

File Upload Report Configuration Guide  
Oracle Banking APIs  
Patchset Release 22.2.3.0.0

Part No. F72988-01

February 2024

**ORACLE®**

## File Upload Report Configuration Guide

February 2024

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

Copyright © 2006, 2024, 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

<b>1. Preface .....</b>	<b>1-1</b>
1.1 Intended Audience.....	1-1
1.2 Documentation Accessibility .....	1-1
1.3 Access to Oracle Support.....	1-1
1.4 Structure .....	1-1
1.5 Related Information Sources .....	1-1
<b>2. File Uploads.....</b>	<b>2-1</b>
2.1 OutsideIn (For MS Excel processing).....	2-1
2.2 Configuration for storing key for decrypting uploaded files and creating encrypted response files 2-2	
2.3 Using Enrichers in File Uploads .....	2-3
<b>3. Reports .....</b>	<b>3-1</b>
3.1 Reports – Internal Report Engine .....	3-1

---

# 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.

The subsequent chapters describe following details:

- Introduction
- Preferences & Database
- Configuration / Installation.

## 1.5 Related Information Sources

For more information on Oracle Banking APIs Patchset Release 22.2.3.0.0, refer to the following documents:

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

## 2. File Uploads

### 2.1 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

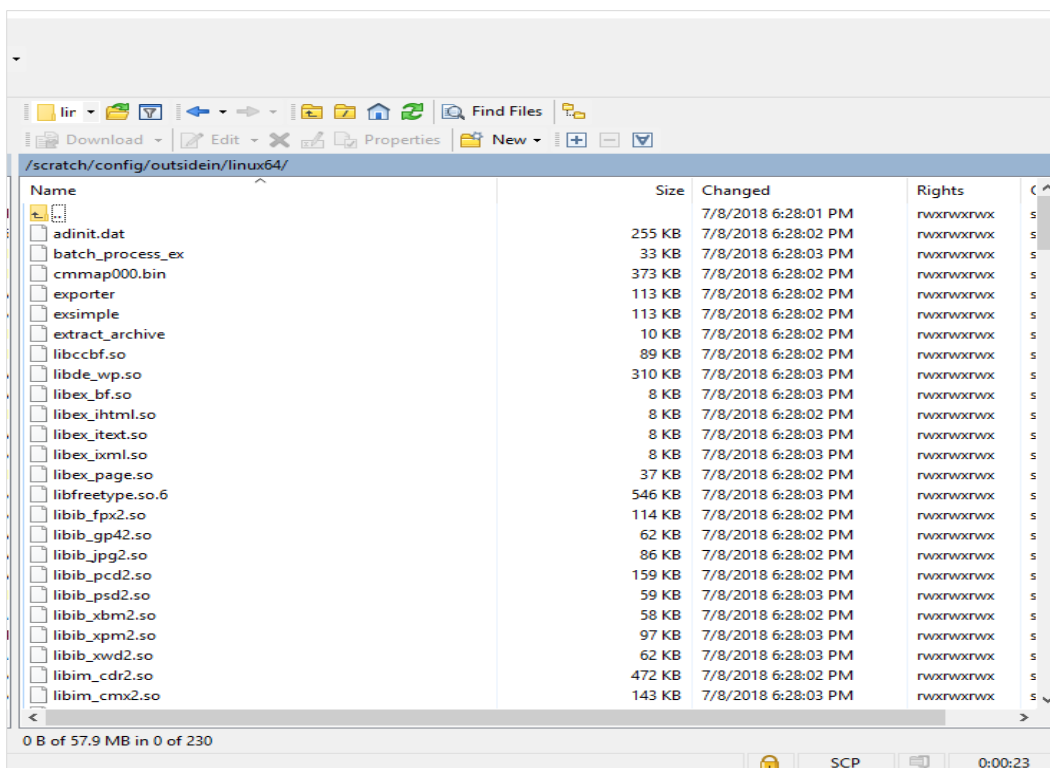
Then copy all contents (except jar & sh files) of 'sdk/demo' directory to config/outsidein/<os>

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



## 2.2 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'. 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 –

The screenshot shows the Oracle WebLogic Server Administration Console. The main area displays a table of Deployments. The following table represents the data shown in the screenshot:

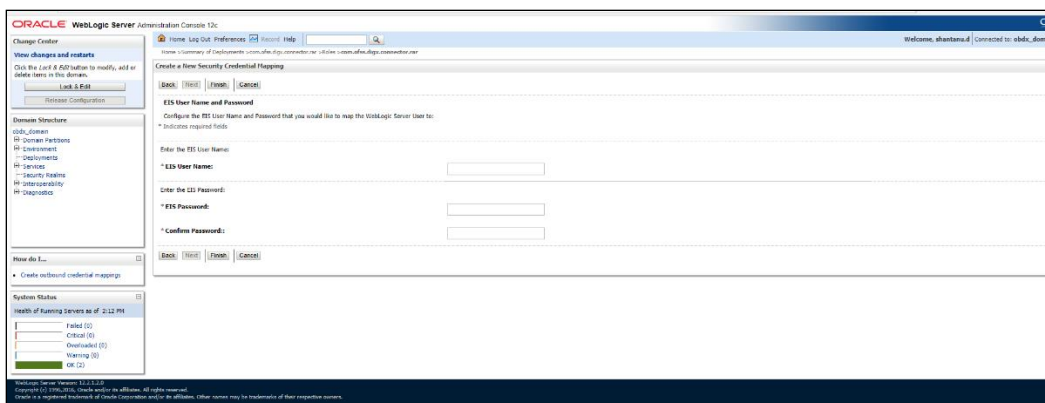
Name	State	Health	Type	Targets	Scope	Domain Partitions	Deployment Order
WLSUser	Active		Library	AdminServer, obdt_cluster	Global		100
WLSUser	Active		Library	AdminServer, obdt_cluster	Global		100
WLSUser	Active		Library	AdminServer, obdt_cluster	Global		100
WLSUser	Active	OK	Enterprise Application	obdt_cluster	Global		0
WLSUser	Active	OK	Enterprise Application	obdt_cluster	Global		0
WLSUser	Active	OK	Resource Adapter	AdminServer, obdt_cluster	Global		100
WLSUser	Active	OK	Enterprise Application	obdt_cluster	Global		100
WLSUser	None to display		Resource Adapter				
WLSUser	None to display						
WLSUser	None to display						
WLSUser	Active	OK	Enterprise Application	obdt_cluster	Global		0
WLSUser	Active	OK	Enterprise Application	obdt_cluster	Global		0
WLSUser	Active	OK	Enterprise Application	obdt_cluster	Global		0
WLSUser	Active	OK	Library	AdminServer, obdt_cluster	Global		0
WLSUser	Active	OK	Web Application	AdminServer, obdt_cluster	Global		5
WLSUser	Active	OK	Enterprise Application	AdminServer	Global		400
WLSUser	Active	OK	Library	AdminServer	Global		100

The screenshot shows the Oracle WebLogic Server Administration Console for the 'com.oracle.digx.connector.jar' module. The 'Outbound Credential Mappings' page is displayed, showing a table of mappings:

WLS User	WLS User	Outbound Connection Pool
WLS User	WLS User	WLS User
Default	Default	WLS User

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

In password field, enter the encryption key



## 2.3 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="beneld~beneName" on beneficiary address field, the parser simply invokes getters on beneld 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=" beneld"/>
<Field name=" beneName "/>
<Field name="beneAddr" enricher="ADRESSENRICHER" enricherDynArgs=" beneld~beneName
"/>

```

Eg.

## Extract (Response) File Enrichers

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <FileDefinition fileType="csv" delimiter="," handler="com.ofss.digx.framework.fileupload.extract.CSVHandler" encryptionClass="">
3
4   <RecordDefinition recordType="H">
5     <Field name="record" label="RECORD" />
6     <Field name="recRefId" label="RECORD REF NO" />
7     <Field name="fileRefId" label="FILE REF NO" />
8     <Field name="digxRefId" label="E-BANKING REF NO" />
9     <Field name="contractRefId" label="CONTRACT REF NO" />
10    <Field name="recStatus" label="RECORD STATUS" />
11    <Field name="errCode" label="STATUS CODE" />
12    <Field name="errMsg" label="STATUS DESCRIPTION" />
13  </RecordDefinition>
14
15  <RecordDefinition query="ResponseList" recordType="B">
16    <Field name="record" no="1" wrapchar="" />
17    <Field name="recRefId" no="2" />
18    <Field name="fileRefId" no="3" />
19    <Field name="digxRefId" no="4" />
20    <Field name="contractRefId" no="5" />
21    <Field name="recStatus" no="6" />
22    <Field name="errCode" no="7" enricher="ERRORMSG" enricherArgs="" />
23    <Field name="errMsg" no="8" />
24  </RecordDefinition>
25
26 </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 OBPM as host, for **file level** uploads in OBDX, the files are generated in PAIN001001/PAIN001001 formats after approval at OBDX end is complete. These files are stored in a directory on OBDX server. For record level, service is used same as of single screen transactions.

## 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

File is copied via REST service to OBPM

### 3. 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

Select Host: Oracle FLEXCUBE Universal Banking 14.4.0.0.00...

Entity : UBS 14.4 HEL Branch

Handoff File Path for UBS	/scratch/deployables/handoff/UBS	Work Area Path	/home/devops/obdx/fileupload
UBS File Handoff Host IP	10.40.90.47	UBS File Handoff Host Port	7203
Response File Path	/scratch/deployables/fileupload	Handoff File Path for FCORE	/scratch/deployables/handoff/FCORE
Maximum Records for File Uploads	10000	Maximum Length of File	5242885
Debtor BIC FI	PPBKGB21456		

Entity Configuration Categories:

- Basic Details
- Host Details
- Bank Details
- Branch Details
- Currency Payments
- SMTP
- Third Party Configuration

Help | Video

[Home](#)

---

## 3. Reports

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

### 3.1 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.

---

[Home](#)