Development Workbench - Source Upgrade Oracle Banking Trade Finance Release 14.5.2.0.0 [August][2021]



Contents

1	Preface	3
1	.1 Audience	3
1	.2 Related Documents	
2	Introduction	4
2	2.1 How to use this Guide	4
3	Overview of Refresh functionality in Oracle FLEXCUBE Development Workbench	4
4	Child Refresh	4
	1 Process Steps	5
4	2 Functionality Demonstration	10
5	Screen Child Refresh	
	5.1 Process Steps	
5	5.2 Functionality Demonstration	
6	Source Refresh	19
6	0.1 Process Steps	20
6	5.2 Functionality Demonstration	
6	5.3 Source refresh is not possible in below scenarios	

1 Preface

This document describes the Refresh functionality available in Oracle FLEXCUBE Development Workbench for Universal Banking and guides the developers on how to use this feature

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 Object Naming conventions	Development Overview Guide
Working knowledge of Web based	Self Acquired
applications	
Working knowledge of Oracle Database	Oracle Documentations
Working knowledge of PLSQL & SQL	Self Acquired
Language	
Working knowledge of XML files	Self Acquired

1.2 Related Documents

<u>05-Development WorkBench Screen Development-II.docx</u> <u>14-Development of Online_Forms.docx</u> <u>15-Development of Call_Form.docx</u> <u>16-Development of Launch Forms and Others_Screens.docx</u> <u>12-Child_and_ScreenChilds_Concept_and_Design.docx</u>

2 Introduction

2.1 How to use this Guide

The information in this document includes:

- Chapter 2, "Introduction"
- Chapter 3, "Overview of Refresh Functionality in Oracle FLEXCUBE Development Workbench"
- Chapter 4 , "Child Refresh"
- Chapter 5, "Screen Child Refresh"
- Chapter 6 , "Source Refresh"

3 Overview of Refresh functionality in Oracle FLEXCUBE Development Workbench

Refresh Functionality allows us to upgrade the existing radxml to its later version keeping the *sub version* specific changes intact. Three kinds of refresh can done using the Tool.

- 1) Child Refresh
- 2) Screen Child Refresh
- 3) Source Refresh

4 Child Refresh

Child Refresh allows the developer to upgrade a child radxml with its latest parent radxml .In doing so; the latest changes done in parent functionId would be reflected in the child functionId while retaining all the changes done in the child level

- This process is to be done within a release. i.e. child functionId has to be refreshed it's the parent function_id from the same release
- It is recommended that this process is done before development cut of the release for all child radxmls within a release. For instance; if development has happened parallel for a child and parent functionId during a release, child refresh should be done before base lining so that child and parent record types are consistent

• All the system units need to be regenerated after Child Refresh. A thorough unit testing is recommended after deployment of all generated units

4.1 Process Steps

Child Refresh process is explained taking two hypothetical functionIds , STDCIFD and as example

STDCIFD – Parent Screen STDCIFDC – Child Screen

Click on Refresh Node from Development Workbench landing page .



Fig 4.1.1: Development Workbench Landing Page

The following window will be launched

ORACLE FLEXCUBE Development Workben	ch for Universal Banking - W	indows Internet Explorer		-	and the first state of the	and had	
ORACLE FLEXCUBE Develops Browser	nent Workbench for Unive	rsal Banking				Windows Op	DEMOUSER
Browser -	Refresh Source File List Source Release Type Refresh Type	KERNEL - Child Refresh Source Refresh Screen Child Refresh File Name	BROWSE Sub Folder	Base File List Base Release Type File Status	Error Description		

Fig 4.1.2: Workbench Refresh Screen

Source File List: Browse and select the text file containing source file list.

ORACLE FLEXCUBE	evelopment Workbench for Universal Banki	ing				DE	MOUSER
Browser +					Windows	s Options	Sign Out
Administration Function Generation Screen Customizer Tracking Changes Notification Triggers Notifications Buik Generation	Rofresh Source Els Fiol	EROWSE		RHEL *	BROWSE	×	
Excel Template Generation Refresh Web Sevices Purge Generation	Upload File	Browse	File Status	Error Des	scription		
	😋 🔵 🗢 🗼 🕨 Computer 🕨 Data (D:)	► RADTOOL ► REFRESH ► SOURCE		- 4 7	Search SOURCE	Q	
	Organize 👻 New folder						
	odt_11.4 ODT_SOURCE ODT_SVN ODTI_14 piSQL prashant Preferences RAD_WASTE RAD_VASTE	SOURCE.txt	Date modified 9/4/2013 3:43 PM 9/4/2013 3:25 PM	Type Text Document XML File	Size 1 KB 27 KB		

Fig 4.1.3: Selecting source file list text file for Child Refresh

Source File list is a text file which contains the absolute path of all the child radxmls to be refreshed.

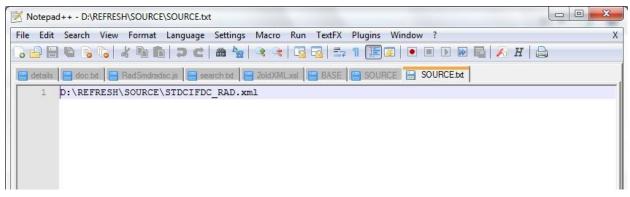


Fig 4.1.4: Content of source.txt file

The figure above shows the content of the source.txt file .Here STDCIFDS is the child radxml which has to be refreshed.

If child refresh of more than one functionId is required, absolute path of each child radxmls has to be specified; each in a new line

Base File List: Browse and select the text file containing base file list

ORACLE FLEXCUBE	Development Workbench for Universal Ba	nking				DE	MOUSER
Browser -					Windows	Options	Sign Out
Administration Function Generation Screen Customizer Tracking Changes Notification Triggers Notifications	Refresti Source File Ltef SOUR Upload	CE.txt BROWSE X		SE.txt	BROWSE	×	
 Bulk Generation Excel Template Generatio Refresh Web Sevices Purge Generation 	n Upload File	Browse	File Status	Error Des	ctiption		
	Choose File to Upload	D:) > RADTOOL > REFRESH > BASE >		• 49 5	iearch BASE		
	Organize Previoued ODT_SOURCE ODT_SVN ODT_SVN ODT11_4 piSQL prestant Preferences RAD_WASTE RADTOOL HTML INC IS LBLXML RAD_OP_FTVKK RADXML RAD_OP_FTVKK RADXML RAD_OP_FTVKK RADXML RAD_OP_FTVKK RADXML RAD_OP_FTVKK RADXML RAD_OP_FTVKK RADXML RADXML RAD_OP_FTVKK RADXML RADXML	Name CUSTOM12 KER122 MM1 BASE.tot STDCIFD_RAD.xml	Date modified 9/4/2013 5:58 PM 9/4/2013 5:59 PM 9/4/2013 5:17 PM 9/4/2013 3:43 PM 9/4/2013 1:36 PM	Type File folder File folder File folder Text Document XML File	Size		

Fig 4.1.5: Selecting base file list text file for Child Refresh

Base File list is a text file which contains the absolute path of all the parent radxmls to be refreshed (here STDCIFD is the parent radxml)

If child refresh of more than one functionId is required, absolute path of each parent radxmls has to be specified; each in a new line

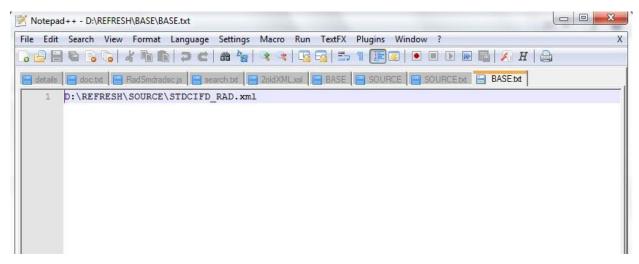


Fig 4.1.6: Content of base.txt file

File Location: Choose file location as client if the path provided is in the client machine .

Refresh Type: Choose Refresh type as Child Refresh

Source Release type and base release type will be disabled for child refresh as the release type of both parent and child is assumed to be same .

Click on Refresh button on lower left portion of the screen and wait for the system to do the process.

Process time will vary depending on the number of files provided, size of each files etc

	File List SOURCE.txt	BROWSE	Base File List		
anges Source Relea			Base Release Type	KERNEL -	
tion	esh Type Child Refresh	•			
ate Generation SI.No	File Name	Sub Folder	File Status	Error Description	*
s 1	STDCIFDC_RAD.xml	REFRESH	Refreshed *		

Fig 4.1.7: File Status after Refresh

After Completion of the process, status will be shown in the screen. File status will be successful if refresh is successful.

Save Mode should be either Client Path or Save path for Refresh activity. Zip mode is not supported. Files will be generated in the Work Directory specified.

Generated Files:

- Refreshed Radxml :A folder named RADXML will be created within the source file path which will contain refreshed files for the particular source(child) radxml.
 For instance, if source file path is D:\REFRESH\SOURCE\ STDCIFD_RAD.xml; refreshed file can be found at D:\REFRESH\SOURCE\RADXML\ STDCIFD_RAD.xml
 For child refresh of multiple files, it is recommended to place all source radxmls in one folder so that generated files could be found at a single location
- 2) Log Files : Following log files will be generated
 i)Refresh Log : This contains the status of all the files refreshed.
 ii)Refresh Report : This file can be used for troubleshooting .

4.2 Functionality Demonstration

In the above Child Refresh process, STDCIFDS is refreshed with the latest STDCIFD.

The figure below shows the preview of STDCIFD and STDCIFDC main screens before refresh

Fønc	tion Ge	eeratioe		•	Main											X			
	S	Functio	tion Loa on K <mark>STC</mark> Pain STC		New 🔄	Custome N	r No ame Type									ianos Judil		×	
	🚞 Prefi	erences		MI	S Char	ige Log		 			 			 					0 00 9
	Dala Dala Scre	ens VS_MAIN			G	Maker hecker Jod No		B	Date Tir Date Tir ecord Sta	ne:					Exit				
1	🛄 Actio	ST_CUST Ins	1						zation Sta			ument v	0101140000000	 IET CIOL K		rang ot Fie	hid	Active	
	Calil Laur Sum	nchForms						 			101	,	uno.	 UT LIVER		- 14 - 19 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -		PROMITE	

Fig 4.2.1: STDCIFD screen before changes

Function Generation		🔶 Main													1	X				" ×
		Et New E	PEnter Que	ery														e 🔳 1	7 🧃	2
Action Loa Function Id Str Save XML Path STD Search	CIFCC		Add	lame iress Type ()	Individua Corpora											iance v	•	- 5	1 27 4 9	
Preferences DataSource ListOfratues DataBlocks DataBlocks Screens CVS_MAIN FieldSets FST_CUST1			Maker Checker Mod No				Date Tim Date Tim ecord State zation State	ie: us							Exit			+		
FST_CUST2			Argument	Name	SC	orce Bloc	*	Source	Field	Arg	ument va	lite	Larg	et Block		arget Fiel	Id	Active	~	
Actions CaliForms LaunchForms Summary																				

Fig 4.2.2: STDCIFDC screen before refresh

From the screen preview it can be noted that in the child screen many changes has been done which had resulted in a very different layout. Many field sets which were part of the parent screen has been made hidden and new field sets containing new fields has been introduced in the child screen.

Now we load STDCIFD in Workbench in the current release and made some modifications to it as required. A new field COUNTRY AND NATION have been introduced

Preview of STDCIFD after the modifications is shown below. Note the newly added field highlighted.

Function Generation	🔶 Main				×
	E New P Enter Query			(in	🗙 🗐 🖗 👹 🛹
Action Load - Function (STOCIFD Save XML Pain STDCIFD_RAI	Customer No Name Type Address		Country Nationality	v soner solutions solutions	
Search					- 🛯 🏹 🗳
Preferences	MIS Change Log			×	
DataStorice DataStorice DataBlocks DataBlocks Screens	Maker Checker	Date Time: Date Time:			
Greats Greats Greats Greats FieldSets FST_CUST1	Mod No	Record Status Authorization Status		Exit	+
F8T_CUST5 Actions CallForms	мующенс вание	2001 CV DIOLK 200	લઇન તરાવ મધ્યપ્રધાનમાં આવેલ	rarger chock rarget Field	Active ^
LaunchForms					

Fig 4.2.3: STDCIFD screen after modifications

Child Refresh of STDCIFDC is done as explained in previous section.

The system units(main packages, language xml.sys js ,xsd's etc) are regenerated by loading the refreshed radxml and deployed .All the units need to be regenerated. Preview of STDCIFDC main screen after refresh is shown below

Ex New Exter Query Customer Na* Address Language	
Type Individual Corporate Country	
MIS	
Maker Date Time: Checker Date Time: Mod No Record Status Authorization Status	t

Fig 4.2.4: STDCIFDC main screen after child refresh

Here we can find that the field added in parent screen has come in the child screen as well. Meanwhile ,other differences we have noticed between the initial parent and child screens has not come up as they were over ridden in the child functionId.

Hence we find that the changes done in the parent has come up in the child while retaining the changes done in the child .Note that only screen layout changes has been explained in this demonstration for ease of understandability ;this is applicable for all nodes(e.g.: cal form, launch form, lovs etc.)

5 Screen Child Refresh

Screen Child Refresh allows the developer to upgrade a screen child radxml with its latest parent radxml .In doing so; the latest changes done in parent functionId would be reflected in the screen child functionId while retaining all the changes done in the screen child level

- This process is to be done within a release. i.e. screen child functionId has to be refreshed with its parent functionId from the same release
- If the parent functionId of the screen child is a child screen ,then it is recommended that child refresh of that screen to be carried out before doing screen child refresh

- It is recommended that this process is done before development cut of the release for all child radxmls within a release. For instance; if development has happened parallel for a screen child and its parent functionId during a release, screen child refresh should be done before base lining so that screen child and parent record types are consistent
- All the system units need to be regenerated after Screen Child Refresh. A thorough unit testing is
 recommended after deployment of all generated units. Note that only frontend units will be
 generated for a screen child functionId.

5.1 Process Steps

For explanation purpose two dummy functionId's has been used : STDCIFD: parent screen STDCIFDC: screen child of STDCIFD

Process steps are similar to child refresh. Refer section 4.2 for more detailed explanation

In the Refresh Page, provide input to fields as:

Source File List: Browse and select the text file containing source file list

Source File list is a text file which contains the absolute path of all the screen child radxmls to be refreshed.

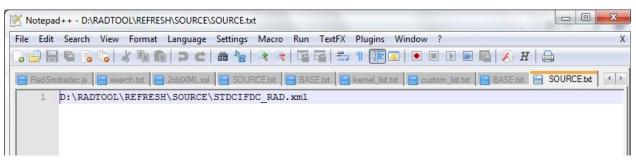


Fig 5.1.1: Content of source.txt file

The figure above shows the content of the source.txt file .Here STDCIFDS is the screen child radxml which has to be refreshed.

If screen child refresh of more than one functionId is required, absolute path of each screen child radxmls has to be specified; each in a new line

Base File List: Browse and select the text file containing base file list

Base File list is a text file which contains the absolute path of all the parent radxmls to be refreshed (here STDCIFD is the parent radxml)

If screen child refresh of more than one functionId is required, absolute path of each parent radxmls has to be specified; each in a new line

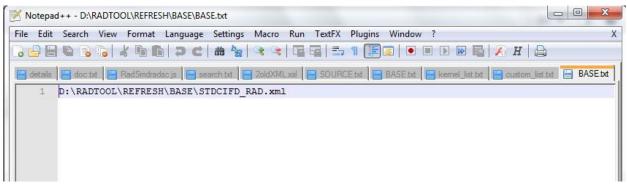


Fig 5.1.2: Content of base.txt file

File Location: Choose file location as client if the path provided is in the client machine.

Refresh Type: Choose Refresh type as Screen Child Refresh

Source Release type and base release type will be disabled for Screen child refresh as the release type of both parent and child is assumed to be same.

Click on Refresh button.

After Completion of the process, status will be shown in the screen. File status will be successful for refresh is successful.

mizer So	urce File List	SOURCE.txt	BROWSE	Base File List	BASE.txt	BROWSE
	elease Type	KERNEL -		Base Release Type	KERNEL -	
	Refresh Type	Screen Child Refr	esh 🔻			
1						
Generation SI.No)	File Name	Sub Folder	File Status	Error Description	on
1		DC_RAD.xml	REFRESH	Refreshed -		
1.3	orboil	00_1000.xm	REIREON	[tonobhod		

Fig 5.1.3: File Status after Screen Child Refresh

After Completion of the process, status will be shown in the screen. File status will be successful if refresh is successful.

5.2 Functionality Demonstration

In the above Child Refresh process, STDCIFDC is refreshed with the latest STDCIFD.

The figure below shows the preview of STDCIFD and STDCIFDC main screens before screen child refresh

Function Generation		— X
		🖩 🗵 🗐 🗸 📦 🗢
Action Load * Function Id STOCIFD Save XML Path STDCIFD_RAI	Function Type Parent Parent	Function Calagory Maintenance V Header Template None V Fooler Template Maint Audit V
Search	Screen Details	- 2 🖓 🗳
Preferences LatChyalues LatChyalues LatChyalues LatChyalues Costenes LatChyalues LatChyalues LatChyalues LatChyalues CostEnes LatChyalues	Screen Name CVS_MAIN Visio	
	Group Id Customer No Relation	
	MIS Change Log Maker Date Time: Checker Date Time: Mod No Record Status Authorization Status	Exit



inction Generation		
Action New * Function Id STDCIFUC Save XML Path	Function Type Corsen Child Parent Function STDC/FD Parent Xml STDC/FD Parent Xml STDC/FD R4 PROWSE	Function Calegory Mainlenance • Header Template None • Foolar Template Maint Audit •
Save AML Parr	Screen Details	- El 🖸 🖗
DalaBlocks DalaBlocks DalaBlocks DalaBlocks DalaBlocks	Screen Name CVS_MAIN	IF Main Screen
Screens CVS_MAIN	E New S Enter Query	
HEADER BOUY TAB_HANN SEC_CUST SEC_GROUP	Customer No Name Name Type Address	Country
 E FOOTER FieldSels 	H I of 1 H Go to Page	
Actions	Group Id Customer No Relation	
	۲. m	
	MIS Change Log	<u> </u>
	Maker Date Time: Checker Date Time:	Exit
	Mod No Record Status Authorization Status	

Fig 5.2.2: Preview of STDCIFDC before screen child refresh

Let us assume that some changes are done in STDCIFD as part of the current release. New field has been added and introduced in the screen. Preview of STDCIFD main screen after changes is shown below

Function Generation							XE	×
Action Load Function Id STDCIFD Save XML Path STDCIFD_RAI		Function Type Parent Parent Function Parent Xml	•		Function Calegory Iv Header Template Iv Fooler Template Iv	lone 💌		
Search	Screen Detaile					<		1 🛛 🗐 📩
Preferances DataSource ListOValues DataBlocks DataBlocks BCSCreens BCSCREAN	E New P Enter Query Customer No Name Type Address			Nationality Language				
FieldSels FieldSels FST_CUST1 FST_CUST2 FST_CUST4 Actions CaliForms LaunchForms Summary	Group Id	Go to Page Customer No	Relation			arget Field	Active	
			m		•			
	MIS Change Log					A		
	Maker Checker		e Time: e Time:		Exit			
	Mod No	Record Authorization				_		

Find the newly added fields (Nationality and Language) placed in a new field set highlighted in the figure

Fig 5.2.3: Preview of STDCIFD after changes

Do screen Child Refresh for STDCIFDC with the latest parent (i.e. STDCIFD with new fields and field set). Regenerate system units for the refreshed radxml and deploy.

Function Generation								. >
						× =	V 🧃	
Action Load Function Id STDCIFDC Save XML Path D:RADTOOLV	Function Type Screen Child Parent Function STDCIFD Parent Xml		Function Category Maintenance Header Template None Footer Template Maint Audit					
Search	Screen Details					-	e 🗖 🄊	
9 DataBlocks 9 Screens 9 CVS_MAIN 9 FieldSets	Screen Name CVS_MAIN Screen Title Main Screen Size Medium • Exit Button Type Default Cancel •	Ξ	I⊄ Main Screen I⊄ Visible					
	♦ Main			×		Populate	+ -	
	New Enter Query				get Field		Active	
	Customer Number Name Type Address		Nationality Language Country					
	Group Id Customer No	Relation		~				
	Checker	III Date Time: Date Time: cord Status		Exit				

Fig 5.2.4: Preview of STDCIFDC after screen child refresh

Here we can find that new fields and field set added in the parent has come in the screen child while the original screen child changes has also been retained.

6 Source Refresh

Source Refresh allows the customer upgrade his existing release with latest release of Flexcube without affecting his custom changes .By using source refresh option all the extensible radxml's of older version can be updated with latest version changes.

- Source Refresh is possible only for the extensible screens. Hence for non extensible screens customization on the screens can't be retained in case of upgrade
- Source Refresh is done for radxmls in different releases.
- All system units needs to be regenerated after source refresh . A thorough unit testing is recommended after deployment of all generated units

- Child and Screen Child Refresh will be done implicitly during Source Refresh if any child/screen child screens are present .Hence if source refresh of any child/screen child has to be done, include parent radxmls also in the source and base file lists
- Select proper release types for source and base while upgrading in Refresh Page.

It is meaningless to do source refresh between two Kernel versions (or two cluster versions etc) as we can replace the entire source with latest version in such scenario. *Hence Source and Base Release types can never be the same for Source Refresh*

Source release type cannot be **Kernel** it can be either Cluster or Custom. Base Release type options will depend on the source release type selected.

Source Release Type	Cluster	Custom
Base Release Type	Kernel	Kernel, Cluster

- If user selects custom as source release type he has option to upgrade his release based on either cluster pack or Kernel.
- If user selects Cluster as source release type we have only one option as base release type i.e. Kernel.

6.1 Process Steps

Consider a bank which is running on 12.0 version of Flexcube .Bank has done custom developments on top of 12.0 Kernel version .Now bank is upgrading to 12.0 sources Here we consider the case of a single functionId (STDCIFD) for demonstration

Process steps are similar to child refresh. Refer section 4.2 for more detailed explanation

In the Refresh Page, provide input to fields as:

Source File List: Browse and select the text file containing source file list

Source File list is a text file which contains the absolute path of all the source release radxmls to be refreshed. Here 11.3 custom radxmls used by bank is the source .

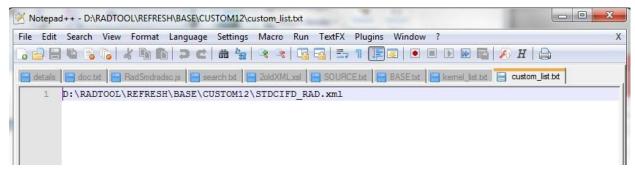


Fig 6.1.1: Content of custom_list.txt file

The figure above shows the content of the source.txt file .Here STDCIFD is the 12.0 custom version radxml which has to be upgraded to 12.2 .If source refresh of more than one functionId is required, absolute path of each source radxmls has to be specified; each in a new line

Base File List: Browse and select the text file containing base file list

Base File list is a text file which contains the absolute path of all base version radxmls with which source has to be refreshed. If source refresh of more than one functionId is required, absolute path of each base version radxmls has to be specified; each in a new line

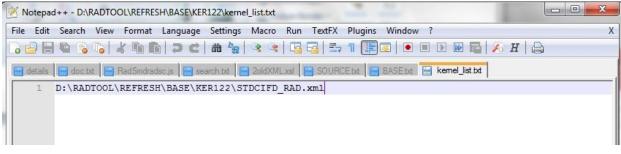


Fig 6.1.2: Content of Kernel_list.txt file

File Location: Choose file location as client if the path provided is in the client machine.

Refresh Type: Choose Refresh type as Source Refresh

Source Refresh Type: Source files are of custom release type (12.0 custom version) ,hence provide source refresh type as custom

Base Refresh Type: Base files are from 12.2 Kernel release. Hence select base release type as Kernel

administration	Refresh						_ ×	
Eunction Generation	Source	File List	custom_list.txt	BROWSE	Base File List	kernel_list.txt BRC	WSE	
Screen Customizer Tracking Changes	Source Relea		Custom -		Base Release Type			
Notification Triggers		esh Type	Source Refresh	-	base Release Type			
Notifications		son type						
Bulk Generation								
Excel Template Generation	SI.No		File Name	Sub Folder	File Status	Error Description	~	
🔤 Refresh 🗀 Web Sevices	1	STDCI	FD_RAD.xml	BASE	Refreshed -			
Block Detail Upload	L.*	ULD UI	D_rowshim	Bride				
Test Case Definition								
Runchart Definition								
Test Case Data Upload								
Create Request								
Test Case Execution								
Runchart Execution								
Test Case Data Bulk Upload								
Execution Report								
Purge Generation								
-								
							-	
						R	efresh Close	

Fig 6.1.3: Release type Selection for Source Refresh

Click on Refresh button.

After Completion of the process, status will be shown in the screen. File status will be successful for refresh is successful.

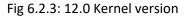
6.2 Functionality Demonstration

In the above section process for upgrading a 12.0 custom release functionId (STDCIFD) with its 12.2 version is explained

The figure below shows the preview of STDCIFD screen as used by the bank; i.e. 12.0 custom version

In custom version, Auto Generate button which was present in 12.0 Kernel version was not required; hence made hidden. Highlighted section shows the original position of Auto Generate button in kernel version of 12.0.

Main	× –
New Enter Query	🔚 🗵 🗐 🖓 🤤 🤤
Customer Number Name Type Address Anto Geo	inerate
	- 2 🗖 🔊
Group Id Customer No Relation	
e	Populate + -
	arget Field Active
MIS Change Log	
Maker Date Time: Checker Date Time:	
Mod No Record Status	Exit
	New Enter Query Customer Number Type Address Auto Ge Group Id Customer No Relation Image: Change Log Maker Date Time: Checker Date Time:



Function Generation	🔶 Main	×	
	New Enter Query		🖫 🗶 🗏 7 🎯 🔿
Action Load Function Id STDCIFD Save XML Path D:RADTOOLV	Customer Number Type	Address	
Search Preferences DataSource Image: STTM_CUSTOMER Image: STTM_CUST_GROUP Image: ListOfvalues Image: DataBlocks Image: ListOfvalues Image: ListOfvalues	Group Id Customer No Relation	· · · · · · · · · · · · · · · · · · ·	Populate + -
CUSTTYPE CNAME ADDR1 CNTY NLTY LANG BTM_MIS BTM_MIS	MIS Change Log Maker Date Time: Checker Date Time: Mod No Record Status	Exit	arget Field Active
BIL CROUP CVS_MAIN CVS_MAIN CVS_MAIN FieldSets Actions CallForms LaunchForms Summary			

Fig 6.2.1 : 12.0 Custom version of STDCIFD screen used by bank

The figure below shows the preview of 12.2 Kernel version of STDCIFD.

Notice some of the changes done in 12.2 Kernel version highlighted in the figure

- 1) Country field is added in the body
- 2) Nationality fields is added in body

Also note that Auto Generate button has been retained in 12.2 Kernel version from 12.0 Kernel

Function Generation	◆ Main				"" X
	E New E Enter Query		x		7 🍯 🔿
Action Load Function Id STDC/FO Save XML Path STDC/FD_RA	Customer No * Name Type Address Country Nationality	Jance	•		
Search	K ◀ 1 of 1 ▶ H Go to Page				X 19
Preferancas DataSource DataSource DataBlocks DataBlocks DataBlock DataBlock DataBlock CUSTIMER CUSTIMER CUSTINO	Group Id Customer No Relation				
CUSTTYPE CRAME ADDR1 CNTY		et Field	1	Active	
INLTY	MIS Change Log				
LANG BTM_MIS BTN_DEM() BTN_DEM() BLK_CROUP	Maker Date Time: Checker Date Time:				
🖃 🧰 Screens 🖃 🧰 CVS_MAIN	Mod No Record Status Authorization Status				
FieldSels					
FST_CUST2 FST_CUST3 Actions					
CallForms LaunchForms Summary					

Fig 6.2.2: 12.2 Kernel version of STDCIFD screen to which bank source has to be upgraded

Do Source Refresh as explained in the previous section.

Regenerate all system units (main package, language xml, sys js ,xsds etc) and deploy in Flexcube server

Compile/Deploy Kernel sources (kernel packages, kernel js etc) from the base release (12.2 here) in Flexcube server.

The figure below shows the preview of the screen after Source Refresh

Function Generation	♦ Main ×				- ×
	New Enter Query		X = 4	V 🧃	4
Action Load Function Id STDCIFD Save XML Path D/RADTOOL# Search	Customer Number * Name Type Address Country Nationality		- 8	1 🗖 🔊)
	Group Id Customer No Relation	arget Field	Populate	+ – Active	-
🗀 Summary	MIS Change Log				
	Maker Date Time: Checker Date Time: Exit				
	Mod No Record Status				

Fig 6.2.4: 12.0 Custom version of STDCIFD after upgrading to 12.2

Here we can observe that changes from 12.2 Kernel are now reflected in the custom version also.

1) Country field in body has come in the refreshed file

2 Nationality field of body has also come up in the refreshed screen from the base version

3) Auto Generate button has not come in the Refreshed screen even though it was present in the base screen. This is because it was made hidden in the custom version. Custom changes are retained

6.3 Source refresh is not possible in below scenarios

- 1. If Parent Function id LOV modified after child refresh if same LOV used in Child function id after next refresh those LOV'S Should be Modified manually
- If Fields data type are changed (examples Increase/Decreased length, Number to varchar2) in custom layer, Later Kernel changed same fields data type those changes would not be reflected in Custom layer. If these changes requires in Custom layer manually (Using ODT Function generation options to refresh data type) needs to change

3. The Following Custom attributes are changed in custom layer, Later Kernel same custom attributes are changes (Add/Deleted) those changes would not be reflected in Custom layer. If these changes requires in Custom layer (Using ODT Function generation options to refresh data type) manually needs to change

Radio Button, Static List values

- 4. There should not be nay naming conflicts in elements across releases (KERNEL, CLUSTER, CUSTOM) or across the parent-child functions For example: same field set should not be created in both Parent and Child Function Hence a standard naming convention has to be followed for each release type/and function type so that names does not conflict. For instance: All Fieldsets in a Custom Parent has to follow convention like FST_U.
- 5. Upgrade feature in not available for Summary Nodes



Development Workbench - Source Upgrade [August] [2021] Version 14.5.2.0.0

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

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

Copyright © 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.