

**Oracle FLEXCUBE Universal Banking ® 12.0.3**  
**Development Workbench – Webservice & Silent Utility for**  
**generation of Webservice Artifacts**  
**Release 1.0**  
**March 2014**



---

## Contents

1	Preface.....	1
1.1	Audience.....	2
2	Introduction.....	2
3	Service XML.....	2
3.1	Process Steps.....	3
4	ODT Silent Utility.....	12
4.1	Prerequisites.....	12
4.2	How to run utility on Windows/Unix.....	13
4.3	Configuration of SilentODTUtility.....	13
4.3.1	<i>SilentOdt.properties</i> .....	13
4.3.2	<i>env_config.xml</i> .....	15
4.3.3	<i>OdtOperations.properties</i> .....	16
4.3.4	<i>GW_CONFIG.properties</i> .....	22
4.4	Generation of Web service Artifacts through SilentOdtUtility .....	24
4.4.1	<i>Log Files</i> .....	28
4.4.2	<i>Ant Build Scripts</i> .....	28
4.4.3	<i>Gateway Property Files</i> .....	29

---

## 1 Preface

This document describes the webservice development using Oracle FLEXCUBE Development Workbench for Universal Banking.

## 1.1 Audience

This document is intended for FLEXCUBE Application developers/users that use Development Workbench to develop various FLEXCUBE components.

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

<i>Proficiency</i>	<i>Resources</i>
FLEXCUBE Functional Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Technical Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Object Naming conventions	<i>Development Overview Guide</i>
Working knowledge of Web based applications	Self Acquired
Working knowledge of Oracle Database	Oracle Documentations
Working knowledge of PLSQL developer	Respective vendor documents
Working knowledge of PLSQL & SQL Language	Self Acquired
Working knowledge of XML files	Self Acquired

## 2 Introduction

This Document explains the steps to create/Modification of Service xml and generating webservice artifacts for building ear file using the Oracle FLEXCUBE Development Workbench for Universal Banking

## 3 Service XML

Oracle FLEXCUBE 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.

ODT assist developers in developing the webservice with the capability of generating the Following artifacts for building ear file.

<b>Files</b>	<b>Description</b>
<Service Name>Src\*Impl.java	IMPL files for service

Files	Description
<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	Config XML's for building the Web service
<Service Name>\<Service Name>\META-INF\MANIFEST.MF	
<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 allows delete operation only.

### 3.1 Process Steps

Login to the Oracle FLEXCUBE ODT using the credentials maintained (refer 02-ODT Administration.docx for creating users)

Map the session to the release and environment as required (Refer 03-ODT Getting Started.docx for detailed explanation)

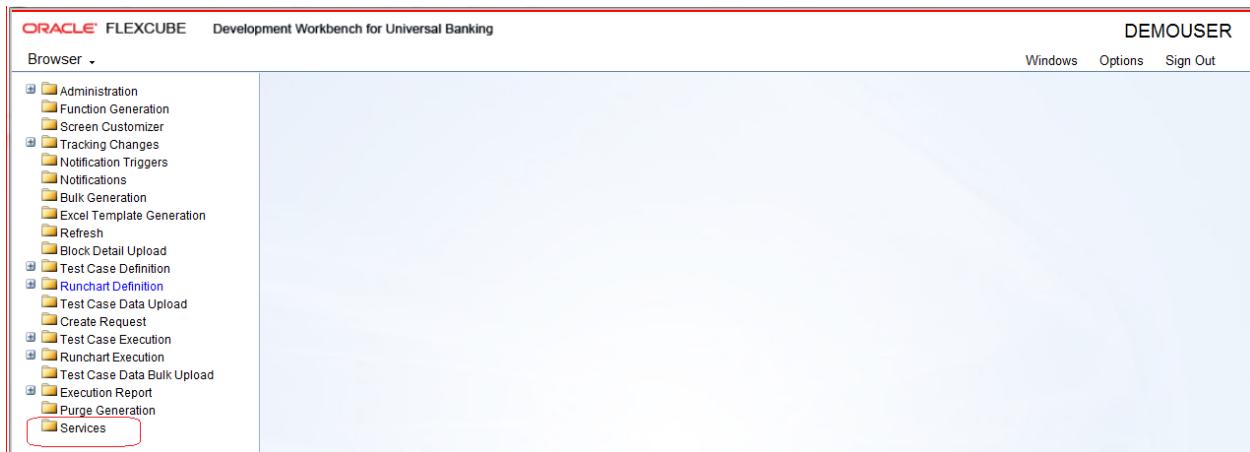


Fig 3.1: ODT Screen Showing Services Option

Click on Service node in the browser tree found in the Landing page of ODT. Services window gets launched, while creating a new Service in ODT, below information needs to be provided in the Header section.

**FLEXCUBE UBS**

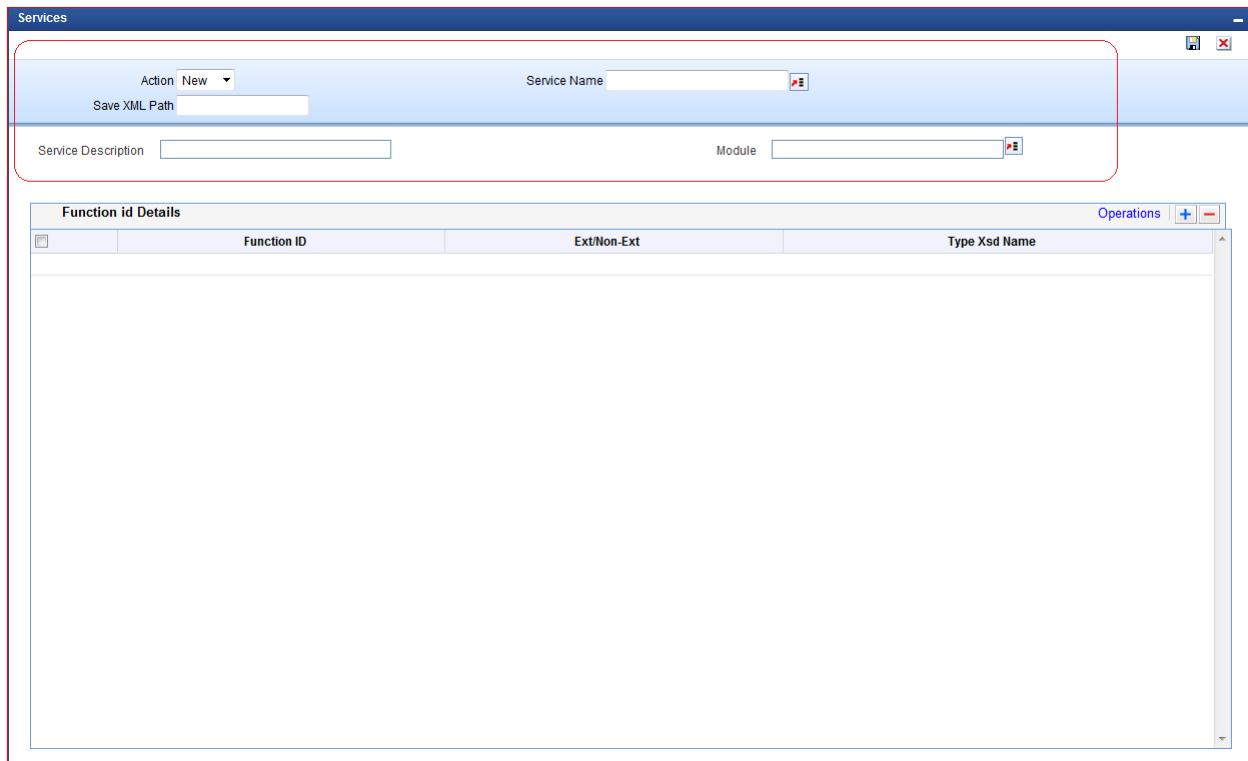


Fig 3.2: Service Screen

While loading an Existing Service in ODT

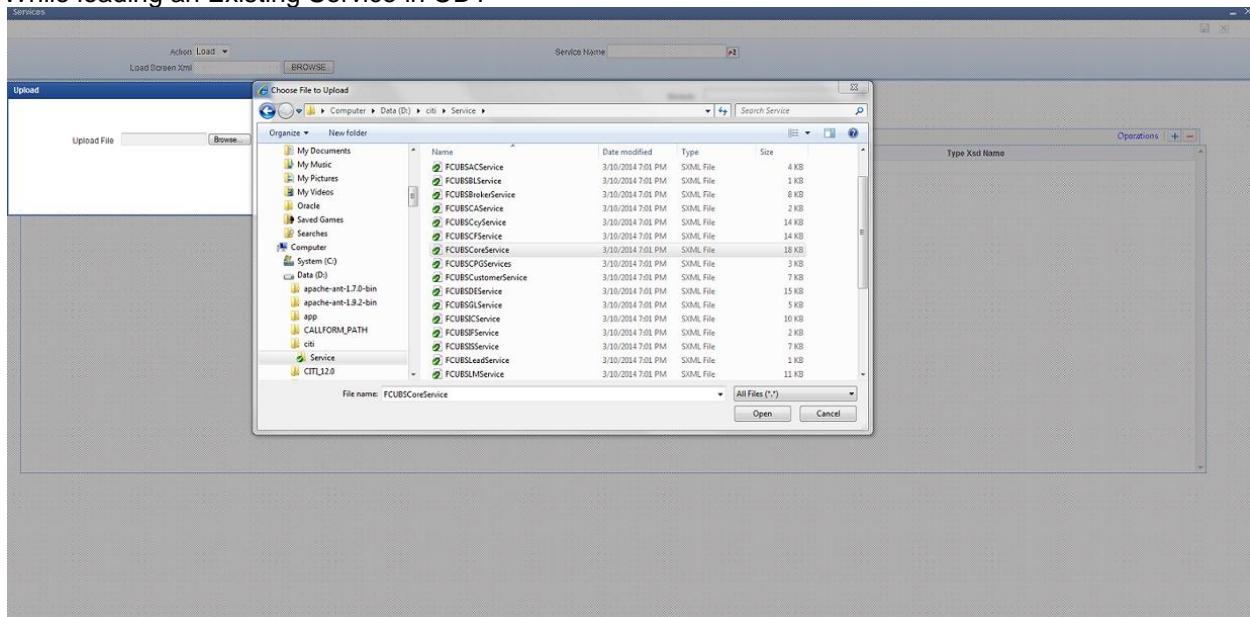


Fig 3.3: Loading Service XML

## FLEXCUBE UBS

The screenshot shows the Oracle Financial Services Service Generation interface. At the top, there are fields for 'Action' (set to 'Load'), 'Service Name' (set to 'FCUBSCoreService'), and 'Module' (set to 'Core'). Below these are fields for 'Service Description' (set to 'Core Service') and 'Operations' (with a '+' and '-' button). The main area is a table titled 'Function id Details' with the following data:

	Function ID	Ext/Non-Ext	Type Xsd Name
	STDCOUNT	Non-Extensible	ST-STCountryCodes-Types.xsd
	CODSCHME	Non-Extensible	CO-RPSchMnt-Types.xsd
	CODBENFT	Non-Extensible	CO-RPBenPlan-Types.xsd
	CODELPLN	Non-Extensible	CO-RPElPlan-Types.xsd
	STDLCHOL	Non-Extensible	ST-LCHoliday-Types.xsd
	CSDDEVPR	Extensible	CS-Project-Types.xsd
	CSDACCTR	Extensible	CS-TransferAccount-Types.xsd
	SMDUSHOL	Extensible	SM-UserHoliday-Types.xsd
	CSDACTRP	Extensible	CS-AccountBranchMaintenance-Types.xsd
	CSDDEVDT	Extensible	CS-Developer-Types.xsd
	CSDECASC	Extensible	CS-Csdecasc-Types.xsd
	CSDSNCS	Extensible	CS-Csdsncsc-Types.xsd
	CSDACLTR	Extensible	CS-AccountClassTransfer-Types.xsd

Fig 3.4: Screen after Loading Service XML.

The Header portion of the Function Generation screens consists of the following fields:

### Action

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 corresponding Service xml has to be loaded using browser option in Save Xml Path; all the header information will get populated.

### Service Name

If the action 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 )

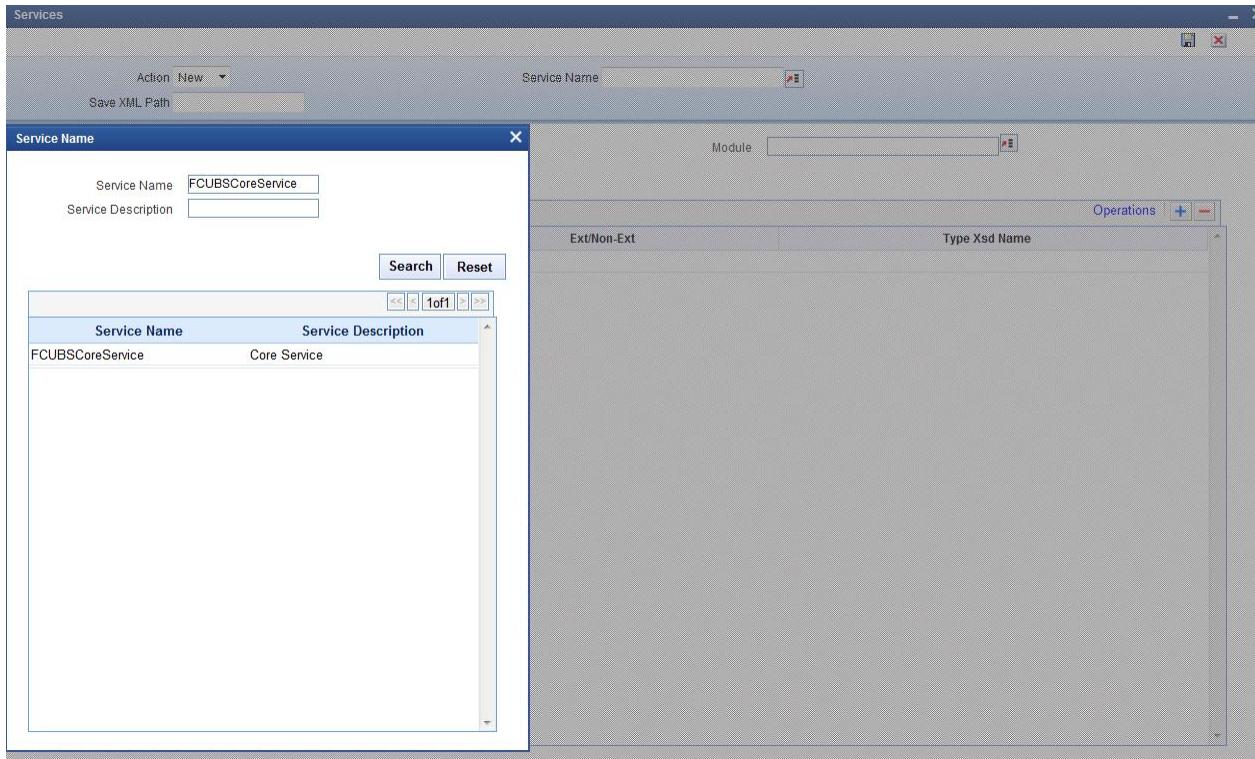


Fig 3.5: Lov to populate Service name.

### Service Description

On Select of service name service description will be populated in service description field

### Module

If the action is selected as New, 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 (Example: Core for CS) in this field**

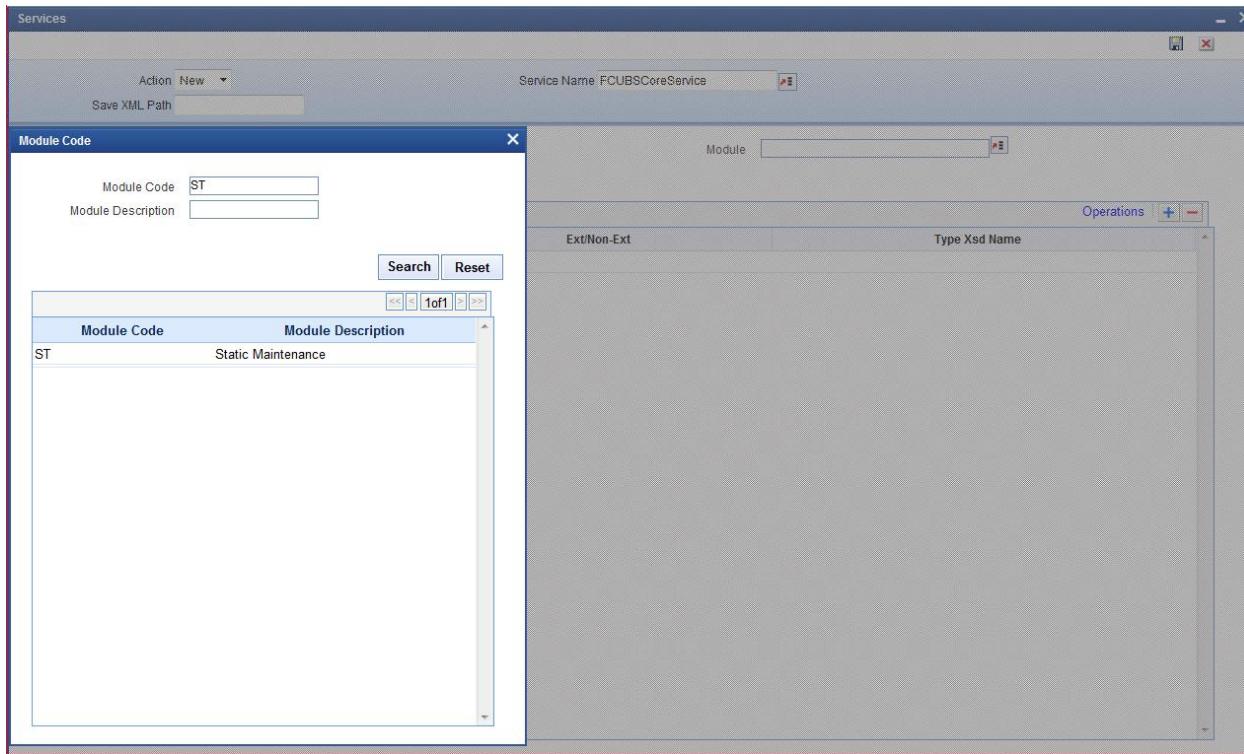


Fig 3.6: Lov to populate Module Code.

### Save Xml Path

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, ODT attaches a Browse button to it so that user can browse the Service xml and load it.

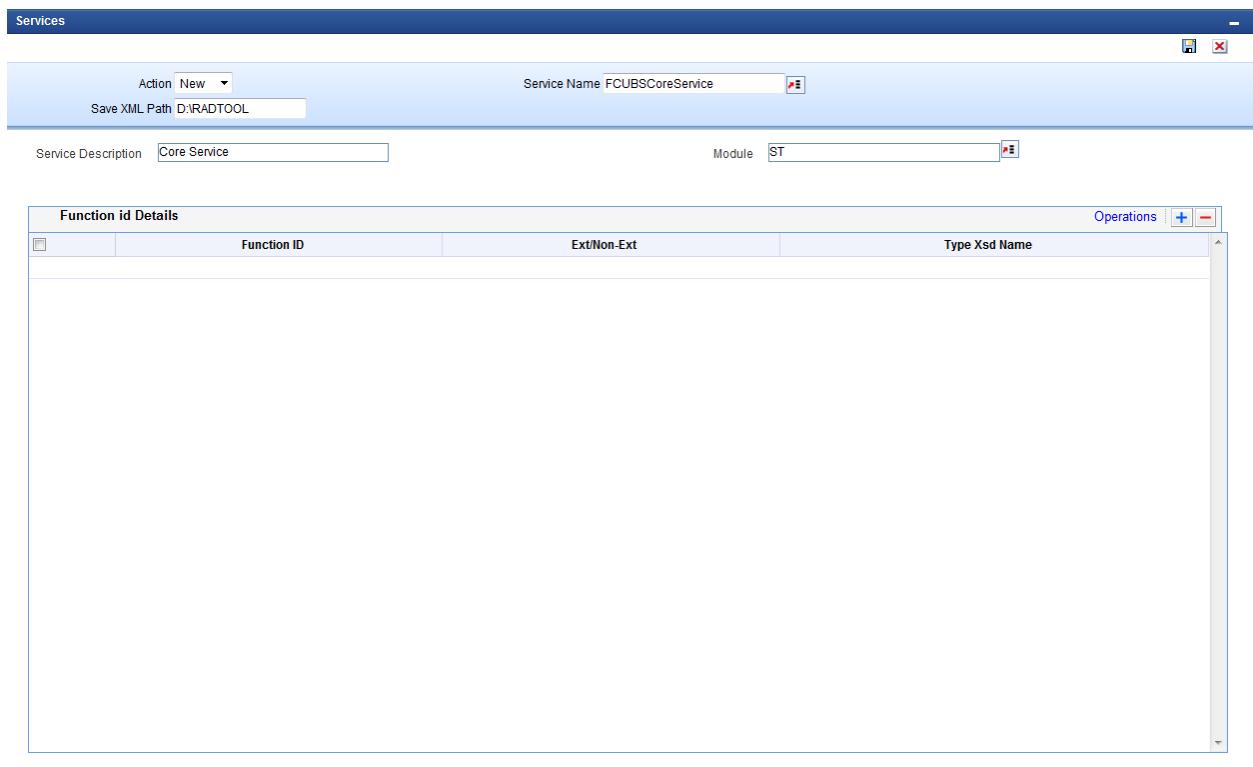


Fig 3.7: Screen to Show save Xml Path.

### Function Id Details:

Developer can attach the Function IDs which are part of this service.  
He can also remove the same from service if not required.

### 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 physical radxml file for the same function Id exists.*

*Example: Select STDCIF and not STGCIF*

## FLEXCUBE UBS

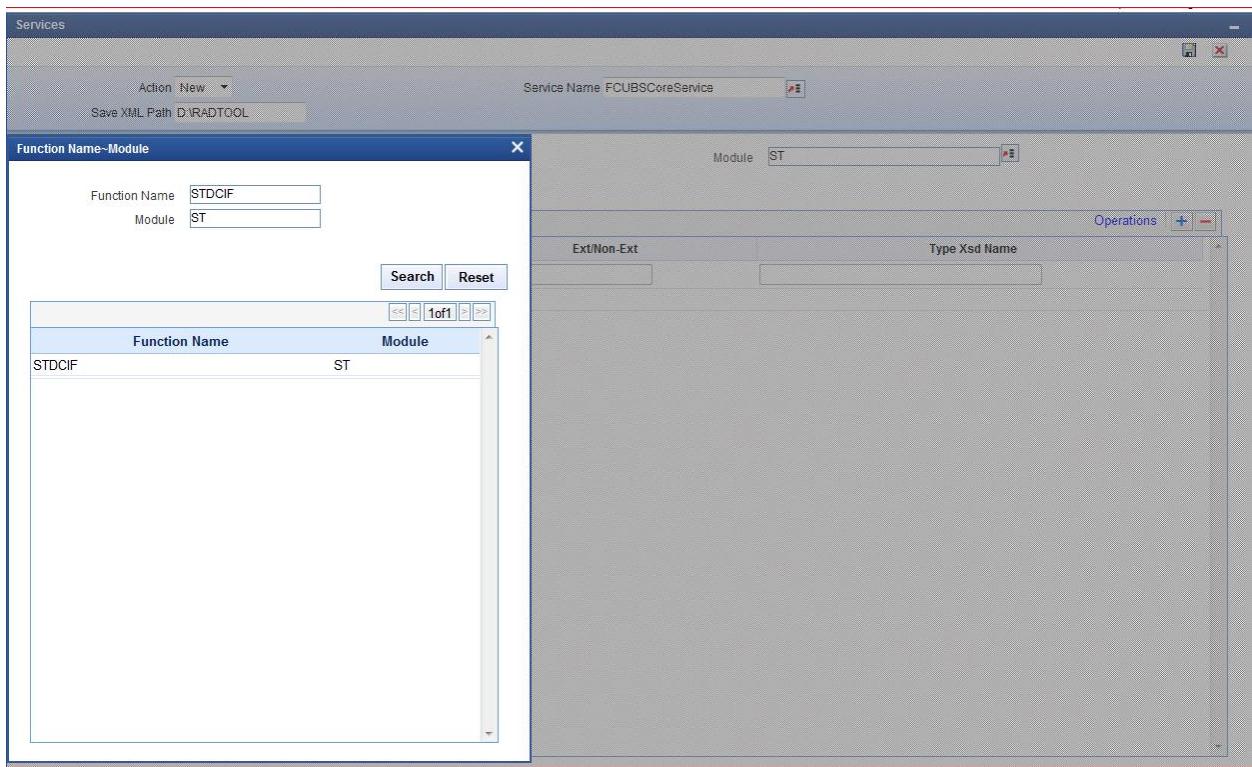


Fig 3.8: Lov to populate Function Id.

### Extensible/Non-Extensible:

On Select of Function id, this field value would be populated

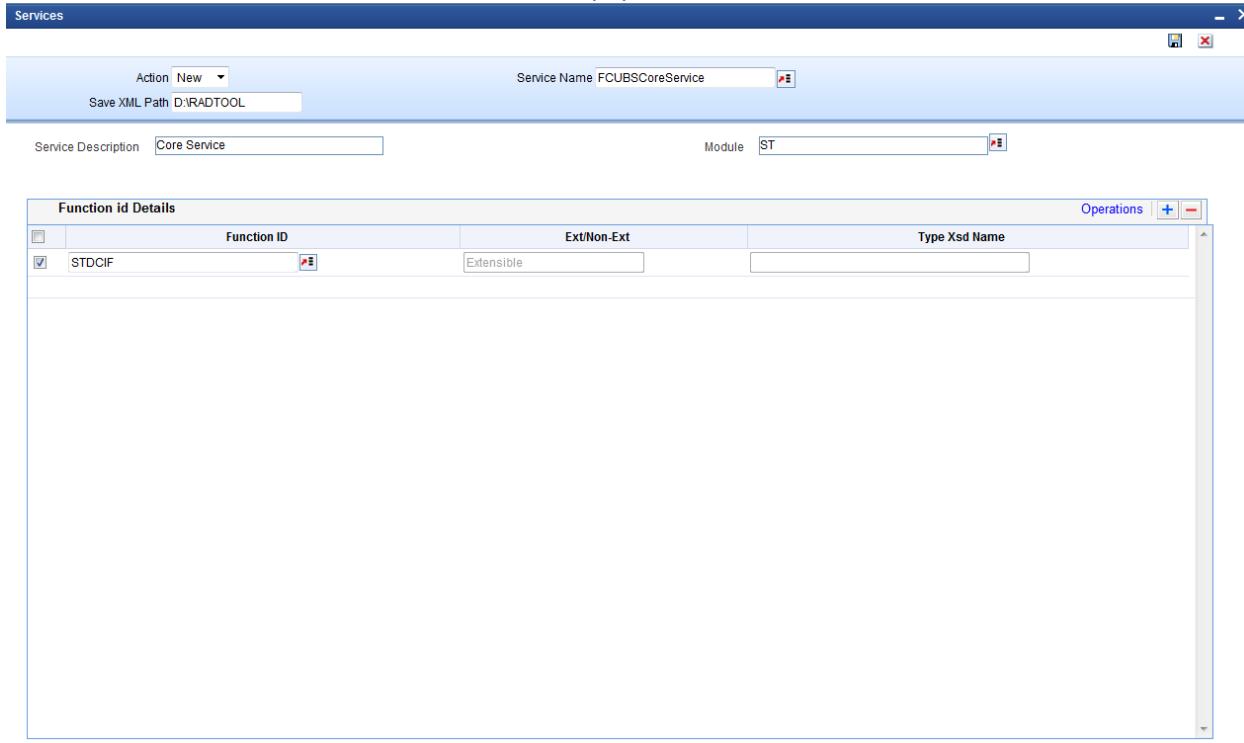


Fig 3.9: Screen to populate Function Id Value.

**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

**Operation:**

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

Select function id checkbox and click on Operations.

Operation details popup screen will be displayed.

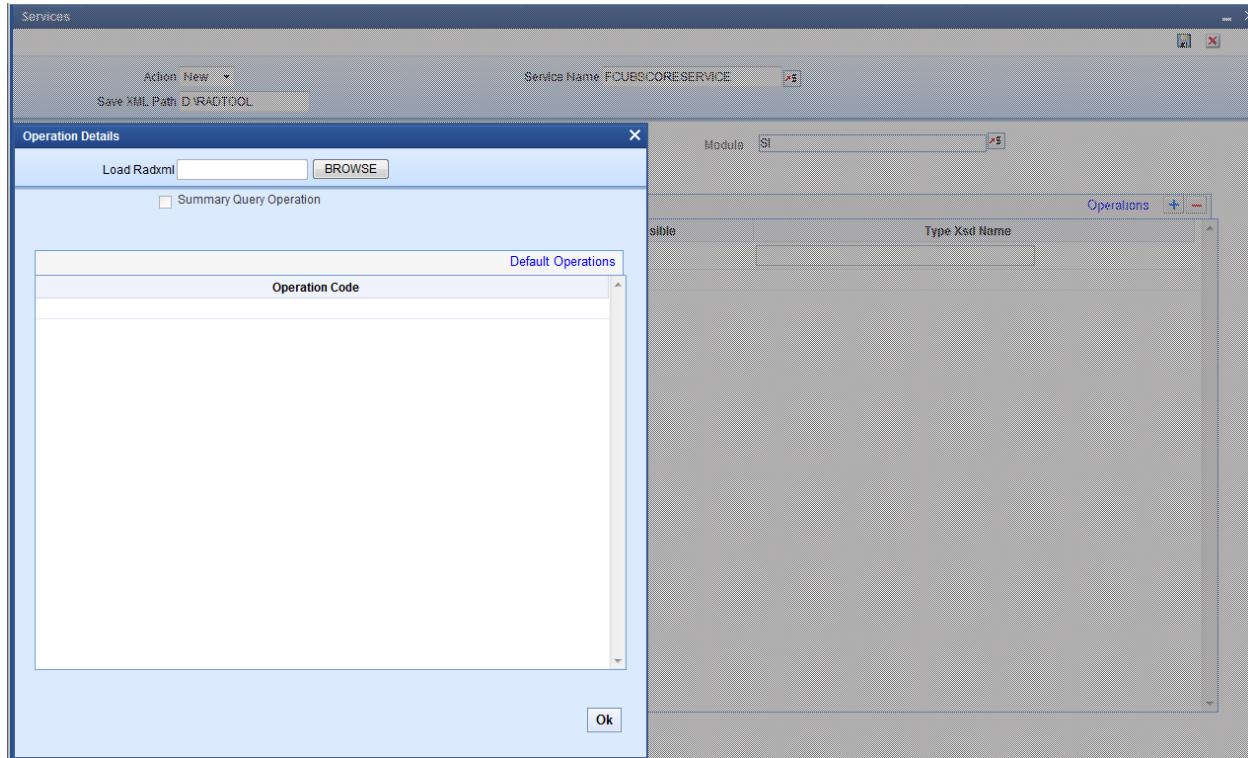


Fig 3.10: Screen to Show Operations.

**Operation Details Screen:**

**Load Radxml**

Operation details screen attaches a Browse button to it so that user can browse the Function id RAD XML and load it to populate operations

## FLEXCUBE UBS

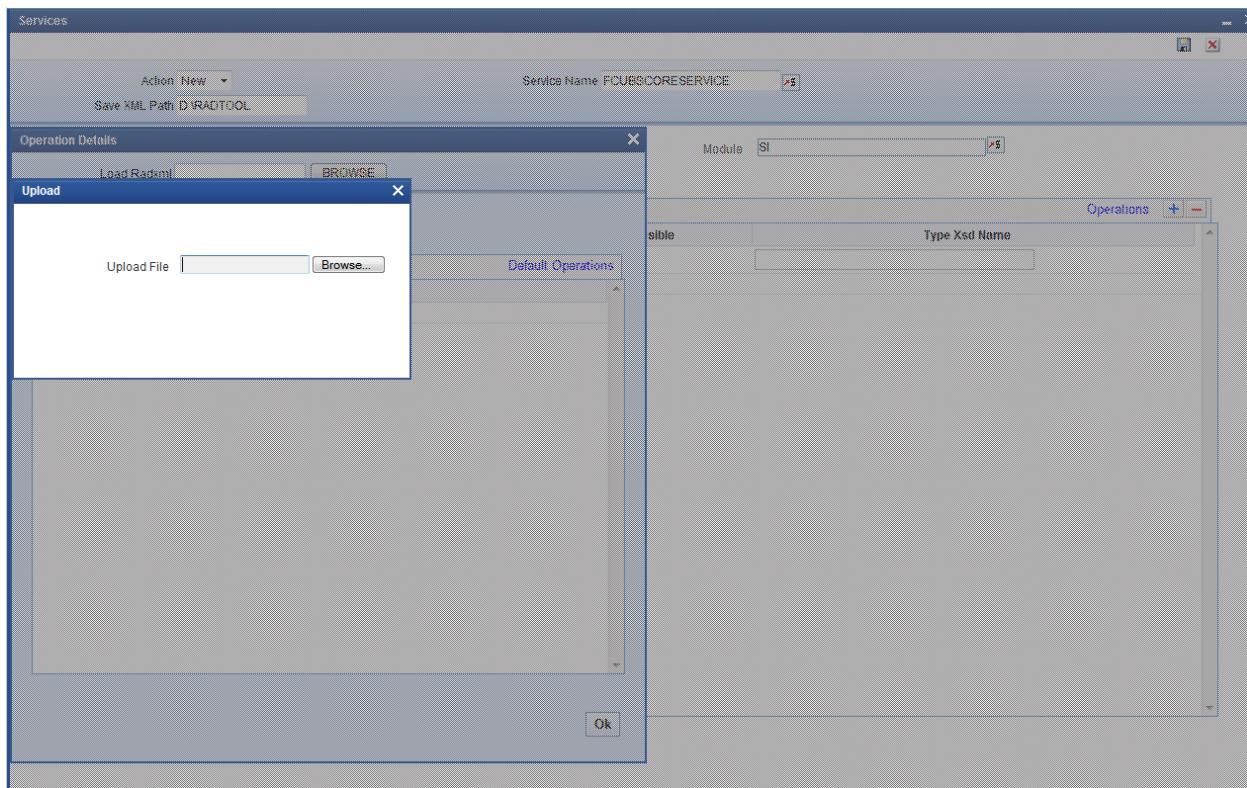


Fig 3.11: Screen to Load Radxml.

### Default Operations:

Click on Default operation. Operation code and Type Xsd Names will be default from loaded RADXML.

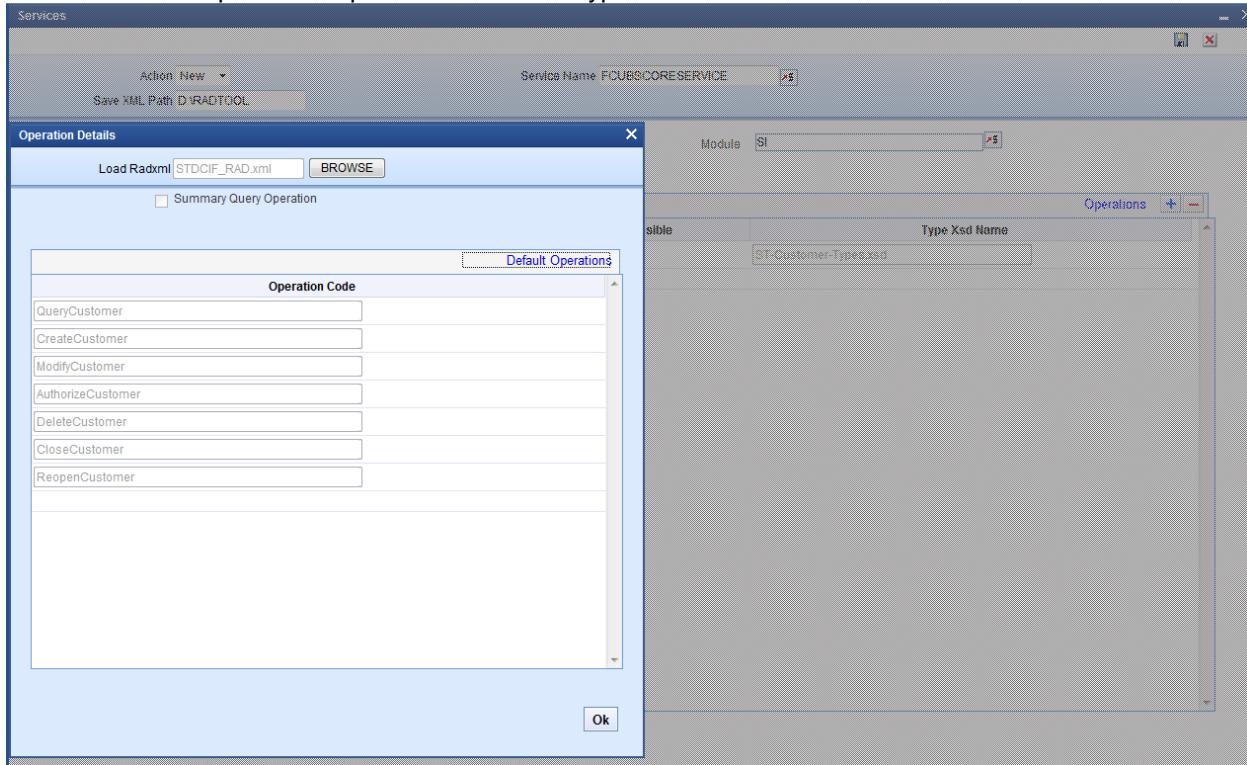


Fig 3.12: Screen to Default Operations..

## Save: ServiceXML

ODT saves all the activities carried out by the developer in an xml file hereby referred to as SXML. Persistence of the WEBSERVICE is achieved through SXML and RADXML. If some changes are required on the webservice in a future release, the same SXML can be loaded and changes can be done on this SXML. ODT can segregate the changes done on different releases and saves the SXML accordingly. SXML will adhere to following naming convention Service Name + .sxml Example: FCUBSCoreService.sxml

## 4 ODT Silent Utility

The Following operations are supported in silent utility of ODT

1. LOGIN
2. SETRELEASE: 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 sequence of operations is as above, then Login Operation, Set Release and Refresh Operations would be processed in respective sequence*

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

### 4.1 Prerequisites

- JDK

***License Information:***

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

***Instructions:***

Installer requires JDK 1.6.0\_24 version to be downloaded in the system and the same Should be set as environmental variable

- Apache Ant 1.7.1

**Instructions:**

Installer requires ANT 1.7.1 version to be downloaded in the system and the same should be set as environmental variable

## 4.2 How to run utility on Windows/Unix

After copying the installer sources and library folder to your local system, make sure you uncheck the read only check box in source properties and apply the same to all the sub folders.  
The screen shot below shows how the source folder in your local system should look like.

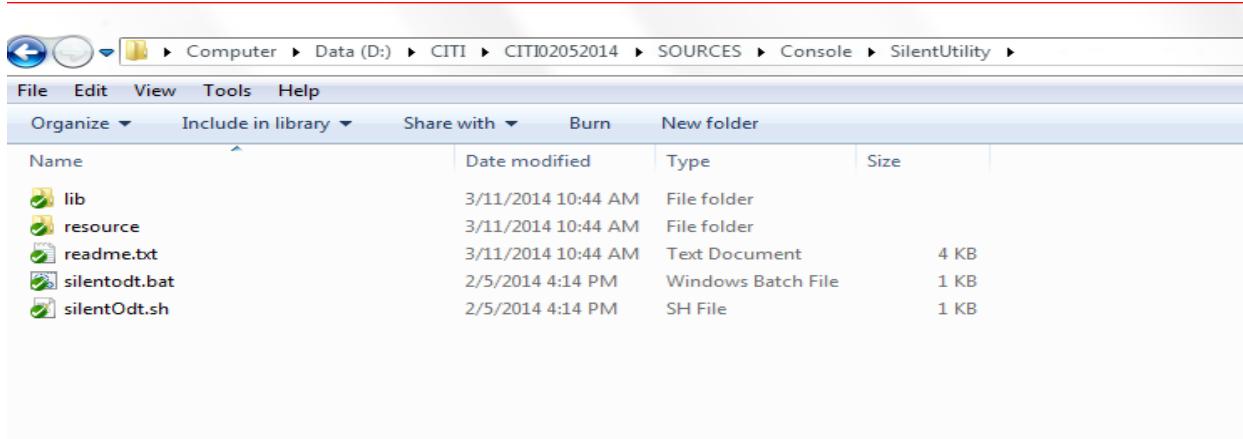


Fig 4.2.1: Source of SilentODTUtility.

## 4.3 Configuration of SilentODTUtility

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

### 4.3.1 SilentOdt.properties

```

1 ##ODT Version --12.0.2/12.0.1/11.4
2 odtVersion=12.0.2
3
4 #ODT DataSource Credentials
5
6 OdtJdbcUrl=jdbc:oracle:thin:@10.184.xx.xx:1521:FCDEMO
7 OdtDbUser=RAD
8 OdtDbPassword=RAD
9
10 ##### Logger Properties#####
11 #logreqd Default set to N
12 logreqd = N
13 #Default Path set User Home directory
14 logpath = D:/RADTOOL/log.txt
15 #LEVEL default set to INFO ; Options : DEBUG/INFO/WARNING/SEVERE
16
17
18 # Provide '/' as File Seperator for all paths
19 ##JAVA_HOME path
20 JAVA_HOME=C:/Program Files/Java/jdk1.6.0_45/
21 ##WEBLOGIC Home Path
22 WEBLOGIC_HOME=D://Oracle/Middleware/
23 ##WebSphere Home Path
24 WAS_HOME=D:/WAS/
25

```

Fig 4.3.1: SilentODT Properties.

<b>ODT Data Source Detail Credentials</b>	
OdtJdbcUrl	Jdbc Url jdbc:oracle:thin:@10.184.xx.xx:1521:FCDEMO
OdtDbUser	DB User name
OdtDbPassword	DP Password
<b>Logger Properties</b>	
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 DEBUG/INFO/WARNING/SEVERE. Provide as DEBUG for writing detailed log. Default value would be set to INFO
<b>System Properties:</b>	
JAVA_HOME	Maintain the Java installed location
WEBLOGIC_HOME	Maintain the oracle weblogic installed location
WAS_HOME	Maintain the IBM websphere installed location

### **ODT Data Source Detail Credentials**

Maintain ODT Database schema details as demonstrated in the figure above.  
Maintain values for ***OdtJdbcUrl***, ***OdtDbUser***, and ***OdtDbPassword***.

### Logger Properties:

**Logreqd:** Y/N. Default set to N,  
If value set to 'Y' logger will be enabled for generation of log  
If value set to 'N' logger will be disabled

**Logpath:** Provide the path where the Logger files will be generated.

**Level:** Provide the Logger Level.  
This can be either DEBUG/INFO/WARNING/SEVERE.  
Provide as DEBUG for writing detailed log.  
*Default value would be set to INFO*

### System Properties:

**JAVA\_HOME:** Maintain the Java installed location.

**WEBLOGIC\_HOME:** *Maintain the oracle weblogic installed location*

**WAS\_HOME:** *Maintain the IBM websphere installed location*

*(JAVA\_HOME, WEBLOGIC\_HOME, WAS\_HOME)These are optional and would be used for generating template Ant scripts*

#### 4.3.2 env\_config.xml

This configuration file is optional. This is used for connecting to the FLEXCUBE Business schema if unable to do the same from data in ODT schema

```

<?xml version = "1.0" encoding = "UTF-8"?>
<odt-configuration>
  <release-details id="FCUBS_12.0.3_CITI_DEV">
    <release-type>KERNEL</release-type>
    <environment id="FCUBS_12.0.3_CITI_DEV_CITI123DEV">
      <jdbc-url>jdbc:oracle:thin:@10.184.xxx|xx:1521:CITI</jdbc-url>
      <db-user>CITIDEV</db-user>
      <db-password>CITIDEV</db-password>
      <language-code>ENG</language-code>
    </environment>
  </release-details>
  <default-destination-path>D:\RADTOOL</default-destination-path>
</odt-configuration>

```

Fig 4.3.2.1: Env Config file

#### Establishing database Connections with FLEXCUBE schema

- 1) Tool Fetches Environment details as maintained in ODT through Environment Maintenance Screen.  
jdbc thin url will be derived as:

jdbc:oracle:thin:@DB\_HOSTNAME:DB\_PORT:DB\_INSTANCE  
username: DB\_SCHEMA

password: DB\_PASSWORD

2) If data provided in ODT is not correct OR if FCUBS DB Server is a clustered database, Jdbc thin URL details can be provided in env\_config.xml  
Sample Data is provided in image above  
Release ID and Environment ID should match the details in ODT

3) Note that env\_config.xml would be considered only if tool is unable to connect to schema using the data in ODT. It cannot be used for over-riding ODT maintained environment details

#### 4.3.3 OdtOperations.properties

Configure the Operations files as per Requirement.

The Following operations are supported in silent utility of ODT

8. LOGIN
9. SETRELEASE: Setting Release and Environment Details
10. BULKGENERATION: Bulk Generation of Radxml units
11. REFRESH: Bulk refresh of radxml
12. SXML\_REFRESH: Bulk refresh of Service Xmls
13. SXML\_UPDATER: Bulk Updater of service Xmls based on the changes in radxml's
14. 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 sequence of operations is as above, then Login Operation , Set Release and Refresh Operations would be processed in respective sequence*

*Note: login and setrelease are mandatory operations to be performed.*

##### 4.3.3.1 Login

**Userid:** Provide the ODT Userid which is created in the ODT Application

**Password:** Provide the ODT Password which is created in the ODT Application

```

1  ##Login to Tool
2  ## 1.operation = LOGIN
3  ## 1.userId=RADTOOL
4  ## 1.password=PASSWORD
5

```

Fig 4.3.3.1: SilentODT Login Properties

***Login should always be the first operation which to be configured as part of any execution***

Login to Tool	
operation	Login
UserId	ODT Userid which is created in the

	ODT Application
password	ODT Password which is created in the ODT Application

#### 4.3.3.2 Set Release

This operation can be used for setting Release and Environment Preferences for SilentODTUtility

**relcode:** Provide the ODT Release Code which is created in the ODT Application

**envCode:** Provide the ODT Environment Code which is created in the ODT Application

**langcode:** Provide the Lang code for above mentioned release code

*Connection to the FLEXCUBE schema would be established based on data maintained in ODT or through the data in env\_config.xml as explained in earlier section*

```

6  ##Set Release and Environment for the User
7  ## 2.operation= SETRELEASE
8  ## 2.relCode=MODEL_BANK
9  ## 2.envCode=MODEL_BANK_DEV_ENV
10 ## 2.langCode=ENG
11
12

```

Fig 4.3.3.2: SilentODT Set Release Properties

Set Release and Environment for User	
operation	SETRELEASE
relCode	ODT Release Code which is created in the ODT Application
envCode	ODT Environment Code which is created in the ODT Application
langCode	Lang code for above mentioned release code

#### 4.3.3.3 Bulk Generation

For generating all radxml artifacts for release in bulk this feature can be used.

**radxmlListFile:** Prepare text file which contains absolute path of all radxml's. Provide 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 of radxml List File or srcPath should be present .If both is present, then 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

**example :** *UIXML, SYS\_JS, MAIN\_SPC, MAIN\_SQL, KERNEL\_SPC, KERNEL\_SQL*

```

12
13 ##Bulk Generation Utility
14 ## 3.operation=BULKGENERATION
15 ## 3.radxmlListFile=D:\ODT123\ABC.TXT      -- A File containing absolute path of all radxmls to be processed
16 ## #3.srcPath=Z:\FCUBS12.0\MAIN           -- Source Path Refers to the path where all radxmls are present. List File
17 ##                                         would be generated by the Tool in this case.
18 ##                                         Note that only either of radxmlListFile or srcPath should be present .
19 ##                                         If both are present ,then radxmlListFile would be considered for Bulk
20 ##                                         Generation
21 ##                                         Generation
22 ##                                         fileType=EXTENSIBLE           -- EXTENSIBLE/NON_EXTENSIBLE/BOTH
23 ##                                         destpath=D:\RADTOOL           -- destination Path
24 ##                                         gen = UIXML,SYS_JS           -- Files to be Generated separated by comma. Possible entries are listed below
25 ##                                         UIXML,SYS_JS,MAIN_SPC,MAIN_SQL,KERNEL_SPC,KERNEL_SQL,CLUSTER_SPC,CLUSTER_SQL,CUSTOM_SPC,CUSTOM_SQL,UPLOAD_SPC,UPLOAD_SQL,
26 ##                                         UPLOAD_TRIGGER,UPLOAD_TABLE_DDL,XSD_FILES,MENU_DETAILS,LABEL_DETAILS,AMEND_DETAILS,SUMMARY_DETAILS,SCREEN_DETAILS,LOV_DETAILS,
27 ##                                         BLOCK_PK_COLS,CALL_FORM_DETAILS,BLOCK_DETAILS,DATASCR_DETAILS,FUNCTION_CALL_FORMS,GATEWAY_DETAILS,NOTIFICATION_DETAILS,
28 ##                                         FUNCTION_PARAMETERS
29 ##                                         NOTIFICATION_TRIGGER,PURGE_DETAILS,ARCHIVE_TBL_DEF
30

```

Fig 4.3.3.3: SilentODT Bulk Generation Properties

<b>Bulk Generation Utility</b>	
<b>Input</b>	<b>Output</b>
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_TRIGGER,UPLOAD_TABLE_DDL,XSD_FILES,MENU_DETAILS,LABEL_DETAILS,AMEND_DETAILS,SUMMARY_DETAILS,SCREEN_DETAILS,LOV_DETAILS,BLOCK_PK_COLS,CALL_FORM_DETAILS,BLOCK_DETAILS,DATASCR_DETAILS,FUNCTION_CALL_FORMS,GATEWAY_DETAILS,NOTIFICATION_DETAILS,FUNCTION_PARAMETERS
srcPath	NOTIFICATION_TRIGGER,PURGE_DETAILS,ARCHIVE_TBL_DEF in destpath
fileType	
gen	
destpath	

#### 4.3.3.4 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.(Please refer the 09-Development\_WorkBench\_Source\_Upgrade.docx)

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

**Refresh Type:** Provide the refresh Type

(CHILD\_REFRESH/SCRCHILD\_REFRESH/SOURCE\_REFRESH)

**srcFileList:** A txt File containing the List of all Sources radxml's. I.e. radxml's which has to be refreshed

**baseFileList:** A txt File containing the List of all base radxml's.

**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

```

28 ##Refresh Utility
29 ## 4.operation=REFRESH
30 ## 4.refreshType=SOURCE_REFRESH      -- Either of CHILD_REFRESH/SCRCHILD_REFRESH/SOURCE_REFRESH
31 ## 4.srcFileList=D:\\REFRESH\\src.txt -- A txt File containing the List of all Sources radxmls. i.e radxmls which
has to be refreshed
32 ## 4.baseFileList=D:\\REFRESH\\base.txt -- A txt File containing the List of all base radxmls.
33 ## 4.srcRelType=CUSTOM               -- Release Type of Source Radxmls; Either of KERNEL/CLUSTER/CUSTOM
34 ## 4.baseRelType=KERNEL              -- Release Type of Base Radxmls;Either of KERNEL/CLUSTER/CUSTOM
35 ##                                         Note that base and Src Release Types should be the same for Child and
36 ##                                         Base Release Type should be atleast one level below Src Release type for
37 ##                                         SOURCE Refresh.
38 ## 4.destpath=D:\\RADTOOL
39

```

Fig 4.3.3.4: SilentODT Refresh Properties

Refresh Utility	
Input	output
refreshType	Refreshed Radxml's in destpath
srcFileList	
baseFileList	
srcRelType	
baseRelType	
destpath	

#### 4.3.3.5 Service XML Bulk Generation

Web service artifacts can be generated through this operation

**sxmlListFile:** Prepare text file which contains absolute path of all Service xml .

**radxmlListFile:** Prepare text file which contains absolute path of all radxmls which are used for those services .

**xsdListFile:** Prepare text file which contains 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 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 coma) . 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 included in the generated Components

**destpath:** Provide the path where the files will be generated.

**validateXsds** : Y/N. Default set to Y,  
*If value set to 'Y' all XSD will be validated by tool*

```

40 ## Service XML Component Generator
41 ## 5.operation=SXML_BULKGENERATION
42 ## 5.sxmlListFile=D:\\ODT123\\TEST\\srcFile.TXT          -- List Of Absolute path of all sxml files in a text file
43 ## 5.radxmxmlListFile=                                    -- A txt File containing the List of all radxmls.
44 ## 5.xsdListFile=                                    -- A txt File containing the List of all xsds. This
parameter is required only if NonExt Operations are Required.
45 ##
46 ## 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
47 ##
48 ## xsdListFile need not be provided
49 ## 5.gen=IMPL_FILE,CONFIG_FILES,WSDL_FILE,XSD_FILES      -- Note that if srcPath is provided,radxmlListFile and
below
50 ## IMPL_FILE,CONFIG_FILES,WSDL_FILE,XSD_FILES,GW_WS_PROP_FILES,ANT_BUILD
51 ## 5.nonExtServicesReqd=Y                                -- Y/N Specifies whether NonExtensible Operations has to
included in the Generated Components
52 ## 5.destpath=D:\\RADTOOL                                -- destination Path
## 5.validateXsds=Y                                     -- validate the xsds Y/N

```

Fig 4.3.3.5: SilentODT Service XML Bulk Generation Properties

<b>Service XML Component Generator</b>	
<b>Input</b>	<b>Output</b>
sxmlListFile	
radxmlListFile	
xsdListFile	IMPL_FILE,CONFIG_FILES,WSDL_FILE,XSD_FILES,GW_WS_PROP_FILES,ANT_BUILD in destpath
srcPath	
nonExtServicesReqd	
destpath	
validateXsds	

#### 4.3.3.6 Service XML Updater

This feature can be used to update the Service XMLs with the latest data from Radxmls. Following details will be updated.

- 1) Any **addition, deletion or modification of operation codes** in function Id would be updated in Service XML
- 2) 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 be added manually in the Service XML through ODT user interface*

**Operation:** SXML\_UPDATE

**sxmlListFile:** Prepare text file which contains absolute path of all Service xml. Provide same file path .

**radxmlListFile:** Prepare text file which contains absolute path of all radxmls which are used for those services . Provide 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

**destpath:** Provide the path where the files will be generated.

**confirmStage : SINGLE\_STAGE\_UPDATE**(Default Value should not be modified by developer)

```

56 ## Service XML Updater
57 ## 6.operation=SXML_UPDATER
58 ## 6.sxmlListFile=D:\\ODT123\\TEST\\srcFile.TXT          -- List Of Absolute path of all sxml files in a text file
59 ## 6.radxmlListFile=                                    -- A txt File containing the List of all radxmls.
60 ## 6.xsdListFile=                                    -- A txt File containing the List of all xsds. This
61 ## parameter is required only if NonExt Operations are Required.
62 ##
63 ## 6.srcPath=Z:\\EXEC\\FLEXCUBE_Kernel\\FCUBS_12.0.0\\MAIN  -- Src Path. Tool will create radxmlListFile and
64 ## xsdListFile by itself from the srcPath if provided
65 ## xsdListFile need not be provided
66 ## 6.destpath=D:\\RADTOOL                                -- destination Path
67 ## 6.confirmStage=SINGLE_STAGE_UPDATE                  --
68 ## SINGLE_STAGE_UPDATE/UPDATE_FROM_STAT_FILES/STAT_FILE_GEN
69 ## one step process.
70 ## If any New function id is found mapped to a
71 ## Service, it will not be updated.
72 ## STAT_FILE_GEN : First Stage if Updation carried in
73 ## User can decide whether to update new
74 ## UPDATE_FROM_STAT_FILES : Second Step . Utility will
75 ## information provided by User in the Stat files
76 ## generated in previous Stage.Stat Files has to be placed in the destPath
77 #

```

Fig 4.3.3.6: SilentODT Service XML Updater Properties

<b>Service XML Updater</b>	
<b>Input</b>	<b>output</b>
sxmlListFile	updated Service XML's in destpath
radxmlListFile	
xsdListFile	
srcPath	
destpath	
confirmStage	

#### 4.3.3.7 Service XML Refresh

Refresh Functionality allows us to upgrade the existing service xml to its later version keeping the sub version specific changes intact.

**srcFileList:** Prepare text file which contains absolute path of all Service xml, same file should be provide i.e. service xml which has to be refreshed

**baseFileList:** Prepare text file which contains absolute path of all base service xmls. Provide same file path .

*For instance, for a custom development team ; all the latest Kernel files has to 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.

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

Fig 4.3.3.7: SilentODT Service XML Refresh Properties

Service XML Refresh	
operation	Refreshed Service XML's in destpath
srcFileList	
baseFileList	
srcRelType	
baseRelType	
destpath	

#### 4.3.4 GW\_CONFIG.properties

This properties files parameters are used for 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

**EJB\_APP\_NAME:** Provide 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.

*Example.: D:/Kernel11.1/GW\_WS/config/gw\_ws\_logger.properties(Windows path)  
/oraint1/kernel//Gateway/GWWS/config/gw\_ws\_logger.properties (Linux or UNIX)*

**GW\_WS\_LOGGER\_FILE\_PATH:** Provide the location where debug files will be written

*Example : D:/Kernel11.1/GW\_WS/log/ (Windows)  
/oraint1/kernel/FC120INS\_DEBUG/Gateway/GWWS/log (Linux or Unix)*

**GW\_WS\_PROP\_FILE\_PATH:** Path where property file is placed in the server. This will be referred in web.xml of web service property file

*Example: D:/Kernel11.1/GW\_WS/prop/ (Windows)  
/oraint1/kernel/FC120INS\_DEBUG/Gateway/GWWS/prop (Linux or UNIX)*

```
#####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
```

Fig 4.3.4.1: SilentODT GW Config Properties

GW_CONFIG	
EJB_APP_NAME	Provide Name of the deployed EJB Application
EJB_APP_SERVER	Provide Name of the deployed EJB Application

EJB_JNDI_NAME	<i>EJB JNDI Name</i> 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.
GW_WS_LOGGER_FILE_PATH	Provide the location where debug files will be written
GW_WS_PROP_FILE_PATH	Path where property file is placed in the server. This will be referred in web.xml of web service property file.
XSD_PATH	XSD Path.

#### 4.4 Generation 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 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*

Double click the batch file silentOdt.bat/sh present in the ODT source.

## FLEXCUBE UBS

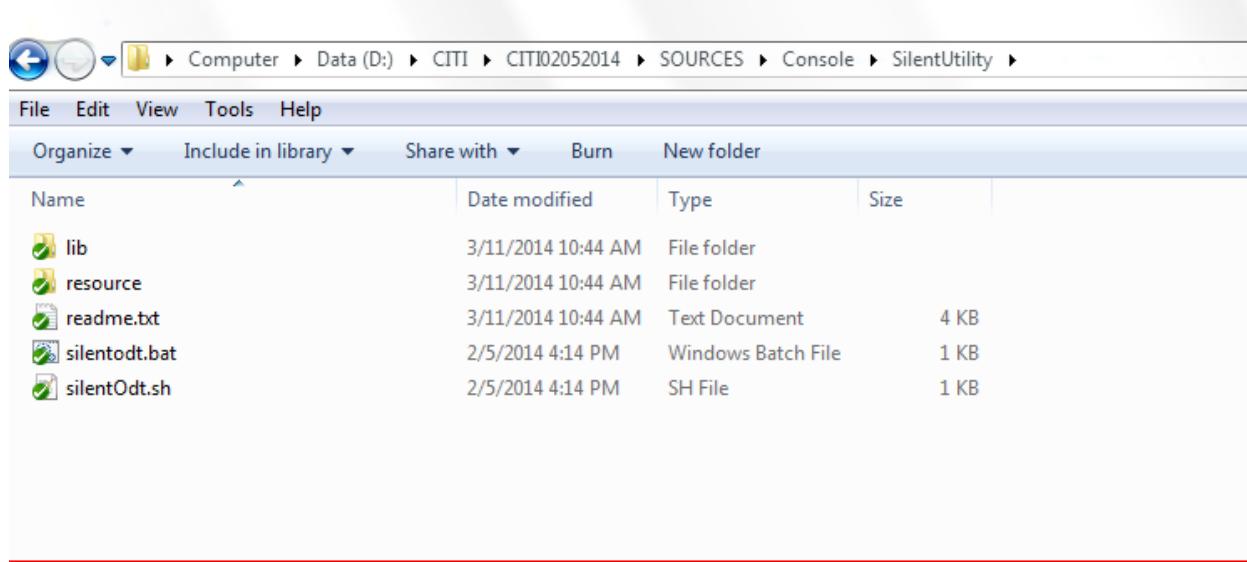


Fig 4.4.1: SilentODT Sources

This displays the screen as follows.

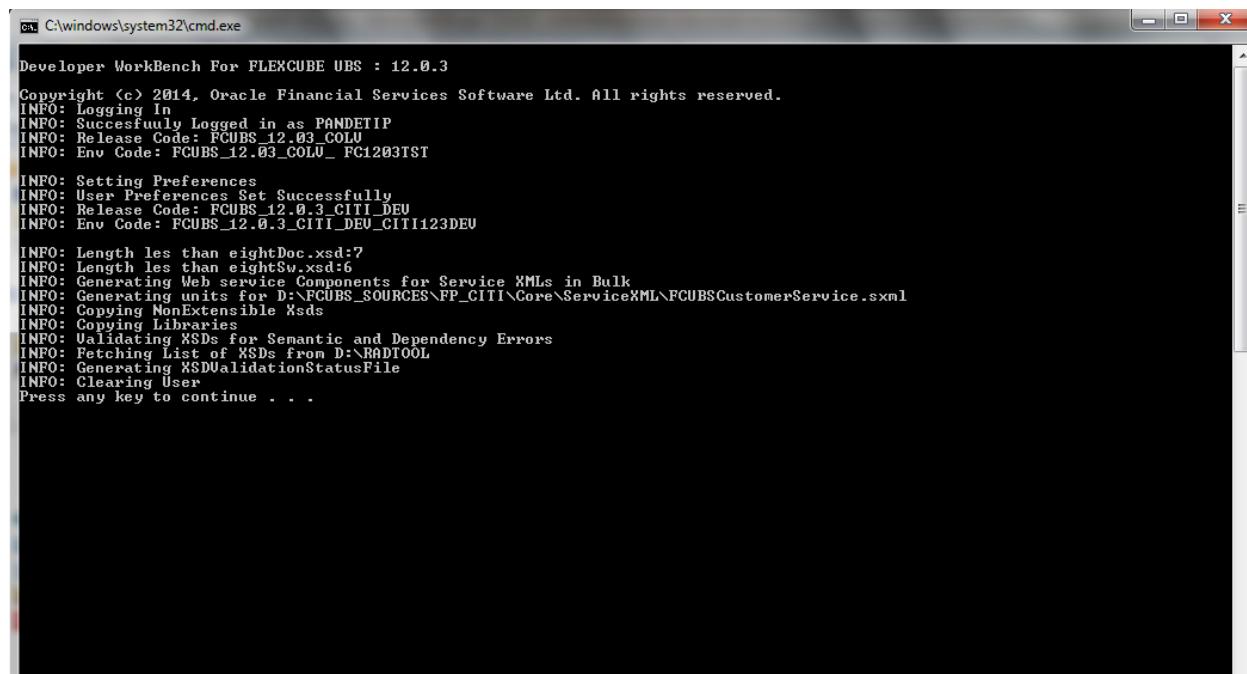


Fig 4.4.2: SilentODT Running in Command Prompt

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

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 )

```

Folder PATH listing for volume Data
Volume serial number is FEF0-E959
D:.
    1394603209212ListXSD.txt
    log.txt
    ServiceGenerationStatus.csv

    Core
        Gateway
            COMMON
                FCUBS_REQ_ENU.xsd
                FCUBS_RES_ENU.xsd
                SubSys-Cscofacm-Types.xsd
                SubSys-Cscupdoc-Types.xsd
                SubSys-CustDiaryDetails-Types.xsd
                SubSys-CustJoint-Types.xsd
                SubSys-Custmis-Types.xsd
                SubSys-Custtext-Types.xsd
                SubSys-Fields.xsd
                SubSys-LinkedEntity-Types.xsd
                SubSys-Stccrdsa-Types.xsd
                SubSys-Stccuacc-Types.xsd
                SubSys-Stensadv-Types.xsd
                SubSys-Stensck-Types.xsd
                SubSys-Stensfib-Types.xsd
                SubSys-Txn-Fields.xsd
                UBS-Messaging.xsd

            Services
                FCUBSCustomerService
                    WASAntBuild.xml
                    WLANNTBuild.xml
                    Config
                        application_WS_FCUBSCustomerService.xml
                        web.xml
                        weblogic.xml
                        webservices_FCUBSCustomerService.xml
                        web_WS_FCUBSCustomerService.xml
                    lib
                        commons-codec-1.2.jar
                        wscommon.jar
                    Src
                        FCUBSCustomerServiceImpl.java
                    Wsdl
                        FCUBSCustomerService.wsdl

            XSD
                CA-AmtBlk-Types.xsd
                CA-AuthorizeAmtBlk-Req-Full-MSG.xsd
                CA-AuthorizeAmtBlk-Req-IO-MSG.xsd
                CA-AuthorizeAmtBlk-Res-Full-MSG.xsd
                CA-AuthorizeAmtBlk-Res-PK-MSG.xsd
                CA-CloseAmtBlk-Req-Full-MSG.xsd
                CA-CloseAmtBlk-Req-IO-MSG.xsd
                CA-CloseAmtBlk-Res-Full-MSG.xsd
                CA-CloseAmtBlk-Res-PK-MSG.xsd
                CA-CreateAccountStructure-Req-Full-MSG.xsd
                CA-CreateAccountStructure-Req-IO-MSG.xsd
                CA-CreateAccountStructure-Res-Full-MSG.xsd
                CA-CreateAccountStructure-Res-PK-MSG.xsd
                CA-CreateAmtBlk-Req-Full-MSG.xsd
                CA-CreateAmtBlk-Req-IO-MSG.xsd

```

Fig 4.4.3: SilentODT Generated Files Tree Structure

#### 4.4.1 Log Files

- **ServiceGenerationStatus.csv**

Generation status will be saved in above mentioned file.

This will be generated in the destination path

- **XSDValidationErrors.csv**

XSD Validation errors, if any, will be saved in 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 status is Success for all services in ServiceGenerationStatus.csv and XSDValidationErrors.csv is not generated***

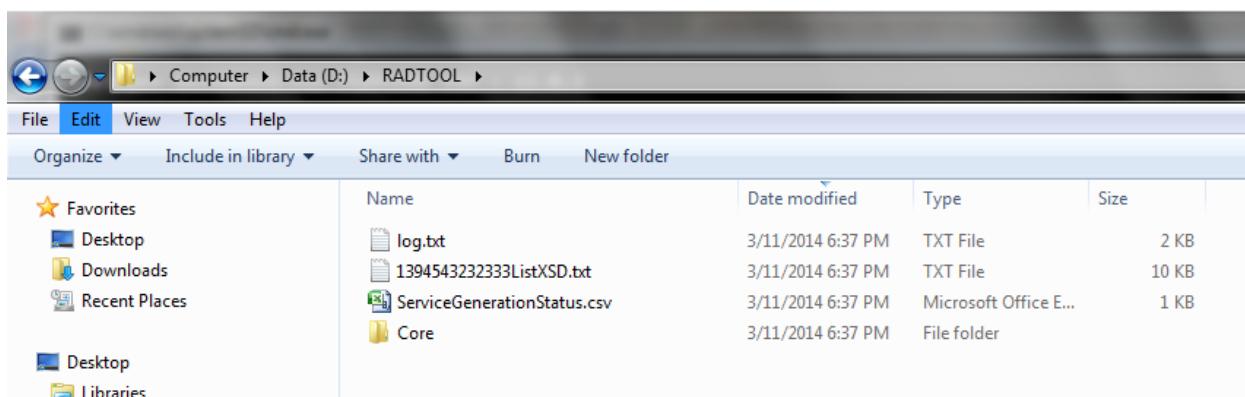


Fig 4.4.1.1: SilentODT Generated log Files.

#### 4.4.2 Ant Build Scripts

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

## FLEXCUBE UBS

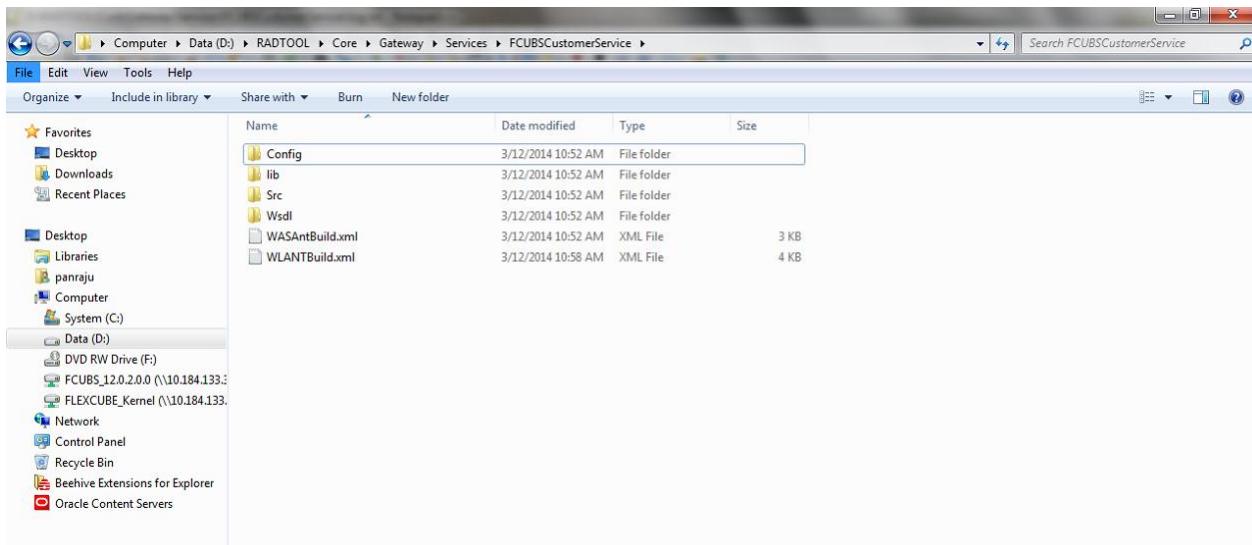


Fig 4.4.1.2: SilentODT Generated Files Ant Build Files.

### 4.4.3 Gateway Property Files

Gateway property files would be generated in GW\_WS folder inside destination folder.  
.Following files would be generated.

- gw\_ws\_logger.properties
- GW\_WS\_Prop.properties

Development Workbench –  
Web service & Silent Utility for Generation of web service artifacts  
March 2014

1.0

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/](http://www.oracle.com/financial_services/)

Copyright © 2014-2014 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-Webservice Development* 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-Tracking Changes* 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.