Oracle® Banking Enterprise Limits and Collateral Management

Web service & Silent Utility for generation of Web service Artifacts

Release 14.8.0.0.0
G32539-01
April 2025

Oracle Banking Enterprise Limits and Collateral Management Web service & Silent Utility for generation of Web service Artifacts, Release 14.8.0.0.0

G32539-01

Contents

1

Preface

1.1 Purpose

This guide is designed to help acquaint you with the Oracle Banking Enterprise Limits and Collateral Management (ELCM) application. This guide provides answers to specific features and procedures that the user need to be aware of the module to function successfully.

User can further obtain information specific to a particular field by placing the cursor on the relevant field and pressing <F1> on the keyboard.

1.2 Audience

This guide is intended for the following User/User Roles:

Table 1-1 Audience

Role	Function
Back office data entry clerk	Input functions for funds
Back office managers/officers	Authorization functions
Product Managers	Product definition and authorization
End of day operators	Processing during end of day / beginning of day

1.3 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

1.4 Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance.

1.5 Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1.6 Basic Actions

Table 1-2 Basic Actions

Action	Description
Approve	Used to approve the initiated report. This button is displayed, once the user click **Authorize**.
Audit	Used to view the maker details, checker details, and report status.
Authorize	Used to authorize the report created. A maker of the screen is not allowed to authorize the report. Only a checker can authorize a report, created by a maker.
Close	Used to close a record. This action is available only when a record is created.
Confirm	Used to confirm the performed action.
Cancel	Used to cancel the performed action.
Compare	Used to view the comparison through the field values of old record and the current record. This button is displayed in the widget, once the user click **Authorize**.
Collapse All	Used to hide the details in the sections. This button is displayed, once the user click **Compare**.
Expand All	Used to expand and view all the details in the sections. This button is displayed, once the user click **Compare**.
New	Used to add a new record. When the user click **New**, the system displays a new record enabling to specify the required data.

Table 1-2 (Cont.) Basic Actions

Action	Description
OK	Used to confirm the details in the screen.
Save	Used to save the details entered or selected in the screen.
View	Used to view the report details in a particular modification stage. This button is displayed in the widget, once the user click **Authorize**.
View Difference only	Used to view a comparison through the field element values of old record and the current record, which has undergone changes. This button is displayed, once the user click **Compare**.
Unlock	Used to update the details of an existing record. System displays an existing record in editable mode.

1.7 Related Documents

For more information refer to the Oracle Banking manuals on:

- Development of Launch Forms and Others Screens
- Enterprise Collaterals User Guide
- Enterprise Limits and Collaterals Common User Guide

1.8 Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1.9 Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

1.10 Acronyms and Abbreviations

The list of the acronyms and abbreviations used in this guide are as follows:

Table 1-3 Acronyms and Abbreviations

Acronyms	Abbreviations
CIF	Customer Information File

Table 1-3 (Cont.) Acronyms and Abbreviations

Acronyms	Abbreviations
CASA	Current Account and Savings Account
DDA	System that holds the CASA account and balances
ELCM	Enterprise Limits and Collateral Management
ECA	External Credit Approval
FCUBS	Oracle FLEXCUBE Universal Banking Solution
GW	Gateway
HTTP	Hyper Text Transfer Protocol
ID	Identification Number
Mark EOTI	Mark End of Transaction Input
Mark TI	Mark Transaction Input
OFSAA	Oracle Financial Services Analytical Applications
ORMD	Oracle Revenue and Billing Management
PK	Primary Key
RDBMS	Relational Data Base Management System
SMS	Security Services
UI	User Interface
VD	Value Date
XML	Extensible Mark-up Language
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformations

1.11 Symbols and Icons

The lists of symbols, buttons and shortcut key that are used in the application to perform various tasks are covered in this topic.

Table 1-4 Symbols and Icons

Icons	Function
	Perform search
	Minimize
	Navigate to the next record
	Navigate to the previous record
	Toggle OFF
	Toggle ON
	Delete
	Click this icon to add a new row.

Table 1-4 (Cont.) Symbols and Icons

Icons	Function
	Click this icon to delete an existing row.
	List view
	Maximize
	Navigate to the first record
	Navigate to the last record
	Advance search
	Search record
	Save the record
	Reset the record
	Clear the record

Table 1-5 Symbols and Icons - Audit Details

Icons	Function
	A user
	Branch details
	Date and Time

1.12 Prerequisite

Specify the **User ID** and **Password**, and login to **Home** screen.

2 Introduction

This document explains the steps to create/modify the Service XML and generate web service artifacts for building EAR files using the Oracle FLEXCUBE Universal Banking Development Workbench.

3
Service XML

This topic provides an overview of the Service XML.

Oracle FLEXCUBE Universal Banking Development Workbench provides the developer with a user-friendly console for defining a gateway service of FCUBS. One Service XML corresponds to one Gateway Service. All the function IDs which are part of the particular service would be captured in the Service XML along with the operation details.

Open Development Tool assist developers in developing the web service with the capability of generating the Following artifacts for building EAR file:

Table 3-1 Webservice

Files	Description
<Service Name>Src*Impl.java	IMPL files for service
<Service Name>WSDL*.wsdl	WSDL files for service
<Service Name>Config*.xml	Config files
<Service Name>XSD*.xsd	Service specific XSD's
<Service Name>Common*.xsd's	Common XSD's (call forms) part of service | • **<Service Name>\<Service Name>\META-INF\application.xml** • **<Service Name>\<Service Name>\META-INF\MANIFEST.MF**	Config XML's for building the Web service
<Service Name>\<Service Name>\commons-codec-1.2.jar	Utility Jar for building the web service
<Service Name>\<Service Name>\wscommon.jar	Utility Jar for building the web service
Sample Ant file	For building service ear file

> **Note:**
>
> Non-extensibility function ID's operations don't allow add/modify any existing service; it will allow delete operation only.

4

Create a New Service in Open Development Tool

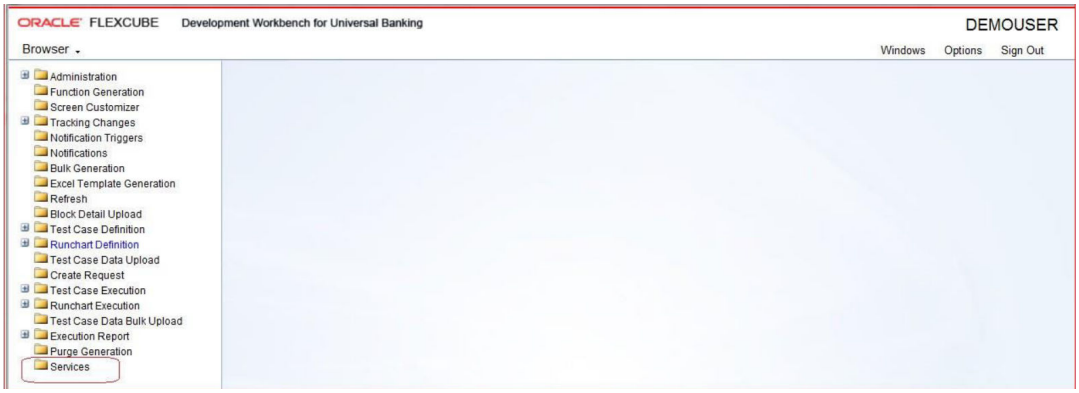
This topic provides systematic instructions to create a new Service in Open Development Tool.

1. On Development Workbench Login page, specify the **Username** and **Password** and log in to the Oracle FLEXCUBE Universal Banking Open Development Tool using the credentials maintained.

   Refer to the Development Workbench – Administration for creating users.

   Development Workbench For Universal Banking screen displays.

Figure 4-1 Development Workbench For Universal Banking



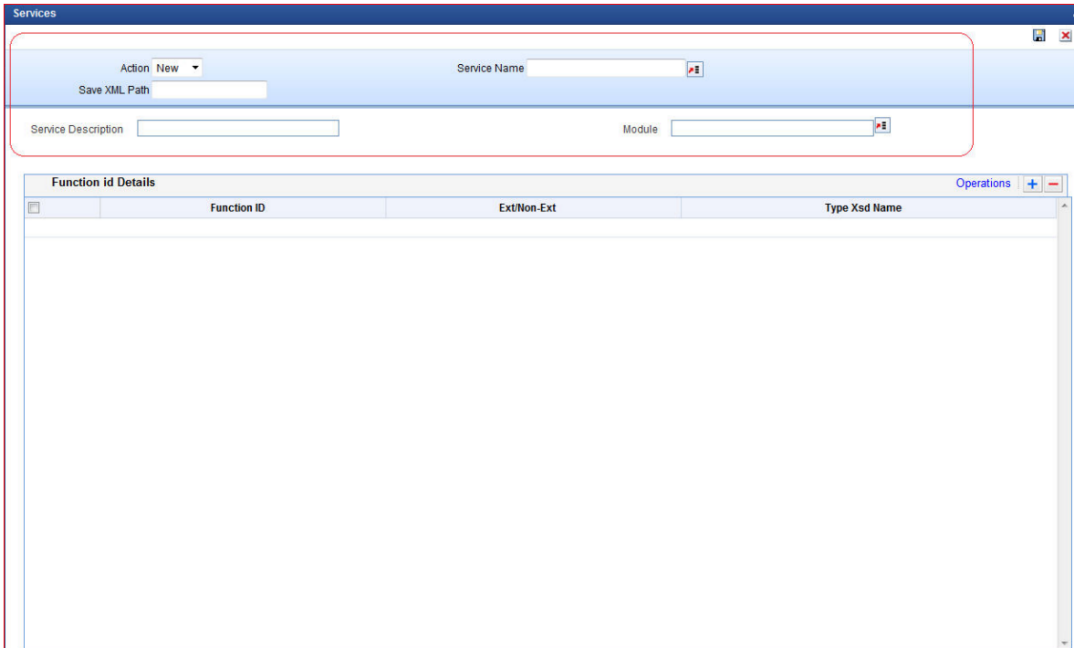
2. Map the session to the release and environment as required.

   Refer Development Workbench – Getting Started for detailed explanation.

3. **Development Workbench For Universal Banking** landing page, click on the **Services** node under the **Browser** menu.

   Services screen displays.

Figure 4-2 Services



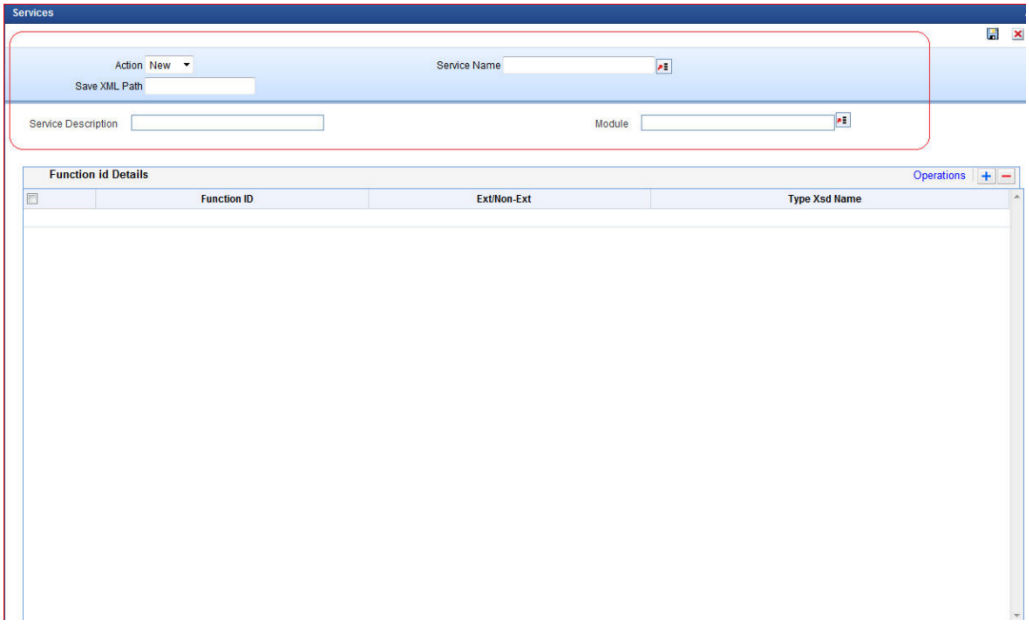
4. Specify the below details in the Header section of the **Services** screen.

   a. Specify the **Action** field.

      New and **Load** options are provided for this field. For a new Service development, select the action as **New**. If the **Action** is **Load** then the corresponding Service XML has to be loaded using the **Browser** option in **Save XML Path**; all the header information will get populated.

      On selection of **Action** field as **New**, **Service Name** window displays.

Figure 4-3 Services



b. Select the **Service Name** from service LOV and click on the **Search** button.

   If the **Action** field is selected as **New**, **Service Name** has to be selected from service LOV (Service LOV will fetch values from **GWTM_SERVICES_MASTER** For new service, service name needs to be added in **GWTM_SERVICES_MASTER** of business schema).

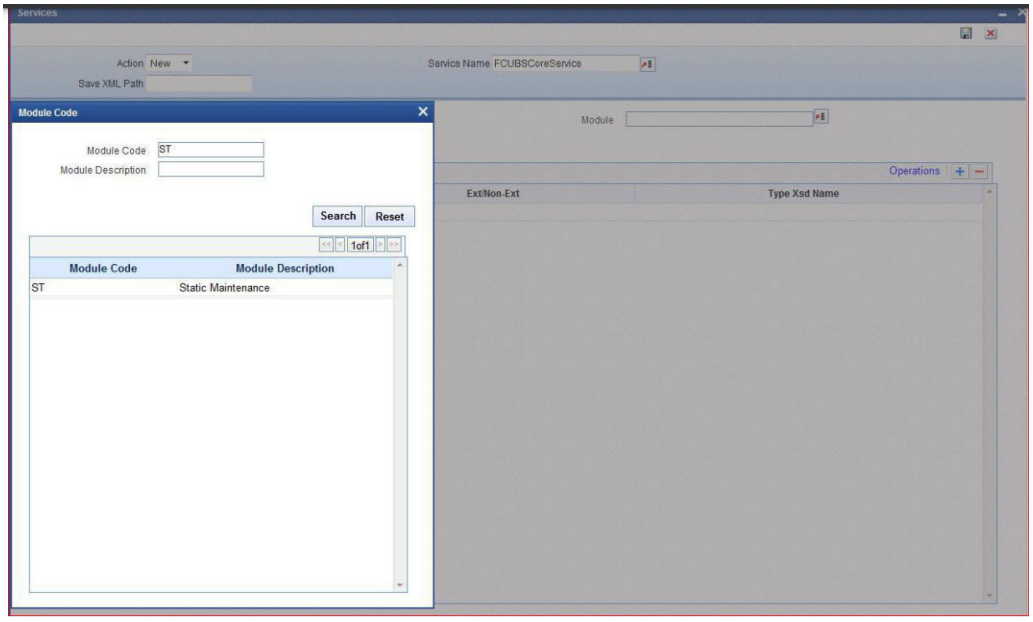
   On selection of the **Service Name**, service description will be populated in the **Service Description** field.

c. At **Module** field, click on the **List of Values** icon.

   If the **Action** is selected as **New**, the **Module** has to be selected from **Module LOV** (Module LOV will fetch values from **SMTB_MODULES** of Business schema). Module Name need not always be from the LOV. Note that Artifacts would be generated based on the Module Name specified. Hence provide source Folder module names (For example, Core for CS) in this field.

   Module Code window displays.

Figure 4-4 Module Code



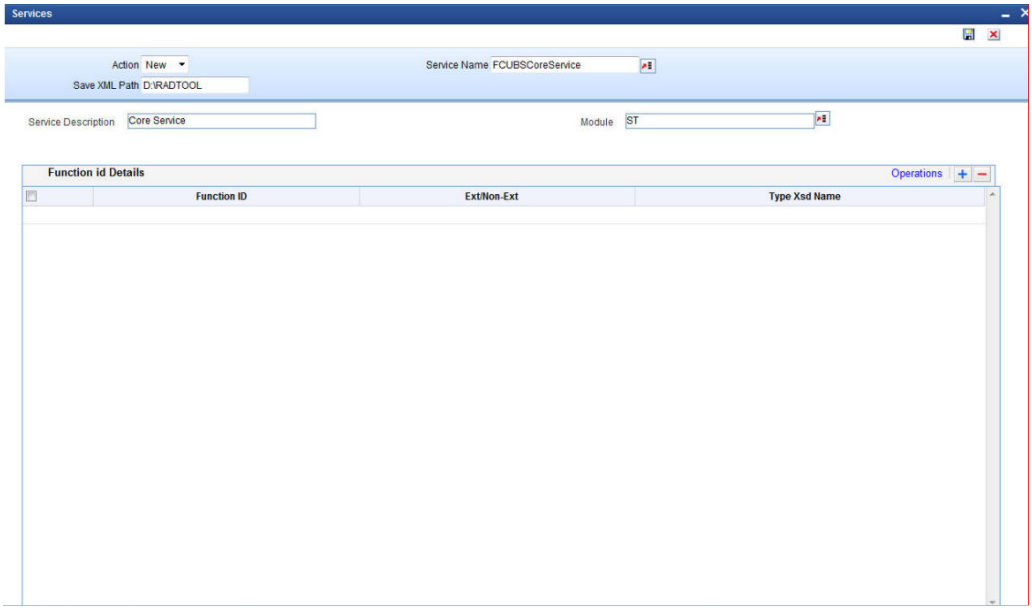
d. Select the **Module Code** and click on the **Search** button.

   Module displays in the **Services** screen.

e. Specify the **Save XML Path** field.

   If the **Action** is **New**, **Save XML Path** is optional. If provided, then the generated units will be saved in the path mentioned. Note that the value in the Save XML Path will be used only if the Save Format is Client Path and if the User has given **CURRENT_DIRECTORY** in the User Preferences Work Directory. The label description of the field will change depending on the action. If the **Action** is **Load**, Open Development Tool attaches a **Browse** button to it so that the user can browse the Service XML and load it.

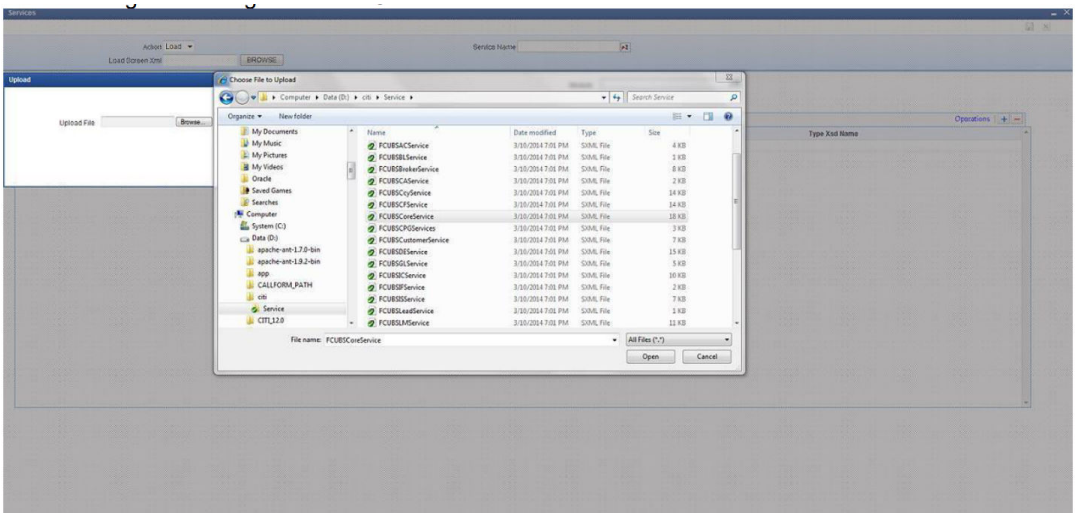
Figure 4-5 Services- Save XML Path



5. To load an existing service in Open Development Tool, click on the **Browse** button at the **Save XML Path** field.

   Upload window displays to choose an existing service.

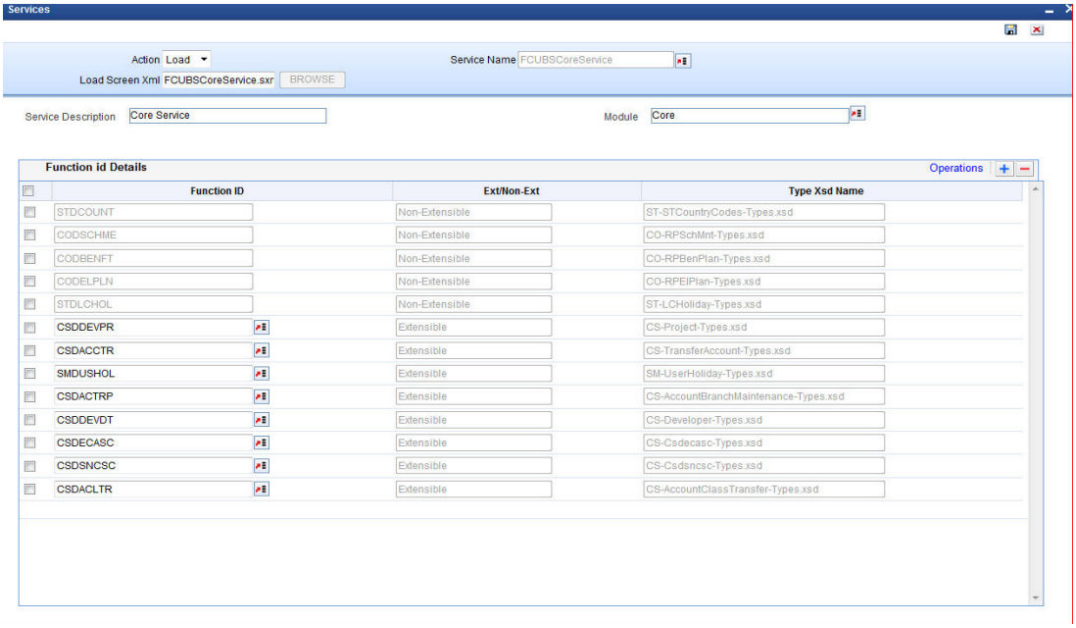
Figure 4-6 Upload- Choose File to Upload



6. Select an existing service SXML file and click on the **Open** button.

   Services screen displays with function ID details of the selected service.

Figure 4-7 Services- Function ID Details



7. For a creation of new service, specify the below function ID details in the **Services** screen.

   Function ID Details: The developer can attach the Function IDs that are part of the service or remove the same from the service if not required.

Table 4-1 Function ID Details

Field	Description
Function ID	Select the function ID from function ID LOV for adding the function ID for that service. Function ID LOV will populate data from **SMTB_MENU**. Make sure that **FC_FUNCTION_ID** values are selected for function ID so that a physical RADXML file for the same function ID exists. For example, Select **STDCIF** and not **STGCIF**
Extensible/Non-Extensible	On select of the **Function ID** field, this field value would be populated.
Type Xsd Name	**Type Xsd Name** would be defaulted along with **Operations** for an Extensible function ID. For Non-Extensible function ID, **Type Xsd Name** has to be explicitly mentioned in the field.

Figure 4-8 Function Name- Module

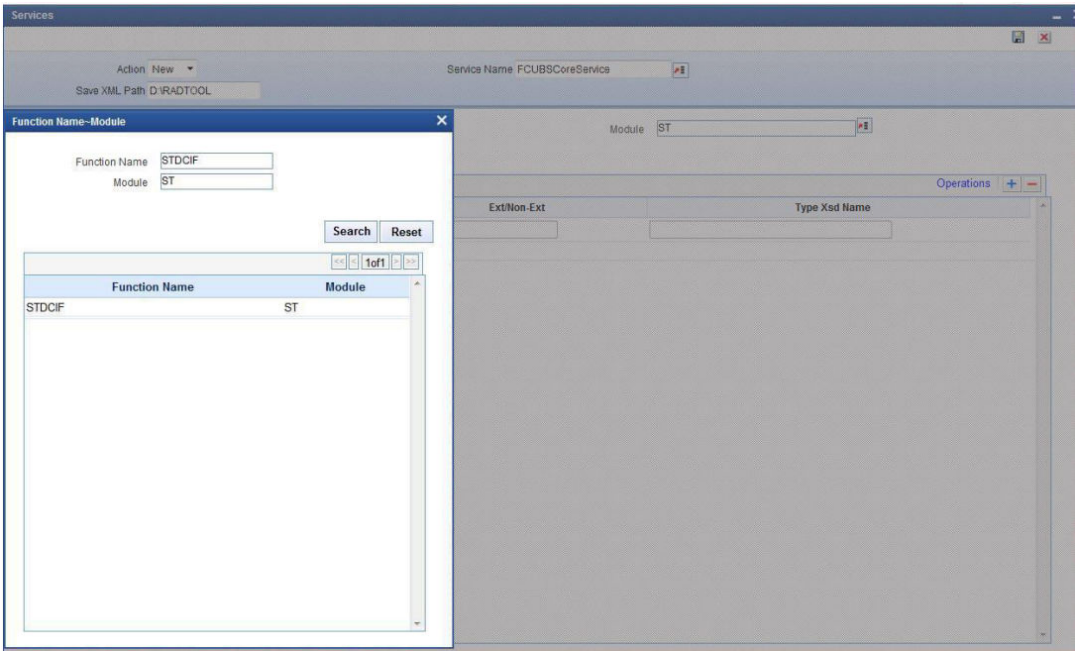
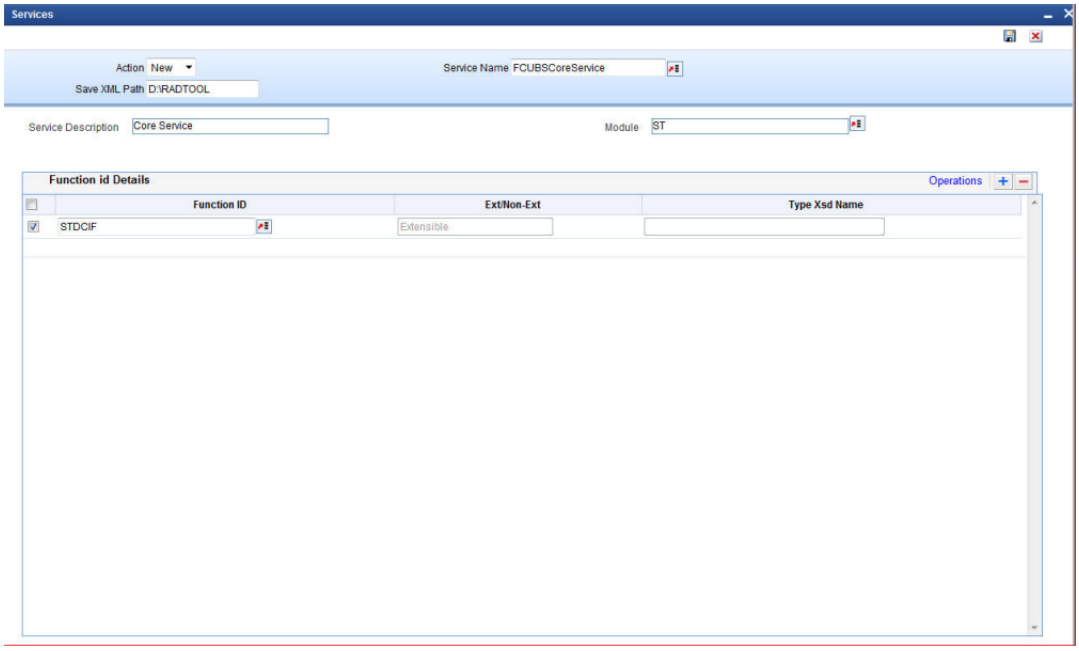


Figure 4-9 Services- Extensible/Non-Extensible Field

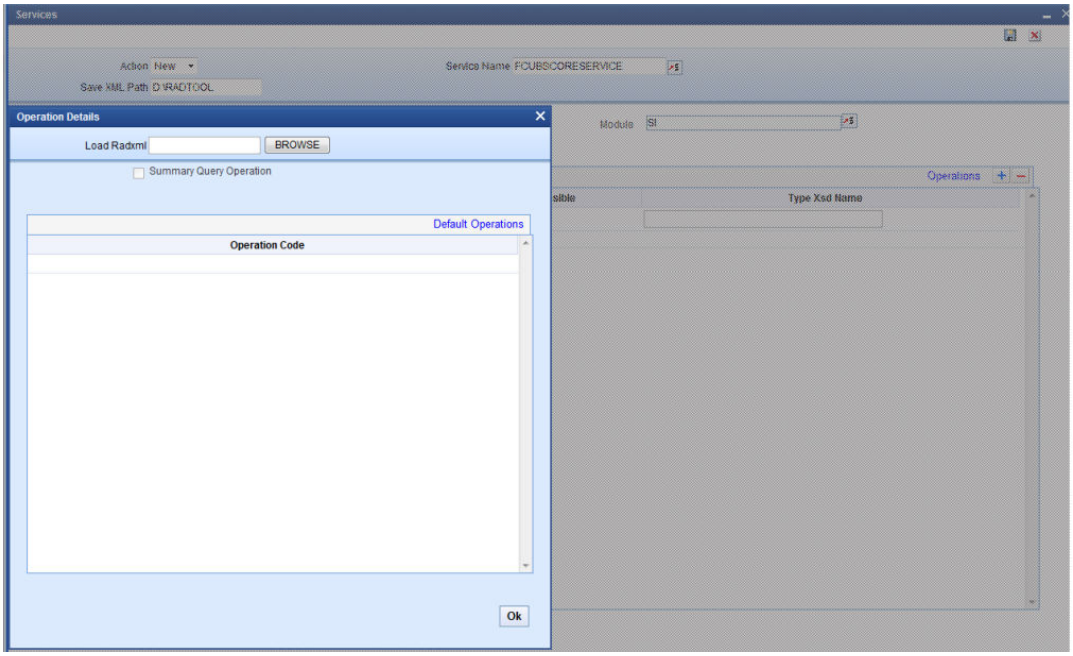


8. Select **Function ID** checkbox and click on **Operations**.

   Operation Codes would be defined in each RADXML which has to have defaulted in the Service XML as well.

   Operation Details window displays.

Figure 4-10 Operation Details



9. Specify the details in the **Operation Details** window.

Table 4-2 Operation Details

Field	Description
Load Radxml	**Operation Details** screen attaches a **Browse** button to it so that the user can browse the function ID Radxml and load it to populate operations.
Default Operations	Click on the **Default Operations**, **Operation Code** and **Type Xsd Name** will have defaulted from loaded RADXML. Click on the **OK** button.

Save ServiceXML: Open Development Tool saves all the activities carried out by the developer in an XML file hereby referred to as **SXML**. The persistence of the WEBSERVICE is achieved through SXML and RADXML. If some changes are required on the web services in a future release, the same SXML can be loaded and changes can be done on this SXML. Open Development Tool can segregate the changes done on different releases and save the SXML accordingly. SXML will adhere to following naming convention:

Service Name + .sxml

For example, **FCUBSCoreService.sxml**

5

Open Development Tool Silent Utility

This topic provides an overview of the operations in the Open Development Tool Silent Utility.

The following operations are supported in the silent utility of Open Development Tool:

1. **LOGIN**
2. **SET RELEASE**: Setting Release and Environment Details
3. **BULKGENERATION**: Bulk Generation of RADXML's units
4. **REFRESH**: Bulk refresh of RADXML's
5. **SXML_REFRESH**: Bulk refresh of Service XML's
6. **SXML_UPDATER**: Bulk Updater of service XML's based on the changes in RADXML's
7. **SXML_BULKGENERATION**: Bulk Generation of web service artifacts

Execution of Operation will be as per the sequence maintained in **OdtOperations.properties**.

Example:

1. Operation = **LOGIN**
2. Operation= **SETRELEASE**
3. Operation= **REFRESH**

If the sequence of operations is as above, then Login Operation, Set Release, and Refresh Operations would be processed in the respective sequence.

> **Note:**
>
> Login and set release are mandatory operations to be performed.

Prerequisites

Refer to the #unique_28 topic for the detailed information.

How to run utility on Windows/Unix

After copying the installer sources and library folder to the local system, make sure to uncheck the read-only check box in source properties and apply the same to all the sub folders.

The below-given figure shows the source folder in the local system:

6

Configuration of SilentODTUtility

This topic provides an overview of the configuration process of Silent Open Development Tool Utility.

All Configuration files can be found inside the resource folder of the utility.

> **Note:**
>
> Copy **jaxb-xjc.jar** to the lib folder. This can be obtained from application server libraries.

For example, In Weblogic 12c
<Oracle_Home>\oracle_common\modules\com.sun.xml.bind.jaxb-jxc.jar

Also make sure **ojdbc6.jar** is available under lib folder.

SilentOdt.properties

The sample property file has been given below. Refer to the details mentioned for each property in the below table. Some of them are encrypted using **ODTPassEncryption.bat** (**ODTPassEncryption.sh** for UNIX). Use 16 characters length of the symmetric key for encryption (Preferably Alphanumeric) which will be prompted for input from the user when the encryption utility is launched. The same symmetric key must be mentioned in the property file as well.

Figure 6-1 File attachment- SilentOdt.properties

```
#################################################################
##ODT Version  ==> 12.2/12.1/12.0.2/12.0.1/11.4 etc
##type         ==> FCUBS/FCIS/ELCM  ; specifies the product
##release      ==> Release of the product specified
#################################################################
odtVersion=12.2
#FCUBS,FCIS,ELCM,PAYMENTS
type=FCUBS
release=FCUBS_12.2.0.0.0
productDesc=Oracle FLEXCUBE Universal Banking
ReleaseMonth=May
ReleaseYear=2016

###ODT DataSource Credentials
#####################################################
OdtJdbcUrl=O40fjmATqNKRRdNFP9UR3eeXUUMaPnZJ1gtHVtXkIyVEgM1qCkuNqcIs96vR4NFq
OdtDbUser=ODT121
OdtDbPassword=ITgBkLEJpGO6AuYE6jJkmg==
SymmetricKey=oraclefinancials

############ Logger Properties#############################
#logreqd Default set to N
#Default Path set to User Home directory, if not provided
#LEVEL ==>DEBUG/INFO/WARNING/SEVERE   ; default value is INFO
#########################################################
logreqd = Y
logpath = D:/DESTTEMPDIR/ODT/log.txt
level = DEBUG


##########System Properties#####################################################
##JAVA_HOME is mandatory
##WEBLOGIC and WebSphere Home would be required only if ANT scripts are being generated.
##Use Backward Slash(\) for File Separator
#################################################################################
JAVA_HOME=C:\Program Files\Java\jdk1.8.0_73
WEBLOGIC_HOME=D:\Oracle\Middleware
WAS_HOME=D:\WAS
```

Table 6-1 Open Development Tool Data Source Detail Credentials

Open Development Tool Data Source Detail Credentials	Description
OdtJdbcUrl	Encrypted JDBC URL, Sample JDBC URL (before encrypting) **jdbc:oracle:thin:@10.184.xx.xx:1521:FCDEMO**
OdtDbUser	DB User Name
OdtDbPassword	Encrypted Password.
SymmetricKey	The key is used when encrypting the JDBC URL and password. This should have exactly 16 characters. **Note:** User must use the same key for both JDBC URL and password encryption.

Table 6-2 Logger Properties

Logger Properties	Description
Logreqd	**Y/N**. Default set to **N**.
Logpath	Provide the path where the Logger files will be generated.
Level	Provide the Logger Level. This can be either **SEVERE/WARNING/INFO/CONFIG/FINE/FINER/ FINEST**. Provide as **FINEST** for writing detailed log. The default value would be set to **INFO**.

Table 6-3 System Properties

System Properties	Description
JAVA_HOME	Maintain the Java(JDK) installed location.
WEBLOGIC_HOME	Maintain the Oracle WebLogic installed location.
WAS_HOME	Maintain the IBM Websphere installed location.

ODTOperations.properties

Configure the Operations file as per requirement. The following operations are supported in the Silent Utility of Open Development Tool:

1. **LOGIN**
2. **SETRELEASE**: Setting Release and Environment Details
3. **BULKGENERATION**: Bulk Generation of Radxml units
4. **REFRESH**: Bulk refresh of RADXML
5. **SXML_REFRESH**: Bulk refresh of Service XMLs
6. **SXML_UPDATER**: Bulk Updater of service Xmls based on the changes in RADXML's
7. **SXML_BULKGENERATION**: Bulk Generation of web service artifacts

Execution of Operation will be as per the sequence maintained in **OdtOperations.properties**.

For example,

1. Operation = **LOGIN**
2. Operation= **SETRELEASE**
3. Operation= **REFRESH**

If the sequence of operations is as above, then Login, Set Release, and Refresh Operations would be processed in the respective sequence.

> **Note:**
>
> **LOGIN** and **SETRELEASE** are mandatory operations to be performed.

1. **LOGIN**
   LOGIN should always be the first operation to be configured as part of any execution.

Figure 6-2 LOGIN Operation

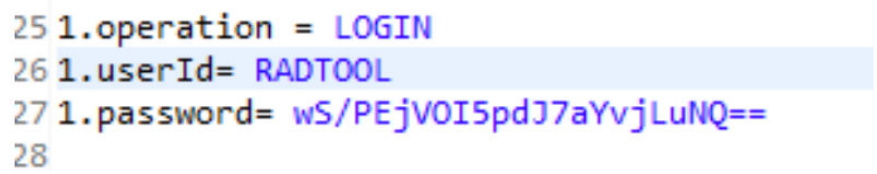


Table 6-4 Login to Tool

Field	Description
Operation	LOGIN
User Id	Open Development Tool User Id which is created in the Open Development Tool Application.
Password	Open Development Tool Password which is created in the Open Development Tool Application. Encrypted using **ODTPassEncryption.bat**.

2. **SETRELEASE**
This operation can be used for setting Release and Environment Preferences for SilentODTUtility. Connection to the Oracle FLEXCUBE Universal Banking schema would be established based on data maintained in Open Development Tool or through the data in **env_config.xml** as explained in an earlier section.

Figure 6-3 SETRELEASE Operation

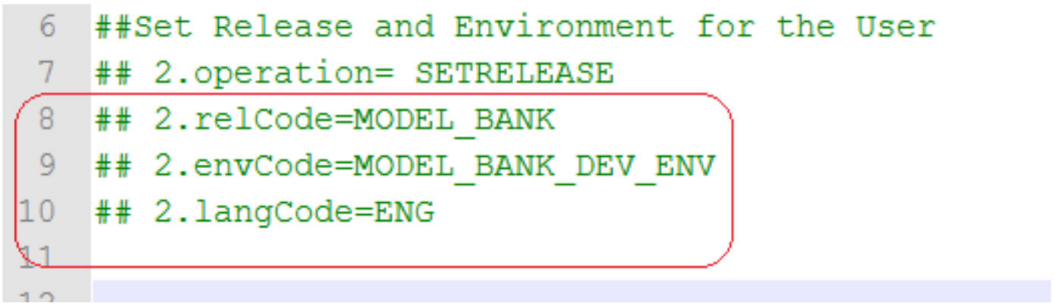


Table 6-5 Release and Environment Details for User

Field	Description
Operation	SETRELEASE
relCode	Open Development Tool Release Code which is created in the Open Development Tool Application.
envCode	Open Development Tool Environment Code which is created in the Open Development Tool Application.
langCode	Lang code for the above-mentioned release code.

3. **BULKGENERATION**
For generating all RADXML artifacts for release in bulk this feature can be used.

Table 6-6 BULKGENERATION

Field	Description
radxmlListFile	Prepare text file which contains absolute path of all RADXML's. Provide the same file path.
srcPath	Source path refers to the path where all RADXML's are presented. List File would be generated by the tool in this case. Note that only either **radxmlListFile** or **srcPath** should be present. If both are present, then the **radxmlListFile** parameter would be considered for Bulk Generation.
fileType	• **EXTENSIBLE**: artifacts generated only for extensible screens. • **NON_EXTENSIBLE**: artifacts generated only for non-extensible screens. • **BOTH**: artifacts for all files would be generated.
destpath	Provide the path where the files will be generated.
gen	Provide the type of files to be generated. For example, UIXML, SYS_JS, MAIN_SPC, MAIN_SQL,KERNEL_SPC, KERNEL_SQL

Figure 6-4 SilentODT Bulk Generation Properties

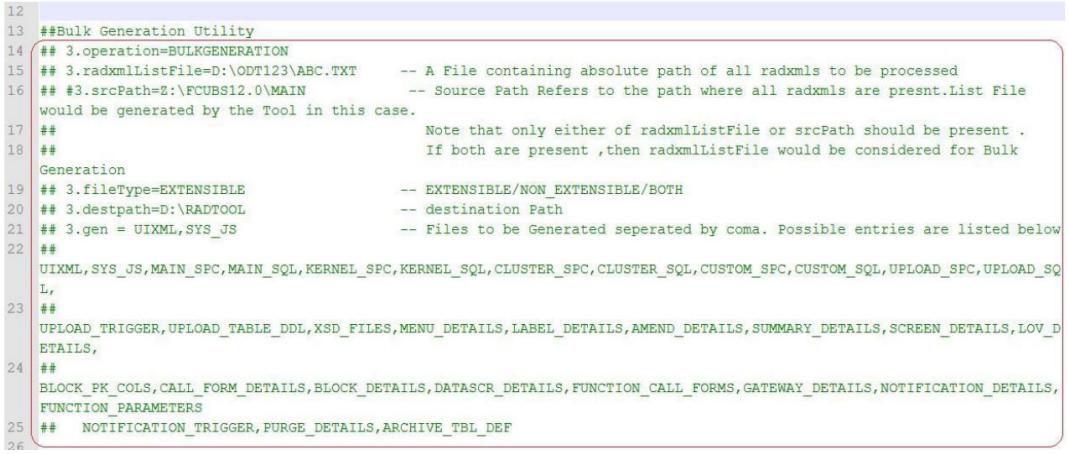


Table 6-7 Bulk Generation Utility

Input	Output
radxmlListFile	UIXML,SYS_JS,MAIN_SPC,MAIN_SQL,KERNEL_SPC,KERNEL_SQL, CLUSTER_SPC,CLUSTER_SQL,CUSTOM_SPC,CUSTOM_SQL, UPLOAD_SPC,UPLOAD_SQL,UPLOAD_TRIGG
srcPath	
fileType	
gen	

Table 6-7 (Cont.) Bulk Generation Utility

Input	Output
destpath	ER,UPLOAD_TABLE_DDL, XSD_FILES,MENU_DETAILS,LABEL_DETAILS, AMEND_DETAILS, SUMMARY_DETAILS,SCREEN_DETAILS,LOV_ DETAILS, BLOCK_PK_COLS,CALL_FORM_DETAILS,BLO CK_DETAILS, DATASCR_DETAILS,FUNCTION_CALL_FORM S,GATEWAY_DETAILS, NOTIFICATION_DETAILS,FUNCTION_PARAME TERS NOTIFICATION_TRIGGER,PURGE_DETAILS,A RCHIVE_TBL_DEF in destpath.

4. **REFRESH**
Refresh Functionality allows developers to upgrade the existing RADXML to its later version keeping the sub-version-specific changes intact. Three kinds of refresh can be done using the Tool. (Refer to the Development Workbench - Source Upgrade)

   a. **Child Refresh**

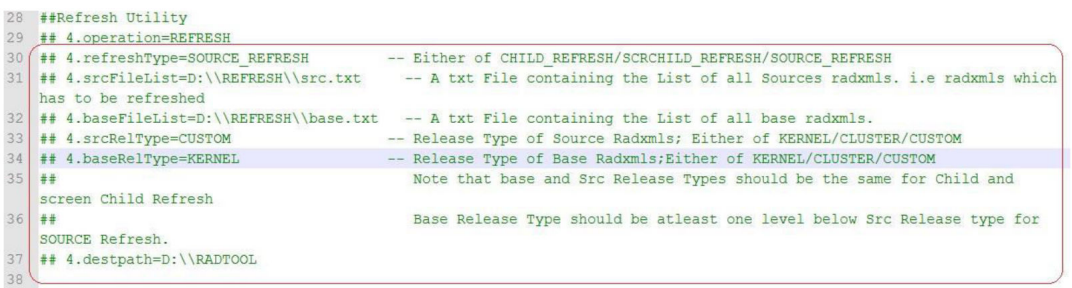
   b. **Screen Child Refresh**

   c. **Source Refresh**

Table 6-8 REFRESH

Field	Description
Refresh Type	Provide the refresh type (**CHILD_REFRESH/ SCRCHILD_REFRESH/SOURCE_REFRESH**).
srcFileList	A text file containing the list of all source RADXMLs. i.e. RADXMLs have to be refreshed.
baseFileList	A text file containing the list of all base RADXMLs.
srcRelType	Provide the release type of source RADXMLs list(**KERNEL/CLUSTER/CUSTOM**).
baseRelType	Provide the release type of base RADXMLs list (**KERNEL/CLUSTER/CUSTOM**).
destpath	Provide the path where the files will be generated.

Figure 6-5 SilentODT Refresh Properties



```
##Refresh Utility
## 4.operation=REFRESH
## 4.refreshType=SOURCE_REFRESH          -- Either of CHILD_REFRESH/SCRCHILD_REFRESH/SOURCE_REFRESH
## 4.srcFileList=D:\\REFRESH\\src.txt     -- A txt File containing the List of all Sources radxmls. i.e radxmls which has to be refreshed
## 4.baseFileList=D:\\REFRESH\\base.txt   -- A txt File containing the List of all base radxmls.
## 4.srcRelType=CUSTOM                   -- Release Type of Source Radxmls; Either of KERNEL/CLUSTER/CUSTOM
## 4.baseRelType=KERNEL                  -- Release Type of Base Radxmls;Either of KERNEL/CLUSTER/CUSTOM
##                                         Note that base and Src Release Types should be the same for Child and screen Child Refresh
##                                         Base Release Type should be atleast one level below Src Release type for SOURCE Refresh.
## 4.destpath=D:\\RADTOOL
```

Table 6-9 Refresh Utility

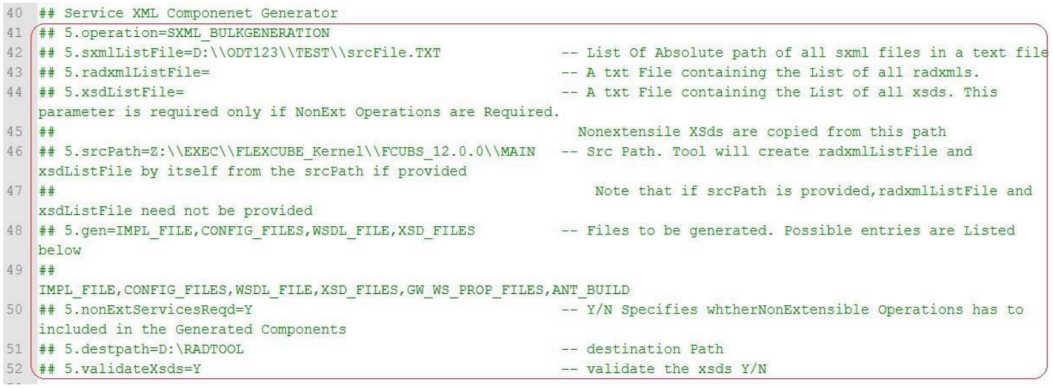
Input	Output
refreshType	Refreshed Radxml's in destpath.
srcFileList	Refreshed Radxml's in destpath.
baseFileList	Refreshed Radxml's in destpath.
srcRelType	Refreshed Radxml's in destpath.
baseRelType	Refreshed Radxml's in destpath.
destpath	Refreshed Radxml's in destpath.

5. **Service XML Bulk Generation**
Web service artifacts can be generated through this operation.

Table 6-10 Service XML Bulk Generation

Field	Description
sxmlListFile	Prepare text file which contains absolute path of all Service XML.
radxmlListFile	Prepare a text file that contains the absolute path of all RADXML which are used for those services.
xsdListFile	Prepare a text file that contains the absolute path of all XSDs which are used for those services. Non-extensibility/Common XSDs are copied from this path.
srcPath	Provide source folder path which is an option (tool will create **radxmlListFile** and **xsdListFile** by itself from the **srcPath**). Note that if **srcPath** is provided, **radxmlListFile** and **xsdListFile** need not be provided.
gen	Provide type of files to be generated (separated by comma). Options are IMPL_FILE, CONFIG_FILES,WSDL_FILE,XSD_FILES,GW_WS_PROP_FILES,ANT_BUILD.
nonExtServicesReqd	**Y/N**, Specifies whether NonExtensible Operations has to include in the generated components.
destpath	Provide the path where the files will be generated.
validateXsds	**Y/N**, Default set to **Y**. If the value is set to **Y** all XSD will be validated by the tool.

Figure 6-6 SilentODT Service XML Bulk Generation Properties



```
## Service XML Componenet Generator
## 5.operation=SXML_BULKGENERATION
## 5.sxmlListFile=D:\\ODT123\\TEST\\srcFile.TXT            -- List Of Absolute path of all sxml files in a text file
## 5.radxmlListFile=                                       -- A txt File containing the List of all radxmls.
## 5.xsdListFile=                                          -- A txt File containing the List of all xsds. This
parameter is required only if NonExt Operations are Required.
##                                                          Nonextensile XSds are copied from this path
## 5.srcPath=Z:\\EXEC\\FLEXCUBE_Kernel\\FCUBS_12.0.0\\MAIN  -- Src Path. Tool will create radxmlListFile and
xsdListFile by itself from the srcPath if provided
##                                                             Note that if srcPath is provided,radxmlListFile and
xsdListFile need not be provided
## 5.gen=IMPL_FILE,CONFIG_FILES,WSDL_FILE,XSD_FILES         -- Files to be generated. Possible entries are Listed
below
##
IMPL_FILE,CONFIG_FILES,WSDL_FILE,XSD_FILES,GW_WS_PROP_FILES,ANT_BUILD
## 5.nonExtServicesReqd=Y                                  -- Y/N Specifies whtherNonExtensible Operations has to
included in the Generated Components
## 5.destpath=D:\RADTOOL                                   -- destination Path
## 5.validateXsds=Y                                        -- validate the xsds Y/N
```

Table 6-11 Service XML Component Generator

Input	Output
sxmlListFile	IMPL_FILE,CONFIG_FILES,WSDL_FILE,XSD_FILES,GW_WS_PROP_FILES,ANT_BUILD in destpath
radxmlListFile	
xsdListFile	
srcPath	
nonExtServicesReqd	
destpath	
validateXsds	

6. **Service XML Updater**
This feature can be used to update the Service XMLs with the latest data from RADXMLs. The following details will be updated:

   a. Any addition, deletion, or modification of operation codes in function Id would be updated in Service XML.

   b. If any function Id is removed from the service (specified in RADXML); then the same would be removed from Service XML. Note that if any new function Id is attached to the service (in RADXML); then the same will not be updated in the Service XML. This has to add manually in the Service XML through the Open Development Tool user interface.

Table 6-12 Service XML Updater

Field	Description
Operation	SXML_UPDATE
sxmlListFile	Prepare a text file that contains the absolute path of all Service XML. Provide the same file path.
radxmlListFile	Prepare a text file that contains the absolute path of all RADXMLs which are used for those services. Provide the same file path.
srcPath	Provide source folder path. This field is optional (tool will create **radxmlListFile** and **sxmlListFile** by itself from the **srcPath**). Note that if **srcPath** is provided, **radxmlListFile** and **sxmlListFile** need not be provided.

Table 6-12 (Cont.) Service XML Updater

Field	Description
destpath	Provide the path where the files will be generated.
confirmStage	SINGLE_STAGE_UPDATE (Default Value should not be modified by the developer)

Figure 6-7 SilentODT Service XML Updater Properties

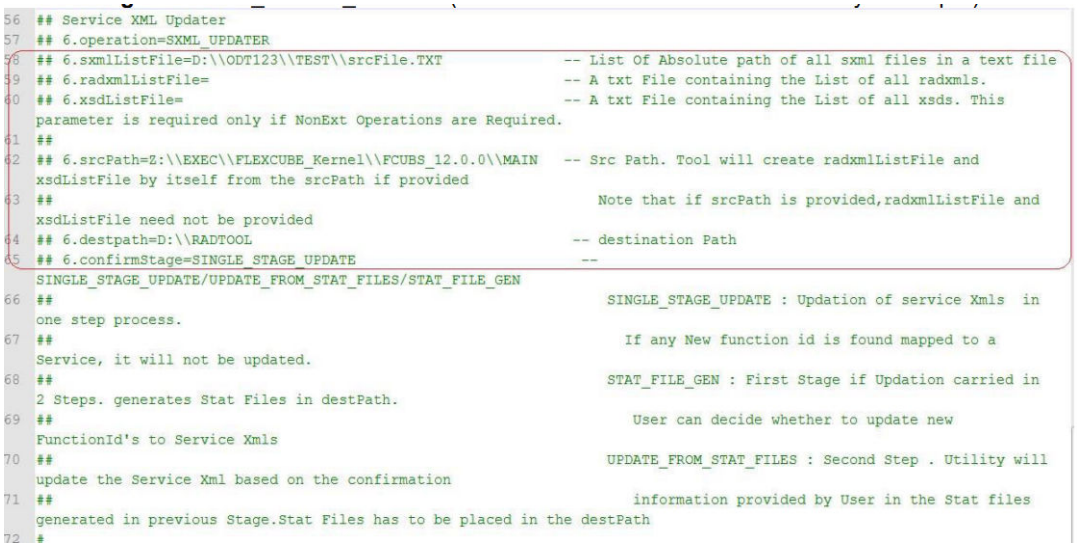


Table 6-13 Service XML Updater

Input	Output
xsdListFile	Updated Service XML's in destpath.
sxmlListFile	Updated Service XML's in destpath.
radxmlListFile	Updated Service XML's in destpath.
srcPath	Updated Service XML's in destpath.
destpath	Updated Service XML's in destpath.
confirmStage	Updated Service XML's in destpath.

7. **Service XML Refresh**
Refresh Functionality allows us to upgrade the existing service XML to its later version keeping the sub-version-specific changes intact.

Table 6-14 Service XML Refresh

Field	Description
srcFileList	Prepare a text file that contains the absolute path of all Service XML, the same file should be provided i.e. service XML which has to be refreshed.

Table 6-14 (Cont.) Service XML Refresh

Field	Description
baseFileList	Prepare a text file that contains the absolute path of all base service XMLs. Provide the same file path. For instance, for a custom development team; all the latest Kernel files have to be mentioned in **baseFileList** while the custom Service XMLs to be refreshed has to be mentioned in the **srcFileList**.
srcRelType	Provide the release type of Source RADXMLs list (**KERNEL/CLUSTER/CUSTOM**).
baseRelType	Provide the release type of base RADXMLs list (**KERNEL/CLUSTER/CUSTOM**). **baseRelType** should be at least one level below **srcRelType** for SOURCE Refresh. For Instance, for a custom development team, **srcRelType** would be CUSTOM and **baseRelType** can be either **KERNEL/CLUSTER** depending on the base source type.
destpath	Provide the path where the files will be generated.

Figure 6-8 SilentODT Service XML Refresh Properties

```
## Service XML Refresh
## 7.operation=SXML_REFRESH
## 7.srcFileList=D:\\REFRESH\\src.txt     -- A txt File containing the List of all Sources sxmls. i.e sxmls which has
to be refreshed
## 7.baseFileList=D:\\REFRESH\\base.txt   -- A txt File containing the List of all base sxmls.
## 7.srcRelType=CUSTOM                  -- Release Type of Source sxmls; Either of KERNEL/CLUSTER/CUSTOM
## 7.baseRelType=KERNEL                 -- Release Type of Base sxmls;Either of
KERNEL/CLUSTER/CUSTOM##
##                                        Base Release Type should be atleast one level below Src Release type for
SOURCE Refresh.
## 7.destpath=D:\\RADTOOL
```

Table 6-15 Service XML Refresh

Input	Output
operation	Refreshed Service XML's in destpath.
srcFileList	Refreshed Service XML's in destpath.
baseFileList	Refreshed Service XML's in destpath.
srcRelType	Refreshed Service XML's in destpath.
baseRelType	Refreshed Service XML's in destpath.
destpath	Refreshed Service XML's in destpath.

GW_CONFIG.properties

These properties file parameters are used for the generation gateway web service properties files. This configuration file is optional. Provide only if Gateway Web service property files (**GW_WS_PROP_FILES**) is being generated as part of Service XML Bulk generation operation.

Table 6-16 GW_CONFIG.properties

Field	Description
EJB_APP_NAME	Provide the Name of the deployed EJB Application.
EJB_APP_SERVER	Provide Application server name in which ear deployed.
EJB_JNDI_NAME	**EJB JNDI Name** is the reference name of the EJB by which the EJB has been deployed.
EJB_SERVER_URL	Application server IP Address & port where the EJB application is deployed.
EJB_SERVER_USERNAME	User Name of the application server where the EJB application is deployed.
EJB_SERVER_PASSWORD	Password of the application server where the EJB application is deployed.
GW_WS_LOGGER_PROP_FILE_PATH	Provide the Location of Logger Property File path in the server where web service is to be deployed. Provide path including the file name. For example, • Windows path: `D:/Kernel11.1/GW_WS/config/gw_ws_logger.properties` • Linux or UNIX: `/oraint1/kernel//Gateway/GWWS/config/gw_ws_logger.properties`
GW_WS_LOGGER_FILE_PATH	Provide the location where debug files will be written. For example, • Windows: `D:/Kernel11.1/GW_WS/log/` • Linux or Unix: `/oraint1/kernel/FC120INS_DEBUG/Gateway/GWWS/log`
GW_WS_PROP_FILE_PATH	Path where property file is placed in the server. This will be referred to in **web.xml** of the web service property file. For example, • Windows: `D:/Kernel11.1/GW_WS/prop/` • Linux or UNIX: `/oraint1/kernel/FC120INS_DEBUG/Gateway/GWWS/prop`
XSD_PATH	XSD Path

Figure 6-9 SilentODT GW Config Properties

```
######GW_EJB Properties######Used for Creation GW Property File ###############
################################################################################
#EJB_APP_NAME - Name of the deployed EJB Application
EJB_APP_NAME=GWEJB
# EJB_APP_SERVER(WEBLOGIC,WEBSPHERE)
EJB_APP_SERVER=WEBLOGIC
#Reference name of the EJB by which it has been deployed. This should be '<EJB_APPLICATION_NAME>/ejb/GW_EJB_Bean'
EJB_JNDI_NAME=GWEJB/ejb/GW_EJB_Bean

EJB_SERVER_URL=http://localhost:7010
EJB_SERVER_USERNAME=weblogic
EJB_SERVER_PASSWORD=weblogic1

##Location of Logger Property File path in the server where Webservice is to be deployed.
##Provide Path Including File Name
GW_WS_LOGGER_PROP_FILE_PATH=/home/ora11gas/Gateway11.0/GW_WS/config/gw_ws_logger.properties
##Location where Debug Files will be written
GW_WS_LOGGER_FILE_PATH=/home/ora11gas/Gateway11.0/GW_WS/log
#Location of Gateway Property File. If not provided ; assumed to be same as of Logger property File Path
GW_WS_PROP_FILE_PATH=/home/ora11gas/Gateway11.0/GW_WS/prop

XSD_PATH=/home/ora11gas/Gateway11.0/GW_WS/XSD
```

7

Prerequisites

Before the run utility operation in Open Development Tool, ensure the installation of the following technologies:

- **JDK**
  License Information: JDK is distributed by Sun Microsystems, Inc under Java Development Kit Binary Code License agreement.

  Instructions: The installer requires **JDK 1.7.xx_xx** version to be downloaded in the system and the same Should be set as an environmental variable.
- **Apache Ant 1.7.1**
  Instructions: The installer requires **ANT 1.7.1** version to be downloaded in the system and the same should be set as an environmental variable.

8

Generation of Web service Artifacts through SilentOdtUtility

This topic provides an overview of the generation process of Web service Artifacts through SilentOdtUtility.

To generate web service artifacts for a service, configure the property files of the utility as explained in the previous section. Following operations needs to be configured in **OdtOperations.properties** in the respective sequence:

1. **LOGIN**
2. **SETRELEASE**
3. **SXML_BULKGENERATION**

Provide following values for **gen** parameter of SXML_BULKGENERATION IMPL_FILE, CONFIG_FILES, WSDL_FILE, XSD_FILES, GW_WS_PROP_FILES, ANT_BUILD. To generate Web service Artifacts through SilentOdtUtility, the user has to double click the batch file **silentOdt.bat/sh** present in the Open Development Tool source and command prompt displays.

Figure 8-1 SilentODT Sources Folder

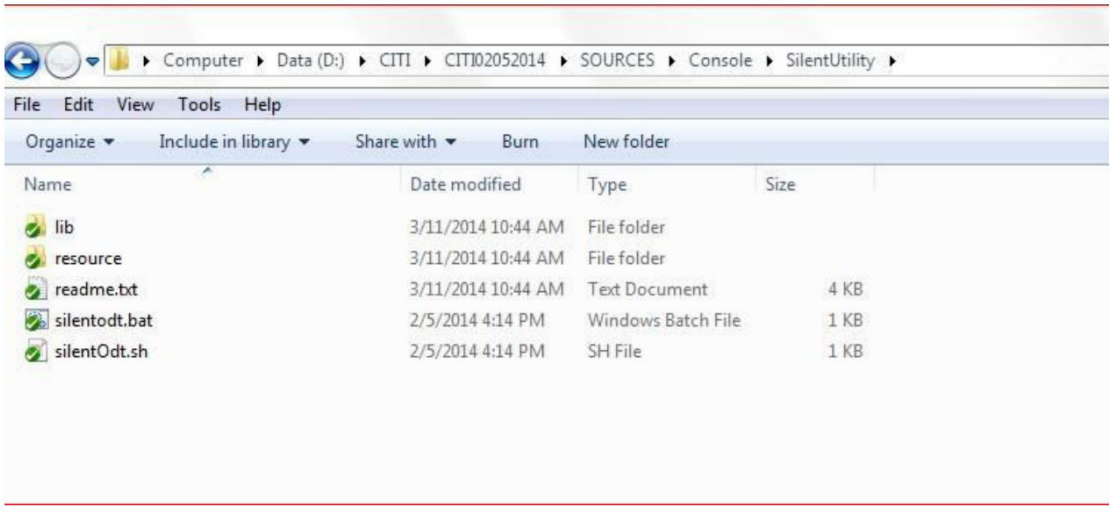
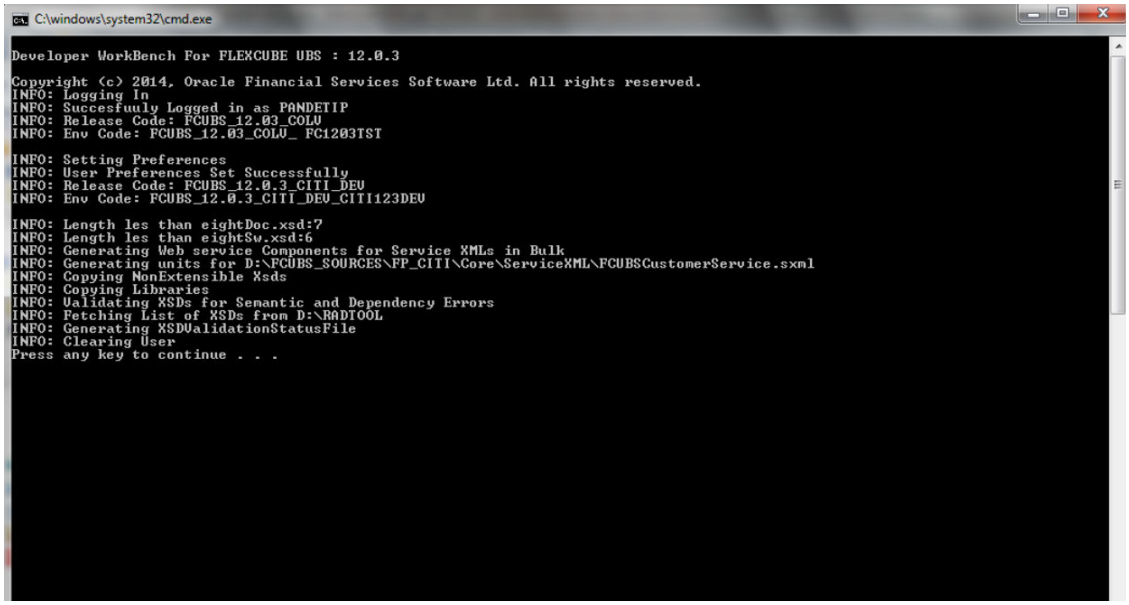


Figure 8-2 SilentODT Running in Command Prompt



After the successful built operation, the following Service artifacts files will be generated in the destination directory specified.

Table 8-1 Service Artifacts files

Files	Description
*<Service Name>Src*Impl.java*	IMPL files for service
<Service Name>WSDL.wsdl*	WSDL files for service
<Service Name>Config.xml*	Config files
<Service Name>XSD.xsd*	Service specific xsd's
<Service Name>Common.xsd's*	Common XSD's (call forms) part of service | • *<Service Name>\<Service Name>\META-INF\application.xml* • *<Service Name>\<Service Name>\META-INF\MANIFEST.MF*	Config XML's for building the Web service
<Service Name>\<Service Name>\commons-codec-1.2.jar	Utility Jar for building the web service
<Service Name>\<Service Name>\wscommon.jar	Utility Jar for building the web service
Sample Ant file	For building service ear file (Can be modified by Dev team as per Folder structure)

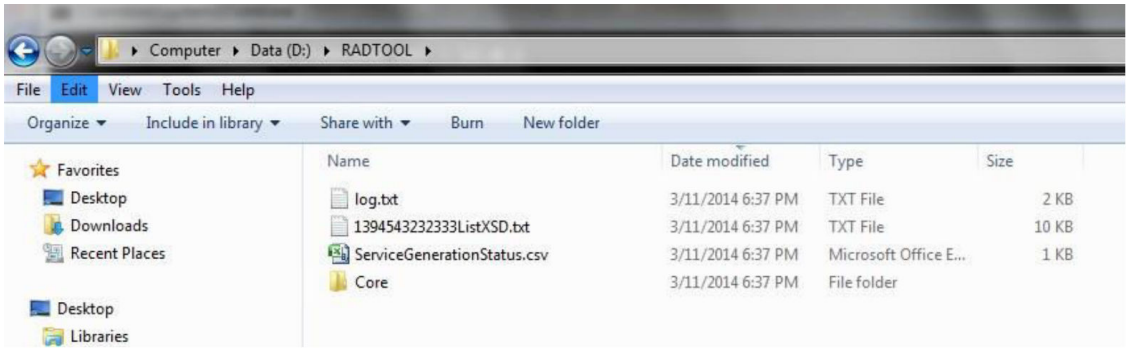
Figure 8-3 SilentODT Generated Files Tree Structure

Log Files

- **ServiceGenerationStatus.csv**
  Generation status will be saved in the above-mentioned file. This will be generated in the destination path.
- **XSDValidationErrors.csv**
  XSD Validation errors, if any, will be saved in the above-mentioned file. This will be generated in the destination path.
- **Utility Log File**
  Log File of the utility would be generated in the path configured in **SilentOdt.properties**. This can be used in case of any troubleshooting.

Proceed only if the status is **Success** for all services in **ServiceGenerationStatus.csv** and **XSDValidationErrors.csv** is not generated.

Figure 8-4 SilentODT Generated log Files Folder

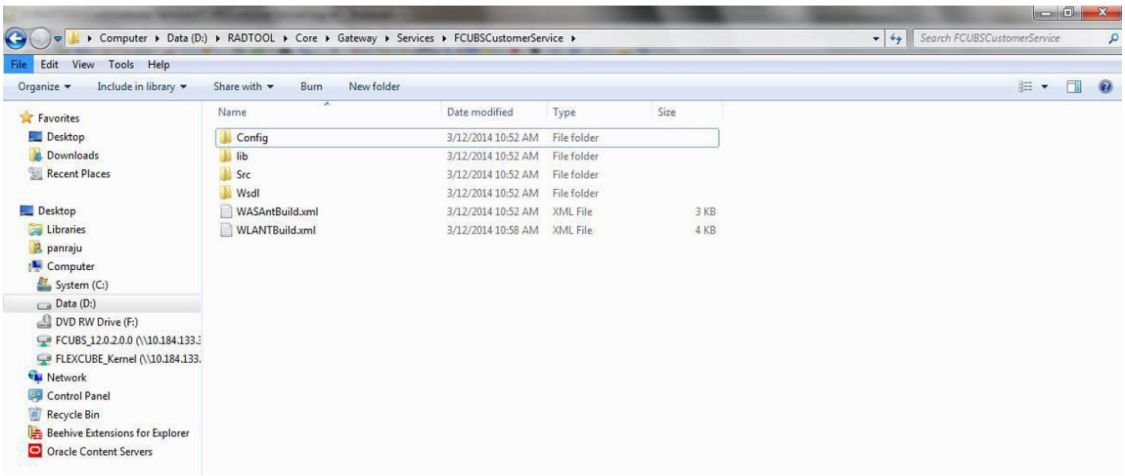


Ant Build Scripts

The tool will generate the sample ant scripts for WebLogic and web sphere application server. Developer can write ant script based on sample ant script or same ant file can be used for building ear file.

- **WLANTBuild.xml** – Ant script for web logic server
- **WASAntBuild.xml** – Ant Script for web sphere server

Figure 8-5 SilentODT Generated Files Ant Build Files Folder



Gateway Property Files

Gateway property files would be generated in the GW_WS folder inside the destination folder. Generated files:

- **gw_ws_logger.properties**
- **GW_WS_Prop.properties**