

Oracle® Banking APIs

File Upload Configuration Guide



Innovation Release 25.1.1.0.0

G46838-01

October 25.1

ORACLE®

Copyright © 2006, 2025, 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	i
Audience	i
Documentation Accessibility	i
Critical Patches	i
Diversity and Inclusion	ii
Conventions	ii
Related Resources	ii
Screenshot Disclaimer	ii
Acronyms and Abbreviations	ii

1 File Uploads

1.1 Using Enrichers in File Uploads	1
-------------------------------------	---

2 Reports

Index

Preface

- [Purpose](#)
- [Audience](#)
- [Documentation Accessibility](#)
- [Critical Patches](#)
- [Diversity and Inclusion](#)
- [Conventions](#)
- [Related Resources](#)
- [Screenshot Disclaimer](#)
- [Acronyms and Abbreviations](#)

Purpose

This guide is designed to help acquaint you with the Oracle Banking 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.

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	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

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

1.1 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`. To configure a new enricher we have to add an entry of the fully-qualified name of the new enricher in `META-INF\services\com.ofss.digx.framework.fileupload.enrichment.IEnrichment` file. Currently OBDX support only Java enrichers.
- Enrichers can be in any package but must implement the `IEnrichment` interface and should be annotated with `@Enricher(value = " {ENRICHMENT_ID}")`, where `ENRICHMENT_ID` is the id of the enricher used in template. Custom enrichers should also be annotated with `@Custom`. Custom enricher with the same `ENRICHMENT_ID` as of base will override the base enricher.

Example

- 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. `ACCTCURR` enrichment id would be looked for and `AccountCurrencyEnricher` class is invoked. This enricher derives the `debitAccountCurr`. Hence this attribute must be present in the record DTO with its setters defined.

```

6 usages
@Enricher(value = "ACCTCURR")
@Base
public class AccountCurrencyEnricher implements IEnrichment {

    /**

    @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,

Example. `enricherDynArgs="beneId~beneName"` on beneficiary address field, the parser simply invokes getters on `benId` and `benName` 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

```
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. Example, 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. Copying the file to host system

File is copied via REST service to OBPM

2. 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

Basic Details	Handoff File Path for UBS	/scratch/deployables/handoff/UBS	Work Area Path	/home/devops/obdx/fileupload
Host Details	UBS File Handoff Host IP	10.40.90.47	UBS File Handoff Host Port	7203
Bank Details	Response File Path	/scratch/deployables/fileupload	Handoff File Path for FCORE	/scratch/deployables/handoff/FCORE
Branch Details	Maximum Records for File Uploads	10000	Maximum Length of File	5242885
Currency Payments	Debtor BIC FI	PPBKGB21456		
SMTP				
Third Party Configuration				

Help

2

Reports

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

Index

R

Reports, [1](#)

U

Using Enrichers in File Uploads, [1](#)