 👻		Function Ty	Parent -		Function Category Transaction	-			
Function Id LCDTRONL		Parent Function	on		Header Template None 🔻				
Save XML Path LCDTRONL_F	BROWSE	Parent Xi	nl		Footer Template None	•			
earch	Preferences								5
Preferences		Head Office Fur	ction	Module	LC 🗾				
DataSource     ListOfValues		🔲 Logging Requir	ed	Module Description	Letters Of Credit				
DataBlocks		Auto Authorizati	on	Branch Program Id					
B Creens		📋 Tank Modificatio	ns	Process Code					
🗄 🚞 FieldSets		🗖 Field Log Requi	red	SVN Repository URL					
i Actions Actions		Multi Branch Ace	ess	Transaction Block Name	BLK_CONTRACT_DETAILS -				
a LaunchForms		Excel Export Re	quired	Transaction Field	BRANCH				
Carl Summary				Name					
						Contro	l String	+ -	]
	E Fu	nction Id		Module *	Module Description			*	i.
	LCDTRONL		LC	×=	Letters Of Credit				L
	LCSTRONL		LC		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.

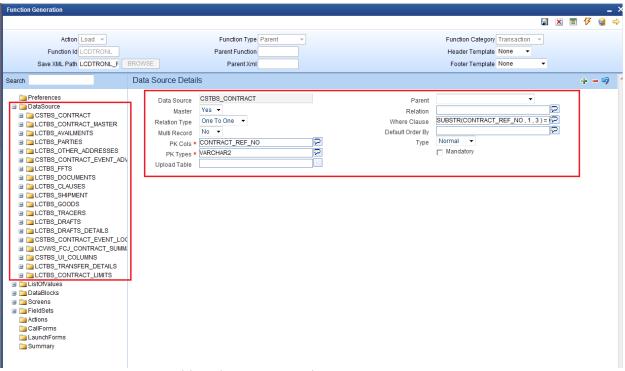


Fig 4.3 Adding data sources and maintaining properties

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
- 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: LCVWS\_FCJ\_CONTRACT\_SUMMARY*

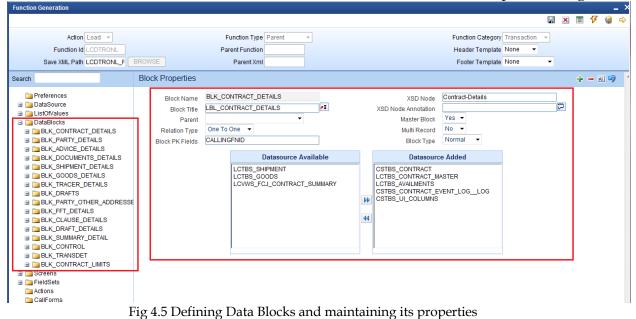
Function Generation				_ ×
				🗏 7 🧐 🔿
Action Load 👻	Funct	n Type Parent 👻	Function Category Transaction	
Function Id LCDTRONL	Parent	unction	Header Template None 👻	
Save XML Path LCDTRONL_F	BROWSE Pa	ent Xml	Footer Template None	
Search	Data Source Field Details		F	Refresh 🗕 🗐 🤺
Preferences  DataSource  Data	Column Name Block Name Field Name		Data Type CHAR MaxLength 1 Dad Table Column Not Required in Upload Tables	

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				-
				🖫 🗷 🗏 7 🧃 🖣
Action Load	Function Type	Parent 💌	Function Category Transaction	n 👻
Function Id LCDTRONL	Parent Function		Header Template None	•
Save XML Path LCDTRONL_F	BROWSE Parent Xml		Footer Template None	•
Search	Block Field Properties			- 🛛 🗔 🦃
Preferences	Field Name * PRTYTYP	XSD 1		Required
DataSource	Field Label LBL_PRTYTYP	XSD Annotat		Visible
🗉 🚞 ListOfValues	DataSource LCTBS_PARTIES		Size * 3	Read Only
DataBlocks	Column Name * PARTY_TYPE	Maximum Len		-
BLK_CONTRACT_DETAILS     BLK_PARTY_DETAILS		Minimum Va		Calender Text
CONTREFNO5	a dia type	Maximum Va		Popup Edit Required
PRTYTYP	Display Type Lov 👻	Maximum Decim		Uppercase Only
PARTYDESC	Item Type Database Item 👻	TextArea Ro		LOV Validation
DARTYCIFID	Parent Field 🔹	To the other		Required
CUSTNAME	Related Block	TextArea Colum		
CUSTADDLIN1	Related Field	Default Va		🔲 Not Required In Xsd
CUSTADDLIN2	LOV Name LOV_PART_TYPE	<ul> <li>Preview Va</li> </ul>		Report Parameter
	Off Line LOV Name	✓ Mas	kid 🚬 🛀	
COUNTRYCD	Fieldset Name FST_CONTRACT_PAR	RTIES		
CUSTREFNO		Determ Fights Determined Fights		
CUSTREFDATE	Custom Attributes Events Bind Variables	Return Fields Related Field		
🚞 LANGCD	Return Fields Mapping		Default	t From Lov Definition
DISSBANK	Query Column	Block Name	Return Field	J Name 🔺
ESN	PARTY_TYPE	BLK_PARTY_DETAILS -	PRTYTYP	•
	ITEM_VAL_DESC	BLK_PARTY_DETAILS	PARTYDESC	•
BLK_DOCUMENTS_DETAILS	N HEM_WE_DEGO			
BLK_SHIPMENT_DETAILS				
BLK_GOODS_DETAILS				
BLK_TRACER_DETAILS				
BLK_DRAFTS				
BLK_PARTY_OTHER_ADDRE				
BLK_CLAUSE_DETAILS     DELK 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	<b>BLOCK FIELD NAME</b>
CONTRACT_STATUS	TXNSTAT
AUTH_STATUS	AUTHSTAT
SUBSYSTEM_STAT	SUBSYSSTAT

inction	Generation												
motion	Seneration										×	<b>=</b> 7	
										un			<b>9</b>
	Action Load 👻				Function Type	Parent .	-	Functi	on Category Tr	ansaction 👻			
	Function Id LCDTROM	IL			Parent Function			Head	er Template No	one 👻			
	Save XML Path LCDTRON	IL E	BROWSE		Parent Xml			Foot	er Template No	one 🔻			
arch	REVOLVE		Block H	Field Proper	ties							- 🗷 📮	2 🗉
		-		Field Name *	CONSTAT		100 T	CONSTAT			Requ	ired	
							XSD Tag	CONSTAN					
				Field Label	LBL_CONSTAT	<u>~=</u>	XSD Annotation		~	,	Visibl		
	FREQ			DataSource	CSTBS_CONTRACT		Field Size *	1		Г	Read	Only	
	INEXTREINDT		С	olumn Name *	CONTRACT_STATUS		Maximum Length	1		г	- Caler	nder Text	
	ALLOWREPAY			Data Type *	Char -		Minimum Value					p Edit Req	awir
	CLOSTYP				Text -		Maximum Value						
	TRANSBLE			Display Type			Maximum Decimals			ſ	- Uppe	rcase Only	y
	MAYCONFIRM			Item Type	Database Item 👻					F		/alidation	
	REMARK			Parent Field		-	TextArea Rows				Requ		
	RELLCREF		F	Related Block		•	TextArea Columns			ſ	Input	by LOV On	nly
	BTN_DEFAULT			Related Field	•		Default Value		×=	Г	Not R	equired In	a X:
	CONREFNOLOG						Preview Value			r	Repo	rt Paramet	eter
	MAKER			LOV Name		· ·	MaskId		<b>#</b> E	1			
	MAKDTTIME		Off Lin	e LOV Name		•							
	CHKR		Fi	eldset Name	FST_PRODDET								
	CHKDTTIME		Queter	Attributes Ev	ents Related Field								
- I	TXNSTAT		Custon	Aundules	ents Related Field								
- 1	CONSTAT											+ -	
[	authstat				ttribute Name		Attribute Value	A				~	
	BTN_NEXT			А	undute name		Attribute value	Active		Position			
	BTN_PREVIOUS	E											
	availesn												
	BTN_CHARGES												
	BTN_SETT												
	BTN_TAX		5										
	BTN_COLLATERAL												
	BTN_EVENTS												
	BTN_LINKAGES												
	BTN_UDF												
	🚞 BTN_MIS												
	🚞 OF												
	VERSIONLBL											*	
	BACKTOBACK												
	ACKNREVCD												
	ACKDT												

## 4.5 Screens

Design the screen layout based on the requirement

						<b>E</b>	K 🔳 4	۶ 🧃
Action Load 👻		Function Type Parer	nt 💌		Function Category	Transaction 👻		
Function Id LCDTRONL		Parent Function			Header Template	None 🔻		
Save XML Path LCDTRONL_F	BROWSE	Parent Xml			Footer Template	None -		
earch	Screen Details						- A	] 🗖 🕻
Preferences DataSource DataSource DataBlocks DataBlocks CVS_MAIN CVS_MAIN	ooroonnamo	CVS_MAIN LBL_CONTRACT_DETAILS Large Default Cancel	v v	I Main Scr I Visible	een			
							+	]
TAB_MAIN     SEC_MAIN	Argument Na	ame Source Block	Source Field	Argument Value	Target Block	Target Field	Active	*
SEC_TOL	CONTREF		<b>_</b>		BLK_CONTRACT_DETAILS -	CONREFNO -	Yes 👻	
SEC_CUST	ESN				BLK CONTRACT DETAILS -	LATEVNSEQNO -	Yes 👻	-
B → TAB_PREFERENCES B → TAB_PARTIES B → TAB_PARTIES_LIMIT B → TAB_SHIPMENT C → TAB_OCUMENTS B → TAB_TRACERS								-

\_ ×

#### Fig 4.7 Designing Screens and providing Screen Properties

Note the following while creating screens

• One Screen should be identified as the main screen.

			🔚 🗵 🗏 🖗 🧐
Action Load -	Function Type Parent		Function Category Transaction
Function Id LCDTRONL	Parent Function		Header Template None 👻
Save XML Path LCDTRONL_F	ROWSE Parent Xml		Footer Template None -
rch	Tab Details		Dependent Fields 🐥 🗕 🗷 🗳
Preferences DataSource DataSource DataStofValues DataBlocks Serens	Screen Name CVS_MAIN Tab Name TAB_MAIN Tab Label LBL_MAIN 25 Tab Type Data •	♥ Visible	
Actions     CallForms     LaunchForms     Summary			

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*

					🗉 🗐 🗐
Action Load 👻		Function Type Parent	<b>•</b>	Function Category Transaction	
Function Id LCDTRONL		Parent Function		Header Template None 👻	
Save XML Path LCDTRONL_F	BROWSE	Parent Xml		Footer Template None 👻	
irch	Section Details				- A
Preferences DataSource ListONalues DataBlocks Screens	Section Name Section Label	SEC_TOL	Visible		
🖃 🚞 CVS_MAIN	Partition Detail	S			+-
⊞ 🚞 HEADER ⊒ 🚞 BODY	Partition SI N	lo	Partition Name	Width Sub-partiti	ions ^
🖃 🚞 TAB_MAIN	1	PART_TOL1		66 🕶 2 💌	
SEC_MAIN	2	PART_TOL2		33 🔻 👻	
SEC_TOL					
SEC_CUST					7
SEC_STAT					Ŧ
SEC_STAT					Ŧ
BEC_STAT B D TAB_PREFERENCES CAB_PARTIES					<b>T</b>
SEC_STAT					T
BEC_STAT TAB_PREFERENCES TAB_PARTIES TAB_PARTIES TAB_PARTIES_LIMIT					T
□ SEC_STAT □ TAB_PREFERENCES □ TAB_PARTIES □ TAB_PARTIES_LIMIT □ TAB_SHIPMENT					T
SEC_STAT     SEC_STAT     SEC_STAT     STAB_PREFERENCES     TAB_PARTIES     TAB_PARTIES_LIMIT     TAB_SHIPMENT     TAB_SHIPMENT     TAB_TRACERS     TAB_ADVICES					T
SEC_STAT     SEC_STAT     SEC_STAT     SEC_STAT     STAB_PREFERENCES     TAB_PARTIES_LIMIT     TAB_PARTIES_LIMIT     TAB_SHIPMENT     TAB_DOCUMENTS     TAB_TRACERS     SEC_STAT_SCORES     SEC_STAT_SCORES					T
SEC_STAT     SEC_STAT					(v)
SEC_STAT     SEC_STAT     TAB_PREFERENCES     TAB_PARTIES_LIMIT     TAB_SHIPMENT     TAB_SHIPMENT     TAB_DOCUMENTS     TAB_TRACERS     TAB_TRACERS     FOOTER     FOOTER     FOOTER     CVS_DRAFT     CVS_PRE_CLOSE					Υ.
SEC_STAT					Ŧ
SEC_STAT B TAB_PREFERENCES TAB_PARTIES TAB_PARTIES_LIMIT TAB_SHIPMENT TAB_TRACERS TAB_TRACERS TAB_ADVOCES TAB_ADVOCES CVS_DRAFT CVS_DRAFT CVS_DRAFT CVS_TRANSDET FieldSets					Ŧ
SEC_STAT					Ŧ
SEC_STAT     SEC_STAT					Ŧ
SEC_STAT					Ŧ

Multiple Screens can be designed if required.

# 4.6 Field Sets

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

						1 🗉 🖗	· 🧐
Action Load 👻	Function Type Parent		Function C	ategory Transaction	-		
Function Id LCDTRONL	Parent Function		Header Te	emplate None 🔻			
Save XML Path LCDTRONL_F BROWS	SE Parent Xml		Footer Te	emplate None	•		
Search Field	dset Properties					-	AT 🌍
Screens     FieldSets	Fieldset Name FST_PROD	ourounname	CVS_MAIN	•	1.5	orizontal Fi	eldset
ST_PROD		Ocidenti oluon	Header	•		eadOnly	
ST_PRODDET	Data Block BLK_CONTRACT_DETAILS		TAB_HEADER	-	∏ Na	avigation B	utton
ST_REF	Multi Record No v	ocononnanno	SEC_HEADER2	<b>•</b>	Vis Vis	sible	
FST_USER_REF FST_AMEND	View Type Single -	Partition Name	PART1	-			
FST_CHARGES_FROM_ISB	Fieldset Height	Number Of Rows					
FST_CREDIT							
ST_OTHER		_			_		
FST_PREF	Data Block Fields	E Fie	eld Set Fields S	Subpartition Name			
FST_REV_DET	BTN_DRAFTS	PRDCD		<b>~</b>			
ST_PREADVDESC	BTN_COMMISSION PRTYTYP	BTN_P	с. Г	<b>~</b>			
FST_BUTTONS	CRDLIN	CALLINGE		•			
FST_DRAFT_DETAILS	LINECID	IV CALLING	INID	·			
E FST_CUSTOMER	INCAMDNO EVENTCD						
FST_DRAWEE_DETAILS	BTN_CHARGES	44					
ST_INSURANCE_DETAILS	BTN_SETT						
FST_GOODS	BTN_TAX BTN_COLLATERAL						
FST_SHIPMENT	BTN_EVENTS						
FST_FOOTER_1	BTN_LINKAGES						
ST_CONTRACT_PARTIES							
ST_CONTRACT_ADVICES							
FST_CONTRACT_DOCUMENT							
FST_CONTRACT_TRACERS							
ST_CONTRACT_OTHER_ADI							
FST_CONTRACT_CLAUSES							
FST_CONTRAT_DRAFTS_DET							
Difference FST_VER							
FST_CONTROL							
FST_TRANS							
FST_GAURENTEE							
ST_GOAR							

\_ X

### 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

		Function Type	Parent		unction Category Transacti	on 👻
Action Load   Function Id LCDTRON		Parent Function				<b>→</b>
Save XML Path LCDTRON		Parent Xml			Footer Template None	•
Save AME Path LCDTROP					Footer remplate None	
ch	Form Actions					E
Preferences	XSD Type Ider	ntifier Contract		Service Name FCU	BSLCService	<b>*</b> =
i DataSource	Operati	on Id Contract				
DataBlocks						
🚞 Screens 🚞 FieldSets						+-
Actions	Web Service	Action Code	Oper	ration Code	Action Stage Type	Amendables
CallForms		QUERY	QueryContract			Amendables
Summary		NEW	CreateContract			Amendables
		MODIFY	ModifyContract		V	Amendables
		AUTHORIZE	AuthorizeContract			Amendables
		DELETE	DeleteContract			Amendables
		CLOSE	CloseContract			Amendables
		REOPEN	ReopenContract		$\overline{\mathbf{v}}$	Amendables
		REVERSE	ReverseContract			Amendables
		ROLLOVER	RolloverContract		V	Amendables
		CONFIRM				Amendables
		LIQUIDATE				Amendables
		SUMMARYQUERY				

- 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 LCDTRONL     Parent Function     Header Template     None       Save XML Path     LCDTRONL_F     BROWSE     Parent Xml     Footer Template     None       earch     Launch Form Details       Preferences       B     DataSource       C DataSource       B     DataBlocks       B     DataBlocks       B     Screen Argume       C SDEVENT     Yes	
arch Launch Form Details	
Preferences CataSource	
DataSource     Screen Argume       DataDivalues     Screen Argume       DataBlocks     Function ID       Screens     Active	
ListOfValues     Screen Argume       DataBlocks     Function ID     Active       Screens     Constraint     Constraint	
Screens	
Actions MSDALMSG Yes  CallForms	
LaunchForms Summary	

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

Function Generation				
			🔛 🐹 🛛	🗏 7 🥃 🧇
Action Load -	Function Type Parent V	tion Calegory Transa	ction 👻	
Function Id LODTRONL	Parent Function Heat	der Templale None	-	
Save XML Path LCDTRONL_F BROWSE	Parent Xml Foo	oler Template	<b>*</b>	
Search	n Form Details			G)
				,
i Preferences DataSource				
ListOfValues			Screen Argume	ints 🕂 🗕
DataBlocks      Screens	Function ID		Active	^
🗉 🧰 FieldSets 🛛 🕅	CSDEVENT		Yes 💌	
CaliForms	MSDALMSG		Yes *	
LaunchForms	MSEMSPRV		Yes 💌	
Canal Summary	Call Form Arguments	×		
	Populate Reset	4		
	Argument Name Source Block Source Field Argument Value     CONTREF BLK_CONTRACT_DETAILS      CONREFNO			
	ACTION_CODE   EXECUTEQUERY			
	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	F BROWSE	Parent Xml		Footer Template	•
irch	Call Form Details				
🔁 Preferences					
🛅 ListOfValues				Screen Arguments Dep	pendent Fields + -
) 🧰 DataBlocks ) 🧰 Screens	Function ID	Parent Data Block	Parent DataSource	Relation	Relation Type
FieldSets	CFCTRCOM	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻
Actions	CFCTRCHG	BLK_CONTRACT_DETAILS	▼ LCTBS_CONTRACT_MASTER ▼	LCTBS_CONTRACT_MASTER.CONT	One To One 🔻
CallForms	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

MIČTRMIS: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										-	
								×	V	9	
Action Load -		Function Type Parent 🔹			Functi	on Category Transaction	-				
Function Id LCDTRONL		Parent Function			Head	er Template None 👻					
Save XML Path LCDTRONL_F	BROWSE	Parent Xml			Foot	er Template	•				
Search	Summary Details								[	<b>3</b> 9	
Preferences DataSource ListOfValues ListOfValues CataBlocks ListOfValues FieldSets Actions CaliForms	Title Data Blocks Data Source Summary Type Summary Screen Size	LBL_SUMMARY BLK_SUMMARY_DETAIL • LCWVS_FCJ_CONTRACT_SUN • Summary • Medium •	/1	Default Where Clause Default Order By Multi Branch Where Clause Main Summary Screen			(SEL V				
🚘 LaunchForms 🔁 Summary	Data Block Fields C	Ustom Buttons Fields Ordering	_	Fields Selected	Query	LOV Name					
	CLOSDT	Data Diock Fields	-	AUTHSTAT	Query	LOV Name	[	1			
	SETLMTH			CONSTAT	<b>v</b>		÷.				
	EFFDT CIFID			 CONREFNO			<u> </u>				
	MAXCONTA		ÞÞ	PRDCD	<b>v</b>						
	CURRAVAI		44	 CONTCCY	R I		-				
	USLIAB			CONTAMT			-				
				USEREFNO			-				
				EXTREFNO			⊸,				
	1									]	

# 4.11 Preview

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

🖹 New 🖻 Enter Query			
Product Code *	Contract Reference	Operation Code *	-
Product Description	User Reference	Source Code FLEXCUBE	
1 rodde Besenpron	Source Reference	Version Number	
Product Type	v		
Main Preferences Parties Parties Limits	Shipment Documents Tracers Advic	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	
	2	Pre-Advice Date	
		Reference To Pre-	E
		advice	
Credit		- Guarantee Details	
Туре 🔍 👻	Credit Available With	Type of guarantee	
		Guarantee	
Mode	- Details	Guarantee	
Revolving Detail			
Revolves in		Automatic Reinstatement Remarks	
Units		Cumulative Default	
	Next Reinstatement	Loan for Collateral	
Frequency	Date	Partial Closure	
Reimbursement Undertaking			
Undertaking Expiry Date	Availed Undertaking Amount		
Undertaking Amount	Amount		
Status			
Drafts   Commission   Charges   St	attlement   Tax   Collateral   Ever	ents   Linkage Details   Fields   MIS   Transfer Details   BC Linkages	
		ents   Linkage Details   Fleids   Mis   Hansler Details   BC Linkages   ents   Message Preview   Import License	
Maker	Date Time	Status	
Checker		Authorization Status	E.M.
	Date Time		Exit

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

	Authorization Status	•		Contract St	tatus	•	
	Contract Reference		7	Product C	Code	<u>7</u>	
	Currency	<b>*</b>		Contract Am	ount	<b>x</b> =	
	Branch		<b>a</b> =	Operation C	Code	-	
eco	rds per page 15 🔻 🚺	◀ 1 of 1 ▶ ▶	Go to Page				
	Authorization Status	Contract Status	Contract Reference	Product Code	Currency	Contract Amount	User Refe
							Exit

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

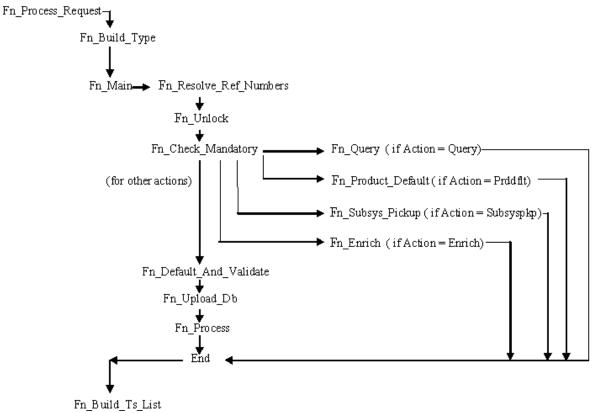


Fig 4.10 Flow of control in an Online main package

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 [June] [2021] Version 14.5.1.0.0

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 © 2020, 2021 Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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