Oracle® FLEXCUBE Investor Servicing

Development Workbench - Service XML Development

Release 14.7.6.0.0
G31220-01
April 2025

ORACLE

Oracle FLEXCUBE Investor Servicing Development Workbench - Service XML Development, Release 14.7.6.0.0

G31220-01

Contents

Preface

Preface

Oracle FLEXCUBE Investor Servicing is a comprehensive mutual funds automation software from Oracle® Financial Servicing Software Ltd.©.

You can use the system to achieve optimum automation of all your mutual fund investor servicing processes, as it provides guidelines for specific tasks, descriptions of various features and processes, and general information.

This topic contains the following sub-topics:

- Purpose
- Audience
- Documentation Accessibility
- Critical Patches
- Diversity and Inclusion
- Conventions
- Screenshot Disclaimer
- Prerequisite
- Related Resources

Purpose

This manual is designed to help FLEXCUBE Application developers/users to familiarize with ORACLE FLEXCUBE Development Workbench for Investor Servicing.

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:

Table 1 Proficiency and Resources

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

Table 1 (Cont.) Proficiency and Resources

Proficiency	Resources
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 and SQL Language	Self-Acquired
Working knowledge of XML files	Self-Acquired

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 customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

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.

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.

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.

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.

Prerequisite

Specify **User ID** and **Password**, and log in to **Home Screen**.

Related Resources

The functions of ORACLE FLEXCUBE Development Workbench for Investor Servicing system is organized into various guides, each discussing a component.

For more information, see these Open Development Tool documents:

- *Open Development Tool Installation*
- *Development Workbench - Getting Started*
- *Development Workbench - Administration*
- *Development Workbench - Screen Development I*
- *Development Workbench - Screen Development II*
- *Development Workbench - Screen Customizer*
- *Development Workbench - Notifications*
- *Development Workbench - Bulk Generation*
- *Development Workbench - Source Upgrade*
- *Development Workbench - Tracking Changes*
- *Child and Screen Childs - Concept and Design*
- *Development of Maintenance Form*
- *Development of Online Form*
- *Development of Call Form*
- *Development of Launch Forms and Other Screens*
- *Development of Dashboard Form*
- *Development Workbench Service XML Development*
- *Development Workbench Performance Tuning Enhancements*
- *Development Workbench - Rest Services Development*

1
Service XML Development

This topic describes the webservice development using Oracle FLEXCUBE Development Workbench for Investor Servicing.

This topic explains the steps to create/Modification of Service xml and generating webservice artifacts for building ear file.

This topic contains the following sub-topics:

2
Service XML

This topic provides an overview of components of the Service XML.

Oracle FLEXCUBE Development Workbench provides the developer with a user friendly console for defining a gateway service of **FCIS**. 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.

Artifacts of Service XML

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

Table 2-1 Files and Description

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>\METAINF\application.xml <Service Name>\<Service Name>\METAINF\MANIFEST.MF	Config XML's for building the Web service
<Service Name>\<Service Name>\commonscodec-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.

- Generate Service XML
  This topic provides the systematic instructions to generate Service XML.

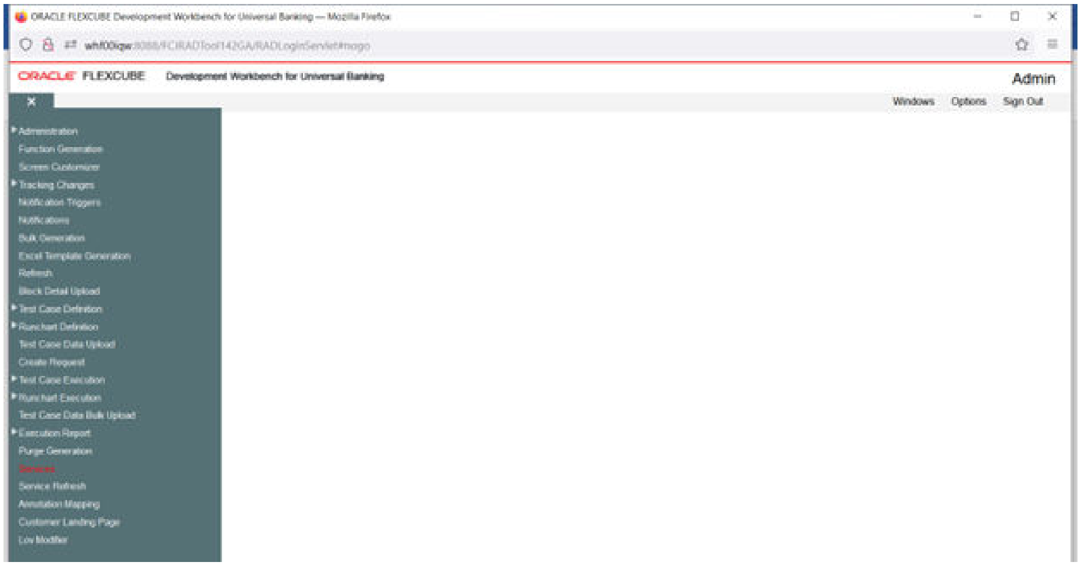
2.1 Generate Service XML

This topic provides the systematic instructions to generate Service XML.

1. Log in to **Development Workbench for Universal Banking**.

   The **Development Workbench for Universal Banking** Home Page is displayed.

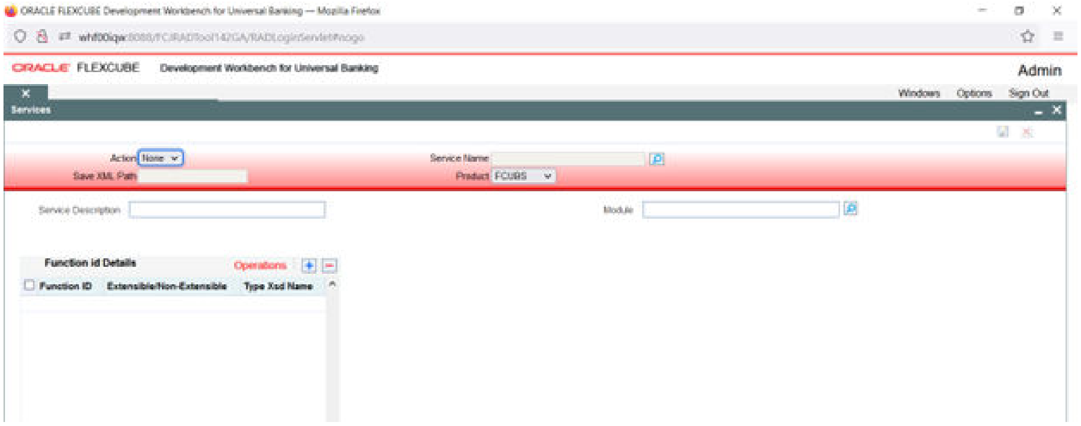
   Figure 2-1 Development Workbench for Universal Banking

2. Click the **Services** node in the **Browser** menu in the Landing page of ODT.

   The **Services** screen is displayed.

   Figure 2-2 Services Screen

   
   ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox
whf00iqw:8088/FCIRADTool142GA/RADLoginServlet#nogo
ORACLE FLEXCUBE
Development Workbench for Universal Banking
Admin
Windows
Options
Sign Out
Services
Service Name
Save XML Path
Service Description
Module
Function id Details
Operations
Function ID
Extensible/Non-Extensible
Type Xsd Name

   Action: None

   Product: FCUBS

3. Specify the details in the **Services** screen while creating a new service in ODT.

   For more information on fields, refer to the field description table.

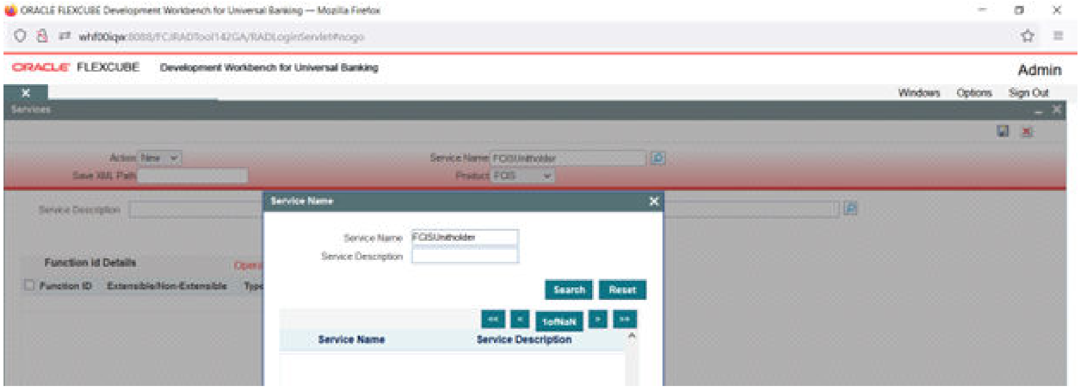
Table 2-2   Services - Field Description

Field	Description
Action	Specify the **Action** from the drop-down list. The list displays the following values: • **New** • **Load** Select the **Action** as **New** for a new Service development. Select the **Action** as **Load** to load the corresponding Service xml using `browser` option in Save Xml path.
Load Screen XML	Use `Browse` option to load XML.
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).
Service Description	On selection 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.

4. Select the **Service Name** from the service LOV, if the action is selected as **New**.

   The **Service Name** window populates.

Figure 2-3   Service Name



Action: New
Service Name: FCISUnitholder
Product: FCIS

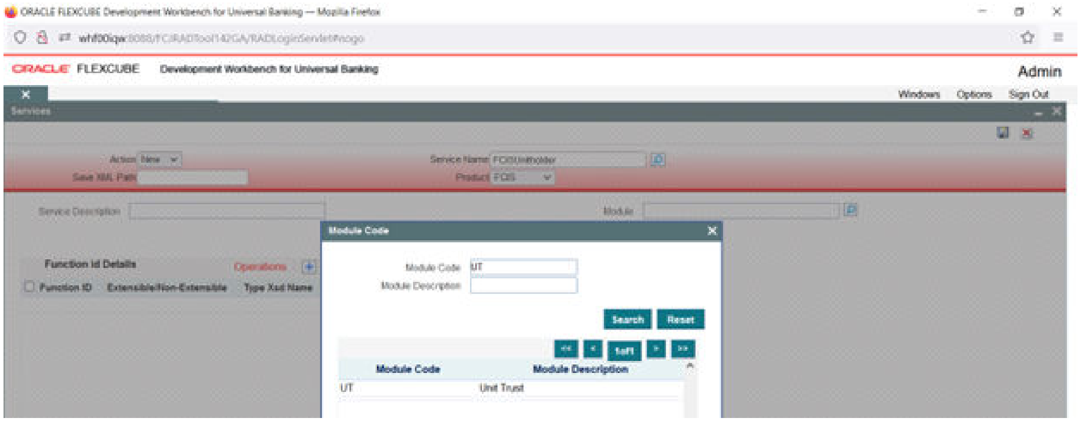
Service Name: FCISUnitholder

5. Select the **Module** from the Module LOV, if the action is selected as **New**.

   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. On selection of module, module code and module description will be populated in module description field.

   The **Module Code** window populates.

Figure 2-4   Services_Module Code



ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox
whf00iqw:8088/FCJRADTool142GA/RADLoginServlet#nogo
ORACLE FLEXCUBE
Development Workbench for Universal Banking
Admin
Windows
Options
Sign Out
Services
Function Id Details
Operations
Search
Reset
1 of 1

Action: New
Save XML Path:
Service Name: FCISUnitholder
Product: FCIS
Service Description:
Module:

□ Function ID	Extensible/Non-Extensible	Type Xsd Name

Module Code

Module Code: UT
Module Description:

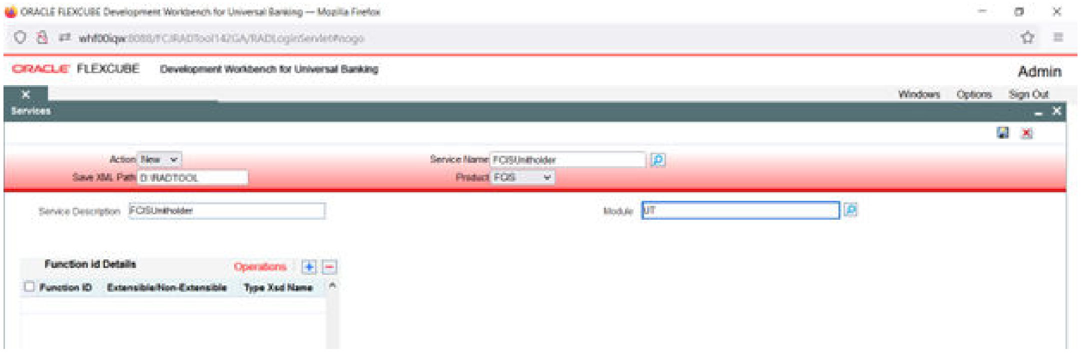
Module Code	Module Description
UT	Unit Trust

6. **Save XML Path:** Save the xml path if the action is selected as **New**. It is optional and 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 the **Browse** button to it so that user can browse the Service xml and load it.

Figure 2-5   Services_Save Xml Path



ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox
whf00iqw:8088/FCJRADTool142GA/RADLoginServlet#nogo
ORACLE FLEXCUBE
Development Workbench for Universal Banking
Admin
Windows
Options
Sign Out
Services
Function Id Details
Operations

Action: New
Save XML Path: D:\RADTOOL
Service Name: FCISUnitholder
Product: FCIS
Service Description: FCISUnitholder
Module: UT

□ Function ID	Extensible/Non-Extensible	Type Xsd Name

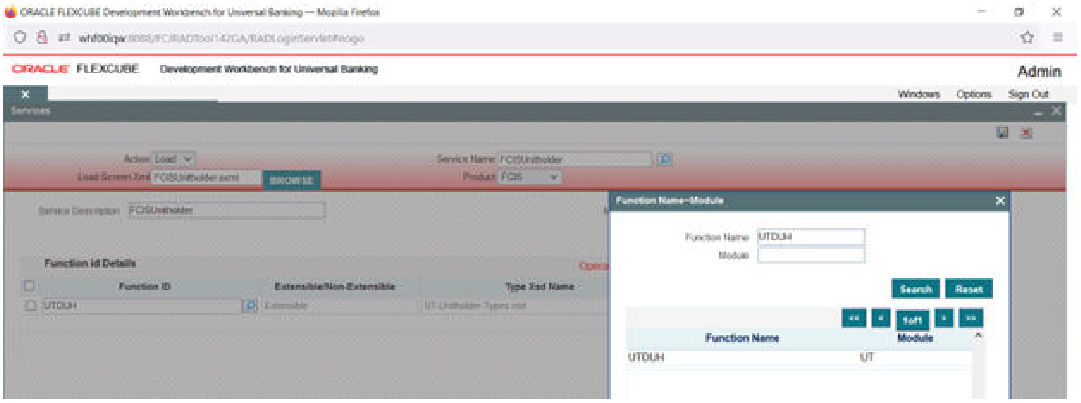
7. Attach the **Function ID** which are part of this service. You can also remove the same from service if not required.

8. Select the **Function ID** from Function Id LOV to add the same 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 **UTDUH** and not **UTGUH**.

   The **Function Name-Module** window populates.

Figure 2-6 Function Name-Module


ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox
whf00iqw:8088/FCJRADTool142GA/RADLoginServlet#nogo
ORACLE FLEXCUBE Development Workbench for Universal Banking
Admin
Windows Options Sign Out
Services
BROWSE
Function Id Details
Operations
Function Name-Module
Search
Reset
1of1

Action: Load
Load Screen Xml: FCISUnitholder.sxml
Service Name: FCISUnitholder
Product: FCIS
Service Description: FCISUnitholder

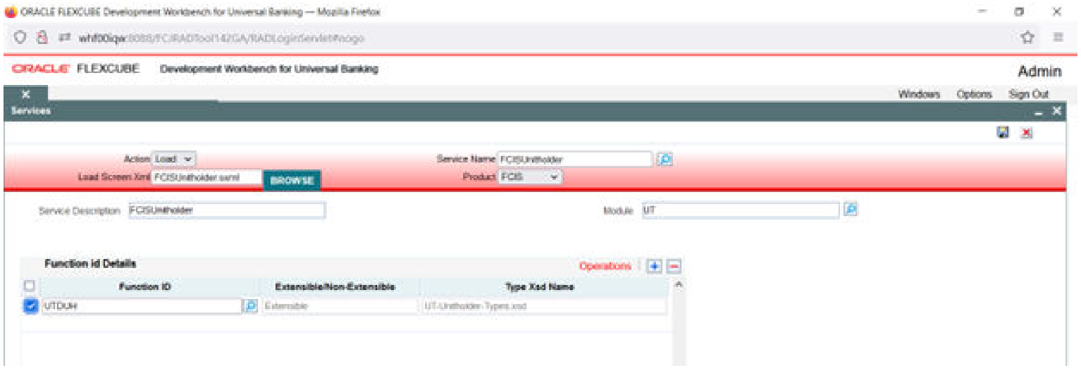
☐	Function ID	Extensible/Non-Extensible	Type Xsd Name
☐	UTDUH	Extensible	UT-Unitholder-Types.xsd

Function Name: UTDUH
Module:

Function Name	Module
UTDUH	UT

9. On selection of Function Id, this field value **Extensible/Non-Extensible** would be populated.

Figure 2-7 Screen to populate Function Id


ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox
whf00iqw:8088/FCJRADTool142GA/RADLoginServlet#nogo
ORACLE FLEXCUBE Development Workbench for Universal Banking
Admin
Windows Options Sign Out
Services
BROWSE
Function Id Details
Operations

Action: Load
Load Screen Xml: FCISUnitholder.sxml
Service Name: FCISUnitholder
Product: FCIS
Service Description: FCISUnitholder
Module: UT

☐	Function ID	Extensible/Non-Extensible	Type Xsd Name
☑	UTDUH	Extensible	UT-Unitholder-Types.xsd

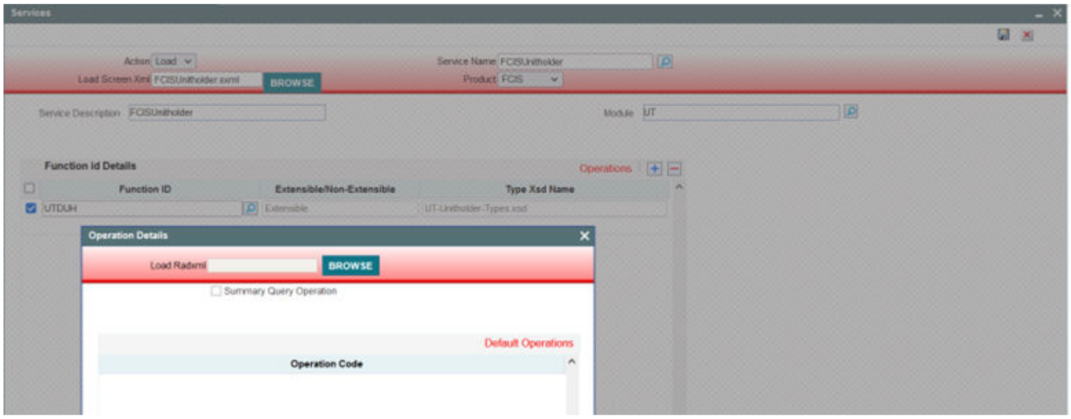
10. **Type XSD name** will be defaulted along with operations for an Extensible Function Id. For Non-Extensible Function Id, **Type XSD name** has to be explicitly mentioned in the field.

11. Select the **Function ID** checkbox and click **Operations**.

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

    The **Operation Details** pop-up screen is displayed.

Figure 2-8 Services_Operation Details



Action: Load
Load Screen Xml: FCISUnitholder.sxml
Service Name: FCISUnitholder
Product: FCIS
Service Description: FCISUnitholder
Module: UT

☐	Function ID	Extensible/Non-Extensible	Type Xsd Name
☑	UTDUH	Extensible	UT-Unitholder-Types.xsd

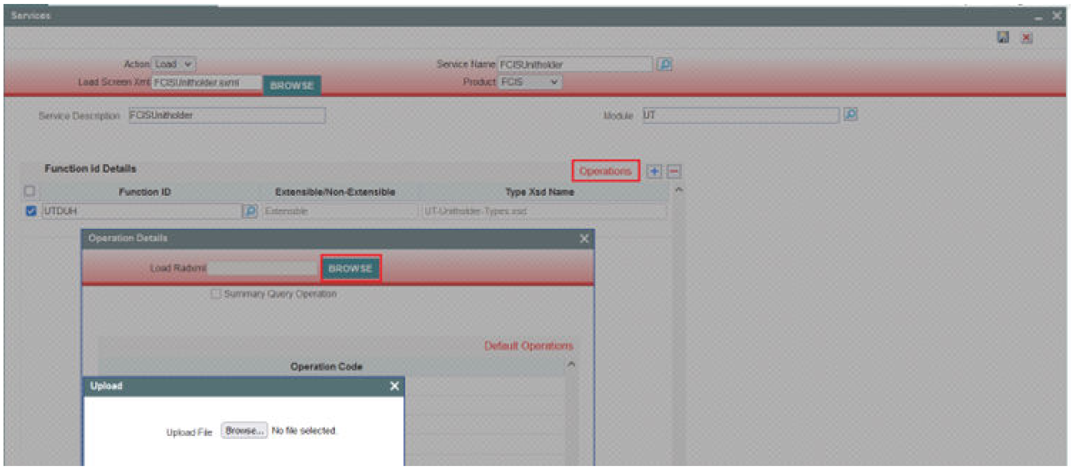
Load Radxml:
☐ Summary Query Operation

Operation Code

12. In the **Operation Details** screen, click the **Browse** button to it so that the user can browse the Function id RAD XML and load it to populate operations.

    The **Upload** window populates.

Figure 2-9 Services_Operation Details_Upload



Action: Load
Load Screen Xml: FCISUnitholder.sxml
Service Name: FCISUnitholder
Product: FCIS
Service Description: FCISUnitholder
Module: UT

☐	Function ID	Extensible/Non-Extensible	Type Xsd Name
☑	UTDUH	Extensible	UT-Unitholder-Types.xsd

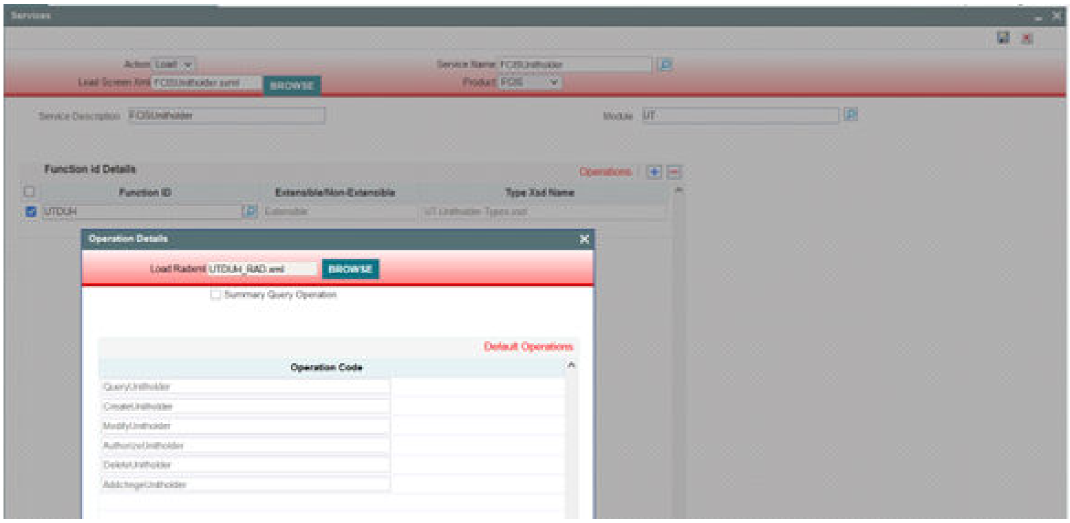
Load Radxml:
☐ Summary Query Operation

Upload File: Browse... No file selected.

13. Click the **Default operations** in the **Operation Details** Screen. **Operation Code** and **Type Xsd Names** will be defaulted from loaded RADXML.

    The **Services_Operation Details_Default Operations** screen is displayed.

Figure 2-10 Services_Operation Details_Default Operations



Action: Load
Load Screen Xml: FCISUnitholder.sxml
Service Name: FCISUnitholder
Product: FCIS
Service Description: FCISUnitholder
Module: UT

☐	Function ID	Extensible/Non-Extensible	Type Xsd Name
☑	UTDUH	Extensible	UT-Unitholder-Types.xsd

Load Radxml: UTDUH_RAD.xml
☐ Summary Query Operation

Operation Code
QueryUnitholder
CreateUnitholder
ModifyUnitholder
AuthorizeUnitholder
DeleteUnitholder
AddchangeUnitholder

14. Save the Service XML. 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 follow the below mentioned naming convention as **Service Name + .sxml**.

    Example: FCISUnitholder.sxml

3
ODT Silent Utility

This topic gives an overview on operations supported in silent utility of ODT.

The Following operations are supported in silent utility of ODT:

- **LOGIN**
- **SETRELEASE**: Setting Release and Environment Details
- **BULKGENERATION**: Bulk Generation of RADXML's units
- **REFRESH**: Bulk refresh of RADXML's
- **SXML_REFRESH**: Bulk refresh of Service XML's
- **SXML_UPDATER**: Bulk Updater of service XML's based on the changes in RADXML's
- **SXML_BULKGENERATION**: Bulk Generation of web service artifacts.

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

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 the respective sequence.

> **Note:**
>
> Login and Set Release are mandatory operations to be performed.

- Prerequisites
  This topic gives an overview on the prerequisites of ODT Silent Utility.
- Run utility on Windows/Unix
  The topic briefs on run utility operation in Windows/Unix.
- Configure SilentODTUtility
  The topic briefs the steps to configure SilentODTUtility.
- Generation of Web service Artifacts through SilentOdtUtility
  The topic provides the instructions to generate Web service Artifacts through SilentOdtUtility

3.1 Prerequisites

This topic gives an overview on the prerequisites of ODT Silent Utility.

Before the run utility operation in ODT, 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 8 Update 321 version to be downloaded in the system and the same should be set as an environmental variable.

- **Apache Ant 2.0**
  Instructions: The installer requires ANT 2.0 version to be downloaded in the system and the same should be set as an environmental variable.

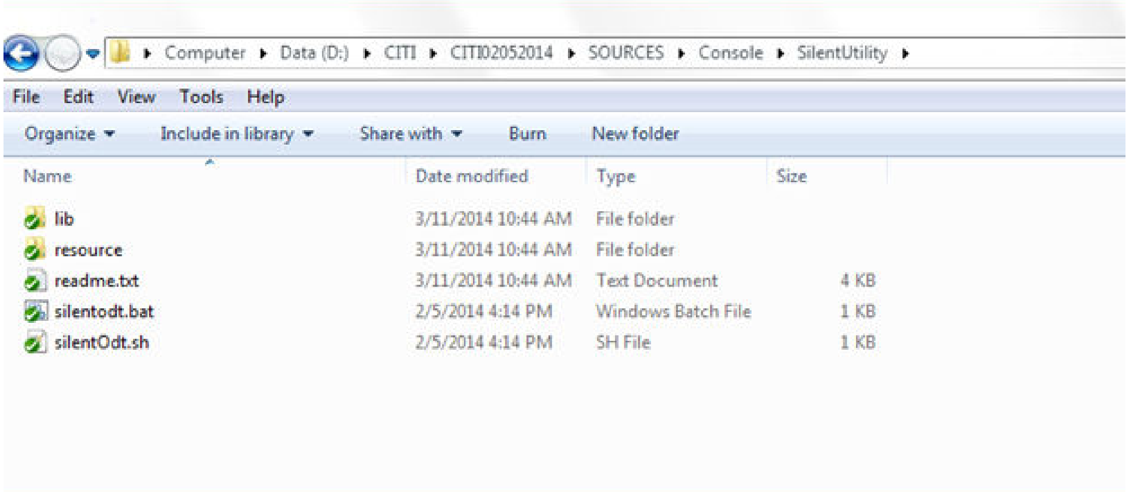
3.2 Run utility on Windows/Unix

The topic briefs on run utility operation in Windows/Unix.

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

The screenshot below shows how the source folder in your local system should look like.

Figure 3-1 Path of SilentODTUtility Source


Computer ▸ Data (D:) ▸ CITI ▸ CITI02052014 ▸ SOURCES ▸ Console ▸ SilentUtility ▸
File
Edit
View
Tools
Help
Organize ▾
Include in library ▾
Share with ▾
Burn
New folder

Name	Date modified	Type	Size
lib	3/11/2014 10:44 AM	File folder	
resource	3/11/2014 10:44 AM	File folder	
readme.txt	3/11/2014 10:44 AM	Text Document	4 KB
silentodt.bat	2/5/2014 4:14 PM	Windows Batch File	1 KB
silentOdt.sh	2/5/2014 4:14 PM	SH File	1 KB

3.3 Configure SilentODTUtility

The topic briefs the steps to configure SilentODTUtility.

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

> **Note:**
>
> - Copy jaxb-xjc.jar to 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.

This topic contains the following sub-topics:

- SilentOdt.properties
  This topic briefs on SilentOdt.properties.
- ODTOperations.properties
  This topic explains the ODTOperations.properties in detail.
- GW_CONFIG.properties
  This topic briefs GW_CONFIG.properties.

3.3.1 SilentOdt.properties

This topic briefs on SilentOdt.properties.

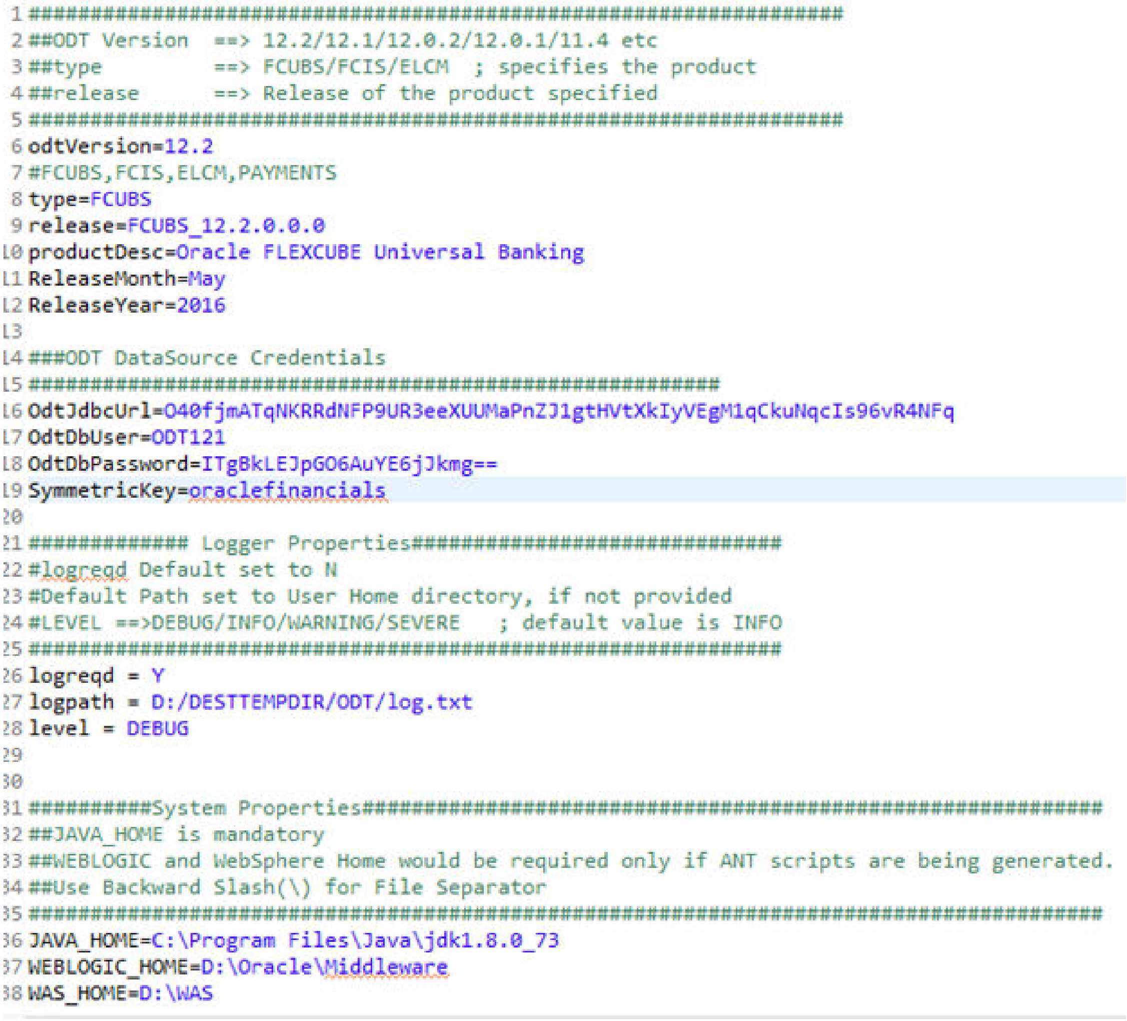
SilentOdt.properties

The sample property file has been given below.

Refer 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 symmetric key for encryption (Preferably Alphanumeric) which will be prompted for input from user when the encryption utility is lunched. The same symmetric key must be mentioned in the property file as well.

Figure 3-2   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
```

Refer to the ODT Dta Source Crendentials and Properties table for more details.

Table 3-1   ODT Dta Source Crendentials and Properties Details

ODT Dta Source	Crendentials and Properties
ODT Data Source Detail Credentials	This section displays the following fields.
OdtJdbcUrl	Jdbc Url jdbc:oracle:thin:@10.184.xx.xx:1521:FCDEMO
OdtDbUser	DB User name
OdtDbPassword	DP Password
Logger Properties	This section displays the following fields.
Logreqd	**Y/N**. Default set to **N**.
Logpath	Provide the path where the Logger files will be generated.

Table 3-1 (Cont.) ODT Dta Source Crendentials and Properties Details

ODT Dta Source	Crendentials and Properties
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	This section displays the following fields.
JAVA_HOME	Maintain the Java installed location
WEBLOGIC_HOME	Maintain the oracle weblogic installed location
WAS_HOME	Maintain the IBM websphere installed location

3.3.2 ODTOperations.properties

This topic explains the ODTOperations.properties in detail.

ODTOperations.properties

Configure the Operations files as per Requirement.

The Following operations are supported in silent utility of ODT:

- **LOGIN**
- **SETRELEASE**: Setting Release and Environment Details
- **BULKGENERATION**: Bulk Generation of Radxml units
- **REFRESH**: Bulk refresh of radxml
- **SXML_REFRESH**: Bulk refresh of Service Xmls
- **SXML_UPDATER**: Bulk Updater of service Xmls based on the changes in radxml’s
- **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= **REST_GENERATION**

If the sequence of operations is as above, then **LOGIN**, **SETRELEASE**, and **REST_GENERATION** Operations would be processed in the respective sequence.

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

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

For Example:

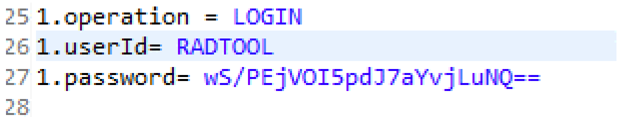
1. Operation = **LOGIN**
2. Operation= **SETRELEASE**
3. Operation= **REST_GENERATION**

If the sequence of operations is as above, then **LOGIN**, **SETRELEASE**, and **REST_GENERATION** Operations would be processed in the respective sequence.

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

1. **LOGIN**

Figure 3-3 SilentODT Login Properties



```
1.operation = LOGIN
1.userId= RADTOOL
1.password= wS/PEjVOI5pdJ7aYvjLuNQ==

```

Figure 3-4 SilentODT Login Properties

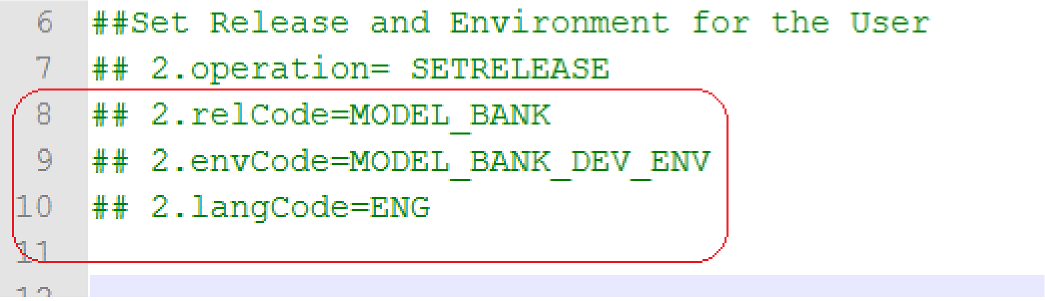
Table 3-2 Log in to Tool

SilentODT Login Properties	Description
Operation	LOGIN
User Id	Specify the ODT User Id which is created in the ODT Application
Password	Specify the ODT Password which is created in the ODT Application. Encrypted using ODTPassEncryption.bat. Refer section **Configure SilentODTUtility** for more details about encryption.

2. **SETRELEASE**

> **Note:**
>
> 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.

Figure 3-5   SilentODT Set Release Properties



```
##Set Release and Environment for the User
## 2.operation= SETRELEASE
## 2.relCode=MODEL_BANK
## 2.envCode=MODEL_BANK_DEV_ENV
## 2.langCode=ENG
```

Table 3-3   Set Release and Environment for User

SilentODT Set Release Properties	Description
Operation	SETRELEASE
relCode	Specify the ODT Release Code which is created in the ODT Application.
envCode	Specify the ODT Environment Code which is created in the ODT Application.
langCode	Specify the Lang code for the mentioned release code.

3. **BULKGENERATION**
   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 will be generated by the tool in this case.

   > **Note:**
   >
   > Either of radxml List File or srcPath should be present. If both are present, then radxmlListFile parameter will be considered for Bulk Generation.

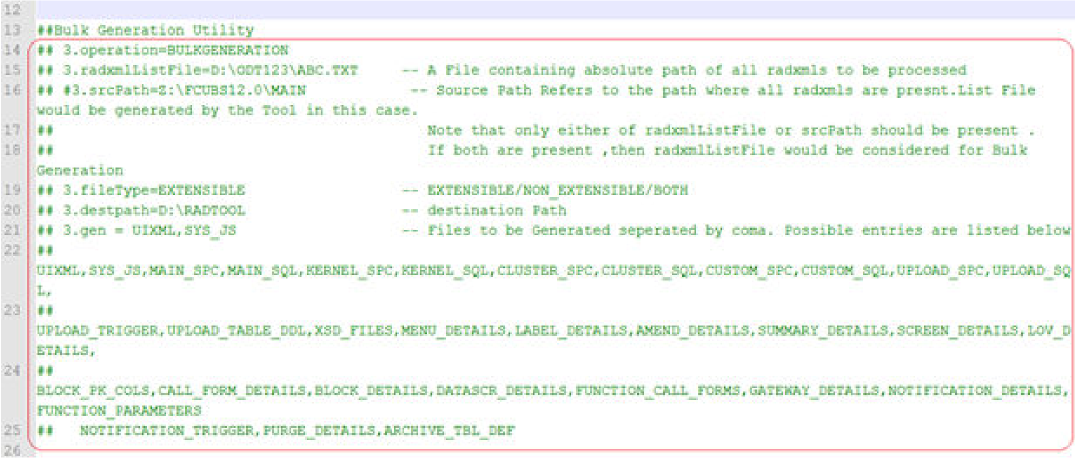
   fileType:

   - **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: Specify the path where the files will be generated.

   srcPath: Specify the type of files to be generated. For Example: **UIXML, SYS_JS, MAIN_SPC, MAIN_SQL,KERNEL_SPC, KERNEL_S**

Figure 3-6   SilentODT Bulk Generation Properties



```
##Bulk Generation Utility
## 3.operation=BULKGENERATION
## 3.radxmlListFile=D:\ODT123\ABC.TXT     -- A File containing absolute path of all radxmls to be processed
## #3.srcPath=Z:\FCUBS12.0\MAIN            -- Source Path Refers to the path where all radxmls are presnt.List File
would be generated by the Tool in this case.
##                                          Note that only either of radxmlListFile or srcPath should be present .
##                                          If both are present ,then radxmlListFile would be considered for Bulk
Generation
## 3.fileType=EXTENSIBLE                   -- EXTENSIBLE/NON_EXTENSIBLE/BOTH
## 3.destpath=D:\RADTOOL                   -- destination Path
## 3.gen = UIXML,SYS_JS                    -- Files to be Generated seperated by coma. Possible entries are listed below
##
UIXML,SYS_JS,MAIN_SPC,MAIN_SQL,KERNEL_SPC,KERNEL_SQL,CLUSTER_SPC,CLUSTER_SQL,CUSTOM_SPC,CUSTOM_SQL,UPLOAD_SPC,UPLOAD_SQ
L,
##
UPLOAD_TRIGGER,UPLOAD_TABLE_DDL,XSD_FILES,MENU_DETAILS,LABEL_DETAILS,AMEND_DETAILS,SUMMARY_DETAILS,SCREEN_DETAILS,LOV_D
ETAILS,
##
BLOCK_PK_COLS,CALL_FORM_DETAILS,BLOCK_DETAILS,DATASCR_DETAILS,FUNCTION_CALL_FORMS,GATEWAY_DETAILS,NOTIFICATION_DETAILS,
FUNCTION_PARAMETERS
##    NOTIFICATION_TRIGGER,PURGE_DETAILS,ARCHIVE_TBL_DEF
```

Table 3-4   Bulk Generation Utility

Bulk Generation Utility Input	Bulk Generation Utility Output
radxmlListFile srcPath fileType gen destpath	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 NOTIFICATION_TRIGGER,PURGE_DETAILS,ARCHIVE_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 done using the Tool.(Please refer the 09-Development_WorkBench_Source_Upgrade.docx).

   - Child Refresh
   - Screen Child Refresh
   - 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.

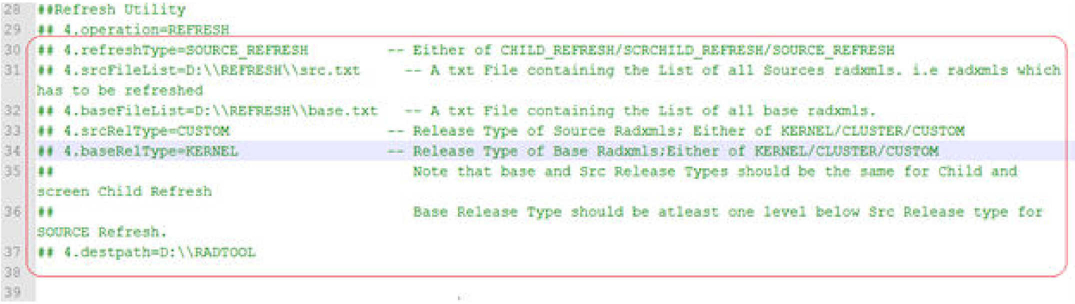
   baseFileList: A txt File containing the List of all base radxml’s.

   srcRelType: Specify the release type of Source Radxmls list(KERNEL/CLUSTER/CUSTOM).

   baseRelType: Specify the release type of base Radxmls list (KERNEL/CLUSTER/CUSTOM).

   destpath: Specify the path where the files will be generated.

Figure 3-7   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 3-5   Refresh Utility

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

5. **SXML_BULKGENERATION**
   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: Specify the source folder path which is optional (Tool will create radxmlListFile and xsdListFile by itself from the srcPath).

   > **Note:**
   >
   > If srcPath is provided, radxmlListFile and xsdListFile need not be provided.

   gen: Specify the type of files to be generated (separated by coma). The options available are:

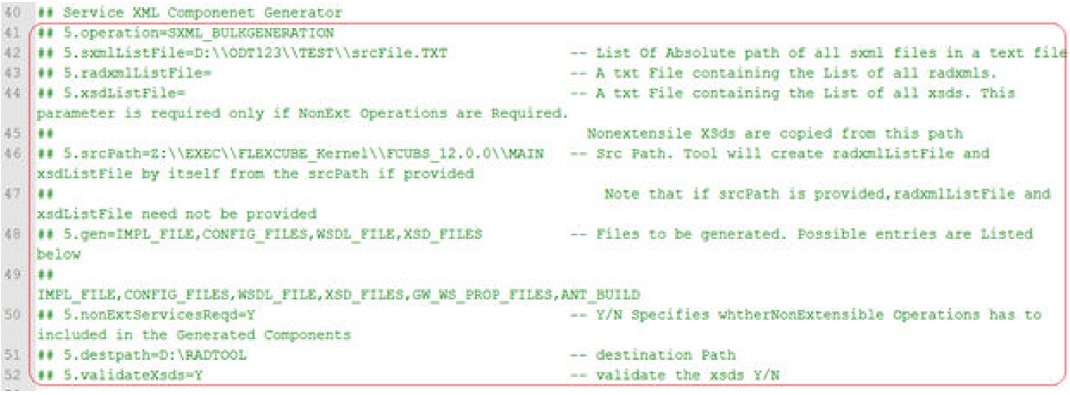
   - IMPL_FILE
   - CONFIG_FILES
   - WSDL_FILE,XSD_FILES
   - GW_WS_PROP_FILES
   - ANT_BUILD

   nonExtServicesReqd: Specify as Y/N whether Non-Extensible operations have to be included in the generated Components.

   destpath: Specify the path where the files will be generated.

validateXsds: Specify as Y/N and the default value is `Y`. If the value is set to `Y`, all XSD will be validated by the tool.

Figure 3-8   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 3-6   SilentODT Service XML Bulk Generation Properties

Service XML Component Generator Input	Service XML Component Generator Output
sxmlListFile radxmlListFile xsdListFile srcPath baseRelType destpath validateXsds	IMPL_FILE,CONFIG_FILES,WSDL_FILE,XSD_FILES,GW_WS_PROP_FILES,ANT_BUILD in destpath

6. **SXML_UPDATER**
   This feature can be used to update the Service XMLs with the latest data from Radxmls. Following details will be updated. Any addition, deletion or modification of operation codes in function Id would be updated in Service XML.

   - If any function Id is removed from the service (specified in radxml); then the same would be removed from Service XML.
   - 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 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. Specify the same file path.

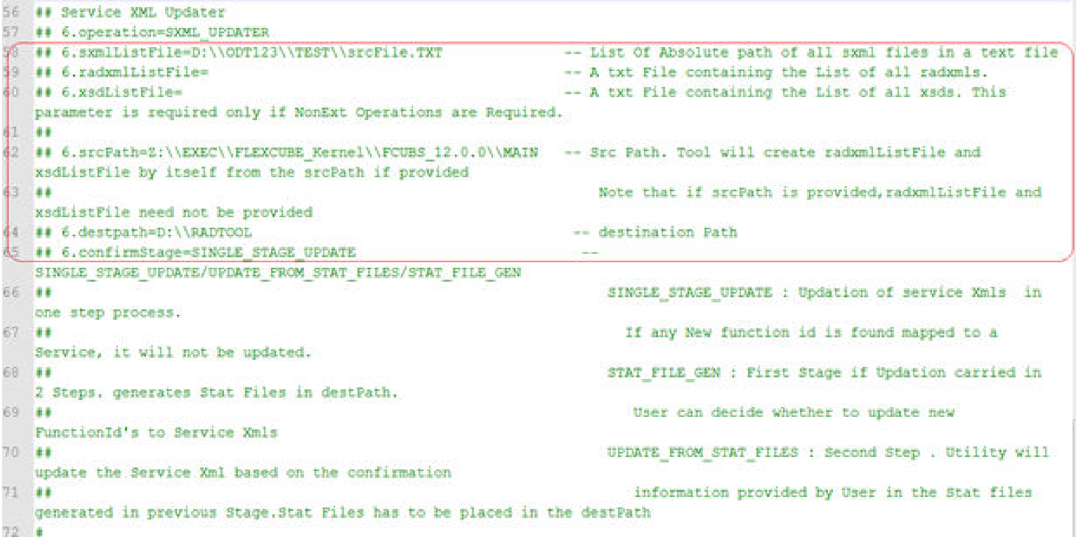
   srcPath: Specify the source folder path which is optional (Tool will create radxmlListFile and SxmlListFile by itself from the srcPath).

> **Note:**
>
> If srcPath is provided, radxmlListFile and sxmlListFile need not be provided.

destpath: Specify the path where the files will be generated.

confirmStage: SINGLE_STAGE_UPDATE(Default Value should not be modified by developer).

Figure 3-9 SilentODT Service XML Updater Properties



```
## Service XML Updater
## 6.operation=SXML_UPDATER
## 6.sxmlListFile=D:\\ODT123\\TEST\\srcFile.TXT              -- List Of Absolute path of all sxml files in a text file
## 6.radxmlListFile=                                        -- A txt File containing the List of all radxmls.
## 6.xsdListFile=                                           -- A txt File containing the List of all xsds. This
parameter is required only if NonExt Operations are Required.
##
## 6.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
## 6.destpath=D:\\RADTOOL                                     -- destination Path
## 6.confirmStage=SINGLE_STAGE_UPDATE                          --
SINGLE_STAGE_UPDATE/UPDATE_FROM_STAT_FILES/STAT_FILE_GEN
##                                                              SINGLE_STAGE_UPDATE : Updation of service Xmls  in
one step process.
##                                                                If any New function id is found mapped to a
Service, it will not be updated.
##                                                              STAT_FILE_GEN : First Stage if Updation carried in
2 Steps. generates Stat Files in destPath.
##                                                                 User can decide whether to update new
FunctionId's to Service Xmls
##                                                              UPDATE_FROM_STAT_FILES : Second Step . Utility will
update the Service Xml based on the confirmation
##                                                                 information provided by User in the Stat files
generated in previous Stage.Stat Files has to be placed in the destPath
#
```

Table 3-7 SilentODT Service XML Updater Properties

Service XML Updater Input	Service XML Updater Output
sxmlListFile radxmlListFile xsdListFile srcPath destpath confirmStage	updated Service XML's in destpath

7. **SXML_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.

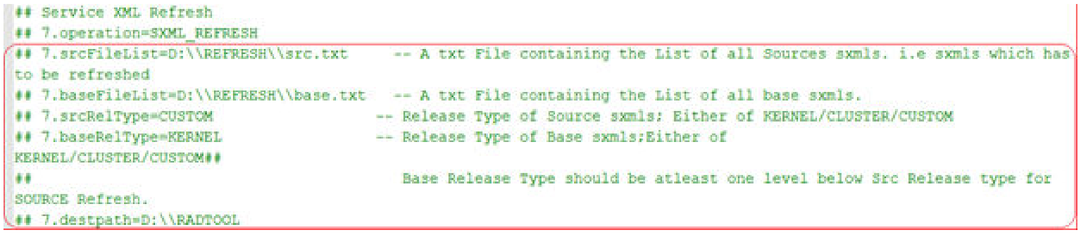
> **Note:**
>
> For instance, for a custom development team; all the latest Kernel files has to be mentioned in baseFileList; while the custom Service XMLs to be refreshed has to be mentioned in the srcFileList.

srcRelType: Specify the release type of Source Radxmls list(KERNEL/CLUSTER/CUSTOM).

baseRelType: Specify the release type of base Radxmls list (KERNEL/CLUSTER/CUSTOM). The baseRelType should be at least one level below srcRelType for SOURCE Refresh.

destpath: Specify the path where the files will be generated.

Figure 3-10   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 3-8   SilentODT Service XML Refresh Properties

Service XML Refresh Input	Service XML Refresh Output
operation srcFileList baseFileList srcRelType baseRelType destpath	Refreshed Service XML’s in destpath

3.3.3 GW_CONFIG.properties

This topic briefs GW_CONFIG.properties.

GW_CONFIG.properties

The parameters of this property files are used to generate gateway web service property files. This configuration file is optional. Specify only if Gateway Web service property files (GW_WS_PROP_FILES) is being generated as part of Service XML Bulk generation operation.

EJB_APP_SERVER: Specify the application server name in which ear is deployed.

EJB_JNDI_NAME: EJB JNDI Name is the reference name of the ejb by which the ejb has been deployed.

EJB_SERVER_URL: Specify the application server IP Address & port where the EJB application is deployed.

EJB_SERVER_USERNAME: Specify the user name of the application server where the EJB application is deployed.

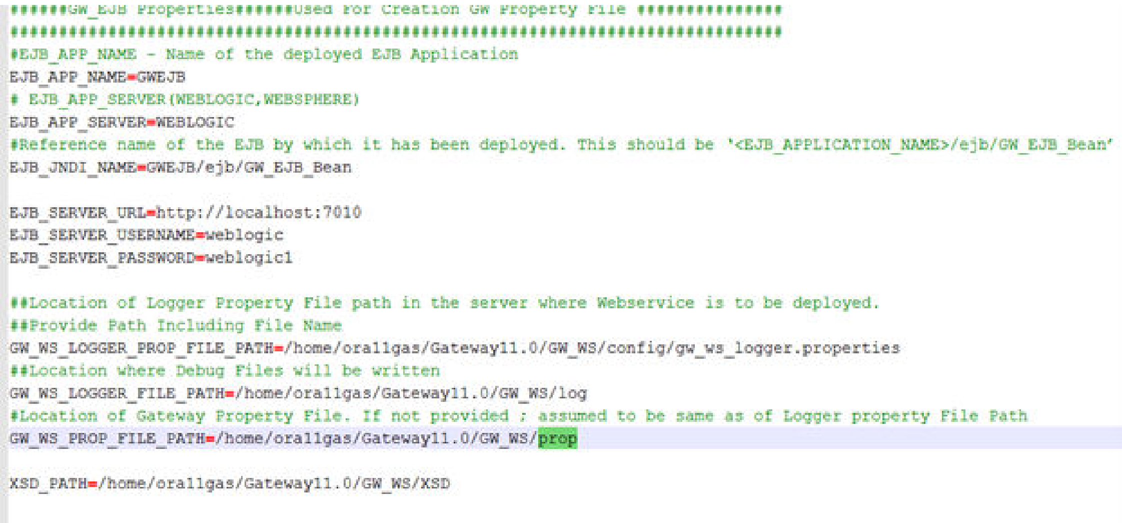
EJB_SERVER_PASSWORD: Specify the password of the application server where the EJB application is deployed.

GW_WS_LOGGER_PROP_FILE_PATH: Specify the location of Logger Property File in the server where web service is to be deployed. Specify the path including the file name. For 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: Specify the location where debug files will be written.

GW_WS_PROP_FILE_PATH: Specify the path where property file is placed in the server. This will be referred in web.xml of web service property file. For Example: D:/Kernel11.1/GW_WS/prop/ (Windows) /oraint1/kernel/FC120INS_DEBUG/Gateway/GWWS/prop (Linux or UNIX).

Figure 3-11 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
```

Table 3-9 SilentODT GW Config Properties

SilentODT GW Config Properties	Description
EJB_APP_NAME	Specify the name of the deployed EJB Application.
EJB_APP_SERVER	Specify the name of the deployed EJB Application.
EJB_JNDI_NAME	Specify the EJB JNDI Name is the reference name of the ejb by which the ejb has been deployed.
EJB_SERVER_URL	Specify the Application server IP Address & port where the EJB application is deployed.
EJB_SERVER_USERNAME	Specify the User Name of the application server where the EJB application is deployed.
EJB_SERVER_PASSWORD	Specify the Password of the application server where the EJB application is deployed.

Table 3-9 (Cont.) SilentODT GW Config Properties

SilentODT GW Config Properties	Description
GW_WS_LOGGER_PROP_FILE_PATH	Specify 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	Specify the location where debug files will be written. For Example: D:/Kernel11.1/GW_WS/log/ (Windows) /oraint1/kernel/FC120INS_DEBUG/Gateway/GWWS/log (Linux or Unix).
GW_WS_PROP_FILE_PATH	Specify the path where property file is placed in the server. This will be referred in web.xml of web service property file. **For Example**: D:/Kernel11.1/GW_WS/prop/ (Windows) /oraint1/kernel/FC120INS_DEBUG/Gateway/GWWS/prop (Linux or UNIX).

3.4 Generation of Web service Artifacts through SilentOdtUtility

The topic provides the instructions to generate 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.

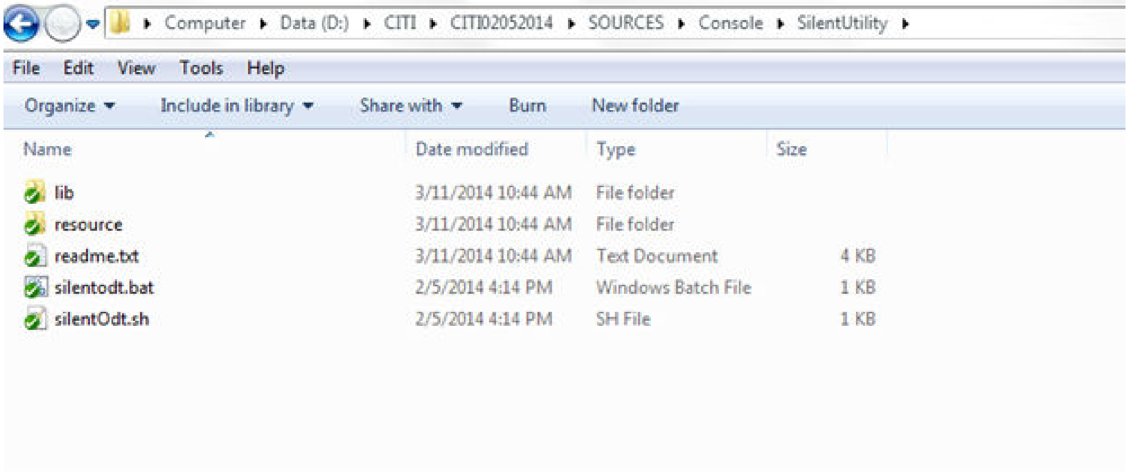
The following operations needs to be configured in OdtOperations.properties in respective sequence.

- **LOGIN**
- **SETRELEASE**
- **SXML_BULKGENERATION**

Specify the 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.

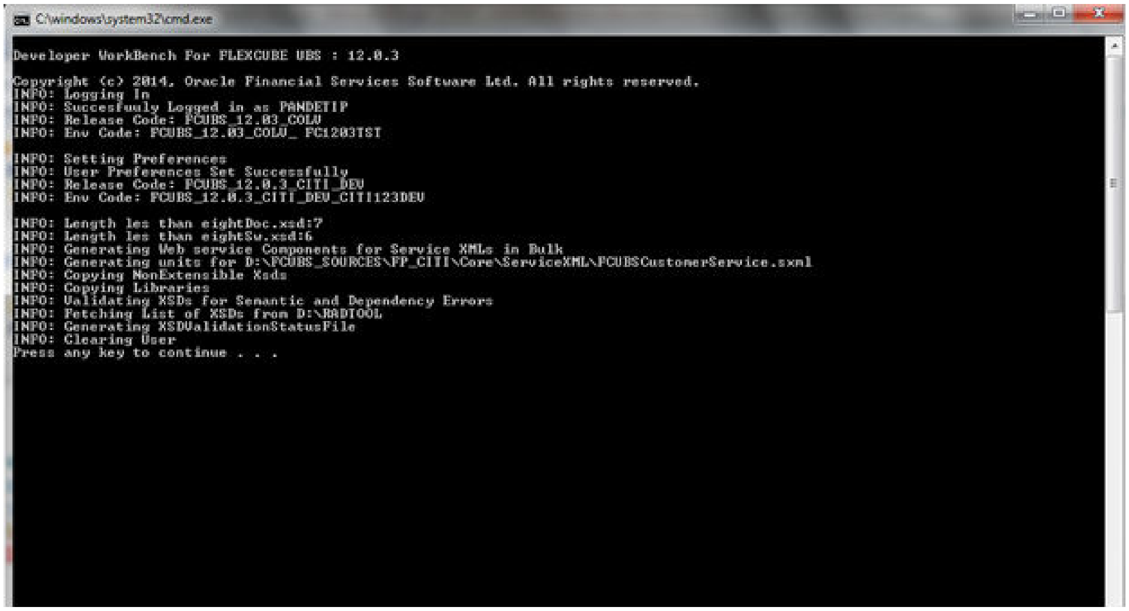
Figure 3-12   SilentODT Sources



Computer ▸ Data (D:) ▸ CITI ▸ CITI02052014 ▸ SOURCES ▸ Console ▸ SilentUtility ▸
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ Burn New folder

Name	Date modified	Type	Size
lib	3/11/2014 10:44 AM	File folder	
resource	3/11/2014 10:44 AM	File folder	
readme.txt	3/11/2014 10:44 AM	Text Document	4 KB
silentodt.bat	2/5/2014 4:14 PM	Windows Batch File	1 KB
silentOdt.sh	2/5/2014 4:14 PM	SH File	1 KB

Figure 3-13   SilentODT Running in Command Prompt



```
Developer WorkBench For FLEXCUBE UBS : 12.0.3

Copyright (c) 2014, Oracle Financial Services Software Ltd. All rights reserved.
INFO: Logging In
INFO: Succesfuuly Logged in as PANDETIP
INFO: Release Code: FCUBS_12.03_COLU
INFO: Env Code: FCUBS_12.03_COLU_ FC1203TST

INFO: Setting Preferences
INFO: User Preferences Set Successfully
INFO: Release Code: FCUBS_12.0.3_CITI_DEV
INFO: Env Code: FCUBS_12.0.3_CITI_DEV_CITI123DEV

INFO: Length les than eightDoc.xsd:7
INFO: Length les than eightSw.xsd:6
INFO: Generating Web service Components for Service XMLs in Bulk
INFO: Generating units for D:\FCUBS_SOURCES\FP_CITI\Core\ServiceXML\FCUBSCustomerService.sxml
INFO: Copying NonExtensible Xsds
INFO: Copying Libraries
INFO: Validating XSDs for Semantic and Dependency Errors
INFO: Fetching List of XSDs from D:\RADTOOL
INFO: Generating XSDValidationStatusFile
INFO: Clearing User
Press any key to continue . . .
```

The following service artifacts files will be generated in the destination directory specified after successful built operation.

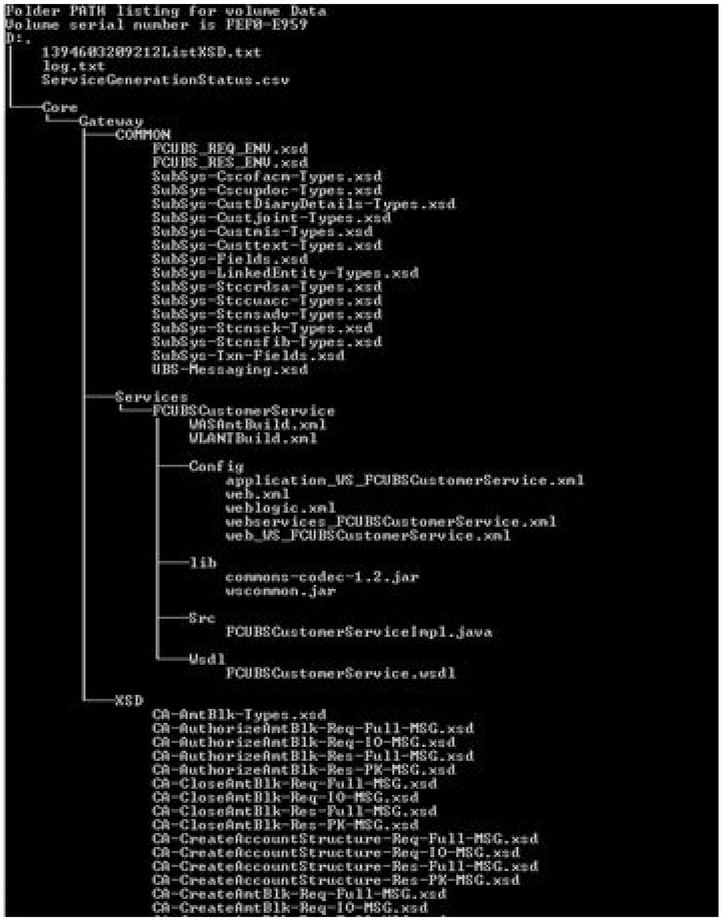
Table 3-10   Files and Description

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

Table 3-10 (Cont.) Files and Description

Files	Description
<Service Name>Common*.xsd's	Common XSD's (call forms) part of service
<Service Name>\<Service Name>\METAINF\application.xml <Service Name>\<Service Name>\METAINF\MANIFEST.MF	Config XML's for building the Web service
<Service Name>\<Service Name>\commonscodec-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 3-14 SilentODT Generated Files Tree 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_ENV.xsd
        │       FCUBS_RES_ENV.xsd
        │       SubSys-Cscofacn-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-Stcnsadv-Types.xsd
        │       SubSys-Stcnsck-Types.xsd
        │       SubSys-Stcnsfib-Types.xsd
        │       SubSys-Txn-Fields.xsd
        │       UBS-Messaging.xsd
        │
        ├───Services
        │   └───FCUBSCustomerService
        │       │   WASAntBuild.xml
        │       │   WLANTBuild.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
```

- Log Files
  This topic gives an overview on Log Files.

- Ant Build Scripts
  This topic gives an overview on Ant Build Scripts.
- Gateway Property Files
  This topic gives an overview on Gateway Property files.

3.4.1 Log Files

This topic gives an overview on Log Files.

The following files are being generated:

- ServiceGenerationStatus.csv
- XSDValidationErrors.csv
- Utility Log File

ServiceGenerationStatus.csv

The generation status will be saved in the above mentioned file. This will be generated in the destination path.

XSDValidationErrors.csv

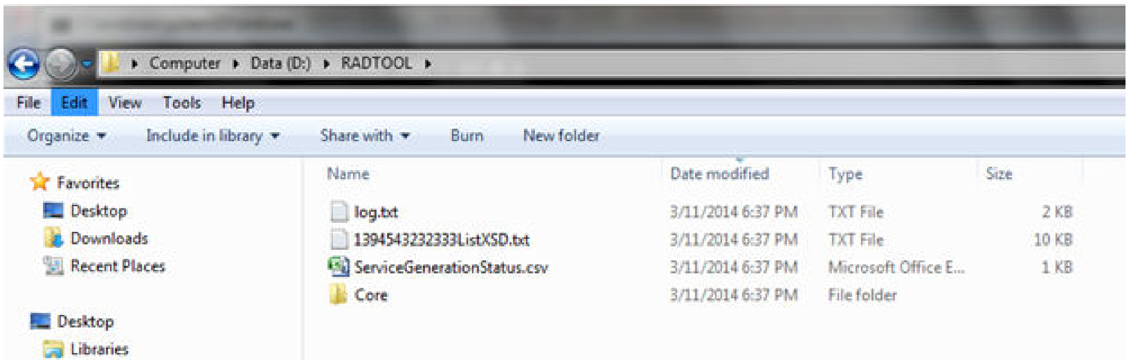
XSD Validation errors, if any, will be saved in above mentioned file. This will be generated in the destination path.

Utility Log File

The Log File of the utility will be generated in the path configured in SilentOdt.properties. This can be used in case of any troubleshooting.

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

Figure 3-15 SilentODT Generated log Files



Name	Date modified	Type	Size
log.txt	3/11/2014 6:37 PM	TXT File	2 KB
1394543232333ListXSD.txt	3/11/2014 6:37 PM	TXT File	10 KB
ServiceGenerationStatus.csv	3/11/2014 6:37 PM	Microsoft Office E...	1 KB
Core	3/11/2014 6:37 PM	File folder	

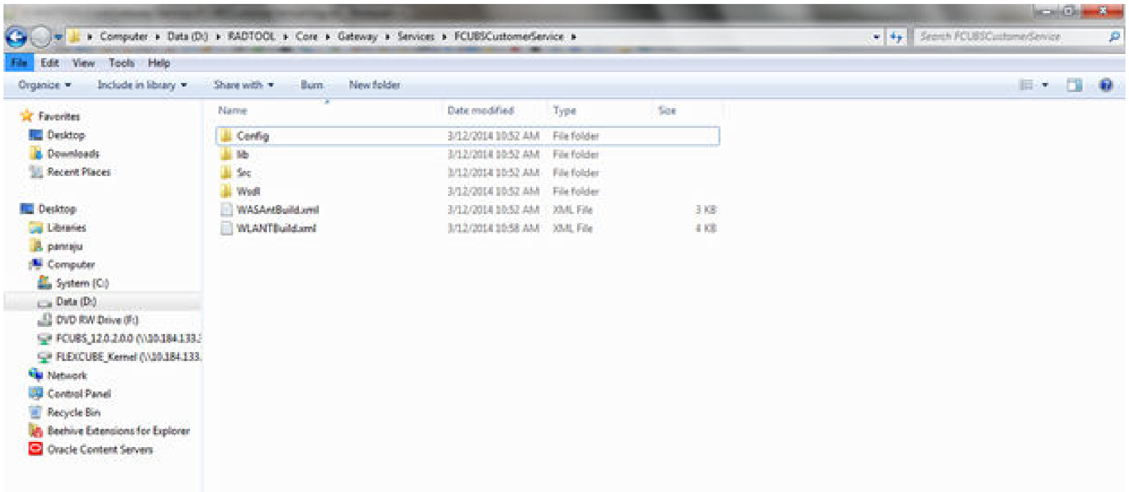
3.4.2 Ant Build Scripts

This topic gives an overview on Ant Build Scripts.

Tool will generate the sample ant scripts for weblogic and websphere 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 weblogic server

Figure 3-16   SilentODT Generated Files Ant Build Files



Name	Date modified	Type	Size
Config	3/12/2014 10:52 AM	File folder	
lib	3/12/2014 10:52 AM	File folder	
Src	3/12/2014 10:52 AM	File folder	
Wsdl	3/12/2014 10:52 AM	File folder	
WASAntBuild.xml	3/12/2014 10:52 AM	XML File	3 KB
WLANTBuild.xml	3/12/2014 10:58 AM	XML File	4 KB

3.4.3 Gateway Property Files

This topic gives an overview on Gateway Property files.

Gateway property files will be generated in GW_WS folder inside the destination path. The following files will be generated.

- gw_ws_logger.properties
- GW_WS_Prop.properties