Oracle Banking APIs

File Upload Report Configuration Guide
Release 19.1.0.0.0

Part No. F18559-01

May 2019

ORACLE®

File Upload Report Configuration Guide

May 2019

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001

www.oracle.com/financialservices/

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

1. Preface **4**

2. File Uploads **5**

2. Reports **14**

1. Preface

1.1 Intended Audience

This document is intended for the following audience:

- Customers
- Partners

1.2 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.

1.3 Access to Oracle Support

Oracle customers 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.

1.4 Structure

This manual is organized into the following categories:

Preface gives information on the intended audience. It also describes the overall structure of the User Manual.

Introduction provides brief information on the overall functionality covered in the User Manual.

The subsequent chapters provide information on transactions covered in the User Manual.

Each transaction is explained in the following manner:

- Introduction to the transaction
- Screenshots of the transaction
- The images of screens used in this user manual are for illustrative purpose only, to provide improved understanding of the functionality; actual screens that appear in the application may vary based on selected browser, theme, and mobile devices.
- Procedure containing steps to complete the transaction- The mandatory and conditional fields of the transaction are explained in the procedure.

If a transaction contains multiple procedures, each procedure is explained. If some functionality is present in many transactions, this functionality is explained separately.

1.5 Related Information Sources

For more information on Oracle Banking APIs Release 19.1.0.0.0, refer to the following documents:

- Oracle Banking APIs Licensing Guide
- Oracle Banking APIs e Installation Manuals

2. File Uploads

(i) OutsideIn (For MS Excel processing)

Outside Inn - This is used for parsing XLS, XLSX in file uploads module. This library is not shipped with OBAPI but needs to be downloaded from below link for required platform (OS on which app server is running)

http://www.oracle.com/technetwork/middleware/webcenter/content/oit-dl-otn-097435.html

Search Export – (Refer Pre requisite installation document for version)

Unzip the downloaded file and copy all contents of 'redist' folder to config/outsidein/<os> directory

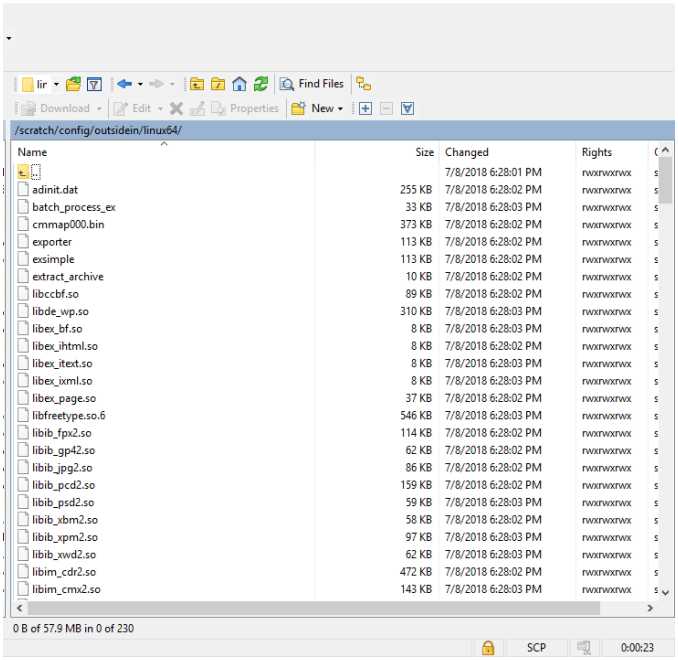
Then copy all contents (except jar & sh files) of 'sdk/demo' directory to config/outsidein/<os>

Use sx.cfg (replace/merge contents if required) shipped in installer from folder config/outsidein/<os>

Confirm/update path → select * from digx_fw_config_all_b where prop_id = 'OUTSIDE_IN_SDK'

Default config/outsidein/linux64

Grant 777 privileges for OutsideIn directory

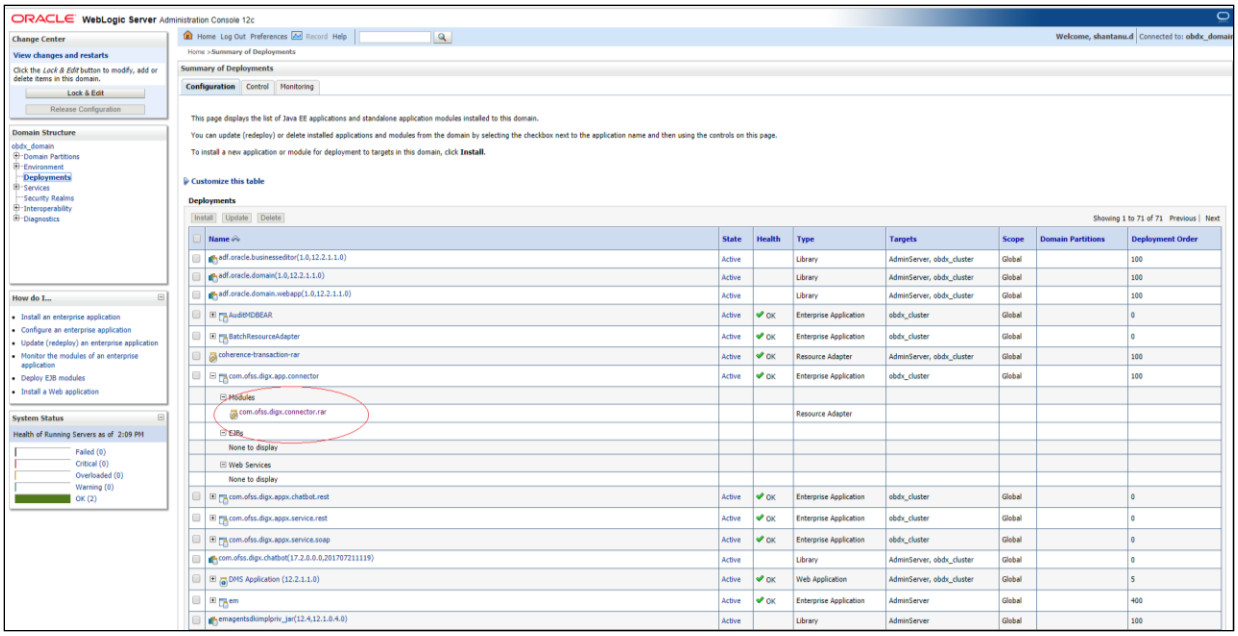


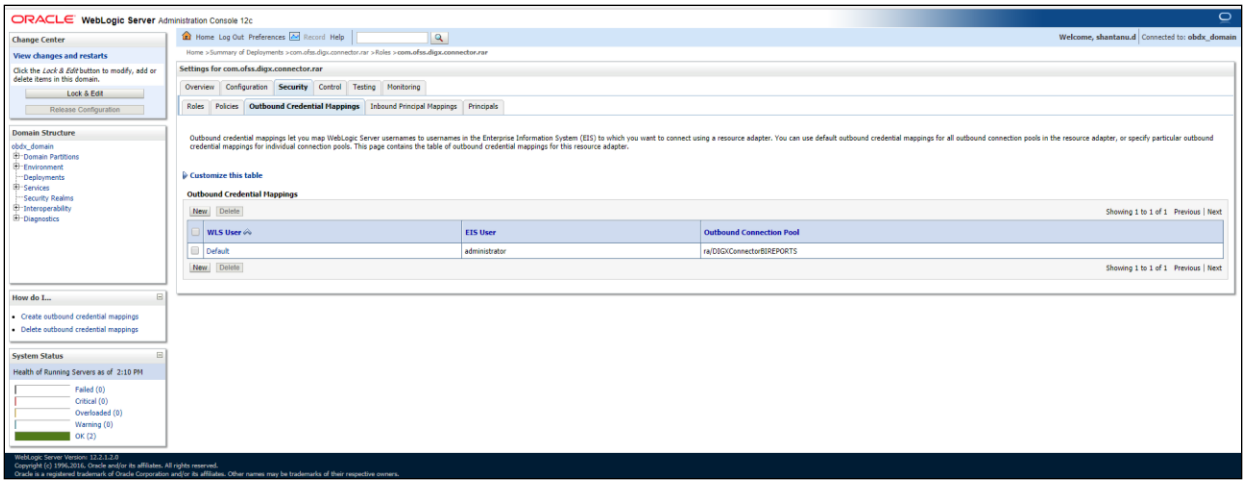
(ii) Configuration for storing key for decrypting uploaded files and creating encrypted response files

The key used for file decryption by default decryptor is stored in database in digx_fw_config_all_b with prop_id as 'ENCRYPTION_KEY'. If this is to be stored in WLS connector update the property as below

update digx_fw_config_all_b set prop_value='KEY_STORE' where prop_id='ENCRYPTION_KEY_LOCATION';

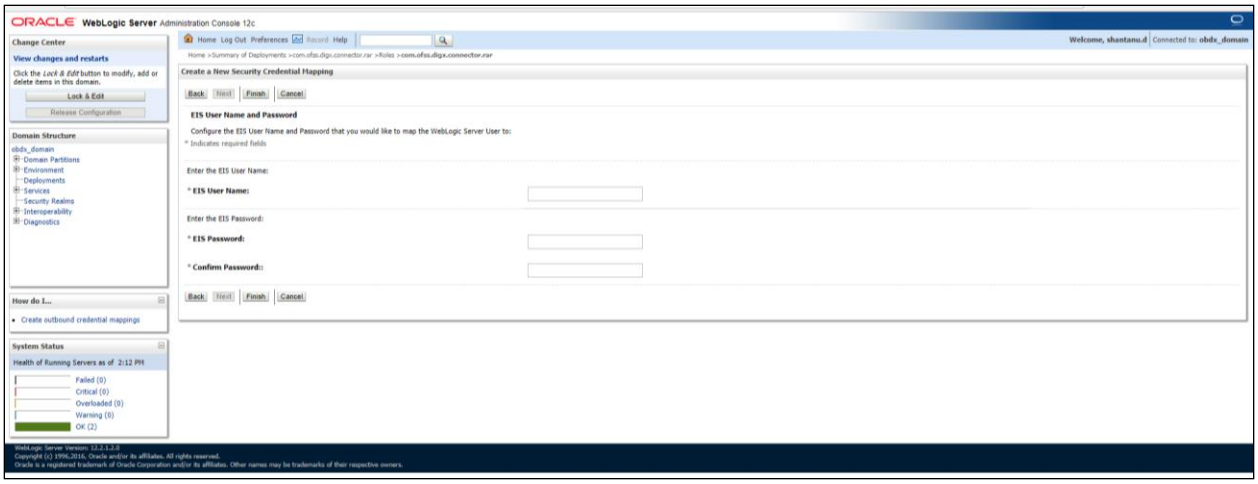
Update the encryption key in connector as below –





Click New > Select ra/DIGXConnectorFILEUPLOAD > Next > Select Default User

In password field, enter the encryption key


ORACLE WebLogic Server Administration Console 12c
Change Center
View changes and restarts
Click the Lock & Edit button to modify, add or delete items in this domain.
Lock & Edit
Release Configuration
Domain Structure
obdx_domain
Domain Partitions
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Create outbound credential mappings
System Status
Health of Running Servers as of 2:12 PM
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (2)
Home Log Out Preferences Record Help
Welcome, shantanu.d Connected to: obdx_domain
Create a New Security Credential Mapping
Back Next Finish Cancel
EIS User Name and Password
Configure the EIS User Name and Password that you would like to map the WebLogic Server User to:
* Indicates required fields
Enter the EIS User Name:
* EIS User Name:
Enter the EIS Password:
* EIS Password:
* Confirm Password:
Back Next Finish Cancel

(iii) Using Enrichers in File Uploads

(For custom defined templates only, not required for out of box templates)

- Enrichers are used to enrich or fetch a value for a given field. Let's say the field is Debit Account Id and enricher is Account Currency, so it means that the currency for that debit account Id needs to be fetched or enriched.
- Enricher can have enricher arguments. These arguments are passed when the enricher is invoked.
- Enrichers are of 2 types
    - Upload File Enrichers
    - Static arguments (enricherArgs) – Value is passed directly from template to enricher as label string
    - Dynamic arguments (enricherDynArgs) – Value is derived from a previous field of the record.
- Extract (Response) File Enrichers

How Enrichers are used in File Upload ?

- In File Upload XML template, the field **which will** enrich other fields must have 'enricher' attribute. This attribute **must not be specified for the fields which would be** enriched.
- The value of this 'enricher' attribute is the 'ENRICHMENT_ID' which is a column in table 'DIGX_FW_ENRICHMENTS_B'. Currently OBAPI support only Java enrichers. Enrichers can be in any package but must implement the 'IEnrichment' interface.
- On the basis of the 'enricher' attribute value mapping is done from table 'DIGX_FW_ENRICHMENTS_B' and the corresponding 'ENRICHMENT_VALUE' column value is fetched and enrich() method of the specified Java class is invoked
  Eg.
- Refer to the following figure of File Template : InternalFT.xml .

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FileDefinition fileName="InternalFT"
    fileHandlerClassName="com.ofss.digx.app.fileupload.handlers.InternalFTFileHandler"
    decryptionClass="" charSet="UTF-8" delimiter="," comments=""
    isFirstRecHeader="false" simpleOrMixed="M" fillchar="" partialProcessing="100" transactionType="ITG">

    <RecordDefinition
        recordHandlerClassName="com.ofss.digx.app.fileupload.handlers.InternalFTRecHandler"
        recordType="B"
        dtoClassName="com.ofss.digx.domain.fileupload.entity.InternalFTDTO"
        multiplicity="-1" maxFields="10" comments=""
        parent="" length="" transaction="ITG"
        mixedIdentifier="A">
        <Field name="mixedIdentifier"/>
        <Field name="partyId"/>
        <Field name="debitAccountId" enricher="ACCTCURR" enricherArgs=""/>
        <Field name="amount" type="CD"/>
        <Field name="amountCurr"/>
        <Field name="valueDate" enricher="DATE" enricherArgs="dd-MM-yyyy"/>
        <Field name="creditAccountId" enricher="ACCTDETAILS"/>
        <Field name="debitNarrative"/>
        <Field name="creditNarrative"/>
        <Field name="purpose"/>
    </RecordDefinition>

    <RecordDefinition
        recordHandlerClassName="com.ofss.digx.app.fileupload.handlers.InternalFTRecHandler"
        recordType="B"
        dtoClassName="com.ofss.digx.domain.fileupload.entity.InternalFTBeneDTO"
        multiplicity="-1" maxFields="10" comments=""
        parent="" length="" transaction="ITGBEN"
        mixedIdentifier="B">
        <Field name="mixedIdentifier"/>
        <Field name="partyId"/>
        <Field name="debitAccountId" enricher="ACCTCURR" enricherArgs=""/>
        <Field name="amount" type="CD"/>
        <Field name="amountCurr"/>
        <Field name="valueDate" enricher="DATE" enricherArgs="dd-MM-yyyy"/>
        <Field name="beneId" enricher="BENE" enricherArgs="INTERNAL"/>
        <Field name="debitNarrative"/>
        <Field name="creditNarrative"/>
        <Field name="purpose"/>
    </RecordDefinition>
</FileDefinition>
```

Static Enrichers

- In above template, the field name ‘debitAccountId’ has a enricher ‘ACCTCURR’ with no enricherArgs. In this case ‘DIGX_FW_ENRICHMENTS_B’ will be queried and search for ‘ACCTCURR’ and ‘AccountCurrencyEnricher’ class is invoked.

  This enricher derives the debitAccountCurr. Hence this attribute must be present in the record DTO with its setters defined.

```
@Override
public HashMap<String, Object> enrich(HashMap<String, Object> parameters) throws Exception {

    SessionContext sessionContext = (SessionContext) ThreadAttribute.get(ThreadAttribute.SESSION_CONTEXT);
    FileUploadPolicyHelper policyHelper = FileUploadPolicyHelper.getInstance();
    policyHelper.fetchAccountId(sessionContext, new Account(parameters.get("value").toString()),
            parameters.get("fileRefId").toString());
    HashMap<String, Object> fields = new HashMap<String, Object>();
    String curr = policyHelper.fetchCurrencyForAccount(new Account(parameters.get("value").toString()),
            parameters.get("fileRefId").toString());
    fields.put("debitAccountCurr", curr == null ? "" : curr);
    fields.put("debitAccountId", parameters.get("value"));
    return fields;
}
```

- The field name ‘valueDate’ has static enricherArgs ‘dd-MM-yyyy’ meaning that the date has to be specifically in ‘dd-MM-yyyy’ format. This value is simply available to the enricher for processing purpose. This enricher does not add any new field but simply modifies the value of the current field.

```
@Override
public HashMap<String, Object> enrich(HashMap<String, Object> parameters) throws Exception {
    DateFormat df = new SimpleDateFormat(parameters.get("enricherArgs").toString());
    Date date = null;
    HashMap<String, Object> fields = new HashMap<String, Object>();
    try {
        df.setLenient(false);
        date = df.parse(parameters.get("value").toString());
        fields.put(parameters.get("field").toString(), new com.ofss.fc.datatype.Date(date));
    } catch (ParseException el) {
        Exception e = new Exception();
        e.setErrorCode(UploadErrorConstants.FU_INVALID_VALUE_DATE);
        throw e;
    }
    return fields;
```

Dynamic Enrichers

If ‘enricherDynArgs’ is specified

Eg. enricherDynArgs=”beneId~beneName” on beneficiary address field, the parser simply invokes getters on beneId and beneName fields and passes the values to the enricher in a map. It should be noted that these fields must be defined previously/above the beneficiary address field, so that parser has already completed the setter operation.

<Field name=" beneId"/>

<Field name=" beneName "/>

<Field name="beneAddr" enricher="ADDRESSENRICHER" enricherDynArgs=" beneId~beneName "/>

Eg.

Extract (Response) File Enrichers

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FileDefinition fileType="csv" delimiter="," handler="com.ofss.digx.framework.fileupload.extract.CSVHandler" encryptionClass="">

   <RecordDefinition recordType="H">
      <Field name="record" label="RECORD" />
      <Field name="recRefId" label="RECORD REF NO" />
      <Field name="fileRefId" label="FILE REF NO" />
      <Field name="digxRefId" label="E-BANKING REF NO" />
      <Field name="contractRefId" label="CONTRACT REF NO" />
      <Field name="recStatus" label="RECORD STATUS" />
      <Field name="errCode" label="STATUS CODE" />
      <Field name="errMsg" label="STATUS DESCRIPTION" />
   </RecordDefinition>

   <RecordDefinition  query="ResponseList" recordType="B">
      <Field name="record" no="1" wrapchar=""/>
      <Field name="recRefId" no="2"/>
      <Field name="fileRefId" no="3" />
      <Field name="digxRefId" no="4" />
      <Field name="contractRefId" no="5" />
      <Field name="recStatus" no="6" />
      <Field name="errCode" no="7"  enricher="ERRORMSG" enricherArgs=""/>
      <Field name="errMsg" no="8" />
   </RecordDefinition>

</FileDefinition>
```

Enrichers can be added to response file templates. The enricher class is invoked in the same way as upload templates. Eg, in above case, localized error message need to be added to extracts from ‘errCode’. Extract enrichers do not support dynamic arguments

File Copy Configuration

In case of FCR/OBPM as host, for **file level** uploads in OBAPI, the files are generated in FCR/OBPM formats after approval at OBAPI end is complete. These files are stored in a directory on OBAPI server. For record level, service is used same as of single screen transactions.

FCR configs

1. Set this path as the value for prop_id = *‘FCORE_HANDOFF_FILE_PATH’* in the *DIGX_FW_CONFIG_VAR_B* table against the required entity (Empty folder with full permission).
2. ***Copying the file to host system using FTP (to rjsin folder)***

Provide the values for the below properties in the MSTPROPERTIES table of host schema:

FU_IPADDRESS	*IP of FCR machine*
FU_FTPFILEPATH	*Filepath of rjsin where FCR will poll and pick files for further processing*
FU_USERNAME	*FTP username of FCR machine.* *Needs to encrypted using AES key as in connecter.*

FU_PASSWORD	*FTP password of FCR machine.* *Needs to encrypted using AES key as in connecter.*

OBPM configs

1. Set this path as the value for prop_id = *'UBS_HANDOFF_FILE_PATH'* in the *DIGX_FW_CONFIG_VAR_B* table against the required entity (Empty folder with full permission).

2. Copying the file to host system using FTP

Provide the values for the below properties in the MSTPROPERTIES table of host schema:

FU_IPADDRESS	*IP of OBPM machine*
FU_FTPFILEPATH	*Filepath of folder where files need to be copied on OBPM machine. Files will be picked by invoking OBPM restful service(from OB OBAPI DX adapter) with configs given below*
FU_USERNAME	*FTP username of OBPM machine.* *Needs to encrypted using AES key as in connecter.*
FU_PASSWORD	*FTP password of OBPM machine.* *Needs to encrypted using AES key as in connecter.*

3. OBPM file upload Restful service configurations
    a. Set the value of the host IP and Port for which the REST API is to be invoked against the prop_id = *'HOST_IP_UBSFU'* and *'HOST_PORT_UBSFU'* in the *DIGX_FW_CONFIG_VAR_B* table against the required entity.
    b. *Provide the values for the below properties in the MSTPROPERTIES table of host schema*:
        - propname = 'FU_FILETYPE' - the type of file.
        - propname = 'FU_HOSTCODE' - the host code.
        - propname = 'FU_RESTFILEPATH' - the filePath provided in the rest payload as on OBPM machine.
        - propname = 'FU_SRCCODE' - the source code .
        - propname = 'FU_TXNBRANCH' - the transaction branch code.

2. Reports

Reports in OBAPI can be used with Internal Reports Engine or Oracle BI.

(i) Reports – Internal Report Engine

In installer scripts, all reports point to Internal report engine, no additional configuration is required.

Note - A8_C2_PENDING_APPROVALS works only with BI.

For API Summary reports, internal engine works for maximum 500 records only. For higher load BI is recommended.

(ii) Reports – BI Configuration

1. Execute below query for those reports which need to pointed to BI

   update digx_rp_definition set provider='BI', allowed_formats='PDF~XLSX' and report_id not in ('A17', 'A01', 'A02', 'A03');

   Update BI webservice URL as

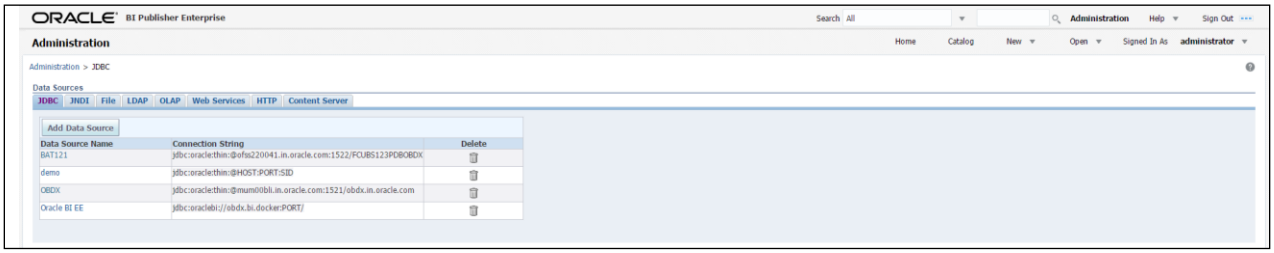
   Update digx_fw_config_out_ws_cfg_b set url='http://<BI Host>:<BI Port>/xmlpserver/services/v2/ReportService?WSDL' where service_id='runReport'

   Note – FATCA & CRS & EBPP reports works only with internal report engine and not with BI

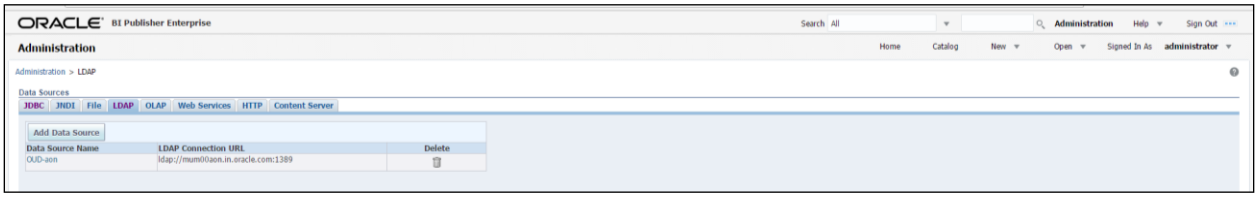
2. Login to BI and navigate to Administration link. Add JDBC data source

   a. OBAPI → Points to OBAPI schema

   b. BAT121 → Points to UBS EXT schema



3. Add OUD data source – OUD-aon (Required only for User Creation Report and using LDAP to store users)



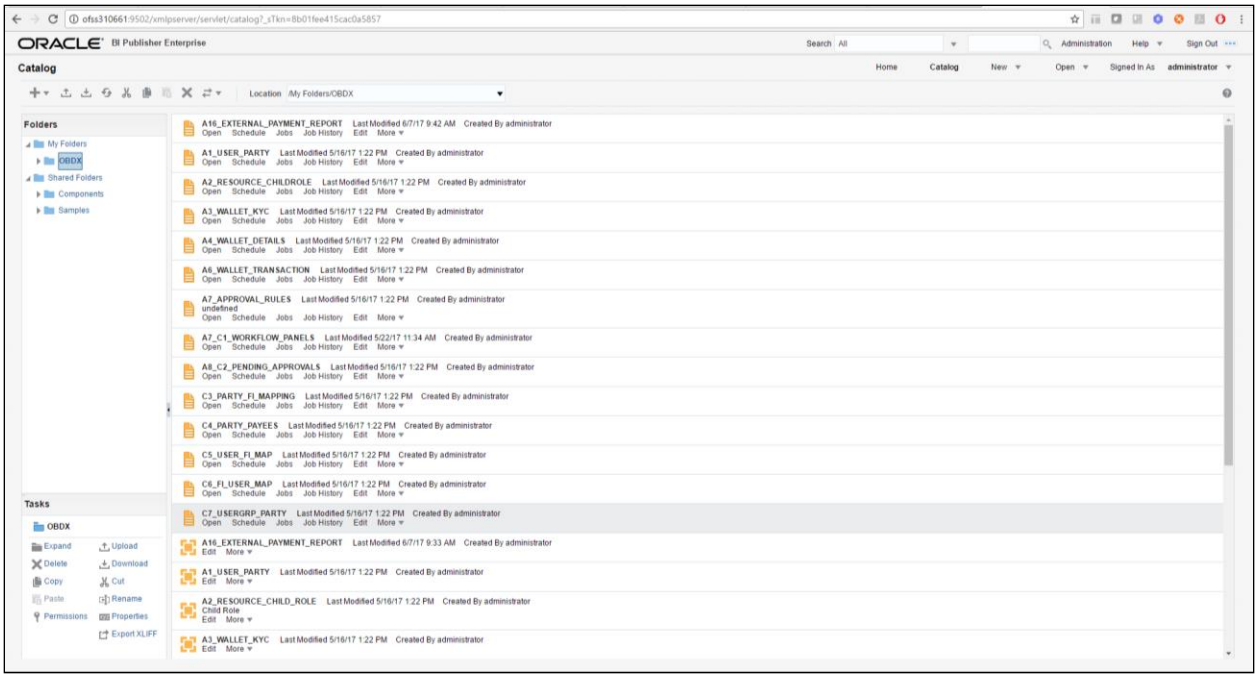
4. Upload all xdoz and xdmz from config/resources/report/obi117 (Some reports may have more then one xdmz's) (All xdoz and xdmz can be copied inside OBAPI.xdrz and uploaded at once. Empty xdrz is supplied in the 'config\resources\report\obi117')

   Eg. A1_USER_PARTY.xdmz – (OUD)

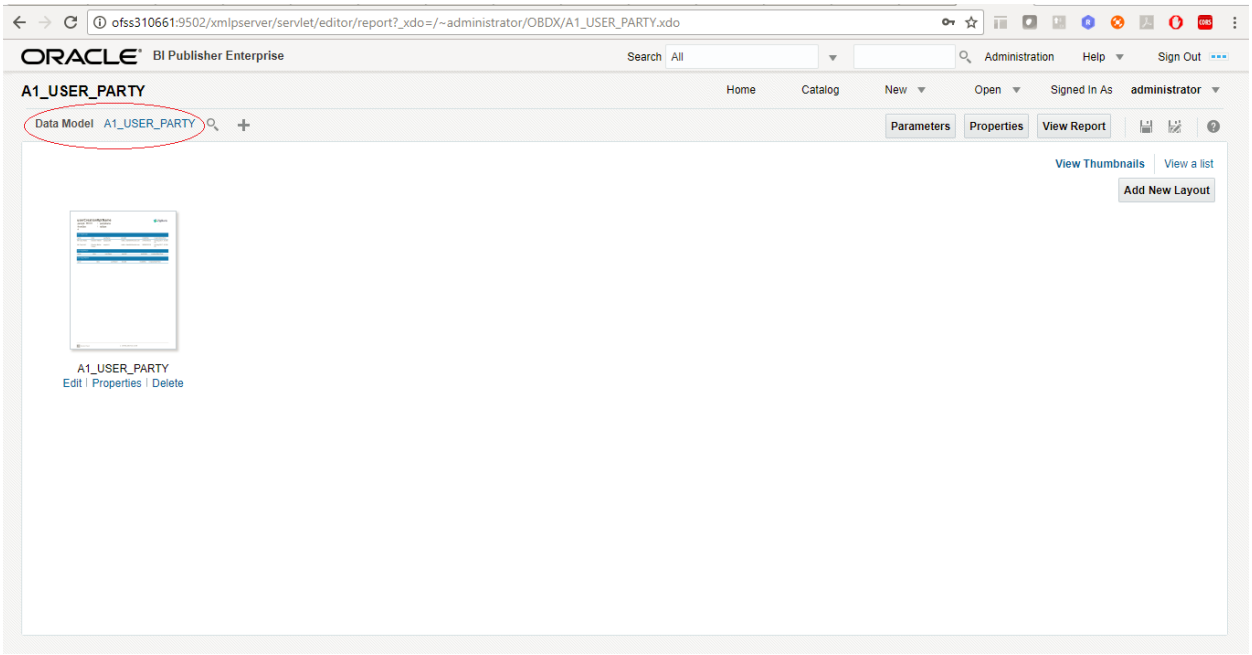
   A1_USER_PARTY_DBAUTH.xdmz – (DB Authenticator)

   A1_USER_PARTY_OPEN_LDAP – (Open LDAP)

   Select the appropriate xdmz and map to xdoz as shown below -



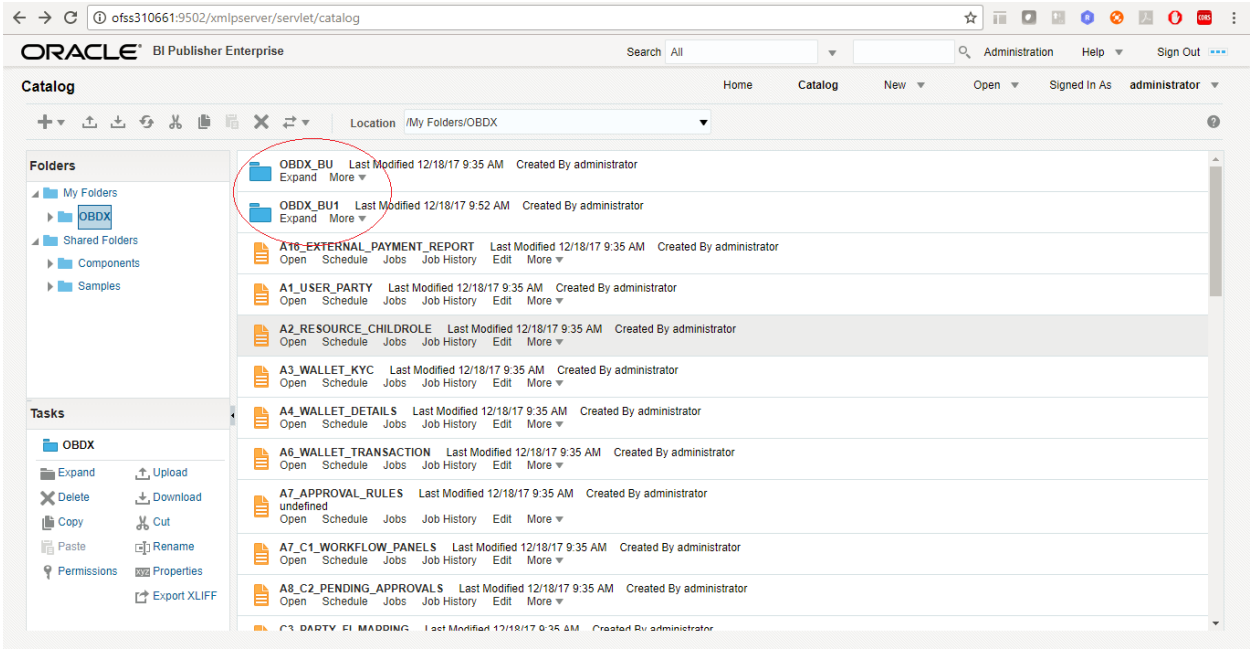
Click Edit → Data model



Select the data model and save.

For multi entity reports, create separate directories as shown below

U3 and U4 are multi entity reports



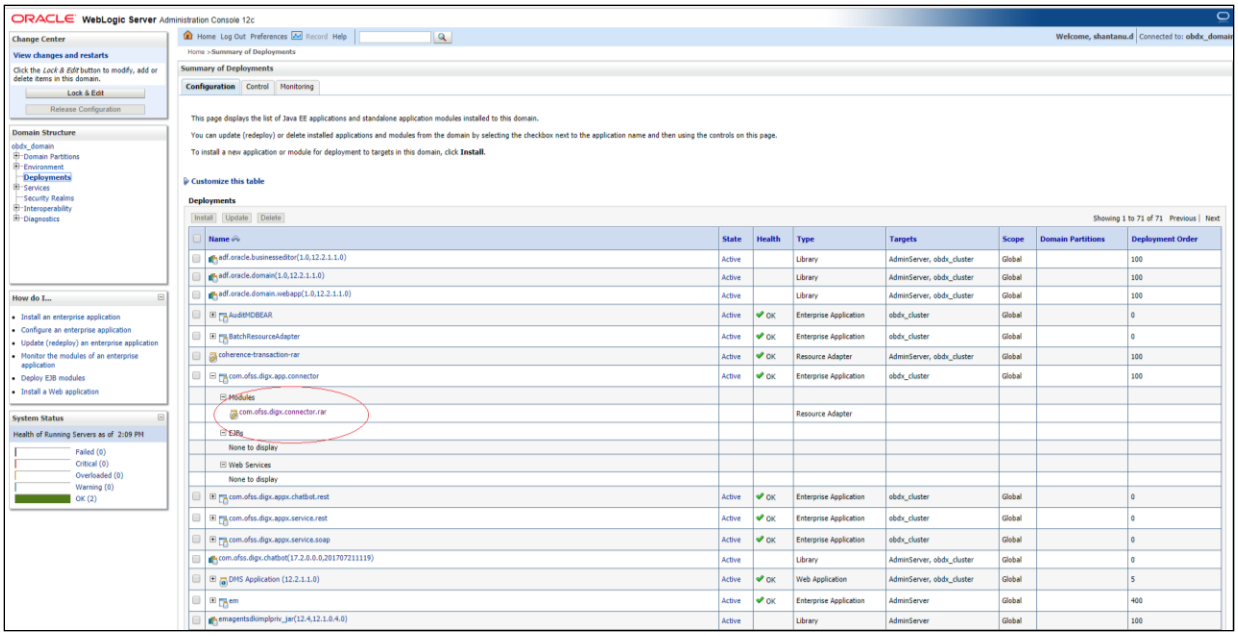
Use separate connections for host in these reports to point to required hosts.

5. Note the user used for BI console and the folder in which these artifacts are uploaded.

   Update the paths if required –

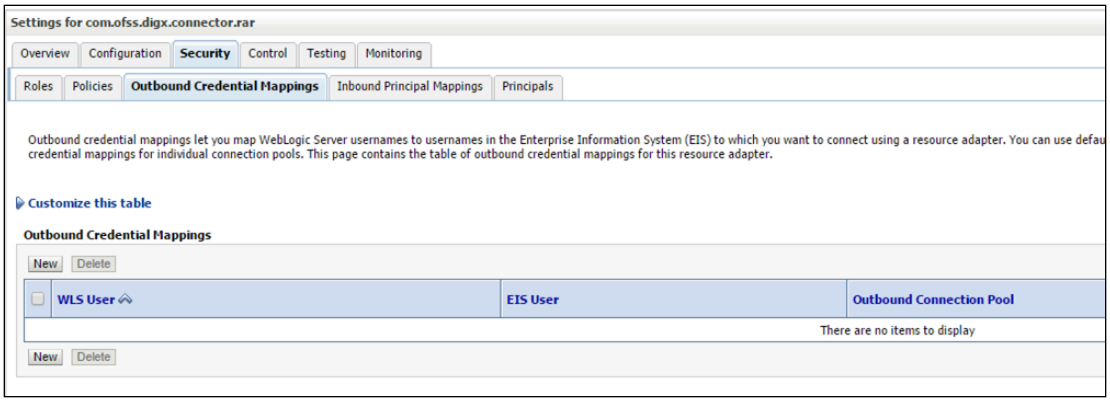
   select * from digx_fw_config_all_b where category_id='reportconfig' and prop_id like 'BI_ABSPATH%'

   Oracle BI Credentials are stored in WLS connector

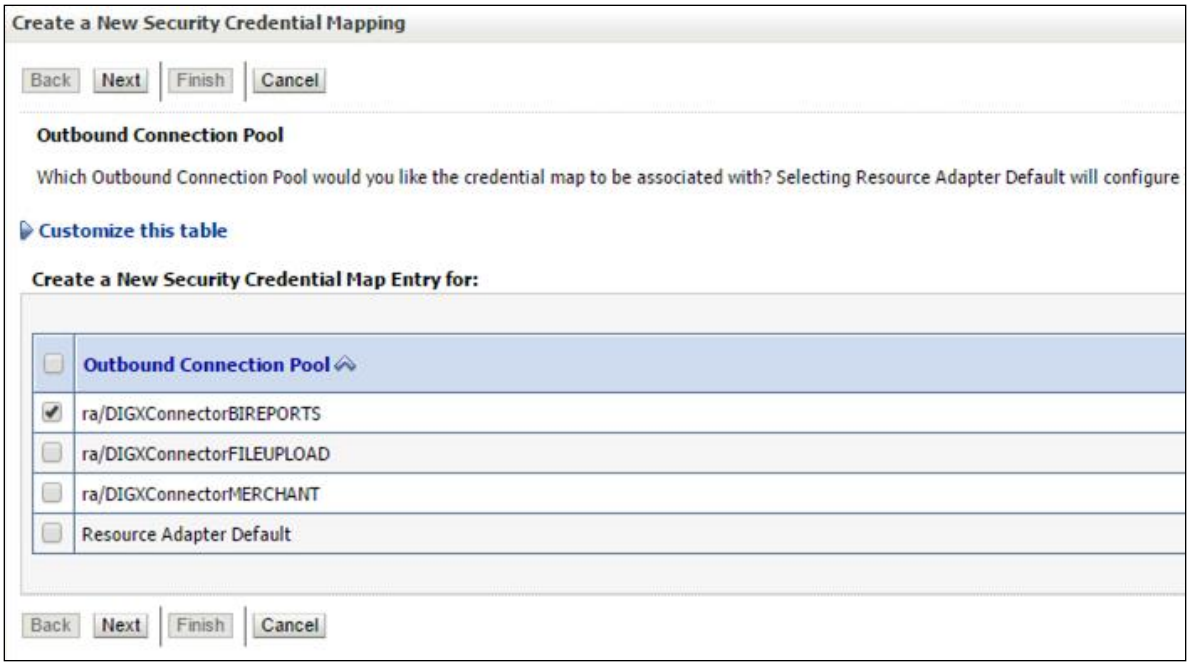


Add outbound credentials for this application, by following below steps.

- Browse to the deployed connector application > Security > Outbound Credential Mapping section

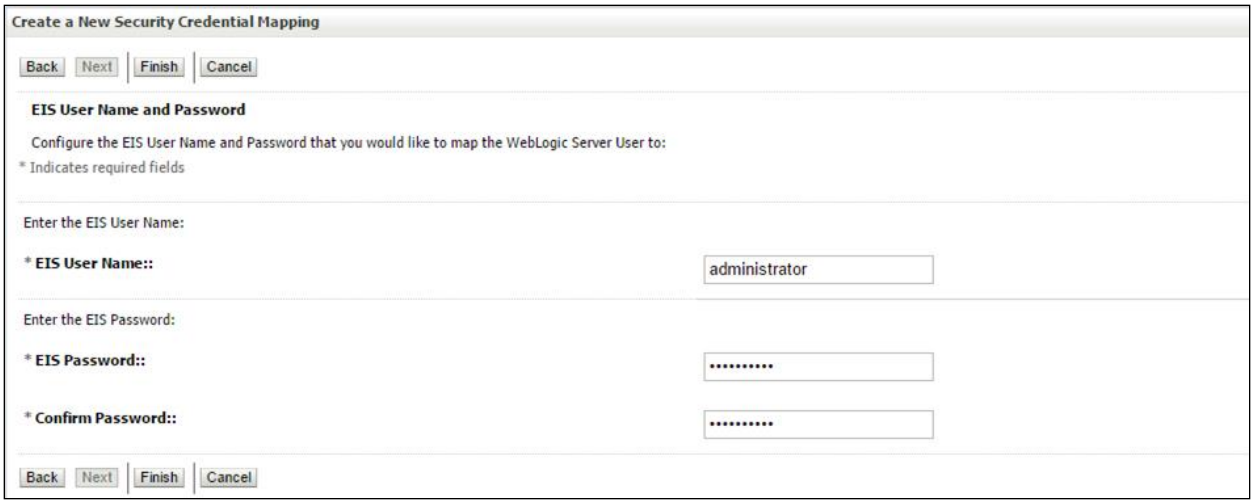


- Click new and select ra/DIGXConnectorBIPREPORTS



- Select Default user option

- Enter administrator credentials of BIP and click Finish


Create a New Security Credential Mapping
Back
Next
Finish
Cancel
EIS User Name and Password
Configure the EIS User Name and Password that you would like to map the WebLogic Server User to:
* Indicates required fields
Enter the EIS User Name:
* EIS User Name::
administrator
Enter the EIS Password:
* EIS Password::
* Confirm Password::
Back
Next
Finish
Cancel