Development Workbench - Source Upgrade Oracle FLEXCUBE Universal Banking Release 14.6.0.0.0 Part No. F56608-01 [May] [2022]



# Contents

1	Pr	eface	3
	1.1	Audience	3
	1.2	Related Documents	3
2	Int	troduction	4
2	2.1	How to use this Guide	4
3	٥v	verview of Refresh functionality in Oracle FLEXCUBE Development Workbench	4
4		nild Refresh	
4	4.1	Process Steps	5
4	4.2	Functionality Demonstration	10
5		creen Child Refresh	
Ę		Process Steps	
Ę	5.2	Functionality Demonstration	16
6		urce Refresh	
(	5.1	Process Steps	20
6		Functionality Demonstration	
6		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 applications	Self Acquired
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:

- <u>Chapter 2 , "Introduction"</u>
- 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 .

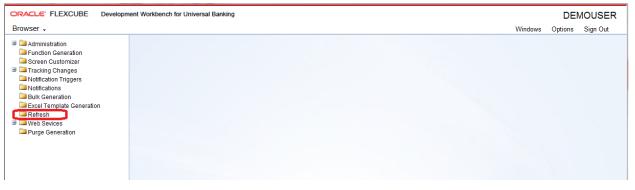


Fig 4.1.1: Development Workbench Landing Page

The following window will be launched

ORACLE FLEXCUBE Development Workben	ch for Universal Banking - Windows Internet Exp	lorer		The I have been been been been been been been be		_ 0 <mark>_ X</mark>
ORACLE' FLEXCUBE Develop	ment Workbench for Universal Banking				DE	MOUSER
Browser -				Windows	Options	Sign Out
<ul> <li>Administration         <ul> <li>Function Generation</li> <li>Screen Customizer</li> <li>Tracking Changes</li> <li>Notification Triggers</li> <li>Notifications</li> <li>Bulk Generation</li> <li>Recel Template Generation</li> <li>Refresh</li> <li>Web Sevices</li> <li>Purge Generation</li> </ul> </li> </ul>	Refresh Source Rile List Source Release Type Refresh Type Source Release Type Source Release Type Source Refresh Source Refres	esh Sub Folder	Base File List KE			

Fig 4.1.2: Workbench Refresh Screen

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

ORACLE' FLEXCUBE	Development Workbench for Universal Banking	DEMOUSER
Browser -		Windows Options Sign Out
Administration     Function Generation     Screen Customizer     Tracking Changes     Notification Triggers     Notifications     Bulk Generation     Excel Template Generation     Refresh     Web Sevices     Purge Generation	Upload File Browse	Base File List BROWSE BROWSE File Status Error Description
	Correction (Distance) > Computer > Data (D:) > RADTOOL > REFRESH > SOURCE	✓ 4+ Search SOURCE P
	Organize  New folder	
	odt_11_4     Name     ODT_SOURCE     ODT_SVN     ODT_SVN     ODT_11_4     pISQL     prshant     Preferences     RAD_WASTE     Name	Date modified Type Size 9/4/2013 3:43 PM Text Document 1 KB 9/4/2013 3:25 PM XIML File 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.

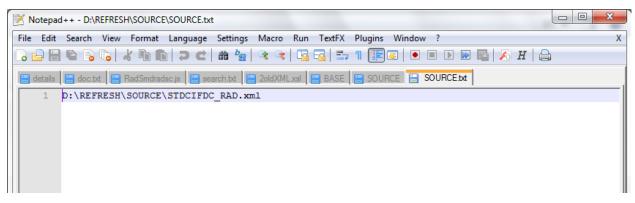


Fig 4.1.4: Content of source.txt file

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	evelopment Workbench for Universa	al Banking					DEI	MOUSE
Browser 🗸						Windows	Options	Sign Out
🗉 🗀 Administration	Refresh					***	* ×	
Function Generation Screen Customizer Tracking Changes	Source File List SC Upload	OURCE.txt	BROWSE		iE.txt RNEL ¥	BROWSE		
Notification Triggers     Notifications     Bulk Generation     Excel Template Generation     Refresh     Web Sevices     Purge Generation	Upload File	e	Browse	File Status Error Description				
	Choose File to Upload	ata (D:) 🕨 R	ADTOOL + REFRESH + BASE +		- <del>(</del>	Search BASE	×	1
	Organize   New folder					!≡ ▼ [	1 0	
	ODT_SOURCE	*	Name	Date modified	Туре	Size		
	DDT_SVN		LUSTOM12	9/4/2013 5:58 PM	File folder			
	ODT11_4		KER122	9/4/2013 5:59 PM	File folder			
	J pISQL		MY1	9/4/2013 5:17 PM	File folder			
	prashant Preferences		BASE.txt	9/4/2013 3:43 PM	Text Document	1 KB		
	RAD_WASTE	=	STDCIFD_RAD.xml	9/4/2013 1:36 PM	XML File	35 KB		
	INC Is							
	INC INC							
	inc Js							
	INC JS BLBLXML							
	INC JS ↓ LBLXML ↓ RAD_OP_FTVKK							

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

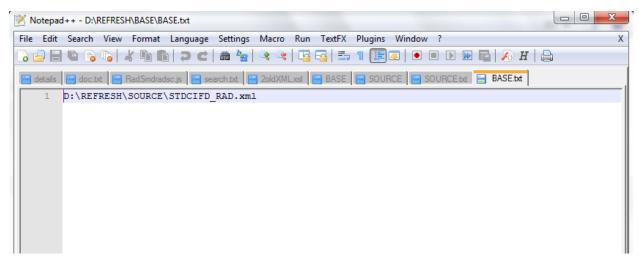


Fig 4.1.6: Content of base.txt file

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

💷 🥅 A daviala karkina	Refresh					_ ×
Administration     Function Generation     Screen Customizer     Tracking Changes     Notification Triggers     Notifications     Bulk Generation	Source File Li: Source Release Typ Refresh Typ	KERNEL -	BROWSE	Base File List Base Release Type	BASE.txt BROWSE	
Excel Template Generation	SI.No	File Name	Sub Folder	File Status	Error Description	*
<ul> <li>Refresh</li> <li>Web Sevices</li> <li>Purge Generation</li> </ul>		CIFDC_RAD.xml	REFRESH	Refreshed		
					Refresh	Close

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

Fenc	tion Generation			🔶 Main											×
				🖹 New 🏟 En	ter Query									1 🗏 7 📢	) 🥪
	Functio Save XML F	tion Load n Id STOCH Palh STDCIF	0	C	ustomer No Name Type Address							10	nce 💌		œ 🍐
Searc	ch													- 🔬 🟹	-9 - 1
	DalaSource			MIS   Change	e Log										
) ھ	ListOfValues DalaBlocks				laker ecker		Date 1	Time:							
9 (	Screens						Date 1	īme:				Exit			
	CVS_MAIN  FieldSets			Мо	d No		Record S								
	FST_CUST	I					Authorization S	tatus						+ -	
t	Actions CallForms			E AIG	DITIGHT BARON	3901	UP DIUCK	Source re	eng.	endinusian equa	 arger siock	9gnsr	t Field	Active	
	LaunchForms Summary														
:	_ our offers														

Fig 4.2.1: STDCIFD screen before changes

Function Generation	🔶 Main				*** ×
	🗗 New 🔄 Enter Query			📓 🗵 🗐 🐬	🧐 🚽
Action Load - Function Id STDCIFEC Save XML Path STDCIFEC_R	Customer No 🖗 Name Address Type 💿 Indivi			ance 🔹 Judii	
Search	Country	prate			a 🧐 🔺
Preferences					
DalaSource     ListOfValues	Maker Checker	Date Time:			
DataBlocks     Screens     CVS_MAIN		Date Time:		Exit	
FieldSels	Mod No	Record Status Authorization Status			
FST_CUST2				+	
FST_CUST3	Argument Name	Source Block Source Held	Argument Value   Arget Block	Larget Field Active	
CallForms					

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					"" ×
	🗗 New 🤄 Enter Query				🖪 🗷 🗏 🐬	🍯 🚽
Action Load - Function Id STDCIFD Save XML Path STDCIFD_RAI	Customer No Name Type Address		Country Nationality		vance - udit -	
Search					- AI	<b>i</b> 🧐 👌
Preferences	MIS   Change Log			<b></b>		
e 🖕 DalaSource e 🖕 ListOfValues e 💭 DalaBlocks e 🖕 Screens	Maker Checker	Date Time: Date Time:		Exit		
CVS_MAIN     FieldSels     FST_CUST1	Mod No	Record Status Authorization Status		EXIL	+ -	-
Actions CaliForms	Асуансисисизи	2081C4 DIOCK 24	onceven solmest some	rarger over rarg	et 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

A 11-1-			X
🔶 Main			
🖹 New 🔄 Enter Query			
Customer No Address Type Country	<ul> <li>Individual</li> <li>Corporate</li> </ul>	Name Language	
MIS			
Maker Checker	Date Time:		
Onecker	Date Time:		Exit
Mod No	Record Status Authorization Status		

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.

l++ - D:\R		REFRESH\SOURC	E\SOURCE.txt	:							x
Search	View Fo	rmat Language	Settings	Macro Ru	in TextFX	Plugins W	indow ?				Х
<b>B</b>	6 8	₩ 6   7 C	iii b	🤹 🗲   🛛		ា 厓 ዾ		🕨 🔤 🔊	$H \mid \langle$	3	
dradsc.js	📒 search.t	xt 📙 2oldXMLx	si 🔚 SOUR	CE.txt 📙 B	ASE.txt 📙	kernel_list.txt	😑 custom_lis	t.txt 🔚 BASE	txt 😑	SOURCE.txt	4 >
D:\RAD	COOL\RE	FRESH\SOURCI	STDCIFD	C_RAD.xm	1						
	Search	Search View Fo	Search View Format Language	Search View Format Language Settings	dradsc js 🔚 search.txt 🔚 20ldXMLxsl 🔚 SOURCE.txt 🔚 E	Search View Format Language Settings Macro Run TextFX	Search View Format Language Settings Macro Run TextFX Plugins W	Search View Format Language Settings Macro Run TextFX Plugins Window ?	Search View Format Language Settings Macro Run TextFX Plugins Window ?	Search View Format Language Settings Macro Run TextFX Plugins Window ?	Search View Format Language Settings Macro Run TextFX Plugins Window ?

Fig 5.1.1: Content of source.txt file

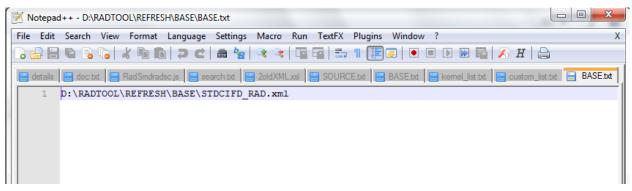
The figure above shows the content of the source.txt file .Here STDCIFDS is the screen child radxml which has to be refreshed.

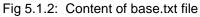
If screen child refresh of more than one functionId is required, absolute path of each screen child radxmls has to be specified; each in a new line

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





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.

Changes on Triggers ons	Source Rele Refr	ase Type KERNEL - esh Type Screen Child Refre	esh 👻	Base Release	Type KERNEL -	
neration mplate Generation	SI.No	File Name	Sub Folder	File Status	Error Description	<b>^</b>
ices eneration	1	STDCIFDC_RAD.xml	REFRESH	Refreshed -		
						÷

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

Acton Load  Acton Load  Parcel Function Gategory Maniferance  Parcel Function Gategory Maniferance  Parcel Function Gategory Maniferance  Hade' Template Manifer  Function Gategory Maniferance  Gategory  Gat	Function Generation						- ×
Functional of STECHE   Save XAL Path STDORD_RAL   Exaction   Scene Details   Perfeteences   DataSource   LaunchForms   Summary     Mater   DataSource   LaunchForms   Summary     Mater   DataSource   LaunchForms   Summary     DataSource     Mater   DataSource   LaunchForms     Summary     DataSource     Mater     DataTime:     Mater     DataTime:     Mater     DataTime:     DataSource     DataSource     Stream Table     Stream Table     Stream Table     Stream Table     Customer No     Relation     Image: DataBook     Stream Table						۶ 🌒	\$
Prinforences   DataSource   DataSource   DataSource   DataSource   Soreen Tain Main   Servens   Costomer No   PeidSels   Active   Type   Address   Summary     If 1 of 1 bit point Date   If 1 of 1 bit point Date   If 1 of 1 bit point Date     If 1 of 1 bit point Date <	Function Id STDCIFD	Parent Function Head	ler Templale None 👻				
BataSource   Screents   Screents   Collonarity   Collonarity <tr< th=""><th>Search</th><th>Screen Details</th><th></th><th></th><th>···· A</th><th>I 🖾 🦃</th><th>^</th></tr<>	Search	Screen Details			···· A	I 🖾 🦃	^
Mills     Change Log     *       Maker     Date Time:     *       Checker     Date Time:     Exit       Mod No     Record Status     Exit	DataSource ListOHatives ClastBlocis Screens G. CVS_MAIN G. FriddSets Actions CaliForms LaunchForms	Screen Title     Main     P5     Visible       Main     Address     Name     Name       Type     Address     Address       Group Id     Customer No     Relation	101	a ,			
Mills     Change Log     *       Maker     Date Time:     *       Checker     Date Time:     Exit       Mod No     Record Status     Exit							
		Maker Date Time: Checker Date Time: Mod No Record Status					

Fig 5.2.1: Preview of STDCIFD before changes

Function Generation		> 🗵 🗵 🗗 🕼 🜩
Action New + Function Id STDCIFDC Save XML Path	Function Type Screen Child  Parent Function STOCIFD Parent Xml STOCIFD Parent Xml STOCIFD_RAI	Function Calegory Mainlenance  Header Template None  Fooler Template Maint Audit
Search	Screen Details	AI 🖾 崎
DataBlocks      DataBlocks      BLK_CUSTOMER      BLK_GROUP	Screen Name CVS_MAIN  Recommendation Recommendatio Recommendatio Recommendation Recommendation Recommendation R	een X
Screens     CVS_MAIN	Et New 🛱 Enter Query	
HEADER     BODY     TAB_MAN	Customer No Country Name	· · · · ·
SEC_CUST	Address	rget Field Active 🔿
FOOTER     FieldSels	K 4 1 of 1 >> H Go to Page	+ - =
FST_CUST1 FST_CUST2 FST_CUST3 Actions	Group Id Customer No Relation	•
		_
	<u>د</u>	
	MIS   Change Log	•
	Maker Date Time: Checker Date Time: 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						XEV	×
Action Load  Function Id STDCIFD Save XML Path STDCIFD_RAI		Function Type Parent		Function Calegory Mai Header Templale No Fooler Template Mai	10 🔻		
Search	Scroon Dotaile		1	×	1	- *	a 🧐 🔺
<ul> <li>Preferances</li> <li>DataSource</li> <li>ListOf/alues</li> <li>DataBlocks</li> <li>Screens</li> <li>CVS_MAIN</li> </ul>	New E? Enter Query Customer No Name Type Address		Nationality Language				
FieldSals FST_CUST1 FST_CUST2 FST_CUST2 FST_CUST4 Actions CallForms LaunchForms Summary	Group Id	Customer No Relat	on		arget Field	Active	
	MIS   Change Log			*			
	Maker Checker	Date Time: Date Time:		Exit			
	Mod No	Record Status Authorization Status					

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.

Action     Load     Function     Type     Screen     Child     Function     Category     Maintenance       Function     Id     STDCIFDC     Parent     Parent     Header     Template     None     Image: Structure       Save XML Path     DrApDTOOLV     Parent     Xml     Footer     Template     Maint Audit     Image: Structure	
Action Load       Function Type Screen Child       Function Category Maintenance         Function Id STDC/FDC       Parent Function STDC/FD       Header Template None         Save XML Path D/RADTOOLV       Parent Xml       Footer Template Maint Audit         Search       Screen Details	> ₮ 🧃 🔿
B DataBlocks Screen Name VS_MAIN CVS_MAIN FieldSets Screen Size Medium ▼ Exit Button Type Default Cancel ▼ Main Screen Screen Size Medium ▼ Exit Button Type Default Cancel ▼ New Enter Query Customer Number Type Country	
Gutern Rain Guter	AL 🗔 🌍
New Enter Query     get Field       Customer Number     Nationality       Name     Language       Type     Country	
Customer Number Nationality Language	e + -
Name Language	Active
Group Id Customer No Relation	
Mis Change Log Maker Date Time: Checker Date Time: Checker 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 .

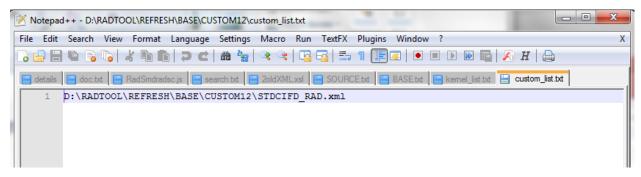


Fig 6.1.1: Content of custom\_list.txt file

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

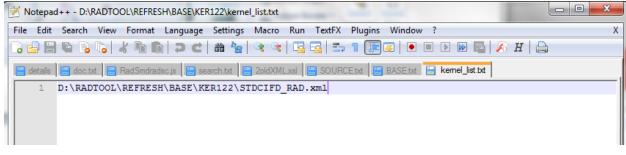


Fig 6.1.2: Content of Kernel\_list.txt file

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					_ ×
Function Generation Screen Customizer	Source	e File List custom_list.txt	BROWSE	Base File List	kernel_list.txt BROWSE	
	Source Rele	ase Type Custom -		Base Release Type	Kernel 🔻	
Notification Triggers		esh Type Source Refresh	•	Dase Release Type		
Notifications	TX01	ean type				
Bulk Generation						
Excel Template Generation						
Refresh	SI.No	File Name	Sub Folder	File Status	Error Description	<u> </u>
Web Sevices	1	STDCIFD_RAD.xml	BASE	Refreshed -		
Block Detail Upload						
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						
Fulge Generation						
						-
	L					
					Refresh	Close
					Removin	

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.

Function Generation	🔶 Main				×				_ ×
	New Enter Query						×	V 🧐	4
Action Load  Function Id STDCIFD Save XML Path D:\RADTOOL\F	Customer Number Type			Name Address Auto Gener	ate				
Search							- (	xi 🔯 🗳	2
Preferences  DataSource  DataSource  DataBlocks  DataBlocks  DataBlock  CUSTNO	Group Id	Customer No	Relation						
CUSTTYPE	•		III		+				
CNAME ADDR1							Populate		4
	MIS Change Log					arget Field		Active	
	Maker Checker		Date Time:						
BTN_DEMO BLK_GROUP	Checker		Date Time:		Exit				
G G Screens	Mod No	Re	cord Status						

Fig 6.2.3: 12.0 Kernel version

Function Generation	♦ Main X				_ ×
	New Enter Query	<b></b>	×	77 📢	3 🔿
Action Load  Function Id STDCIFD Save XML Path D\RADTOOLV	Customer Number Name Address				
Search Preferences DataSource SITM_CUSTOMER DISTTM_CUSTOMER DISTTM_CUST_GROUP DISTEROUP DataBlocks	Group Id Customer No Relation		-	a 🔾	<b>?)</b> ^
BLK_CUSTOMER	★ III				_
CUSTNO CUSTTYPE CNAME	MIS Change Log	arget Field	Populate	Active	
ADDR1	Maker Date Time:				
	Checker Date Time: Exit				
BTM_MIS	Mod No Record Status				
BTN_DEMO     BLK_GROUP     Screens     CYS_MAIN     FieldSets     Actions     CallForms     CallForms     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												J			50	* ×
	E4 New E	Penter Query														7 🌒	\$
Action Load 👻		Customer I						Name					ance	-			
Function Id STDCIFD		Ty Coun				-	Ade	dress	Auto Ge	marata			¥				
Save XML Pain STDCIFD_RAI		National							Autu Ge	arerare			udit	*			
Search		5 5 T			1											🗖 🧐	~
Preferences		of 1 🕨 🕅 🛛									+-						
DalaSource		Group Id	C	Customer No		Relation						<b>^</b>					
ListOfValues																	
DalaBlocks     BLK_CUSTOMER																	
												-					
CNAME ADDR1	•										+				+		
CNTY													et Field		Active	_ ^ _	
INLTY	MIS   Ch	nange Log											<b>I</b>				
iii LANG iii BTM_MIS		Maker			Dat	e Time:											
BTN_DEMO		Checker				<b>_</b> .											
					Date	e Time:						Exit					
Screens     CVS_MAIN		Mod No			Record												
E FieldSets				Au	thorization	n Status											
FST_CUST1																	
FST_CUST2																	
Actions																	
CallForms																	
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		×	1	<ul> <li>§</li> </ul>	⇔
Action Load  Function Id STDCIFD Save XML Path D/RADTOOL	Customer Number * Name Address		-	- <u>Aï</u>	<b>a</b> 5	
<ul> <li>Preferences</li> <li>DataSource</li> <li>ListOfValues</li> <li>DataBlocks</li> <li>Screens</li> <li>FieldSets</li> <li>Actions</li> <li>CallForms</li> <li>LaunchForms</li> </ul>	Group Id Customer No Relation	arget Field	Popula		+ – ctive	
🗀 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 [May] [2022] Version 14.6.0.0.0

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.