Oracle FLEXCUBE Universal Banking ® 12.87.03.0.0 Development Workbench- Screen Development II

June 2017



Contents

1	Pref	ace3
	1.1	Audience
	1.2	Related Documents3
2	Intro	oduction4
3	Gen	erated Files4
	3.1	Front-End Files
	3.2	System Packages7
	3.3	Hook Packages
	3.4	Meta Data9
	3.5	Others11
4	Dep	loyment11
	4.1	Prerequisites:
	4.2	Deploy Files:
5	Rele	ase14
	5.1	SVN Integration

1 Preface

This document describes the generation, deployment and release of units from Oracle FLEXCUBE Development Workbench for Universal Banking. This document explains the process of generating and deploying files from Development Workbench for a function id

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 Language	Self Acquired
Working knowledge of XML files	Self Acquired
Working Knowledge of Development	User Manuals
Workbench for Universal Banking	

1.2 Related Documents

04-Development_WorkBench _ Screen_Development-I.docx

<u>07-Development_WorkBench_Notifications.docx</u>

2 Introduction

This document gives information on the following topics:

- Chapter 2, "Introduction"
- Chapter 3 , "Generated Files"
- <u>Chapter 4 , "Deployment"</u>
- <u>Chapter 5 , "Release"</u>

3 Generated Files

Development Workbench generates following files as shown in the figure below. This screen

can be launched by clicking on icon from the function generation screen. This has to be done after design of screen is completed. Refer <u>04-Development_WorkBench</u> <u>Screen_Development-I.docx</u> for detailed explanation on screen development

on Generation					Window	s Options Sign (
						-
ate						
Front-End Files	System Packages	Hook Packages		Meta Data	Others	
RedXML	Main Package Spec	Kernel Package Spec	Menu Detaits	Cabel Details	IZ Xada	
Screen Xml	Main Package Spec	Kernel Package Spec	P Datasource Detaits	Block PK Columns	Xsd With Annotations	
System JS	Notification Triggers	Cluster Package Spec	LOV Details	Function Call Forms	C Screen Html	
	Upload Package Spec	Cluster Package Body	Block Details	C Gateway Details	I Upload Table Trigger	
	Upload Package Body	Custom Package Spec	Screen Details	Notification Details	Upload Tables Definition	
		Custom Package Body	Z Amendable Details	Punction Parameters	Archive Table Definition	
			Call form Details	Purge Details		
			Summary Details			
SLNo		File Name	File Ty	(De	Status	
						Generate Exit
						and the second second

Fig 3.1: File Generation Screen

For action New , Workbench automatically selects the all files as shown in figure. If function id has been loaded .then hook packages will not be checked

Click on Generate button to generated the files selected

									副無關係
mie									
Freed-Line Pi		ormation	×		Main De			Others	1
	Contraction of the second s			12 Menu Datalis	(F)	Label Details	12 Xoda		
Escreek Xmi Error Description Error Code		Error Code	Datapoorce Datates	192	Block PK Optumne	El Xad We	n Asviotations		
		1 Request successfully Processed	RD-SAVE-007	COV Details		Function Call Forms Galaxian Details	2 Screen	Html Table Tripper	
				Z Screen Details		Notrication Dutails		Table Ingger Tables Delivition	
				CAmendable Details	38	Function Parameters		Table Datakon	
				2 Call form Details 2 Summary Details		Purge Details			
				(K) Settle utility Concepted					
SE.No	-					file Type		Slates	(*)
	FTDTRON		-		LIDOM.		Generated	*	
	FTOTRON		Ok		JS		Cananated	÷	
	FIDTRON		(200)		HTML		Generalad	*	
	FTOTHON				HTML		Generated	*	
	TUTRONL_C	IVS_MAINTAB_MARKING			HTML		Generated	-	
	FTDTRONL_C	DVS_MAINTAB_PARTYDET html			HTML		Centrated	*	
	FTUTRONL_C	VS_MANTAB_OTHERDET.ntml			HTML.		Senerated	*	
	FTDTROM_C	NS_MARHTAB_SETTLROUTE.Nom			HTML				
	FTDTRONL_C	V6_MAIN_TAB_FOOTER ntmi			HTML		Generalad	*	
	FTDTRONL_C	20 STOP_PYMNT_TAB_HEADER.html			HTML:		Demmented	-	
	FTDTRONL_C	DVS_BTOP_PYMNIT_TAB_MAIN 2011			HTML		Geoecalad	7	
	FTOTRONL_C	WE_STOP_FYMENT_THE_FOOTER Intra			HTML.		Gemitated		

Fig 3.2: Files Generated message

3.1 Front-End Files

These files need to be copied to application server in respective folders.

• **RADXML** → This file contains the Data Used to Design screen. Radxml is Input to Workbench.

File Name : <Function_ID>_RAD.xml

• Screen Xml → This file is an XML markup of presentation details, for the designed Function ID specific to a language. XSL Transformation is applied to this XML file by linking it to an XSL file. This results in HTML Markup which is displayed by the browser.

File name : <Function_ID>.xml

• **System JS** → This JavaScript file mainly contains a list of declared variables used by FLEXCUBE Infra.

- o msgxml: This variable is used by the system to build FCUBS Request XML
- o dataSrcLocationArray: This variable is an array of DATA BLOCKS
- relationArray:-This array contains relation and relation type details of blocks.This is used for data binding
- o retflds and bndFlds:- These arrays contains LOV information
- CallFormArray, CallFormRelat, CallRelatType:- These arrays contains callform details, call form relation and relation type
- actionsAmmendArray: This array contains information for enabling fields based on actions

File name : <Function_ID>_SYS.js

• Release Specific JS →

Note: Release Specific JavaScript File will not be generated by Workbench. This file, if required, has to be created by developer.

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 KERNEL release would add functions based on two categories:

- Functions triggered by screen loading events Eg: fnPreLoad_KERNEL(),fnPostLoad_KERNEL()
- Functions triggered by screen action events Eg: fnPreNew_KERNEL (),fnPostNew_KERNEL ()

A developer working in CLUSTER release would add functions based on two categories:

- Functions triggered by screen loading events Eg: fnPreLoad_CLUSTER(),fnPostLoad_CLUSTER()
- Functions triggered by screen action events Eg: fnPreNew_ CLUSTER (),fnPostNew_ CLUSTER ()

In case if any function in KERNEL JavaScript file has to be modified, this can be achieved by overriding the function in CLUSTER JavaScript file.

The functions in this file are generally triggered by screen events. A developer working in CUSTOM release would add functions based on two categories:

- Functions triggered by screen loading events
 Eg: fnPreLoad_CUSTOM(),fnPostLoad_CUSTOM()
- Functions triggered by screen action events Eg: fnPreNew_CUSTOM (),fnPostNew_CUSTOM ()

In case if any function either in KERNEL JavaScript file or CLUSTER JavaScript file has to be modified, this can be achieved by overriding the respective function in CUSTOM JavaScript file

File name : <Function_ID>_<RELEASE_TYPE>.js

3.2 System Packages

- Main Package Specification: The Main Package Spec Contains all the Declarations.
- Main Package Body:

Main Package would contain the all the business logic and persistence code for the functioned. It will also contain call to hook packages .Different flavors of main Package generated are

- 1) Maintenance FunctionId
- 2) Transaction FunctionId
- 3) Others Category FunctionId
- 4) Report Parameter Screen
- 5) Notification
- 6) Purge Entity

The main package has the below stages in case of a maintenance function:

- Converting Ts to PL/SQL Composite Type
- o Checking for mandatory fields
- o Defaulting and validating the data
- o Writing into Database
- Querying the Data from database
- Converting the Modified Composite Type again to TS

Each of these stages has a 'Pre' and 'Post' hooks in the Release Specific Packages.

Note : Main Package has the system-generated code and should not be modified by the developer.

- Upload Package Specification
- Upload Package Body

Upload Packages would support adapter upload flow of any FunctionId. The package will contain code to convert upload table data to function id specific

PL/SQL composite type. The main package of the functioned would be called to upload the data

Notification Triggers

This will be generated only for notification trigger screens. Refer <u>07-</u> <u>Development_WorkBench_Notifications.docx</u> for further reference

3.3 Hook Packages

Kernel, Cluster and Custom Packages are the packages where the respective team can add business logic in appropriate functions using the Pre and Post hooks available.

• Kernel Package Spec

• Kernel Package Body

The Kernel package is solely for the Kernel Team to modify. The Main package has designated calls to the Kernel package for executing any functional checks or validations included in the Kernel Package.

All the user level validations and conditional operations should be included in Fn_Post_Default_and_Validate. This function is called from the Main Package after the execution of Fn_Default_and_Validate. User should avoid putting validations or code in any other function.

In case user needs to add a separate function, the existing Workbench generated structure should not be changed. Instead the user can create a new package e.g. <MODULE>PKS_<FID>_UTILS package. The desired function can be included in this package and the call can be made from the Kernel Package.

- Cluster Package Spec
- Cluster Package Body

The Cluster package is available to the Cluster Team to add any validations or Checks specific to the Cluster Team over and above the Kernel Team. The Kernel Team or the Custom Team should not modify the contents of this package.

- Custom Package Spec
- Custom Package Body

The Custom package is available to the Custom Team only to add any validations or Checks over and above those already present in the Kernel and Cluster Packages.

3.4 Meta Data

Workbench generates **INC** files which contains insert scripts. These are static data required for the functioning of the function id. Developer needs to deploy generated INC's in FLEXCUBE schema.

• Menu Details → Under menu Details below 4 table entry Insert Scripts will get generated.

SMTB_MENU SMTB_FUNCTION_DESCRIPTION SMTB_ROLE_DETAIL SMTB_FCC_FCJ_MAPPING

If any one of table entry missing while designing new function ID, FLEXCUBE won't allow launching function ID.

- **Data source Details** → Generates INC for CSTB_FID_DATA_SOURCES. This data is required for uploading data to FLEXCUBE for the function id through excel.
- Lov Details → Generates INC for CSTB_LOV_INFO. FLEXCUBE will be using this information to get data for List of values provided in the screen
- **Block Details** → Generates INC for CSTB_FID_DATA_BLOCKS. This data is required for uploading data to FLEXCUBE for the function id through excel.
- Screen Details → Generates INC for CSTB_FID_SCREENS.
- Amendable details → Generates INC for below table. These scripts provides information regarding the amendable fields in the screen .Entry has to be present for the required field in these tables in order to make that field amendable through FLEXCUBE and gateway operation

GWTM_AMEND_MASTER GWTM_AMEND_NODES GWTB_AMEND_NODES GWTB_AMEND_FIELDS

• **Call form Details** → Generates INC for CSTB_CALL_FORM_NODES. These will be generated only if the screen is a Call Form. It is required for attaching the particular Call form to any other function id.

- **Summary Details** → Generates INC for CSTB_SUMMARY_INFO. Dat in this table is required for fetching records in the summary screen of the particular function id .
- Label Details → Generates INC for CSTB_FIELD_LABELS and CSTB_OTHER_LABELS. Labels can be used for Language translation of screens. These scripts provides information regarding the label codes attached to each field. These are useful while developing language packs for FLEXCUBE

CSTB_FIELD_LABELS → Contains Field Label details. CSTB_OTHER_LABELS → Contains SCREEN, TAB, FIELDSET, BLOCK etc label details.

Note that Workbench does not generate scripts for CSTB_LABELS. Data in CSTB_LABELS is maintenance required for the Tool.

- **Block Pk Columns** → Generates INC for STTB_AUDIT_PK_COLS.
- **Function Call Forms** → Generates INC for CSTB_FID_CALLFORMS. This data is required for uploading data to FLEXCUBE for the function id through excel.
- Gateway Details → Generates INC for below tables.
 Data in these tables are required for gateway operations (web services) for the particular screen

GWTM_FCJ_FUNCTIONS GWTM_OPERATIONS_MASTER

• Notification Details → Generates INC for below tables. Notification INCs will get generated only for Notification screens

GWTM_NOTIFICATION_TRIGGERS GWTM_NOTIFICATIONS_MASTER

• Function parameters → Generates INC for below tables. This INC gets generates only for Reports screens (i.e. Function Category – Report).

AETB_FUNC_MASTER AETB_FUNC_DETAIL

• **Purge Details** → Generates INC for below tables . These scripts will be generated only for Purge Entity Definitions screens

STTM_PURGE_MASTER STTM_PURGE_TBL_DETAILS STTM_PURGE_FILTERS

3.5 Others

- Xsds → The XSD files are used for the creation of xmls, which are in turn used during the execution of operations through Gateway.
- Xsds with Annotation → Xsds will be generated with annotations. This can be used for document creation of any Web Service
- Screen html → The HTML files are used to take screen shots by converting them to images, which in turn used for Document preparation.
- **Upload Table Trigger** → Trigger will be generated for inserting records into adapter upload process tables (CSTB_EXT_CONTRACT_STAT ,STTB_UPLOAD_MASTER) from upload tables.
- **Upload Table definition** → DDL for upload tables would be generated. If existing upload tables are being re used , the generated DDL's can ignored
- Archive Table definition → DDL for archival tables would be generated for Purge Entity Radxml. Synonyms for the tables would also be generated.

Note: For new screen all file options will be checked automatically. For modification only system generated files will be checked, based on user requirement can opt for other file generation.

4 Deployment

Workbench generated files can be deployed directly to the application server and schema. Front-end files get copied to desired location which user has maintained while adding environment details. Database files get compiled to FLEXCUBE schema which is specified in the working environment.

4.1 Prerequisites:

The below maintenance is required to use the deploy option.

In environment user has two options to copy files based on the operating system FLEXCUBE application is running.

- Windows : File Copy / FileManager
- Unix : FileManager

FileManager Servlet needs to deploy in application server where FLEXCUBE is running.

Refer File manager section in <u>02-Development_WorkBench_Administration.docx</u> to get more details about file manager.

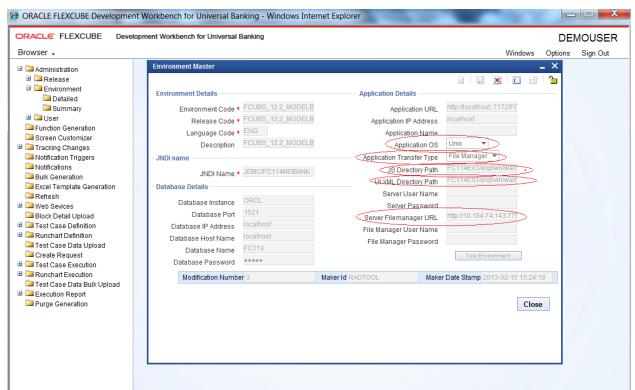


Fig 4.1: The Environment Details

In environment creation below details are mandatory to use deploy option.

- Application Operating System → Unix/Windows
- Application Transfer Type → Windows → File Copy/FileManager
 UNIX → FileManager
- JS Directory Path : Enter the shared path of JS files.
- **Uixml Directory Path:** Enter Shared path of Language Xml files.

Note: If appliaction server is windows user can enter absolute path.

4.2 Deploy Files:

Click Deploy button as shown in below image.

VICLE FLEXCUBE De	evelopment Workbench for Uni	versal Banking			DEMOUS Windows Options Sign (
ion Generation					
Front-End Files	System Packages	Hook Packages	Me	ta Data	Others
Screen Xml System JS	Main Package Spec Main Package Body Notification Triggers Upload Package Spec Upload Package Body	Kernel Package Spec Kernel Package Body Cluster Package Spec Cluster Package Body Custom Package Spec Custom Package Body	 Menu Details Datasource Details LoV Details Block Details Screen Details Screen Details Call form Details Summary Details 	Label Details Isock PK Columns Isock PK Columns Function Call Forms Gateway Details Notification Details Function Parameters Purge Details	Xsds Xsd With Annotations Screen Html V upload Table Trigger V upload Tables Definition Archive Table Definition
SI.No	File I	Name	File Type		Status

Fig 4.2: The deployment Screen

Select required files to deploy .Click deploy button. On success full deployment user will get status as Deployed.

	FLEXCUBE Development Workbench for Universal	Banking				MOUSE
rowser	-				Windows Options	Sign Out
nonon oc	amanon				x =	- 7
decella						
formatio	n		× Me	ta Dala	Others	^
			fenu Details	Label Details	Xada	
	Error Description	Error Code	Dalasource Cletails	Block PK Columns	Xsd With Annolations	
	Request successfully Processed	RD-SAVE-007	.OV Details Block Details	Function Call Forms Gateway Details	Screen Himi	
			Screen Delalis	Notification Details	Upload Tables Definit	ion
			kmendable Delails	Function Parameters	Archive Table Definition	n
			Sall form Cletails Summary Details	Purge Details		
			summary Denaits			*
				File Type	Status	~
		-	SPC		Deployed *	
		Ok	SPC		Deployed 👻	
		Ŭ.	SOL		Deployed *	
			SQL.		Deployed 👻	
5	CST8_FIELD_LABELSFTDTRONLINC		INC		Deployed *	
6	CSTB_OTHER_LABELSFTDTRONL INC		INC		Deployed 💌	L
7	CST8_FID_CALLFORMSFTDTRONLINC		INC		Deployed 💌	
8	COTB_LOV_INFOFTDTRONLINC		INC		Deployed 💌	
9	CSTB_SUMMARY_INFOFTDTRONLINC		INC		Deployed *	
10	CSTB_FID_DATA_BLOCKSFTDTRONLINC		INC		Deployed 👻	
11	CST8_FID_DATA_SOURCESFTDTRONLINC		INC		Deployed *	
	CSTB_FID_SCR_TABSFTDTRONLINC		INC		Deployed V	

Fig 4.3: Deployment of selected files

5 Release

Release feature of Workbench is applicable only for developers within Oracle Financials. It involves the integration the Tool with other in house tools (repository/version control tools)

Oracle FLEXCUBE Development Workbench is integrated with SVN.

5.1 SVN Integration

Now, Workbench provides an interface to release the units generated to SVN

The following prerequisite should be adhered to before attempting to release units to SVN

 SVN Repository URL should be provided in the Preferences Node of Function Id against field SVN Repository URL . Path till the module name has to be given. *Example:- For an FT module RADXML path can be given as* Z:\FCUBS_11.3.0\Soft\MAIN\FT

DRACLE' FLEXCUBE Develo			Windows Internet Ex	plorer					X
Drawaar	pment workbench	for Universal Banking				105 1		EMOU	
Browser 🗸						Windows	s Optior	is Sign	Out
unction Generation							. 🗙 1	= 17 (-
Action Load 🔻			ype Parent 🔻			on Category Transactio			
Function Id FTDTRONL		Parent Funct					-		
Save XML Path FTDTRONL_F	BROWSE	Parent)	Kml		Foot	er Template None	•		
earch	Preferences								9
Dreferences		Head Office Fu	unction	Module	FT	×=			
DataSource		Logging Requ				Transfer			
	Auto Authoriza		tion	Branch Program Id					
		Tank Modificat	tions	Process Code					
🗄 🚞 FieldSets	📋 Field Log Requ		uired	ed SVN Repository URL					
in Actions CallForms	Multi Branch A		cess Transaction Block Name		BLK_	CONTRACT_DETAILS	•		
LaunchForms	Excel Export R		equired Transaction Field		BRN		-		
🚞 Summary				Name					
								Cont	rol St
		Function Id		Module *		Mo	dule Descr	iption	
	FTDTRONL		FT		* E	Funds Transfer			
	FTSTRONL		FT		> E				

Fig 5.2.1: Preferences Screen highlighting SVN Repository url

The following files can be released to SVN through Development Workbench

- 1) RADXML
- 2) Main Package(spc and sql)

From the release screen check on the units to be released to SVN and click OK .SVN credentials has to be provided in the screen . SFR number has to be provided if it is not a simple check in. If checked in successfully, status will be displayed as released.



Development Workbench – Screen Development II June 2017

Oracle Corporation World Headquarters 500 Oracle Parkway Redwood Shores, CA 94065 U.S.A.

Worldwide Inquiries: Phone: +1.650.506.7000 Fax: +1.650.506.7200 www.oracle.com/ financial_services/

Copyright © 2011-2017 Oracle Financial Services Software Limited. All rights reserved.

No part of this work may be reproduced, stored in a retrieval system, adopted or transmitted in any form or by any means, electronic, mechanical, photographic, graphic, optic recording or otherwise, translated in any language or computer language, without the prior written permission of Oracle Financial Services Software Limited.

Due care has been taken to make this document *Development Workbench-Screen Development II* and accompanying software package as accurate as possible. However, Oracle Financial Services Software Limited makes no representation or warranties with respect to the contents hereof and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this *Development Workbench-Screen Development II* and the accompanying Software System. Furthermore, Oracle Financial Services Software Limited reserves the right to alter, modify or otherwise change in any manner the content hereof, without obligation of Oracle Financial Services Software Limited to notify any person of such revision or changes.

All company and product names are trademarks of the respective companies with which they are associated.