

Content and NLP Configurations Guide
Oracle Banking APIs
Patchset Release 21.1.1.0.0

Part No. F40802-01

June 2021

ORACLE®

Configurations for Content and NLP

June 2021

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 © 2006, 2021, 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	1-1
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
2. Introduction	2-1
3. Ideal Scenario	3-1
4. Complex Scenario	4-1
5. Use Case Name for NLP	5-1

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.

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 Patchset Release 21.1.1.0.0, refer to the following documents:

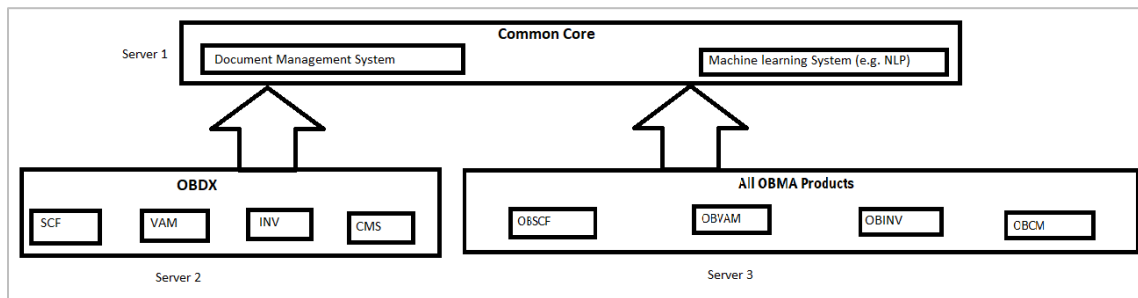
- Oracle Banking APIs Licensing Guide
- Oracle Banking APIs Installation Manuals

2. Introduction

Document Management System (DMS) is a common core product which stores the documents either uploaded from channel banking product (e.g. OBAPI) or uploaded from any Oracle Banking Mid Office products (e.g. OBSCF, OBCM, OBVAM, and so on) and returns document ID for future access. Similarly, Machine Learning System (e.g. NLP) is a common core product with machine learning related functionalities, such as extracting data from image called as Natural Language Processing (NLP).

3. Ideal Scenario

Document Management System (DMS) and Machine Learning System (e.g. NLP) are deployed on a single common core server and are commonly accessed by both OBAPI and OBMA products.



There are no issues in this scenario. OBAPI will interact with common core by picking default configurations.

1. For Content transactions, OBAPI will upload the document in DMS using default HOST IP and PORT (Variable Names in DIGX_FW_CONFIG_VAR_B - MO_HOST_IP, MO_HOST_PORT) and default Headers (Variable Names in DIGX_FW_CONFIG_ADAPTER_PROP_B with **MO as HOST ID - MO_BRANCH_CODE, MO_APP_ID, MO_USER_ID, MO_ENTITY_ID**).
2. For NLP transactions, OBAPI will use the document ID received from DMS and send it to NLP system using default HOST IP and PORT (Variable Names in DIGX_FW_CONFIG_VAR_B - NLP_HOST_IP, NLP_HOST_PORT) and default Headers (Variable Names in DIGX_FW_CONFIG_ADAPTER_PROP_B with **NLP as HOST ID - APP_ID, USER_ID, BRANCH_CODE, ENTITY_ID**).

Now since DMS and NLP systems are installed on the same server, both can interact with each other's DB and thus NLP system will be able to download the document from document ID received.

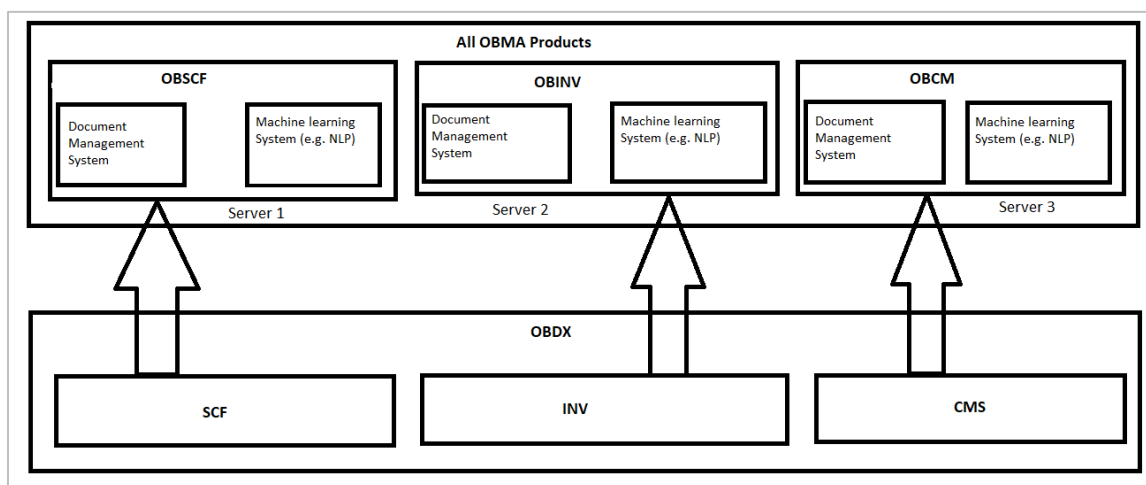
The problem arises in Complex Scenario.

[Home](#)

4. Complex Scenario

Here all OBMA products are installed on separate servers and every OBMA product has DMS and NLP system installed. In such a scenario, documents uploaded on DMS of OBCM will not be accessible by NLP of OBSCF. For NLP of OBSCF, documents must be uploaded on DMS of OBSCF. And for NLP of OBINV, documents must be uploaded on DMS of OBINV and so on.

Note: NLP & DMS of same OBMA products can interact with each other but not across OBMA products.



Thus in OBAPI:

- All content transactions and NLP transactions done for module SCF should interact with DMS and NLP system of OBSCF
- All content transactions and NLP transactions done for module INV should interact with DMS and NLP system of OBINV.

Hence, default configurations cannot be used in this scenario. To achieve this, a configurator table has been added: "DIGX_MO_CONFIGURATION_SELECTOR". For modules, where module specific HOST IP, PORT & HEADERS are to be picked instead of default configurations, entries can be made in this table. Thus using this table, every module can interact with DMS and NLP systems of their respective mid-office products. For such modules, the configuration in this table has been overridden.

Previously, actual configuration values were stored in tables like: DIGX_FW_CONFIG_ADAPTER_PROP_B, DIGX_FW_CONFIG_VAR_B and configuration variables were created in AdapterConstants.java of Host implementation.

With the addition of the DIGX_MO_CONFIGURATION_SELECTOR table, configuration values will still be stored in same tables. The only difference would be that instead of providing configuration variables via AdapterConstants.java, they will be provided by the table.

Note: The DIGX_MO_CONFIGURATION_SELECTOR table contains the variable names of configurations which are to be overridden and not the actual configuration values.

The dummy scripts are listed below for reference:

- Insert into DIGX_MO_CONFIGURATION_SELECTOR
(MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
values
(('SCF','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.create','scfDocumentUpload','SCFAdapterImplConfig',null,'DMS_APP_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
(MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
values
(('SCF','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.read','scfDocumentDownload','SCFAdapterImplConfig',null,'DMS_APP_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
(MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
values
(('SCF','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.listDocuments','scfDocumentsList','SCFAdapterImplConfig',null,'DMS_APP_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
(MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
values
(('SCF','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.delete','scfDocumentDelete','SCFAdapterImplConfig',null,'DMS_APP_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
(MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
values
(('SCF','com.ofss.digx.extxface.nlp.adapter.namedentityrecognizer.INamedEntityRecognizerAdapter.extract','scfNamedEntityExtract','SCFAdapterImplConfig',null,'NLP_APP_ID','USER_ID','BRANCH_CODE','NER_USE_CASE_NAME','ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
(MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
values
(('SCF','com.ofss.digx.extxface.nlp.adapter.namedentityrecognizer.INamedEntityRecognizerAdapter.correct','scfNamedEntityCorrect','SCFAdapterImplConfig',null,'NLP_APP_ID','USER_ID','BRANCH_CODE','NER_USE_CASE_NAME','ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
(MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
values
(('INVOICE','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.create','invoiceDocumentUpload','InvoiceAdapterImplConfig',null,'DMS_APP_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');

- Insert into DIGX_MO_CONFIGURATION_SELECTOR
 (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
 values
 ('INVOICE','com.ofss.digx.extxfce.midoffice.content.adapter.IContentMOAdapter.read','invoiceDocumentDownload','InvoiceAdapterImplConfig',null,'DMS_APP_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
 (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
 values
 ('INVOICE','com.ofss.digx.extxfce.midoffice.content.adapter.IContentMOAdapter.listDocuments','invoiceDocumentsList','InvoiceAdapterImplConfig',null,'DMS_APP_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
 (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
 values
 ('INVOICE','com.ofss.digx.extxfce.midoffice.content.adapter.IContentMOAdapter.delete','invoiceDocumentDelete','InvoiceAdapterImplConfig',null,'DMS_APP_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
 (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
 values
 ('INVOICE','com.ofss.digx.extxfce.nlp.adapter.namedentityrecognizer.INamedEntityRecognizerAdapter.extract','invoiceNamedEntityExtract','InvoiceAdapterImplConfig',null,'NLP_APP_ID','USER_ID','BRANCH_CODE','NER_USE_CASE_NAME','ENTITY_ID');
- Insert into DIGX_MO_CONFIGURATION_SELECTOR
 (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_TYPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID)
 values
 ('INVOICE','com.ofss.digx.extxfce.nlp.adapter.namedentityrecognizer.INamedEntityRecognizerAdapter.correct','invoiceNamedEntityCorrect','InvoiceAdapterImplConfig',null,'NLP_APP_ID','USER_ID','BRANCH_CODE','NER_USE_CASE_NAME','ENTITY_ID');

Table definition is already available in svn, please find below table description:

```
CREATE TABLE DIGX_MO_CONFIGURATION_SELECTOR(  
    MODULE_IDENTIFIER VARCHAR2(100) not null,  
    QUALIFIED_METHOD_NAME VARCHAR2(200) not null,  
    SERVICE_ID VARCHAR2(200) not null,  
    ADAPTER_IMPL VARCHAR2(100) not null,  
    TRANSACTION_TYPE VARCHAR2(100),  
    APPLICATION_ID VARCHAR2(500) not null,  
    USER_ID VARCHAR2(500) not null,  
    BRANCH_CODE VARCHAR2(500) not null,  
    USE_CASE_NAME VARCHAR2(100),  
    ENTITY_ID VARCHAR2(500)  
)
```

Primary Key:

Module Identifier – Module ID for which configuration overriding is being done.

Qualified Method Name – Fully qualified name of the Host service (from Interface) for which configuration overriding is being done.

Remaining fields/columns:

These fields contain the configuration variable names which are typically required to call the Host/mid-office service.

Note: The table and scripts are only required for the Complex Scenario and not for the Ideal Scenario. Thus, these scripts are not committed, but need to be executed as and when required

[Home](#)

5. Use Case Name for NLP

Only for NLP transactions, apart from standard headers, an extra field is required: “Use Case Name”. It represents the NLP use case to be used for your module from the list of use cases created in NLP system.

If the use case is not provided, then NLP system tries to identify the use case based on the document uploaded. Thus to avoid this, the use case name needs to be provided.

This will cause an issue for **Ideal Scenario**. These scripts are not required for ideal scenario. The use case names can then be provided using the following two options:

1. In the request DTO “NamedEntityRecognizerDTO.java” accepted by extract service of NLP, there is a field “useCaseName”. In this field, use case name can be provided. For this, UI should be modified using UI extensibility to send this field in the payload.
2. Make the entries in the table “DIGX_MO_CONFIGURATION_SELECTOR” only for NLP transactions and specify the use case name. For other fields, variable names of default configurations can be used, only for this case.

[Home](#)