Oracle® Banking APIs

File Upload Report Configuration Guide

Patchset Release 22.2.4.0.0
F99653-01
June 2024

Oracle Banking APIs File Upload Report Configuration Guide, Patchset Release 22.2.4.0.0

F99653-01

Copyright © 2006, 2024, Oracle and/or its affiliates.

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.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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 fail-safe, 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.

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Purpose iv
Audience iv
Documentation Accessibility iv
Diversity and Inclusion iv
Conventions v
Related Resources v
Screenshot Disclaimer v
Acronyms and Abbreviations v

1 File Uploads

1.1 OutsideIn (For MS Excel processing) 1-1
1.2 Configuration for storing key for decrypting uploaded files and creating encrypted response files 1-2
1.3 Using Enrichers in File Uploads 1-3

2 Reports

2.1 Reports – Internal Report Engine 2-1

3 List of Topics

Index

Preface

- Purpose
- Audience
- Documentation Accessibility
- Diversity and Inclusion
- Conventions
- Related Resources
- Screenshot Disclaimer
- Acronyms and Abbreviations

Purpose

This guide is designed to help acquaint you with the Oracle Banking APIs application. This guide provides answers to specific features and procedures that the user need to be aware of the module to function successfully.

Audience

This document is intended for the following audience:

- Customers
- Partners

Documentation Accessibility

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

Access to Oracle Support

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

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.

Related Resources

For more information on any related features, refer to the following documents:

- Oracle Banking APIs Installation Manuals

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.

Acronyms and Abbreviations

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

Table 1 Acronyms and Abbreviations

Abbreviation	Description
OBAPI	Oracle Banking APIs

1

File Uploads

- 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)
- 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`.
- Using Enrichers in File Uploads
  (For custom defined templates only, not required for out of box templates)

1.1 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.

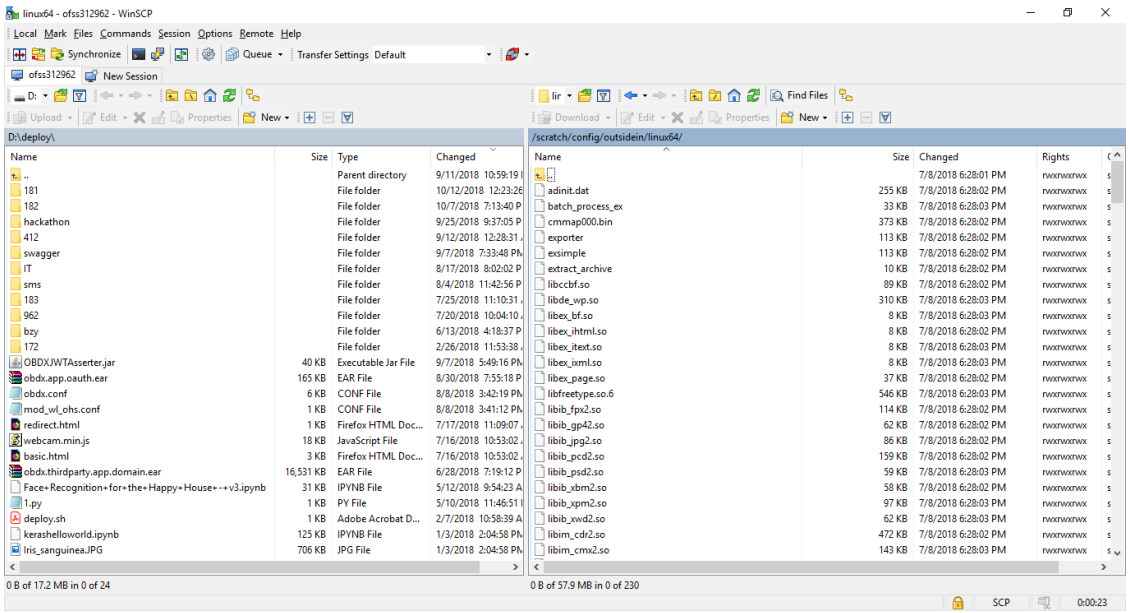
Then copy all contents (except jar & sh files) of **'sdk/demo'** directory to **config/outsidein/<os>**.

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.



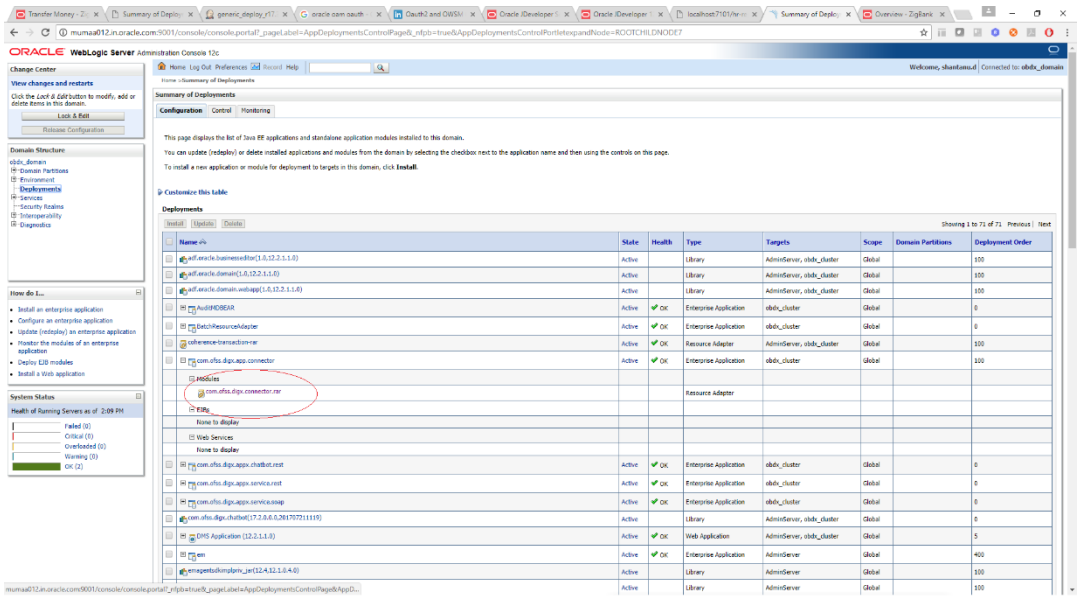
1.2 Configuration for storing key for decrypting uploaded files and creating encrypted response files

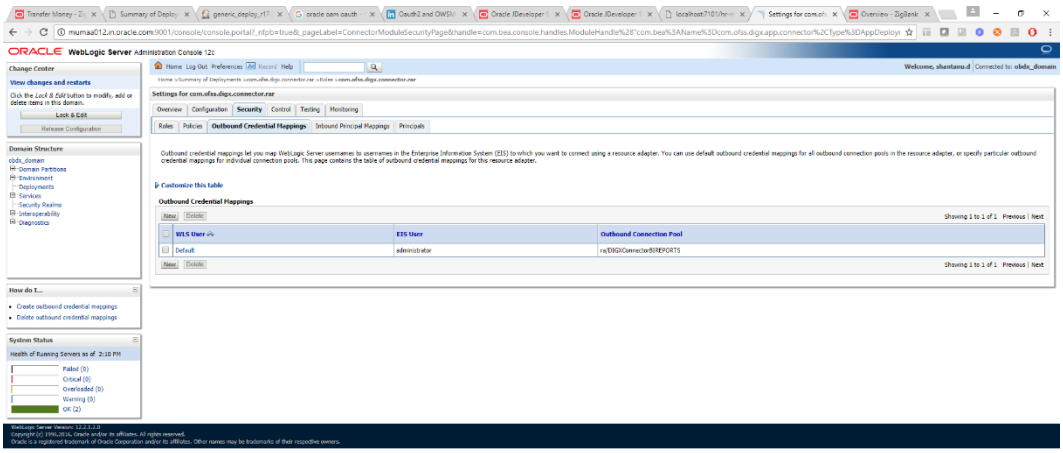
The key used for file decryption by default decryptor is stored in database in `digx_fw_config_all_b` with prop_id as `ENCRYPTION_KEY`.

1. If this is to be stored in WLS connector update the property as below:

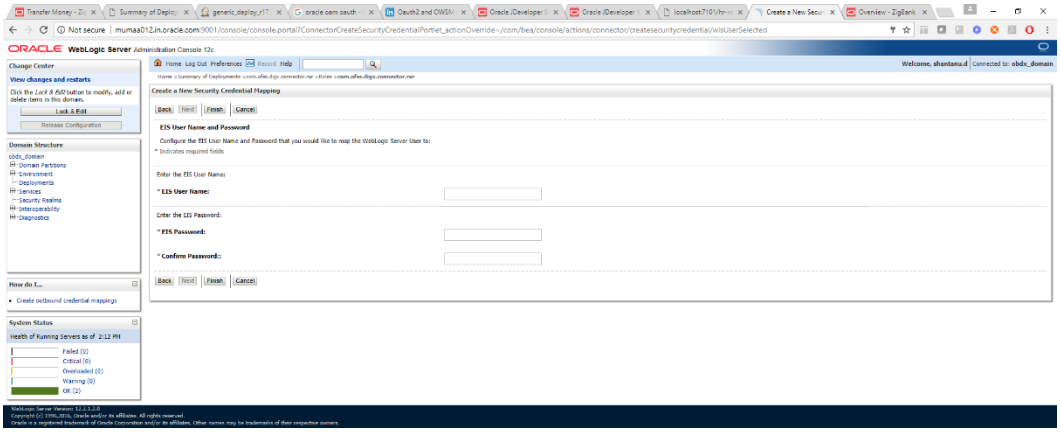
```
update digx_fw_config_all_b set prop_value='KEY_STORE'
where prop_id='ENCRYPTION_KEY_LOCATION';
```

2. Update the encryption key in connector as below:





3. Click **New** → Select **ra/DIGXConnectorFILEUPLOAD** → **Next** → Select **Default User**.
4. In **Password** field, enter the encryption key.



1.3 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 e1) {
        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.

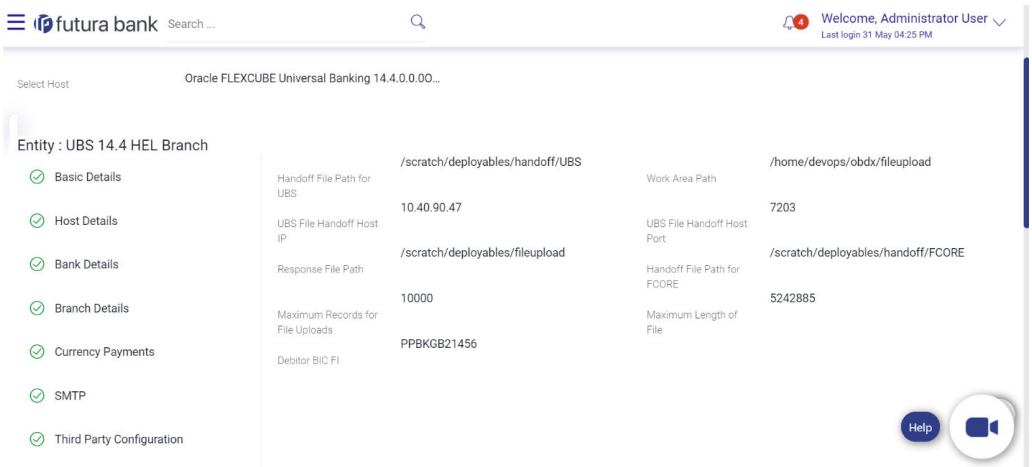
OBPM configs

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 OBAPI 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.
4. Debtor BIC FI Configuration
   a. Provision to set Debtor BIC has been provided at entity level.
   b. The same can be configured in the following path by System Administrator user:
      i. *Toggle menu → Configuration → System Configuration → Click on Continue → Select Entity → Dynamic Module Tab → File Upload*



2
Reports

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

- Reports – Internal Report Engine
  In installer scripts, all reports point to Internal report engine, no additional configuration is required.

2.1 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.

3

List of Topics

This user manual is organized as follows:

Table 3-1 List of Topics

Topics	Description
Preface	This topic provides information on the introduction, intended audience, list of topics, and acronyms covered in this guide.
File Uploads	This topic provides information about OutsideIn (For MS Excel processing), Configuration for storing key for decrypting uploaded files and creating encrypted response files, and Enrichers in File Uploads .
Reports	This topic explains how reports in OBDX can be used with Internal Reports Engine or Oracle BI.

Index

C

Configuration for storing key for decrypting uploaded files and creating encrypted response files, *1-2*

O

OutsideIn (For MS Excel processing), *1-1*

R

Reports, *2-1*
Reports – Internal Report Engine, *2-1*

U

Using Enrichers in File Uploads, *1-3*