# Oracle® Banking APIs Cloud Service

## Content and NLP Configurations Guide

Release 25.1.0.0.0
G28950-01
April 2025

Oracle Banking APIs Cloud Service Content and NLP Configurations Guide, Release 25.1.0.0.0

G28950-01

# Contents

# Preface



## Purpose

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

# 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 Introduction

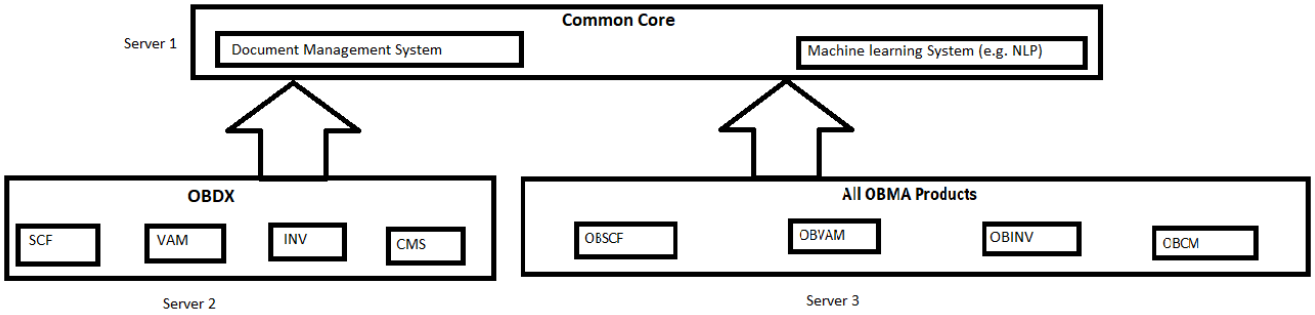
Document Management System (DMS) is a common core product which stores the documents either uploaded from channel banking product (e.g. OBDX) or uploaded from any Oracle Banking Mid Office products (e.g. OBSCF, OBCM, OBVAM, and so on) and returns document ID for future access.

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).

# 2 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 OBDX and OBMA products.

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).




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

1. For Content transactions, OBDX 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, OBDX 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.

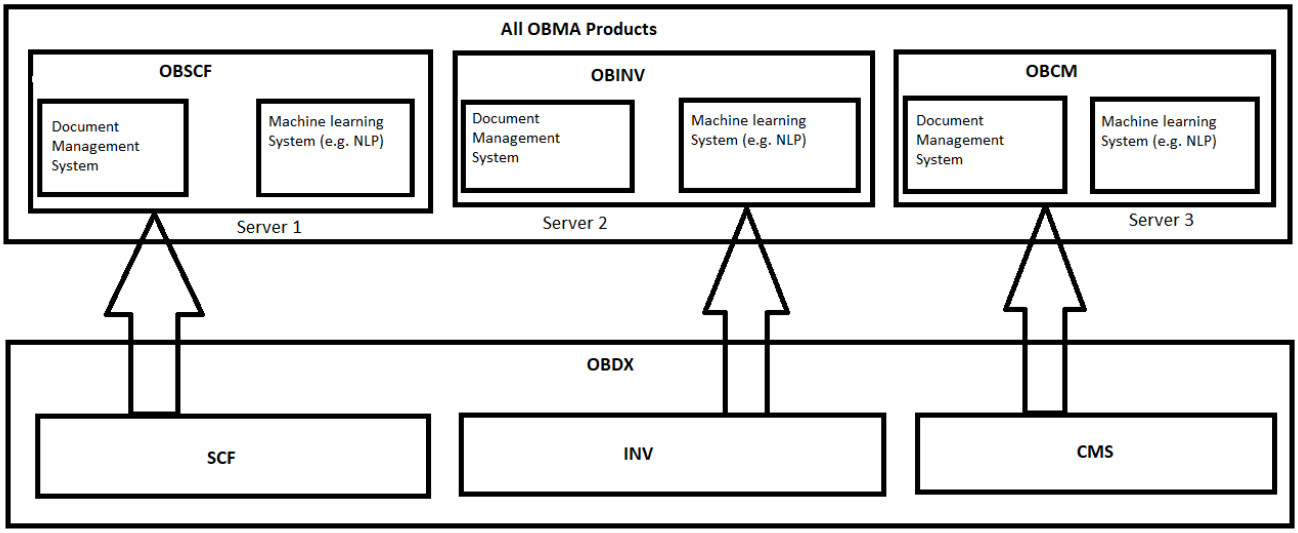
The problem arises in Complex Scenario.

# 3
# 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 OBDX:

- 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_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('SCF','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.cre
  ate','scfDocumentUpload','SCFAdapterImplConfig',null,'DMS_APP_ID','USER_ID','B
  RANCH_CODE',null,'ENTITY_ID');
  ```
- ```
  Insert into DIGX_MO_CONFIGURATION_SELECTOR
  (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('SCF','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.rea
  d','scfDocumentDownload','SCFAdapterImplConfig',null,'DMS_APP_ID','USER_ID','B
  RANCH_CODE',null,'ENTITY_ID');
  ```
- ```
  Insert into DIGX_MO_CONFIGURATION_SELECTOR
  (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('SCF','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.lis
  tDocuments','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_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('SCF','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter.del
  ete','scfDocumentDelete','SCFAdapterImplConfig',null,'DMS_APP_ID','USER_ID','B
  RANCH_CODE',null,'ENTITY_ID');
  ```
- ```
  Insert into DIGX_MO_CONFIGURATION_SELECTOR
  (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('SCF','com.ofss.digx.extxface.nlp.adapter.namedentityrecognizer.INamedEntityR
  ecognizerAdapter.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_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('SCF','com.ofss.digx.extxface.nlp.adapter.namedentityrecognizer.INamedEntityR
  ecognizerAdapter.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_T
  YPE,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_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('INVOICE','com.ofss.digx.extxface.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_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('INVOICE','com.ofss.digx.extxface.midoffice.content.adapter.IContentMOAdapter
  .listDocuments','invoiceDocumentsList','InvoiceAdapterImplConfig',null,'DMS_AP
  P_ID','USER_ID','BRANCH_CODE',null,'ENTITY_ID');
  ```
- ```
  Insert into DIGX_MO_CONFIGURATION_SELECTOR
  (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('INVOICE','com.ofss.digx.extxface.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_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('INVOICE','com.ofss.digx.extxface.nlp.adapter.namedentityrecognizer.INamedEnt
  ityRecognizerAdapter.extract','invoiceNamedEntityExtract','InvoiceAdapterImplC
  onfig',null,'NLP_APP_ID','USER_ID','BRANCH_CODE','NER_USE_CASE_NAME','ENTITY_I
  D');
  ```
- ```
  Insert into DIGX_MO_CONFIGURATION_SELECTOR
  (MODULE_IDENTIFIER,QUALIFIED_METHOD_NAME,SERVICE_ID,ADAPTER_IMPL,TRANSACTION_T
  YPE,APPLICATION_ID,USER_ID,BRANCH_CODE,USE_CASE_NAME,ENTITY_ID) values
  ('INVOICE','com.ofss.digx.extxface.nlp.adapter.namedentityrecognizer.INamedEnt
  ityRecognizerAdapter.correct','invoiceNamedEntityCorrect','InvoiceAdapterImplC
  onfig',null,'NLP_APP_ID','USER_ID','BRANCH_CODE','NER_USE_CASE_NAME','ENTITY_I
  D');
  ```

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

4

# 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.

# Index