Oracle Banking Digital Experience

File Upload Report Configuration
Release 17.2.0.0.0

Part No. E88573-01

July 2017

ORACLE®

File Upload, Report Configuration User Manual

July 2017

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/

Table of Contents

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 Digital Experience Release 17.2.0.0.0, refer to the following documents:

- Oracle Banking Digital Experience Licensing Guide
- Oracle Banking Digital Experience Installation Manuals

2. File Uploads

Outside In (For MS Excel processing)

This is used for parsing XLS,XLSX in file uploads module. This library for Linux x64 is shipped with OBDX. For other platforms, download from

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

Search Export – 8.5.3

Update the path for exepath in sx.cfg located at config/outsidein/linux64

Eg.

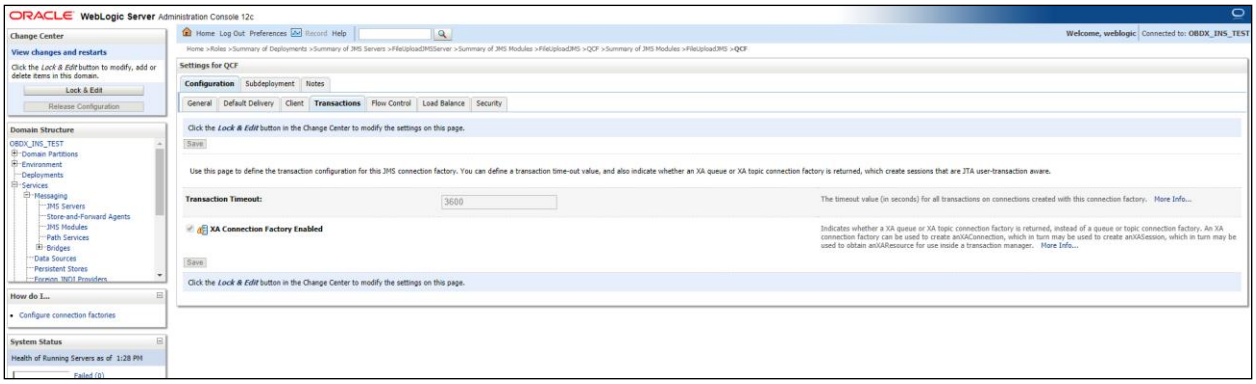
exepath /scratch/container/config/outsidein/linux64/exporter

For other platforms merge the sx.cfg configurations

Grant 777 privileges for OutsideIn directory

Configurations for Connection Factory

Enable **XA Connection Factory Enabled**

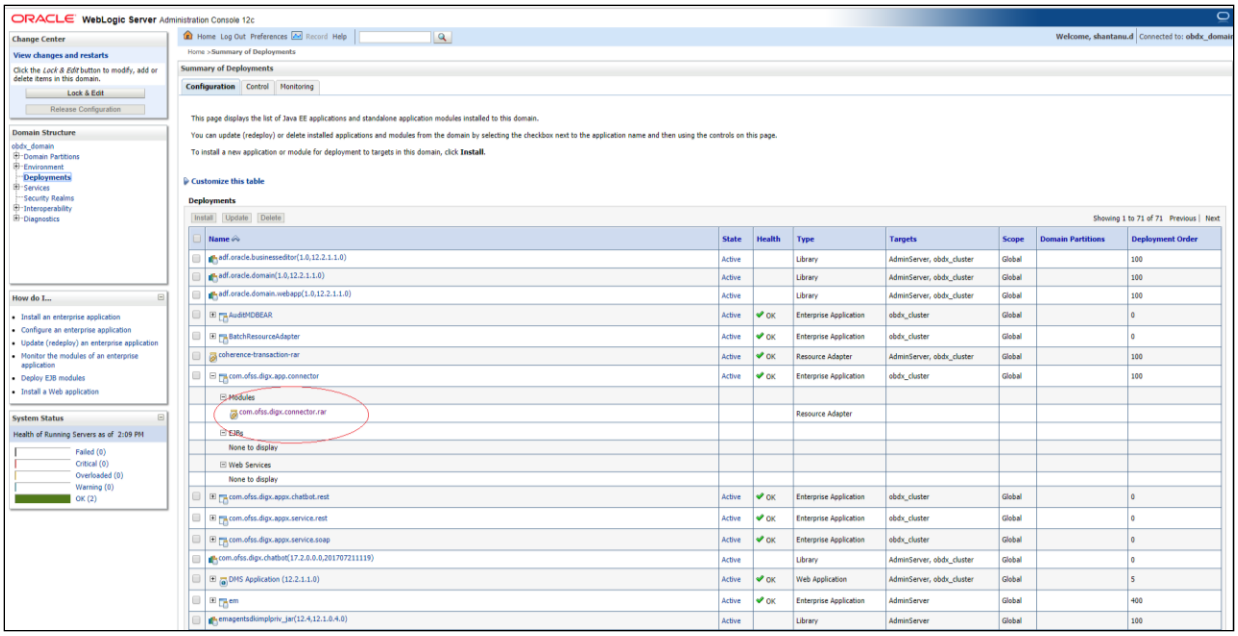


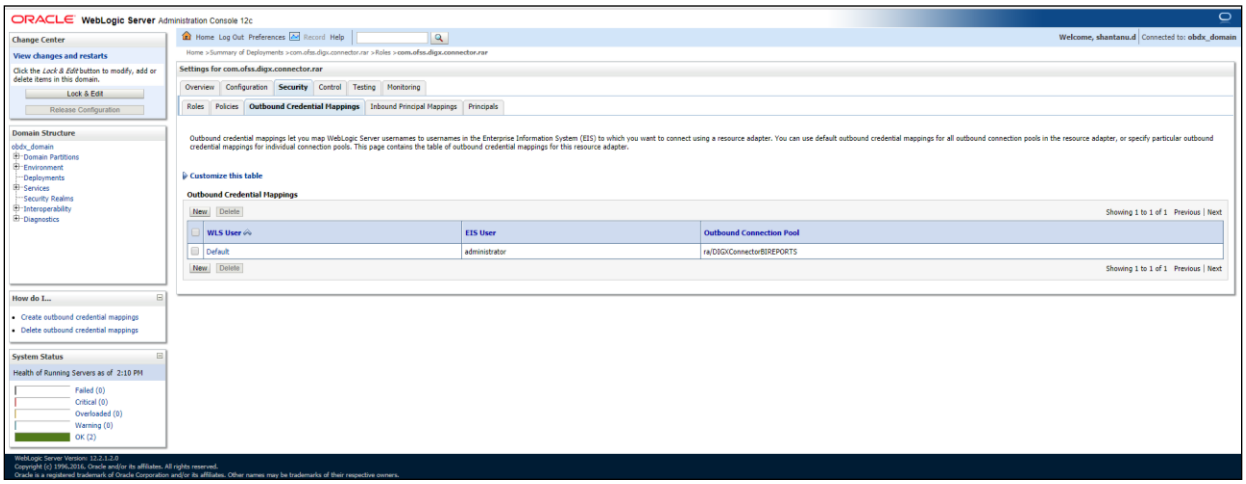
Configuration for storing key for decrypting uploaded files and creating encrypted response files

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';

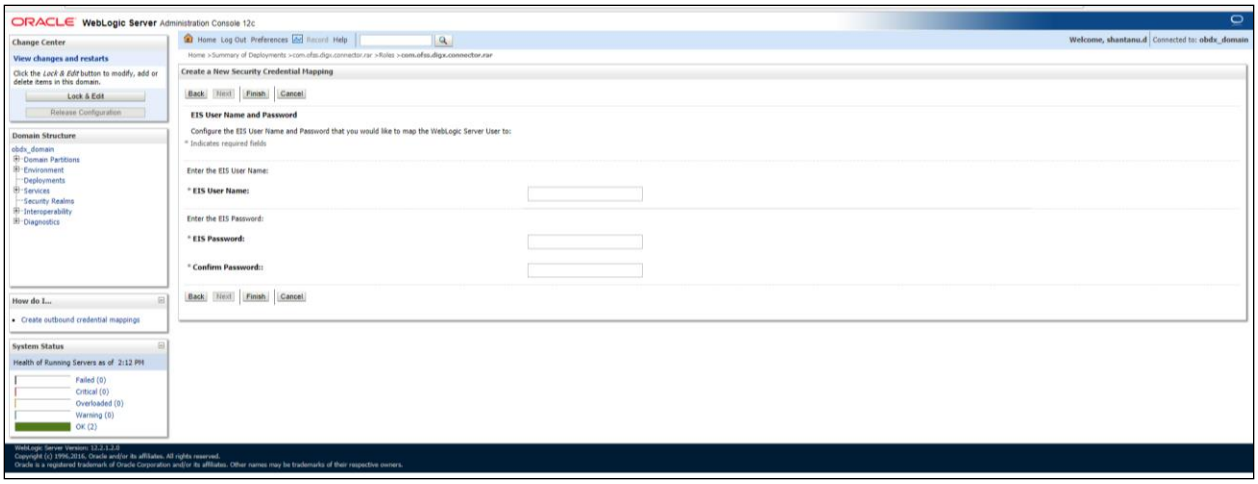
Update the encryption key in connector as below –





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
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

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. Lets 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 OBDX 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.

Eg.

```
<Field name=" beneId"/>

<Field name=" beneName "/>
```

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.

3. Reports

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

Reports – Internal Report Engine

update digx_rp_definition set provider='IN', allowed_formats='PDF' ;

update digx_rp_definition set allowed_formats='PDF~CSV' where report_id in ('A1','A3','A4','A6','A7','A9','C3','C8');

Reports – BI Configuration

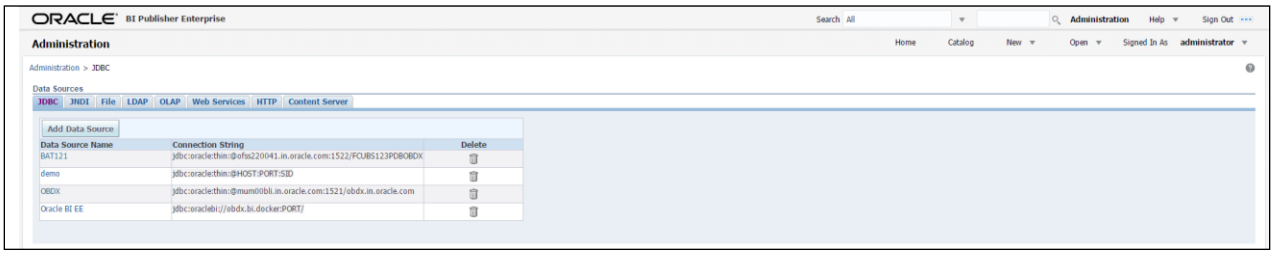
1. Execute below query for those reports which need to pointed to BI

   update digx_rp_definition set provider='BI', allowed_formats='PDF~EXCEL';

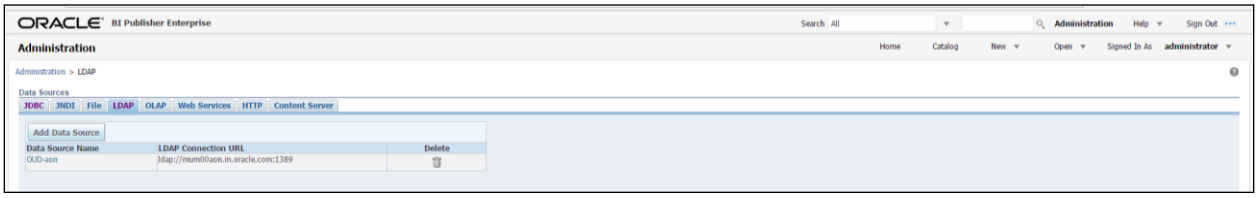
Update BI webservice URL as

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

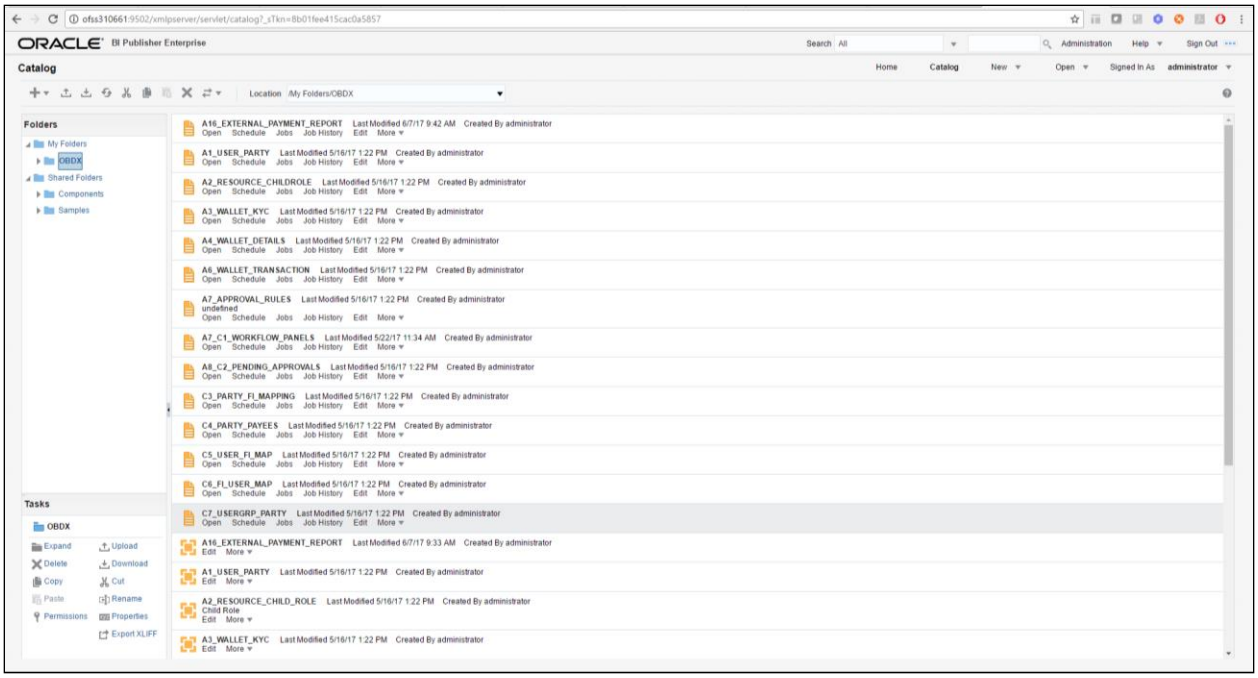
2. Login to BI and navigate to Administration link. Add JDBC data source
   a. OBDX → Points to OBDX schema
   b. BAT121 → Points to UBS EXT schema



3. Add OUD data source – OUD-aon (Required only for User Creation Report)



4. Upload all xdoz and xdmz from config/resources/report/obi117

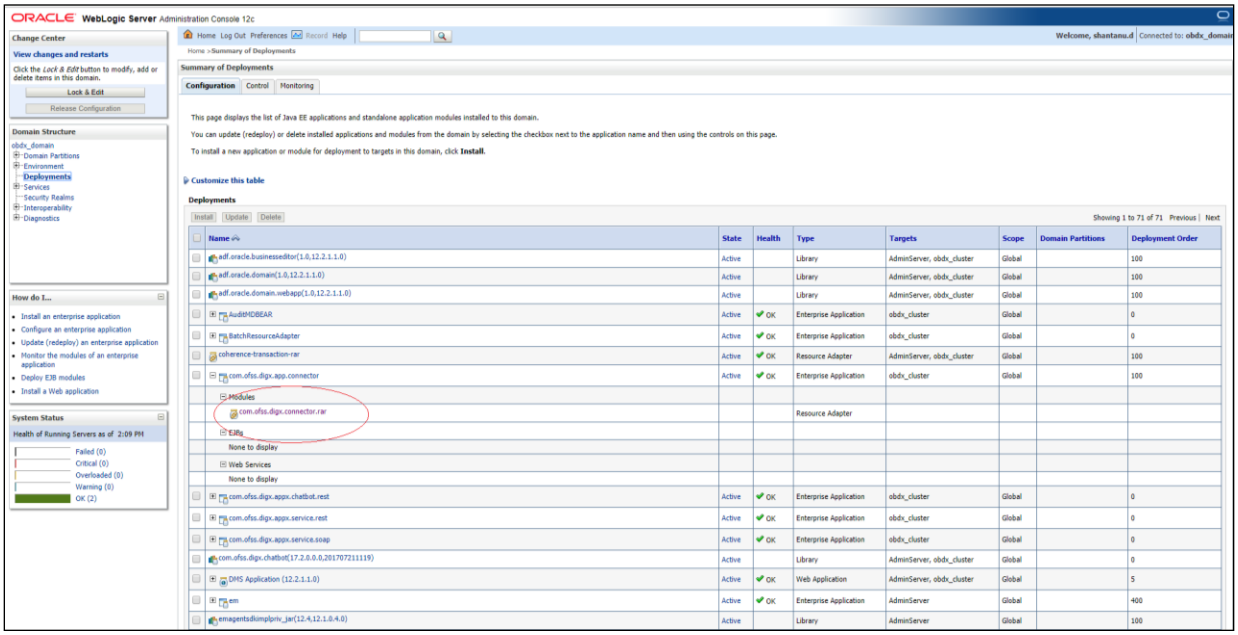


5. Note the user used for BI console and the folder in which these artifacts are uploaded.

   Update the paths if required –

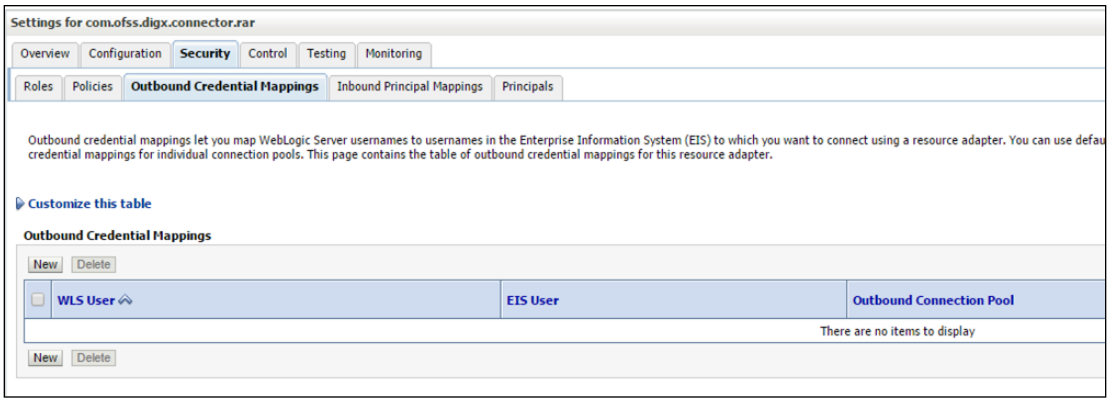
   select * from digx_fw_config_all_b where category_id='reportconfig' and prop_id like 'BI_ABSPATH%'

   Oracle BI Credentials are stored in WLS connector

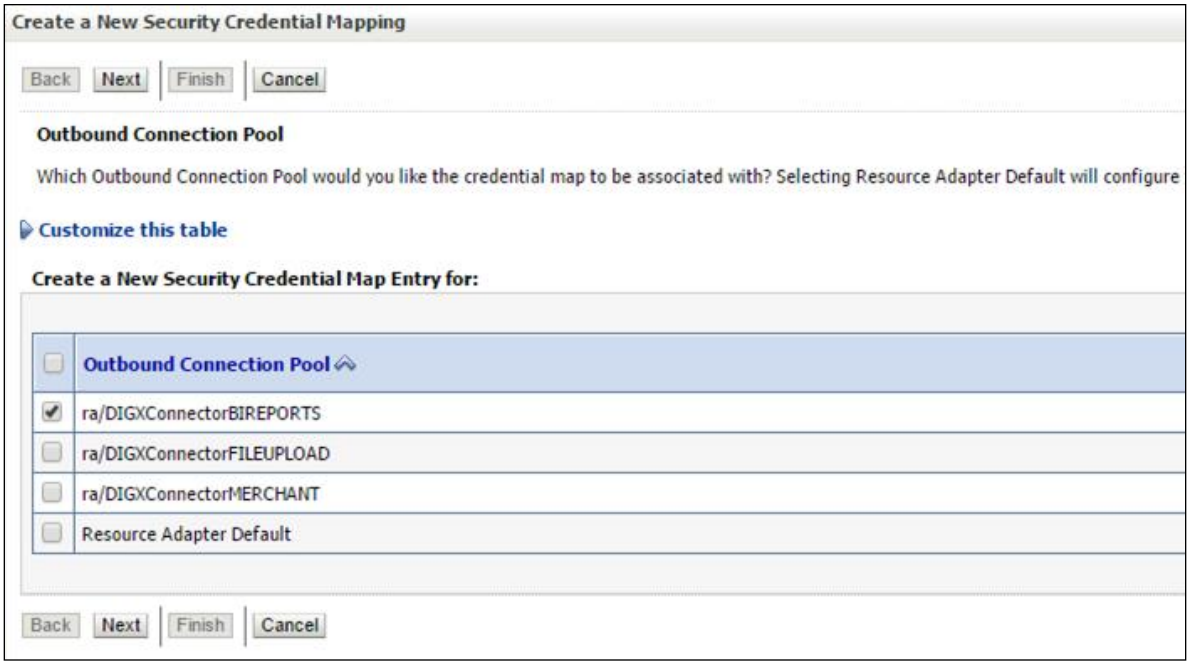


Add outbound credentials for this application, by following below steps.

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