# Development of Online Forms

Oracle Banking Enterprise Limits and Collateral

Management

Release 14.7.0.0.0

Part No. F75024-01

[November] [2022]

FINANCIAL SERVICES

## Contents

1. Prefa	ace	3
1.1	Audience	3
1.2	Related Documents	3
2. Intro	oduction	4
2.1	How to use this Guide	4
3. Over	rview of Online Form	4
4. Scree	en Development	5
4.1	Header Information	5
4.2	Preferences	5
4.3	Data Sources	6
4.4	Data Blocks	8
4.5	Screens	10
4.6	Field Sets	12
4.7	Actions	13
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	Static Scripts	20
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.

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

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

### **1.2** Related Documents

<u>04-Development\_WorkBench\_Screen\_Development-I.docx</u> <u>05-Development\_WorkBench\_Screen\_Development-II.docx</u>

## 2. Introduction

## 2.1 How to use this Guide

The information in this document includes:

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

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.

Functio	Generation					-	×
				×	V	9	4
	Action New  Function Id LCDTRONL Save XML Path	Function Type Parent Parent Function Parent Xml	Function Category Transaction				
Search							1
D D D S Fi Ar C L	eferences staSource stOfValues taBlocks treens aldSets tions allForms unchForms ummary		Form has dan Information				

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**. Function Type :

Parent and Child functionality is supported for Online forms.

## 4.2 Preferences

iv)

Provide the menu details in the Preferences screen

Action Load 👻		Function Type Par	ent 👻		Function Category Transaction	-			
Function Id LCDTRONL		Parent Function			Header Template None 🔻				
Save XML Path LCDTRONL_F	BROWSE	Parent Xml			Footer Template None	•			
earch	Preferences								6
in Preferences		Head Office Function		Module	LC 🗾				
		🗖 Logging Required		Module Description	Letters Of Credit				
DataBlocks		Auto Authorization		Branch Program Id					
B Creens		Tank Modifications		Process Code					
🗄 🚞 FieldSets		Field Log Required		SVN Repository URL					
i Actions CallForms		Multi Branch Access		Transaction Block Name	BLK_CONTRACT_DETAILS	•			
LaunchForms		Excel Export Required		Transaction Field	BRANCH	-			
Carl Summary				Name					
						Contro	l String	+ -	-
	Func	tion Id	M	odule *	Module Description	on		^	6.
	LCDTRONL	LC		×=	Letters Of Credit				1
	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.

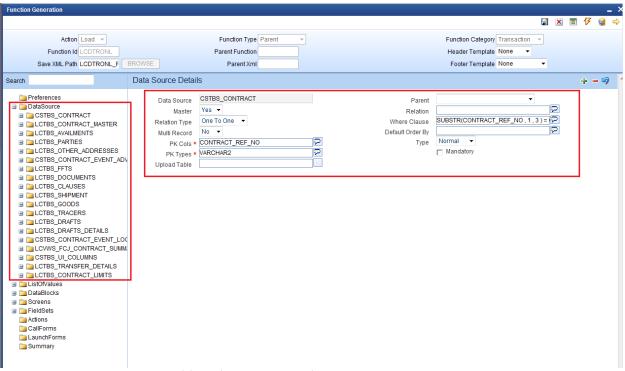


Fig 4.3 Adding data sources and maintaining properties

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*

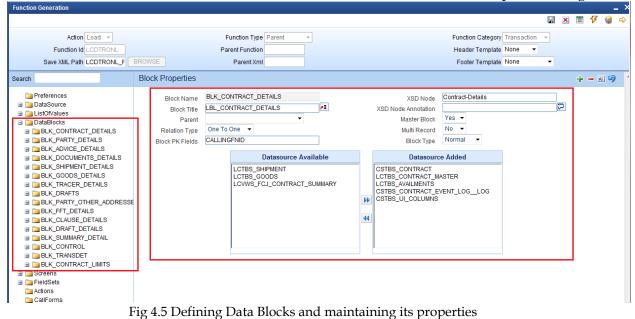
Function Generation					_ ×
					🗙 🗏 🎸 🍯 🔿
Action Load -		ent Function		Function Category Transaction	
	BROWSE			·	
Save XML Path LCDTRONL_F	BROWSE	Parent Xml		Footer Template None 🔻	
Search	Data Source Field Details				Refresh 🗕 🗐 🤺
Preferences DataSource DataSource UDATESTATUS BOOK_DATE BRANCH CONTRACT_REF_NO CONTRACT_STATUS EXTERNAL_REF_NO LATEST_VERSION_NO LATEST_VERSION_NO LATEST_VERSION_NO MODULE_CODE PRODUCT_TYPE SOURCE STOP_DATE USER_DEFINED_STATUS USER_REF_NO CURR_EVENT_CODE USER_DEFINED_STATUS USER_REF_NO CURR_EVENT_CODE USER_ONTRACT_MASTER USELCTBS_AVAILMENTS LCTBS_ONTRACT_MASTER LCTBS_ONTRACT_VAILES LCTBS_ONTRACT_VAILES LCTBS_ONTRACT_VAILES LCTBS_ONTRACT_VAILES LCTBS_ONTRACT_VAILES USER_CONTRACT_VAILES USER_CONTRACT_VAILES CONTRACT_VAILES LCTBS_ONTRACT_VAILES LCTBS_ONTRACT_VAILES LCTBS_ONTRACT_VAILES LCTBS_CONTRACT_VAILES USEN_CONTRACT_VAILES USEN_	Column Name Block Name Field Name		Data Type Max Length Upload Table Column	CHAR	

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		1		-
				🖫 🗶 🗏 7 🤤 🖣
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 None	•
Search	Block Field Properties			- 🛛 🗖 🦃
Preferences	Field Name * PRTYTYP	XSD Ta	ag PRTYTYP	Required
🗄 🧰 DataSource	Field Label LBL_PRTYTYP	XSD Annotatio		Visible
🗉 🚞 ListOfValues	DataSource LCTBS_PARTIES	Field Siz		Read Only
DataBlocks	Column Name * PARTY_TYPE	Maximum Leng	-	-
BLK_CONTRACT_DETAILS     BLK_PARTY_DETAILS		Minimum Valu		Calender Text
CONTREFN05		Maximum Valu		Popup Edit Required
PRTYTYP	Display Type Lov 🗸	Maximum Decima		Uppercase Only
PARTYDESC	Item Type Database Item 👻	TextArea Row		LOV Validation
PARTYCIFID	Parent Field	Testine Octoor		Required
CUSTNAME	Related Block	<ul> <li>TextArea Column</li> </ul>		Input by LOV Only
CUSTADDLIN1	Related Field 👻	Default Valu		Not Required In Xsd
CUSTADDLIN2	LOV Name LOV_PART_TYPE	<ul> <li>Preview Value</li> </ul>		Report Parameter
CUSTADDLIN3	Off Line LOV Name	- Mask	ld 🚬	
	Fieldset Name FST_CONTRACT_PAR	RTIES		
CUSTREFDATE	Custom Attributes Events Bind Variables	Return Fields Related Field		
LANGCD	Return Fields Mapping		Default Fro	m Lov Definition
DISSBANK	Query Column	Block Name	Return Field Nat	me 🔺
ESN	PARTY_TYPE	BLK_PARTY_DETAILS	PRTYTYP 🔻	
BLK_ADVICE_DETAILS	ITEM_VAL_DESC	BLK_PARTY_DETAILS	PARTYDESC -	
BLK_DOCUMENTS_DETAILS	V TEM_VAL_DESC	bengi mini geemieo	TARTIBEOU	
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

Function	1 Generation									
Tuncuon	Tueneration								× =	- 77 (a)
								لصا		
	Action Load -		Function Type	Parent .	-	Function Ca	tegory Transaction	-		
	Function Id LCDTRONL		Parent Function			Header Ter	nplate None 👻			
	Save XML Path LCDTRONL	F BROWSE					nplate None	-		
	Save XML Pain LCDTRONL	F BROWSE	Parent Xml			FoolerTer	nplate None	-		
Search		Block F	ield Properties							R 🗔 🌍
	REVOLVE	*					_			
	REINTYP		Field Name * CONSTAT		XSD Tag	CONSTAT			Require	d
			Field Label LBL_CONSTAT	×=	XSD Annotation				Visible	
			DataSource CSTBS_CONTRACT		Field Size *	< <b>1</b>			Read Or	ılv
		0	olumn Name * CONTRACT_STATUS		Maximum Length	1			Calende	
	NEXTREINDT ALLOWREPAY	C			Minimum Value		7			
			Bula Type				4		Popup E	dit Require
	TRANSBLE		Display Type Text -		Maximum Value		4		Upperca	se Only
			Item Type Database Item 👻		Maximum Decimals				LOV Vali	dation
	REMARK		Parent Field	•	TextArea Rows			10	Require	
	RELLCREF		Related Block	•	TextArea Columns				Input by	LOV Only
	BTN_DEFAULT				Default Value		<b>*</b> =		Not Reg	uired In Xsd
	CONREFNOLOG				Preview Value					arameter
	MAKER		LOV Name	•	Maskid		×E		Report	arameter
		Off Lin	e LOV Name	-	Maskiu					
	CHKR	Fi	eldset Name FST_PRODDET							
	TXNSTAT	Custom	Attributes Events Related Field							
	CONSTAT								+	-
	authstat									~
	BTN_NEXT		Attribute Name		Attribute Value	Active	Position			
	BTN_PREVIOUS	E								
	availesn									_
	BTN_CHARGES									
	DIN_SETT									
	DIN_TAX									
	BTN_COLLATERAL									
	BTN_EVENTS									
	BTN_LINKAGES									
	DIN_UDF									
	DTN_MIS									
	🚞 OF									
	VERSIONLBL									Ŧ
	BACKTOBACK									
	ACKNREVCD									
	CKDT									
	MDDT									

#### 4.5 Screens

Design the screen layout based on the requirement

						<b>F</b>	×	77 🧃
Action Load 🔻		Function Type Pare	nt 👻		Function Category	Transaction 👻		
Function Id LCDTRONL		Parent Function			Header Template	None 👻		
Save XML Path LCDTRONL_F	ROWSE	Parent Xml			Footer Template	None -		
arch	Screen Details						-	AÜ 🗔 🕻
Preferences     DataSource     ListONalues     DataStocks     Screens     CVS_MAIN     MEADER	ocroonitiante	CVS_MAIN LBL_CONTRACT_DETAILS Large Default Cancel		I Main Sc I Visible	reen			
								+ -
	Argument Na	ame Source Block	Source Field	Argument Value	Target Block	Target Field	Active	~
SEC_TOL	CONTREF	· · · · · · · · · · · · · · · · · · ·	-		BLK_CONTRACT_DETAILS -	CONREFNO -	Yes -	- T
SEC_CUST	SN ESN		-		BLK_CONTRACT_DETAILS -	LATEVNSEQNO -	Yes -	i I

\_ ×

#### 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 Xm		Footer Template None -
ch 7	Fab Details		Dependent Fields 🐥 🗕 🗷 🗳
Preferences	Screen Name CVS_MAIN	Visible	
DataSource	Tab Name TAB_MAIN		
istOfValues	Tab Label LBL_MAIN		
DataBlocks			
Creens Screens	Tab Type Data 🔻		
E CVS_MAIN			
BODY			
TAB_MAIN			
SEC_MAIN			
SEC_TOL			
SEC_CUST			
⊞ TAB_PARTIES     ∐ TAB_PARTIES_LIMIT     ☐ TAB_PARTIES_LIMIT			
TAB_FARTES_LIMIT			
TAB_TRACERS			
TAB_ADVICES			
B CVS DRAFT			
B CVS_PRE_CLOSE			
CVS_TRANSDET			
FieldSets			
a Actions			
a CallForms			
aunchForms			

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*

					E 77 🧐
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 🔹	
arch	Section Details				- 🕅 🗳
Preferences DataSource ListONalues DataBlocks Sceens	Section Name Section Label	SEC_TOL	Visible		
🖃 🚞 CVS_MAIN	Partition Details	5			+ -
	Partition SI N	0	Partition Name	Width Sub-partition	IS ^
🖃 🚞 TAB_MAIN	1	PART_TOL1		66 🔻 2 👻	
SEC_MAIN	2	PART_TOL2		33 🔻 👻	
SEC_TOL		_			
SEC_CUST					~
SEC_STAT					Ŧ
SEC_STAT					Ŧ
SEC_STAT TAB_PREFERENCES TAB_PARTIES					*
SEC_STAT					*
TAB_PREFERENCES					T
□ SEC_STAT □ TAB_PREFERENCES □ TAB_PARTIES □ TAB_PARTIES_LIMIT □ TAB_SHIPMENT					Ŧ
BEC_STAT DEC_ST					Ŧ
SEC_STAT					Ŧ
SEC_STAT					Ŧ
SEC_STAT     SEC_STAT     TAB_PREFERENCES     TAB_PARTIES     TAB_SHIPMENT     TAB_SHIPMENT     TAB_SHIPMENT     TAB_TRACERS     TAB_TRACERS     FOOTER     FOOTER     FOOTER     GOVER					v
SEC_STAT					T
SEC_STAT B TAB_PREFERENCES TAB_PARTIES TAB_PARTIES TAB_SHIPMENT TAB_SHIPMENT TAB_TRACERS TAB_TRACERS TAB_ADVOCES TAB_ADVOCES CVS_DRAFT CVS_DRAFT CVS_DRAFT CVS_TRANSDET FieldSets					
SEC_STAT					
SEC_STAT SEC_STAT STAB_PREFERENCES TAB_PARTIES TAB_PARTIES_LIMIT TAB_SHIPMENT TAB_TARCERS TAB_TARCERS TAB_TARCERS CVS_DRAFT CVS_DRAFT CVS_DRAFT CVS_TRANSDET FieldSets Actions CaliForms					
SEC_STAT					

Multiple Screens can be designed if required.

## 4.6 Field Sets

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

				🔚 🗶 🗏 🌮 🥞 考
Action Load 👻	Function Type Parent	Funct	ion Category Transaction	-
Function Id LCDTRONL	Parent Function	Head	ler Template None 🔻	
Save XML Path LCDTRONL_F BROWSE	Parent Xml	Foo	ter Template None	•
Search Fields	et Properties			- 🛛 🌍
🖃 🧫 FieldSets	ieldsetName FST_PROD	Screen Name CVS_MAIN	▼	Horizontal Fieldset
		Screen Portion Header	•	ReadOnly
ST_PRODDET		Tab Name TAB_HEADER	•	Navigation Button
FST_REF	Multi Record No -	Section Name SEC_HEADER2	•	Visible
EST AMEND	View Type Single -	Partition Name PART1	-	
FST_CHARGES_FROM_ISB	ieldset Height	Number Of Rows		
Difference FST_CREDIT				
FST_OTHER				¬ I
FST_PREF	Data Block Fields	Field Set Fields	Subpartition Name	
FST_TERMS	BTN_DRAFTS BTN COMMISSION	▲ PRDCD	<b></b>	
ST_PREADVDESC	PRTYTYP	BTN_P	•	
ST_BUTTONS	CRDLIN	E CALLINGFNID	•	
FST_DRAFT_DETAILS	LINECID INCAMDNO	44		
FST_CUSTOMER	EVENTCD	44		
FST_INSURANCE_DETAILS	BTN_CHARGES BTN_SETT	N		
FST_GOODS	BTN_TAX			
Difference FST_PORT	BTN_COLLATERAL BTN_EVENTS			
ST_SHIPMENT	BTN_LINKAGES	-		
FST_FOOTER_1				
FST_CONTRACT_ADVICES				
FST_CONTRACT_DOCUMENT				
FST_CONTRACT_TRACERS				
FST_CONTRACT_OTHER_ADI				
FST_CONTRACT_FFTS				
ST_CONTRAT_DRAFTS_DET				
FST_VER				
FST_CONTROL				
FST_TRANS				
FST_GAURENTEE				
FST_GOAR				

\_ X

### Fig 4.9 Field Set Properties

Note the following when attaching field to a field set

If a field is not required in the screen, but kept as hidden and value defaulted; then i) 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 -	Fu	Inction Category Transaction	on 👻	
Function Id LCDTRO	unction Id LCDTRONL Parent Function			н	eader Template None	•	
Save XML Path LCDTRO	NL_F BROWSE	Parent Xml		i	Footer Template None	•	
ch	Form Actions						
Preferences	XSD Type Iden	tifier Contract		Service Name FCUE	SLCService	×=	
DataSource ListOfValues DataBlocks Screens	Operatio	on Id Contract					
⊒ FieldSets ⊒ Actions	Web Service	Action Code	Oper	ration Code	Action Stage Type	+ Amendables	-
CallForms	<b>v</b>	QUERY	QueryContract		V	Amendables	
Summary		NEW	CreateContract			Amendables	
		MODIFY	ModifyContract		V	Amendables	
		AUTHORIZE	AuthorizeContract			Amendables	
	V	DELETE	DeleteContract			Amendables	
		CLOSE	CloseContract			Amendables	
		REOPEN	ReopenContract		V	Amendables	
		REVERSE	ReverseContract			Amendables	
		ROLLOVER	RolloverContract			Amendables	
		CONFIRM				Amendables	
		LIQUIDATE				Amendables	
		SUMMARYQUERY			$\overline{\vee}$		

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-PK-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            □ DataSource         □ DataSource         □ DataBlocks         □ DataBlocks         □ DataBlocks         □ Screen S     Function ID     Active	unts the
arch Launch Form Details  Preferences DataSource ListON/alues DataBlocks  Function ID	vonte
Preferences DataSource ListOfValues DataBlocks ListOfValues ListOfVal	vonte d
DataSource     ListOrValues     DataBlocks     Function ID     Active	unte L
ListOfValues         Screen Argum           DataBlocks         Function ID         Active	onte 🕂 🗖
	iento T
□ FleidSets CSDEVENT Yes ▼	
Actions MSDALMSG Yes V	
LaunchForms Ves  Yes	

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

Function Generation		_ ×
		🖬 🗵 🗏 🖗 🧇
Action Load *	Function Type Parent 👻	on Calegory Transaction 👻
Function Id LCDTRONL	Pareni Function Head	er Templale None 👻
Save XML Palh LCDTRONL_F	BROWSE Parent Xml Fool	er Template 👻
Search	Launch Form Details	19 19
🚞 Preferences		
DalaSource     ListOfValues		Screen Arguments
DataBlocks	Function ID	Active
<ul> <li>Screens</li> <li>FieldSels</li> </ul>		Yes •
Actions	MSDALMSG	Yes *
CaliForms	MSDMSPRV	Yes 💌
Summary		•
	Call Form Arguments	×
	Populate Reset	
	Argument Name Source Block Source Field Argument Value	
	CONTREF BLK_CONTRACT_DETAILS V CONREFNO V	
	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

								×	E 17	۶ 🧯
Action Load -		Function Type Parent			Function	Category Transact	on 👻			
Function Id LCDTRONL		Parent Function			Header	Template None	•			
Save XML Path LCDTRONL_F	BROWSE	Parent Xml			Footer	Template	•			
earch	Summary Details									۵
<ul> <li>Preferences</li> <li>DataSource</li> <li>ListOValues</li> <li>DataBlocks</li> <li>Screens</li> <li>FieldSets</li> <li>Actions</li> <li>CallForms</li> </ul>	Title Data Blocks Data Source Summary Type Summary Screen Size	LBL_SUMMARY BLK_SUMMARY_DETAIL • LCVWS_FCJ_CONTRACT_SUN • Summary • Medium •	<b>*I</b>	Default Where Clause Default Order By Multi Branch Where Clause Main Summary Screen	MODULE_C MODULE_C	ODE = "LC" AND EXI	STS (SEL			
aunchForms	Data Block Fields C	ustom Buttons Fields Ordering								
🚞 Summary					-					
Summary		Data Block Fields		Fields Selected	Query	LOV Name		•		
Summary Summary	CLOSDT SETLMTH	Data Block Fields		UTHSTAT	<b>v</b>	LOV Name	•			
Summary	SETLMTH EFFDT	Data Block Fields	_	UTHSTAT ONSTAT	<u>।</u> ज	LOV Name	• •	* E		
Summary	SETLMTH EFFDT CIFID MAXCONT	AMT		UTHSTAT ONSTAT ONREFNO	प         प           प         प	LOV Name	•	•		
Summary	SETLMTH EFFDT CIFID MAXCONT MAXLIABAI CURRAVA	AMT AT		UTHSTAT ONSTAT ONREFNO RDCD	े ज्य य	LOV Name	* * *	•		
Summary	SETLMTH EFFDT CIFID MAXCONT MAXLIABAI	AMT AT	<b>A</b>	UTHSTAT IONSTAT IONREFNO RDCD IONTCCY	प         प           प         प           प         प	LOV Name	* * * *			
Summary	SETLMTH EFFDT CIFID MAXCONT MAXLIABAI CURRAVA	AMT AT		UTHSTAT ONSTAT ONREFNO RDCD ONTCCY ONTAMT	प         प           प         प           प         प           प         प	LOV Name	* * *			
Summary S	SETLMTH EFFDT CIFID MAXCONT MAXLIABAI CURRAVA	AMT AT		UTHSTAT IONSTAT IONREFNO RDCD IONTCCY	प         प           प         प           प         प	LOV Name	* * * *			

## 4.11 Preview

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

E New 🤄 Enter Query	1		
Product Code *	P Contract Reference	Operation Code *	-
Product Description	User Reference	Source Code FLEXCUBE	
1 router B coonputer	Source Reference	Version Number	
Product Type	· · · · · · · · · · · · · · · · · · ·		
Main Preferences Parti	ies Parties Limits Shipment Documents Tracers Advi	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-	E
		advice	
Credit		- Guarantee Details	
Туре	Credit Available With	Type of guarantee	
Mode		Guarantee	
Mode	Details		
Revolving Detail			
Revolves in		Automatic Reinstatement Remarks	
Units		Cumulative Default	
Frequency	Next Reinstatement	Loan for Collateral	
( requerte)	Date	Partial Closure	
- Reimbursement Underta	aking		
Undertaking Expiry	Availed Undertaking		
Date	Availed Ondertaking		
Undertaking Amount			
Status			
			-
Drafts   Commission	Charges   Settlement   Tax   Collateral   Eve	ents   Linkage Details   Fields   MIS   Transfer Details   BC Linkages	<b>^</b>
Split Settlement   Loar	n Preference   Brokerage   All Messages   Docume	ents   Message Preview   Import License	
Maker	Date Time	Status	
Checker		Authorization Status	Exit
	Date Time		LAIL

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

	Authorization Status			Contract S	tatus	•	
	Contract Reference		<u>&gt;</u>	Product (	Code	×	
	Currency	<b>2</b>		Contract Am		<b>N</b> E	
	Branch		<b>N</b>	Operation (	Code	-	
есо	rds per page 15 👻 🚺	1 of 1	Go to Page				
	Authorization Status	Contract Status	Contract Reference	Product Code	Currency	Contract Amount	User Refe
		III					ŀ
							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

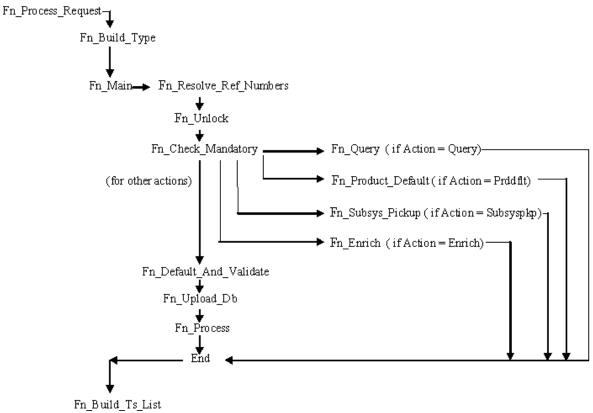


Fig 4.10 Flow of control in an Online main package

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 [November] [2022] Version 14.7.0.0.0

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

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

Copyright © 2007, 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.