

# Oracle® Banking APIs

## Mid-Office Product Setup and Configuration Guide



Release 25.1.0.0.0  
G28241-01  
April 2025

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                     | v  |
| Audience                    | v  |
| Documentation Accessibility | v  |
| Critical Patches            | v  |
| Diversity and Inclusion     | vi |
| Conventions                 | vi |
| Related Resources           | vi |
| Screenshot Disclaimer       | vi |
| Acronyms and Abbreviations  | vi |

## 1 Introduction

---

## 2 Trade Finance

---

|                                                               |     |
|---------------------------------------------------------------|-----|
| 2.1 Oracle Banking Trade Finance Process Management (OBTFFPM) | 2-1 |
| 2.1.1 Mandatory Executions                                    | 2-1 |

## 3 Corporate Lending

---

|                        |     |
|------------------------|-----|
| 3.1 Day One Executions | 3-1 |
|------------------------|-----|

## 4 Supply Chain Finance

---

|                                            |     |
|--------------------------------------------|-----|
| 4.1 Non Customer Onboarding Using Chaining | 4-1 |
| 4.2 OBRH Integration                       | 4-3 |

## 5 Receivables/Payables Management

---

|                      |     |
|----------------------|-----|
| 5.1 OBRH Integration | 5-1 |
|----------------------|-----|

|     |                                   |     |
|-----|-----------------------------------|-----|
| 6   | Virtual Account Management        |     |
| 6.1 | OBRH Integration                  | 6-1 |
| 6.2 | Verify System Configurations      | 6-4 |
| 6.3 | Enumerations                      | 6-4 |
| 6.4 | Adapter Properties                | 6-4 |
| 6.5 | Cloud specific Configurations     | 6-4 |
| 6.6 | OBVAM to OBAPI Error code mapping | 6-4 |
| 7   | Cash Management System            |     |
| 7.1 | OBRH Integration                  | 7-1 |
| 8   | Credit Facility Management        |     |
| 8.1 | Core as Third Party               | 8-1 |
| 8.2 | OBRH Configurations               | 8-1 |
| 9   | Liquidity Management              |     |
| 9.1 | OBRH Configurations               | 9-1 |
| 9.2 | Verify System Configurations      | 9-3 |
| 9.3 | Enumerations                      | 9-3 |
| 9.4 | Adapter Properties                | 9-4 |
| 9.5 | Simulation IC Group maintenance   | 9-4 |
| 9.6 | Cloud Specific Configurations     | 9-5 |
| 9.7 | Host Notifications                | 9-5 |
| 9.8 | OBLM to OBAPI Error code mapping  | 9-6 |
| 10  | User Credential Configuration     |     |
| 11  | OBRH Integration Configuration    |     |
|     | 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 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.

## 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                                                                                                                                |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>boldface</b>        | 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

## Introduction

This document is intended for setting up OBAPI 22.1.0.0.0 with different Mid-Office Products.



# 2

## Trade Finance

- [Oracle Banking Trade Finance Process Management \(OBTFPM\)](#)  
Following things need to be done to enable OBAPI Trade Finance with OBTFPM

### 2.1 Oracle Banking Trade Finance Process Management (OBTFPM)

Following things need to be done to enable OBAPI Trade Finance with OBTFPM

- [Mandatory Executions](#)

#### 2.1.1 Mandatory Executions

Execute the following script at OBAPI database and restart the managed server.

```
../installables/db/OBTFPM/DIGX_FW_CONFIG_ALL_O.sql
```

**Note:**

'%ENTITY\_ID%' should be replaced with entity identifier (For example 'OBDX\_BU').

# 3

## Corporate Lending

- [Day One Executions](#)

### 3.1 Day One Executions

Following script needs to be executed post installation for Corporate Lending with OBCL 14.3.0.0.0 release

```
UPDATE digx_fw_config_all_o
SET PROP_VALUE = CONCAT('OBCL14.1,', (select PROP_VALUE from
digx_fw_config_all_o where PROP_ID
like '&ENTITY_ID')) WHERE PROP_ID LIKE '&ENTITY_ID';
```

Scripts mentioned in below path also needs to be executed:

../installables/db/OBCL/version/DIGX\_FW\_CONFIG\_ALL\_O.sql for OBCL APIs and

../installables/db/OBCLPM/version/DIGX\_FW\_CONFIG\_ALL\_O.sql for OBCLPM APIs.



#### Note:

'%ENTITY\_ID%' should be replaced with entity identifier (For example 'OBDX\_BU')

The list of OBCL and OBCLPM APIs that are integrated with OBAPI using OBRH are as follows:

| Interface ID                   | Transaction Name         | Description                                                                   |
|--------------------------------|--------------------------|-------------------------------------------------------------------------------|
| CORPORATE_LOAN_ROLLOVER        | Loan Rollover            | The API is used to post the roll over for a specific loan to Loans Mid Office |
| CORPORATE_BULK_LOAN_SETTLEMENT | Multiple Loan Settlement | This API is used to post single and bulk loan settlement to Loans Mid Office  |
| LOAN_DISBURSEMENT_DETAILS      | Disbursement Details     | This API is used to fetch disbursement details from Loans Back Office         |
| LOAN_SCHEDULE_DETAILS          | Schedule Details         | This API is used to fetch schedule details from Loans Back Office             |
| LOAN_OUTSTANDING_DETAILS       | Outstanding Details      | This API is used to fetch outstanding details from Loans Back Office          |
| LOAN_DETAILS                   | Loan Details             | This API is used to fetch loan details from Loans Back Office                 |

| Interface ID                            | Transaction Name           | Description                                                                               |
|-----------------------------------------|----------------------------|-------------------------------------------------------------------------------------------|
| LOAN_ACCOUNT_LIST                       | Loan Account List          | This API is used to fetch list of accounts from Loans Back Office                         |
| LOAN_RATEREVISION_DETAILS               | Rate Revision Details      | This API is used to fetch rate revision details from Loans Back Office                    |
| LOAN_FULL_OUTSTANDING_DETAILS           | Full Outstanding Details   | This API is used to fetch full outstanding details from Loans Back Office                 |
| CORPORATE_LOAN_SWIFT_MESSAGE_OR_ADVICES | Swift Message or Advices   | This API is used to fetch corporate loan swift message and advices from Loans Back Office |
| CORPORATE_LOAN_SETTLEMENT_SIMULATION    | Loan Settlement Simulation | This API is used to fetch corporate loan settlement simulation from Loans Back Office     |
| CORPORATE_LOAN_SETTLEMENT               | Loan Settlement            | This API is used to post corporate loan settlement from Loans Back Office                 |
| LOAN_PRODUCT_PROCESSING_CHARGES         | Processing Charges         | This API is used to fetch processing charges of a product from Loans Back Office          |
| LOAN_DRAWDOWN_APPLICATION               | Drawdown Application       | This API is used to post drawdown application to Loans Mid Office                         |
| LOAN_APP_FETCH_APPLICATION_STATUS       | Application Status         | This API is used to fetch status of a drawdown application from Loans Mid Office          |
| LOAN_APP_PRODUCT_DETAILS                | Product Details            | This API is used to fetch product details from Loans Mid Office                           |
| LOAN_APP_PRODUCT_LIST                   | Product List               | This API is used to fetch product list from Loans Mid Office                              |
| LOAN_APP_PRODUCT_SEGMENT                | Product Segment            | This API is used to fetch product segment from Loans Mid Office                           |
| LOAN_APP_PURPOSE_DETAILS                | Purpose List               | This API is used to fetch purpose list from Loans Mid Office                              |

# 4

## Supply Chain Finance

- [Non Customer Onboarding Using Chaining](#)
- [OBRH Integration](#)

### 4.1 Non Customer Onboarding Using Chaining

Now in the standard scenario, the core system contains the Customer data and the OBSCF mid office system contains the Non Customer data. Thus in order to onboard a noncustomer (give channel access) the system needs to inquire in OBSCF mid office.

But till now the system was inquiring only in Core system, which we still need for the onboarding of customers. Thus a concept of chaining is introduced where for a given corporate, the system will first inquire in Core system and if found then the given corporate is a customer but if not found then the system will inquire in OBSCF mid office system and if found there then the given corporate is a noncustomer.

Now, the chaining is not only implemented for 2 levels (calling only 2 systems) but it can be implemented for n levels. Also there is a provision to break a chain at any level or if there is a case that there is an overridden adapter to call a common system containing both customers and noncustomers and not want to call core system and mid office system adapters i.e. not implement/require chaining at all, then this is also possible.

For detail explanation of Chaining, how it works, chaining in case of overridden adapters and many more please refer **Chaining Section in Extensibility Document**.

Now below are the scenarios of how chaining will be used for Non Customer Onboarding in case of different possible implementations at Bank.

Considering, IPartyAdapter has three implementation

i1 - PartyAdapter(UBS), i2 - PartyAdapter(ASP) & i3 - PartyAdapter(TP)

#### **Case 1:**

Bank has both UBS core entity and ASP mid office as well (OBASP).

In this case, the entry for UBS core entity in DIGX\_FW\_CONFIG\_ALL\_O will be like (Assuming OBDX\_BU is the determinant value for UBS core entity)

OBDX\_BU | UBS, ASP, TP



#### **Note:**

Here entry of TP might be for other mid offices system but not for UBS Core and OBSCF mid office as both are present with the bank according to the case.

Thus in case of chaining, it will first inquire in "i1 Adapter", if found then it will stop and return the result. If not found then it will inquire in "i2 Adapter", if found then it will stop and return the result. If not found then it will inquire in "i3 Adapter", where there are maximum chances that it

won't be found because of above note. Thus finally after not able to find in "i3 Adapter", it will throw the error like it used to throw before chaining when not found in core system.

### Case 2:

Bank has UBS core entity but ASP mid office is Third Party.

In this case, the entry for UBS core entity in DIGX\_FW\_CONFIG\_ALL\_O will be like (Assuming OBDX\_BU is the determinant value for UBS core entity)

OBDX\_BU | UBS, TP

Thus in case of chaining, it will first inquire in "i1 Adapter", if found then it will stop and return the result. If not found then it will inquire in "i3 Adapter". Now in case of i3, it will push the request in JMS queue which will be read by one of the middleware implementations and sent to the actual Third party mid office of ASP. If found then the result will returned but if not then it will throw the error like it used to throw before chaining when not found in core system.

\*\* In this case, all the other functionalities of ASP mid office like Onboarding Associated Party will also be achieved by using Third Party implementation of ASP.

### Case 3:

Bank has Third party core entity and Third Party ASP mid office

In this case, the entry for Third Party core entity in DIGX\_FW\_CONFIG\_ALL\_O will be like (Assuming OBDXBU1 is the determinant value for Third Party core entity)

OBDXBU1 | TP

Thus, here there is no scenarios of chaining as always only "i3 Adapter" will be picked. Now in case of i3, it will push the request in JMS queue which will be read by one of the middleware implementations and sent to the actual single system (like common core). Now that single system can have the logic to check the party in core system & ASP system if required.

\*\* In this case, all the other functionalities of ASP mid office like Onboarding Associated Party will also be achieved by using Third Party implementation of ASP.

### Case 4:

Bank has Third Party core entity but ASP mid office is of OBASP

In this case, the entry for Third Party core entity in DIGX\_FW\_CONFIG\_ALL\_O will be like (Assuming OBDXBU1 is the determinant value for Third Party core entity)

OBDXBU1 | TP, ASP

Thus in case of chaining, it will first inquire in "i3 Adapter". Now in case of i3, it will push the request in JMS queue which will be read by one of the middleware implementations and sent to the actual Third party core system. If found then it will stop and return the result. If not found then it will inquire in "i2 Adapter". If found then the result will be returned but if not then then it will throw the error like it used to throw before chaining when not found in core system.

\*\* In this case, all the other functionalities of ASP mid office like Onboarding Associated Party should only be achieved by ASP host implementation (one that is qualified with OBASP). For that, **we need to override the scripts of host adapter in DIGX\_FW\_CONFIG\_ALL\_O such that for ASP functionalities it will always pick the ASP adapter and for other common functionalities like Get Non Customer party, chaining will be applied as explained above.**

We need to execute below script to fetch mid office token required for Purchase Order File Upload.

```
INSERT INTO DIGX_FW_CONFIG_OUT_RS_CFG_B
(SERVICE_ID, CONTEXT_URL, SERVICE_URL, REQUEST_MEDIA_TYPE,
RESPONSE_MEDIA_TYPE, AUTHENTICATION, AUTH_TYPE, CREDENTIAL_STORE_TYPE,
CREDENTIAL_STORE_KEY, CREATION_DATE, LAST_UPDATED_DATE) VALUES
('tokenOBSCF144',
'http://{OBSCF_HOST_IP}:{OBSCF_HOST_PORT}', 'api-gateway/
platojwtauth', 'application/json',
'application/json', 'N', 'Bearer', 'credential_impl', 'OBSCF_14.4', sysdate,
sysdate);
```

## 4.2 OBRH Integration

During Bank Implementation, assuming OBRH is installed and configured as part of OBSCF installation.

For OBAPI and OBSCF integration using OBRH the following configurations need to be done.

1. Carry out all the steps mentioned in **OBRH Integration Configuration** section in this document. The service provider for product OBSCF (Oracle Banking Supply Chain Finance) end-points configured in OBRH is “OBSCF”
2. After all the above steps are completed, user needs to execute some scripts for the host APIs that are to be consumed via OBRH. These are the scripts to pick the third party adapter implementation instead of the host specific implementations as well as to call the OBRH end-point for the configured interfaces. The scripts are available at the following location:

/Installables/Modules/OBSCF/DIGX\_FW\_CONFIG\_ALL\_O.sql

### Note:

'%ENTITY\_ID%' should be replaced with the entity identifier (For Example if the entity during implementation is OBDX\_BU then '%ENTITY\_ID%' should be replaced by 'OBDX\_BU').

3. The list of OBSCF APIs that are integrated with OBAPI using OBRH is as follows:

| OBRH Consumer Service Name | Transaction Name             | Description                                        |
|----------------------------|------------------------------|----------------------------------------------------|
| SCF_FINANCE_CHARGE         | View/Edit Invoice            | Fetches the list of finance charges.               |
| SCF_PURCHASE_ORDER_CREATE  | Purchase Order Creation      | This API is used to create purchase orders.        |
| SCF_PURCHASE_ORDER_UPDATE  | View Purchase Orders         | This API is used to modify purchase order details. |
| SCF_PURCHASE_ORDER_ACCEPT  | Accept/Reject Purchase Order | This API is used to accept purchase orders.        |
| SCF_PURCHASE_ORDER_REJECT  | Accept/Reject Purchase Order | This API is used to reject purchase orders.        |

| OBRH Consumer Service Name      | Transaction Name                     | Description                                                      |
|---------------------------------|--------------------------------------|------------------------------------------------------------------|
| SCF_PURCHASE_ORDER_CANCEL       | View Purchase Orders                 | This API is used to cancel purchase orders.                      |
| SCF_PURCHASE_ORDER_LIST         | View Purchase Orders                 | Fetches the list of purchase orders.                             |
| SCF_PURCHASE_ORDER_READ         | View Purchase Orders                 | Fetches purchase order details.                                  |
| SCF_LINKED_PO_LIST              | View Finance Details                 | Fetches Purchase Orders linked to a finance.                     |
| SCF_PO_FINANCE_CREATE           | Request Finance                      | This API is used to request finance on purchase order(s).        |
| SCF_FINANCE_LIMITS_LIST         | View Limits                          | Fetches the list of finance limits.                              |
| SCF_MAIN_LIST                   | View Limits                          | Fetches the list of supply chain finance maintenances for a key. |
| SCF_PROGRAM_LIST                | View/Edit Program                    | Fetches the list of programs.                                    |
| SCF_PROGRAM_READ                | Vide/Edit Program                    | Fetches program details.                                         |
| SCF_PROGRAMPRODUCT_LIST         | Create Program                       | Fetches the list of Program products.                            |
| SCF_PROGRAMPRODUCT_READ         | Create Program                       | Fetches program product details.                                 |
| SCF_PROGRAM_CREATE              | Create Program                       | Creates Program                                                  |
| SCF_PROGRAM_UPDATE              | Edit Program                         | Edit program details                                             |
| SCF_LINKED_FINANCE_LIST         | View Invoice Details                 | Fetches finances linked to an invoice.                           |
| SCF_FINANCE_CREATE              | Request Finance                      | This API is used to request finance on invoice(s).               |
| SCF_FINANCE_REPAYMENT           | Repay Finance                        | Initiates request to repay finance.                              |
| SCF_FINANCE_SETTLEMENT          | View Finance Details                 | Fetches finance settlement details                               |
| SCF_FINANCE_READ                | View Finance Details                 | Fetches finance details                                          |
| SCF_FINANCE_LIST                | View Finance                         | Fetches list of finances                                         |
| SCF_LINKED_INVOICE_LIST         | View Finance Details                 | Fetches invoices linked to a finance.                            |
| SCF_LINK_INVOICE_TO_PROGR<br>AM | Link Invoice To Program              | Links Invoice(s) to Program                                      |
| SCF_DISCOUNT_OFFER_CREATE       | Create Discount Offer                | Creates Discount Offer                                           |
| SCF_DISCOUNTOFFER_LIST          | View Discount Offer                  | Fetches list of discount offers                                  |
| SCF_DISCOUNT_OFFER_READ         | View Discount Offer Details          | Fetch discount offer details                                     |
| SCF_LINKED_DISCOUNT_OFFER<br>S  | View Receivables/Payables<br>Details | Fetches list of offers linked to an invoice                      |
| SCF_CHARGE_CALCULATION          | Manage Receivables/Payables          | Fetches applicable discount on an invoice                        |

This completes the entire configuration needed for consuming OBSCF APIs in OBAPI through OBRH.

4. In addition to the above, below script needs to be executed for successful creation of Discount Offer.

```
insert into DIGX_FW_TAXONOMY_DATA_TYPE_MAP
(ID,TYPE,DATATYPEID,MINLENGTH,MAXLENGTH,MANDATORY,ERRORCODE,LENGTH_ERROCODE
,MANDATORY_ERRORCODE,
CREATION_DATE,CREATED_BY, LAST_UPDATED_DATE, LAST_UPDATED_BY, OBJECT_VERSION_N
UMBER) values

('com.ofss.digx.app.scf.dto.discountoffer.DiscountOfferRequestDTO.discountO
ffer.invoices.indicator',
'CLASS','FREETEXT',null,null,'N',null,null,null,sysdate,'ofssuser',sysdate,
'ofssuser',1);
```



# 5

## Receivables/Payables Management

- [OBRH Integration](#)

### 5.1 OBRH Integration

During Bank Implementation, assuming OBRH is installed and configured as part of either OBCM or OBSCF installation.

For OBAPI and OBSCFCM integration using OBRH the following configurations need to be done.

1. Carry out all the steps mentioned in **OBRH Integration Configuration** section in this document. The service provider for product OBSCFCM (Oracle Banking Cash Management or Oracle Banking Supply Chain Finance) end-points configured in OBRH is “ASP”, “INV” and “SCFCM”.
2. After all the above steps are completed, user needs to execute some scripts for the host APIs that are to be consumed via OBRH. These are the scripts to pick the third party adapter implementation instead of the host specific implementations as well as to call the OBRH end-point for the configured interfaces. The scripts are available at the following location:

```
/Installables/db/OBSCFCM/version/DIGX_FW_CONFIG_ALL_0.sql
```

#### Note:

‘%ENTITY\_ID%’ should be replaced with the entity identifier (For Example if the entity during implementation is OBDX\_BU then ‘%ENTITY\_ID%’ should be replaced by ‘OBDX\_BU’).

3. The list of OBSCFCM APIs that are integrated with OBAPI using OBRH is as follows:

| OBRH Consumer Service Name | Transaction Name                     | Description                                                                                             |
|----------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------|
| ASP_ASSOCIATEDPARTY_CREATE | Onboard Associated Party             | This API is used to onboard an associated party.                                                        |
| ASP_ASSOCIATEDPARTY_LIST   | View Associated Parties              | Fetches the list of associated parties.                                                                 |
| ASP_ASSOCIATEDPARTY_READ   | View Associated Parties              | Fetches associated party details.                                                                       |
| ASP_ASSOCIATEDPARTY_UPDATE | Upload KYC document for Non-Customer | This API is used to update the document Id for a new associated party that is not a customer of a bank. |

| OBRH Consumer Service Name           | Transaction Name                          | Description                                                                                                                    |
|--------------------------------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| ASP_ASSOCIATEDPARTY_RELATIONSHIP     | Fetch Buyer-Supplier Relationship details | This interface is used to fetch details of buyer-supplier relationship                                                         |
| INV_INVOICES_CREATE                  | Create Receivables/Payables               | This API is used to create invoices.                                                                                           |
| INV_INVOICE_LIST                     | View/Edit Receivables/Payables            | This API is used to fetch invoices                                                                                             |
| INV_INVOICE_READ                     | View/Edit Receivables/Payables            | This API is used to fetch invoice details                                                                                      |
| INV_INVOICES_UPDATE_STATUSES         | Manage Receivables/Payables               | This API allows a user to perform various operations on invoices like Edit, Cancel, Accept, Raise Dispute, Resolve Dispute etc |
| INV_INVOICES_DISPUTE_REASONS         | Manage Receivables/Payables               | This API fetches list of dispute reasons required to raise dispute on an invoice.                                              |
| INV_LIST_COMMODITIES                 | Create Receivables/Payables               | This API fetches list of supplier based commodities.                                                                           |
| INV_CREDIT_NOTE_CREATE               | Create Credit Note                        | This API is used to create credit notes                                                                                        |
| INV_CREDIT_NOTE_LIST                 | View Credit Note                          | This API is used to fetch credit notes                                                                                         |
| INV_CREDIT_NOTE_READ                 | View Credit Note                          | This API is used to fetch credit note details                                                                                  |
| SCFCM_PARAMS_LIST                    | Onboard Associated Party                  | This API fetches application params.                                                                                           |
| SCFCM_PAYMENTS_LIST                  | View Payments                             | This API is used to fetch the list of payments.                                                                                |
| SCFCM_PAYMENT_READ                   | View Payment Details                      | This API is used to fetch payment details.                                                                                     |
| SCFCM_MANUAL_RECONCILIATION          | Manual Reconciliation                     | This API is used to manually reconcile cashflows/invoices with payments                                                        |
| SCFCM_LIST_RECONCILED_TRANSACTIONS   | De-Reconciliation                         | This API is used to fetch the list of reconciled cashflows/invoices.                                                           |
| SCFCM_DERECONCILE                    | De-Reconciliation                         | This API is used to de-reconcile already reconciled cashflows/invoices.                                                        |
| SCFCM_RECONCILIATION_RULE_LIST       | View/Edit Reconciliation Rules            | Fetches list of reconciliation rules maintained for a party.                                                                   |
| SCFCM_RECONCILIATION_RULE_CREATE     | Create Reconciliation Rule                | This API is used to create reconciliation/allocation rule for a party.                                                         |
| SCFCM_RECONCILIATION_RULE_UPDATE     | Edit Reconciliation Rule                  | This API is used to modify reconciliation/allocation rule details for a party.                                                 |
| SCFCM_RECONCILIATION_RULE_READ       | View Reconciliation Rule details          | Fetches reconciliation rule details.                                                                                           |
| SCFCM_LIST_RECONCILIATION_CATEGORIES | View/Edit Reconciliation Rules            | Fetches reconciliation categories                                                                                              |

| OBRH Consumer Service Name           | Transaction Name               | Description                                                                            |
|--------------------------------------|--------------------------------|----------------------------------------------------------------------------------------|
| SCFCM_LIST_RECONCILIATION_ATTRIBUTES | View/Edit Reconciliation Rules | Fetches reconciliation attributes                                                      |
| SCFCM_LIST_ALLOCATED_TRANSACTIONS    | View Payment Details           | This API is used to fetch allocation details of a payment.                             |
| SCFCM_LIST_ALLOCATION_ACCOUNTS       | Manual Allocation              | This API is used to fetch virtual accounts which can be further allocated to payments. |
| SCFCM_MANUAL_ALLOCATION              | Manual Allocation              | This API is used to manually allocate payments to virtual account.                     |

This completes the entire configuration needed for consuming OBSCFCM APIs in OBAPI through OBRH.

We need to execute below script to fetch mid office token required for fetching associated party information for non-customer.

```
INSERT INTO DIGX_FW_CONFIG_OUT_RS_CFG_B
(SERVICE_ID, CONTEXT_URL, SERVICE_URL,
REQUEST_MEDIA_TYPE, RESPONSE_MEDIA_TYPE,
AUTHENTICATION, AUTH_TYPE, CREDENTIAL_STORE_TYPE, CREDENTIAL_STORE_KEY,
CREATION_DATE,
LAST_UPDATED_DATE) VALUES ('tokenASP144', 'http://${ASP_HOST_IP}:${
{ASP_HOST_PORT}}',
'api-gateway/platojwtauth', 'application/json', 'application/json', 'N',
'Bearer',
'credential_impl', 'ASP_14.4', sysdate, sysdate);
```

We need to execute below script to fetch mid office token required for Invoice/Debit-Credit Note File Upload.

```
INSERT INTO DIGX_FW_CONFIG_OUT_RS_CFG_B
(SERVICE_ID, CONTEXT_URL, SERVICE_URL,
REQUEST_MEDIA_TYPE, RESPONSE_MEDIA_TYPE, AUTHENTICATION,
AUTH_TYPE, CREDENTIAL_STORE_TYPE, CREDENTIAL_STORE_KEY, CREATION_DATE,
LAST_UPDATED_DATE)
VALUES ('tokenINV144', 'http://${INV_HOST_IP}:${INV_HOST_PORT}', 'api-gateway/
platojwtauth',
'application/json', 'application/json', 'N', 'Bearer', 'credential_impl',
'INV_14.4',
sysdate, sysdate);
```

# 6

## Virtual Account Management

- [OBRH Integration](#)
- [Verify System Configurations](#)
- [Enumerations](#)
- [Adapter Properties](#)
- [Cloud specific Configurations](#)
- [OBVAM to OBAPI Error code mapping](#)

### 6.1 OBRH Integration

During Bank Implementation, assuming OBRH is installed and configured as part of OBVAM installation.

For OBAPI and OBVAM integration using OBRH the following configurations need to be done.

1. Carry out all the steps mentioned in **OBRH Integration Configuration** section in this document. The service provider for product processor OBVAM (Oracle Banking Virtual Account Management) end-points configured in OBRH is **OBVAM**

All the OBVAM APIs consumed from OBAPI are via OBRH. List is as follows:-

| OBRH Consumer Service Name        | Transaction Name                      |
|-----------------------------------|---------------------------------------|
| abortVirtualAccountClosure        | Terminate Virtual Account Closure     |
| closeRemittance                   | Close Remittance ID                   |
| closeVA                           | Close Virtual Account                 |
| closeVirtualAccountStructure      | Close Virtual Account Structure       |
| createCreditLineLinkage           | Create Internal Credit Line Linkage   |
| createGeneralRates                | Add Generate Rates                    |
| createInternalCreditLine          | Create Internal Credit Line           |
| createInternalTransfer            | Move Money                            |
| createRemittance                  | Create Remittance ID                  |
| createSpecialRates                | Add Special Rates                     |
| createVirtualAccount              | Create Virtual Account                |
| createVirtualAccountStructure     | Create Virtual Account Structure      |
| createVirtualEntity               | Create Virtual Entity                 |
| createVirtualMultiCurrencyAccount | Create Virtual Multi Currency Account |
| deleteCreditLineLinkage           | Delete Internal Credit Line Linkage   |
| deleteInternalCreditLine          | Delete Internal Credit Line           |
| deleteVirtualEntity               | Close Virtual Entity                  |
| deleteVirtualMultiCurrencyAccount | Close Virtual Multi Currency Account  |

| OBRH Consumer Service Name             | Transaction Name                                                     |
|----------------------------------------|----------------------------------------------------------------------|
| downloadVASChildAccounts               | Download Virtual Account Structure                                   |
| editVirtualMultiCurrencyAccount        | Edit Virtual Multi Currency Account                                  |
| fetchBranchDateByBranchCode            | Fetch Branch Date                                                    |
| fetchDefaultInterestRates              | Fetch Default Rates/UDEs                                             |
| fetchDistinctCurrencies                | Fetch Distinct Currencies (Projection)                               |
| fetchEntityBankParameters              | Fetch Entity Bank Parameters                                         |
| fetchInterestHistory                   | Fetch Interest Rates History                                         |
| fetchRateCodes                         | Fetch Rate Codes                                                     |
| fetchRates                             | Fetch Interest Rates (General/Special)                               |
| fetchRemittanceList                    | Fetch Remittance IDs                                                 |
| fetchVAMCountry                        | Fetch Countries (CMC)                                                |
| fetchVAMCurrency                       | Fetch Currencies (CMC)                                               |
| fetchVAMEnabledAccountsForParty        | Fetch VAM Enabled Real Accounts                                      |
| fetchVAStatement                       | Fetch Virtual Account Transactions                                   |
| fetchVAforLinkage                      | Fetch Virtual Accounts for Credit Line Linkage                       |
| fetchVAwithBalance                     | Fetch Virtual Account with Balance and Structure                     |
| fetchValueDatedBalances                | Fetch Value Dated Balances                                           |
| fetchVamChargeDefinitionList           | Fetch Charge Definitions                                             |
| fetchVamChargeList                     | Fetch Charges                                                        |
| fetchVirtualAccountBalances            | Fetch Virtual Account Balance                                        |
| fetchVirtualAccountBranches            | Fetch Virtual Account Branches                                       |
| fetchVirtualAccountList                | Fetch Virtual Accounts                                               |
| fetchVirtualMultiCurrencyAccount       | Fetch Virtual Multi Currency Accounts                                |
| fetchVirtualProduct                    | Fetch Virtual Account Products                                       |
| getAccruedAmountForVirtualAccount      | Fetch Interest Accrued Amount for Virtual Account                    |
| getChildAccountsForVirtualStructure    | Fetch Child Accounts for Parent Account in Virtual Account Structure |
| getIdentificationTypesForVirtualEntity | Fetch Identification Types for Virtual Entity                        |
| getLineAccountUtilization              | Fetch Internal Credit Line Utilization for Virtual Account           |
| getLineUtilization                     | Fetch Internal Credit Line Utilization                               |
| getRemitterCountForVI                  | Fetch Remittance ID count for Virtual Identifier                     |
| getUnmappedVirtualAccounts             | Fetch Virtual Accounts eligible for adding in Structure              |
| getVirtualAccountClosureStatusDetails  | Fetch Virtual Account Closure Status details                         |
| getVirtualAccountStructuresByCustomer  | Fetch Virtual Account Structures                                     |
| listCreditLineLinkage                  | Fetch Internal Credit Line Linkages                                  |
| listInternalCreditLine                 | Fetch Internal Credit Lines                                          |
| listVirtualAccWithStructureCode        | Fetch Virtual Accounts part of a Structure                           |

| OBRH Consumer Service Name             | Transaction Name                                                                         |
|----------------------------------------|------------------------------------------------------------------------------------------|
| listVirtualEntity                      | Fetch Virtual Entities                                                                   |
| listVirtualIdentifier                  | Fetch Virtual Identifiers                                                                |
| modifyVirtualAccountStructure          | Edit Virtual Account Structure                                                           |
| readInternalCreditLine                 | Fetch Internal Credit Line details                                                       |
| readVirtualAccount                     | Fetch Virtual Account details                                                            |
| readVirtualEntity                      | Fetch Virtual Entity details                                                             |
| readVirtualMultiCurrencyAccount        | Fetch Virtual Multi Currency details                                                     |
| reopenRemittance                       | Reopen Remittance ID                                                                     |
| reopeningClosedVirtualAccount          | Reopen Virtual Account                                                                   |
| retryVirtualAccountClosure             | Retry Virtual Account Closure                                                            |
| transferVirtualAccount                 | Fetch Transfer Virtual Accounts for Closure                                              |
| updateCreditLineLinkage                | Edit Internal Credit Line Linkage                                                        |
| updateInternalCreditLine               | Edit Internal Credit Line                                                                |
| updateRemittance                       | Edit Remittance ID                                                                       |
| updateVirtualAccount                   | Edit Virtual Account                                                                     |
| updateVirtualEntity                    | Edit Virtual Entity                                                                      |
| vaForClosure                           | Fetch Virtual Accounts eligible for Closure                                              |
| vaForClosureStatus                     | Fetch Virtual Accounts initiated for Closure along with Status                           |
| vamFetchAdhocCAMTReport                | Fetch Adhoc CAMT Statement                                                               |
| vamFetchAdhocMTReport                  | Fetch Adhoc MT Statement                                                                 |
| vamFetchAdhocPDFReport                 | Fetch Adhoc PDF Statement                                                                |
| vamFetchPreGenReport                   | Fetch Pre-generated Statement                                                            |
| virtualAccountsforInternalTransfer     | Fetch Virtual Accounts for Real Account                                                  |
| fetchVirtualAccountRestrictions        | Fetch Virtual Account Restrictions                                                       |
| maintainVirtualAccountRestriction      | Edit Virtual Account Restrictions                                                        |
| fetchVAMCurrencyWiseBalance            | Fetch currency wise consolidated balance of Virtual Accounts                             |
| listTopFiveVirtualAccountBalances      | Fetch five Virtual Accounts with highest balance for given criteria                      |
| listVirtualIdentifierTransaction       | Fetch list of Virtual Identifier Transactions                                            |
| downloadVirtualIdentifierTransaction   | Download Virtual Identifier Transactions                                                 |
| uploadFeedFile                         | To upload the bulk file via OBRH instead of direct call                                  |
| syncFeedFileStatus                     | To sync the status of uploaded bulk file and its records via OBRH instead of direct call |
| fetchVirtualProductBalanceRestrictions | Fetch Balance restrictions based on Virtual Account Product                              |

This completes the entire configuration needed for consuming OBVAM APIs in OBAPI through OBRH.

## 6.2 Verify System Configurations

Following script helps in listing the VAM specific System Configurations:

```
SELECT prop_id as
      PROPERTY_IN_DATABASE, NVL(SUBSTR(t.UI_definition,
INSTR(t.UI_definition, '"title")+9,
      INSTR(t.UI_definition, '"', '"')-11), t.UI_definition) AS
TITLE_ON_SCREEN, t.prop_value FROM
      digx_fw_config_var_b t where prop_id like '%VAM%' and module =
'OTHERMODULE' and
      determinant_value = '*'; -- Please enter correct determinant value
```

Ensure correct values are maintained against the above properties.

This maintenance can be done from the “System Configuration” admin screen or directly in DB schema.

## 6.3 Enumerations

Following VAM related enumerations are used in OBAPI . They are used to fetch the values on the OBAPI UI.

- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getIdentityTypes';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getCorporateTypes';`

The mapping of these OBAPI values to the corresponding OBVAM values can be found in next section.

## 6.4 Adapter Properties

Certain fields (Eg: Enumerations, Status etc) can have different values in OBAPI as compared to OBVAM.

The mapping of all such values between OBAPI and OBVAM can be found/maintained using the below script:-

```
select * from DIGX_FW_CONFIG_ADAPTER_PROP_B where host_id = 'OBVAM';
```

## 6.5 Cloud specific Configurations

Following additional configurations are required if OBDX and OBVAM are being hosted on cloud:

1. In OBRH, enable Eureka instance for OBVAM service provider.

## 6.6 OBVAM to OBAPI Error code mapping

1. OBVAM to OBAPI error code mappings are present in the database table `DIGX_FW_ERR_COD_MAP` where `MODULE_ID` is “VIRTUAL\_ACCOUNT\_MANAGEMENT”
2. Out of the box, the value in column `EXT_SYSTEM_ID` for all such rows would be **UBS14.5**.

3. The value in column **EXT\_SYSTEM\_ID** for all such rows will have to be modified during implementation, based on the value derived from below query

a. 

```
SELECT
    CONCAT(prop_value, (select prop_value from digx_fw_config_var_b
where prop_id =
    'HOST_VERSION' and determinant_value = '$entity_name$'))
EXT_SYSTEM_ID from
    digx_fw_config_var_b where prop_id = 'HOST_NAME' and
determinant_value = '$entity_name$'; --
    replace $entity_name$ with correct determinant_value.
```



# 7

## Cash Management System

- [OBRH Integration](#)

### 7.1 OBRH Integration

During Bank Implementation, assuming OBRH is installed and configured as part of OBCM installation.

For OBAPI and mid Office OBCM integration using OBRH the following configurations need to be done.

1. Carry out all the steps mentioned in **OBRH Integration Configuration** section in this document. The service provider for mid-office product OBCM (Oracle Banking Cash Management) end-points configured in OBRH is “OBCM”
2. After all the above steps are completed, user needs to execute some scripts for the host APIs that are to be consumed via OBRH. These are the scripts to pick the third party adapter implementation instead of the host specific implementations as well as to call the OBRH end-point for the configured interfaces. The scripts are available at the following location:

```
/Installables/db/OBCM/version/DIGX_FW_CONFIG_ALL_0.sql
```

#### Note:

'%ENTITY\_ID%' should be replaced with the entity identifier (For Example if the entity during implementation is OBDX\_BU then '%ENTITY\_ID%' should be replaced by 'OBDX\_BU').

3. The list of OBCM APIs that are integrated with OBAPI using OBRH is as follows:

| OBRH Consumer Service Name      | Transaction Name                     | Description                                                                    |
|---------------------------------|--------------------------------------|--------------------------------------------------------------------------------|
| CMS_CASHFLOW_TRANSACTION_READ   | View/Edit Expected Cash Flow Details | Fetches Cashflow Details                                                       |
| CMS_CASHFLOW_TRANSACTION_UPDATE | View/Edit Expected Cash Flow Details | This API is used to modify expected cashflow details.                          |
| CMS_CASHFLOW_FETCH              | Cashflow Forecasting                 | This API is used to fetch cashflow forecasting data.                           |
| CMS_CCM_LIST                    | Cash Deposits                        | This API is used to fetch collection maintenance details of a cash management. |
| CMS_DIVISION_CODE_LIST          | Cash & Cheque Deposits               | This API is used to fetch division code details of a given party.              |

| OBRH Consumer Service Name    | Transaction Name                  | Description                                                                                     |
|-------------------------------|-----------------------------------|-------------------------------------------------------------------------------------------------|
| CMS_CASH_DEPOSITS_CREATE      | Cash Deposits                     | This API is used to create multiple cash deposits for a party.                                  |
| CMS_CASH_DEPOSITS_LIST        | View Cash Deposits                | This API is used to fetch cash collections.                                                     |
| CMS_CASH_DEPOSIT_READ         | View Cash Deposit Details         | This API is used to fetch cash collection details.                                              |
| CMS_CHEQUE_LIST               | View Cheque Deposits              | This API is used to fetch cheque collections.                                                   |
| CMS_CHEQUE_READ               | View Cheque Deposit Details       | This API is used to fetch cheque collection details.                                            |
| CMS_BRANCH_DENOMINATION_LIST  | Cash Deposits                     | This API is used to fetch denomination details for a particular branch.                         |
| CMS_BRANCH_CODE_LIST          | Cash & Cheque Deposits            | This API is used to fetch all the branch.                                                       |
| CMS_CASH_DEPOSIT_VALIDATE     | Cash Deposits                     | This API is used to validate cash/cheque number uniqueness.                                     |
| CMS_CMM_LIST                  | Cash Deposits                     | This API is used to fetch maintenance details of a cash management.                             |
| CMS_CASH_WITHDRAWAL_LIST      | View Cash withdrawal              | This API is used to fetch cash withdrawal collections.                                          |
| CMS_CASH_WITHDRAWAL_READ      | View Cash withdrawal details      | This API is used to fetch cash withdrawal collection details.                                   |
| CMS_CASH_WITHDRAWAL_CREATE    | Create Cash withdrawal            | This API is used to create cash withdrawal for a party.                                         |
| CMS_CHEQUE_DEPOSITS_CREATE    | Cheque Deposits                   | This API is used to create multiple cheque deposits for a party.                                |
| CMS_CASHFLOW_CODE_LIST        | Cashflow Forecasting              | This API is used to fetch cash flow code details.                                               |
| CMS_ROUTING_LIST              | Cheque Deposits                   | This API is used to fetch routing details.                                                      |
| CMS_COLLECTION_LIST           | Overview Collection summary       | This API is use to fetch collection summary details for cash/cheque & cash withdraw.            |
| CMS_PDM_LIST                  | Pick up & delivery                | This API is use to fetch pickup and delivery details.                                           |
| CMS_FEEDFILE_SYNC_STATUS      | Cash flow file upload sync status | This API is used to fetch the status of the file that is being uploaded.                        |
| CMS_FETCH_ACCOUNT_DETAILS     | Fetch Account Details             | This API is used to fetch account details                                                       |
| CMS_FETCH_BALANCE_BY_GROUP    | Fetch Balance by group            | This API is used to fetch balance group by, and group can either entity or currency or location |
| CMS_FETCH_BALANCE_BY_IDENTITY | Fetch Balance Details             | This API is used to fetches balance by identity based on selected group                         |

| OBRH Consumer Service Name | Transaction Name                 | Description                                                                                     |
|----------------------------|----------------------------------|-------------------------------------------------------------------------------------------------|
| CMS_FETCH_MAINTENANCE      | Fetch Cash Visibility Management | This API is used to retrieve maintenance details for a cash visibility.                         |
| CMS_COLLECTION_DELETE      | Cancel collection                | This API is used to delete or cancel collection initiated by OBDX user and has in request state |
| CMS_BULK_UPLOAD            | Cash flow file upload            | This API is used to upload bulk cash flow files.                                                |
| CMS_CASHFLOW_FETCH         | Cashflow Forecasting             | This API is used to retrieve cash flow forecasting data.                                        |

This completes the entire configuration needed for consuming OBCM APIs in OBAPI through OBRH.

We need to execute below script to fetch mid office token required for Cashflow/Payment File Upload.

```
INSERT INTO DIGX_FW_CONFIG_OUT_RS_CFG_B
(SERVICE_ID, CONTEXT_URL, SERVICE_URL, REQUEST_MEDIA_TYPE,
RESPONSE_MEDIA_TYPE, AUTHENTICATION,
AUTH_TYPE, CREDENTIAL_STORE_TYPE, CREDENTIAL_STORE_KEY, CREATION_DATE,
LAST_UPDATED_DATE)
VALUES ('tokenOBCM144', 'http://${OBCM_HOST_IP}:${OBCM_HOST_PORT}', 'api-
gateway/platojwtauth',
'application/json', 'application/json', 'N', 'Bearer', 'credential_impl',
'OBCM_14.4',sysdate, sysdate);
```

# 8

## Credit Facility Management

- [Core as Third Party](#)
- [OBRH Configurations](#)

### 8.1 Core as Third Party

During Bank Implementation, in the case that the core system is Third Party and Mid-Office is OBCFPM and ELCM (Oracle Banking Credit Facility Management), then the entry in DIGX\_FW\_CONFIG\_ALL\_O will be: Determinant value for Third Party Entity | TP1.0, OBCFPM14.4,OBCFPM14.3, ELCM14.4,ELCM14.3

For example, if the determinant value for the Third Party Entity is OBDXBU1 then the entry will look like: OBDXBU1 | TP1.0, OBCFPM14.4,OBCFPM14.3, ELCM14.4,ELCM14.3

### 8.2 OBRH Configurations

During Bank Implementation, assuming OBRH is installed and configured as part of OBCFPM installation.

For OBAPI and mid Office OBCFPM integration using OBRH the following configurations need to be done.

1. Carry out all the steps mentioned in **OBRH Integration Configuration** section in this document. The service provider for mid-office product OBCFPM (Oracle Credit facility Management) end-points configured in OBRH is “OBCFPM”
2. After all the above steps are completed, user needs to execute some scripts for the host APIs that are to be consumed via OBRH. These are the scripts to pick the third party adapter implementation instead of the host specific implementations as well as to call the OBRH end-point for the configured interfaces. The scripts are available at the following location:

```
/installables/db/OBCFPM/DIGX_FW_CONFIG_ALL_O.sql  
/installables/db/ELCM/DIGX_FW_CONFIG_ALL_O.sql
```

#### Note:

'%ENTITY\_ID%' should be replaced with the entity identifier (For Example if the entity during implementation is OBDX\_BU then '%ENTITY\_ID%' should be replaced by 'OBDX\_BU').

3. The list of OBASP APIs that are integrated with OBAPI using OBRH is as follows:

| Interface ID                      | Transaction Name         | Description                                                                      |
|-----------------------------------|--------------------------|----------------------------------------------------------------------------------|
| CF_LIABILITY_READ                 | Credit Facility Overview | This API is used to fetch the liability details of particular liability.         |
| CF_LIABILITY_LIST                 | Credit Facility Overview | This API is used to fetch the liability details.                                 |
| CF_COLLATERALGROUP_READ           | Collateral Summary       | This API is used to fetch the collateral group details of particular collateral. |
| CF_COLLATERALGROUP_LIST           | Collateral Summary       | This API is used to fetch the collateral group details.                          |
| CF_COLLATERAL_LIST                | Collateral Summary       | This API is used to fetch the collateral details.                                |
| CF_COLLATERAL_READ                | Collateral Summary       | This API is used to fetch the collateral details of particular collateral.       |
| CF_FACILITY_LIST                  | Facility Summary         | This API is used to fetch the facility details                                   |
| CF_FACILITY_CATAGORY_TYPE_SERVICE | Facility Summary         | This API is used to fetch the facility category of particular facility.          |
| CF_FACILITY_UTILIZATION           | Facility Details         | This API is used to fetch the facility history details                           |
| CF_COLLATERALTYPES_LIST           | Collateral Evaluation    | This API is used to fetch the collateral types.                                  |
| CF_FACILITYCATEGORY_LIST          | Apply new Facility       | This API is used to fetch the facility category.                                 |
| CF_DOCUMENT_READ                  | Apply new Facility       | This API is used to fetch the facility category.                                 |
| CF_FETCH_APPLICATION_STAT<br>US   | Apply new Facility       | This API is used to fetch the document.                                          |
| CF_FETCH_APPLICATION_STAT<br>US   | Apply new Facility       | This API is used to fetch the application status.                                |
| CF_FACILITY_UPDATE                | Apply new Facility       | This API is used to create and update facility.                                  |
| CF_COLLATERAL_OFFER               | Application Tracker      | This API is used to accept or reject applications.                               |
| CF_EVALUATE_COLLATERAL            | Collateral Evaluation    | This API is used to evaluate collateral.                                         |
| CF_REEVALUATE_COLLATERAL          | Collateral Revaluation   | This API is used to revalue collateral.                                          |
| CF_COLLATERAL_READ_MULTIPLE       | Collateral Read Multiple | Collateral Read Multiple                                                         |

# 9

## Liquidity Management

- [OBRH Configurations](#)
- [Verify System Configurations](#)
- [Enumerations](#)
- [Adapter Properties](#)
- [Simulation IC Group maintenance](#)
- [Cloud Specific Configurations](#)
- [Host Notifications](#)
- [OBLM to OBAPI Error code mapping](#)

### 9.1 OBRH Configurations

During Bank Implementation, assuming OBRH is installed and configured as part of OBLM installation.

For OBAPI and OBLM integration using OBRH the following configurations need to be done.

1. Carry out all the steps mentioned in **OBRH Integration Configuration** section in this document. The service provider for product processor OBLM (Oracle Banking Liquidity Management) end-points configured in OBRH is “**OBLM**” (this step is not required to be repeated after each patch-set).
2. After all the above steps are completed, user needs to execute some scripts for the host APIs that are to be consumed via OBRH. These are the scripts to pick the third party adapter implementation instead of the host specific implementations as well as to call the OBRH end-point for the configured interfaces. The scripts are available at the following location:
  - a. `/installables/db/OBLM/version/DIGX_FW_CONFIG_ALL_O.sql` (part of base installer)
  - b. `/patch_incrementals/modules/OBLM/22.2.1.0.0_DIGX_FW_CONFIG_ALL_O.sql` (May'23 PS)
  - c. `/patch_incrementals/modules/OBLM/22.2.2.0.0_DIGX_FW_CONFIG_ALL_O.sql` (Nov'23 PS)
  - d. `/patch_incrementals/modules/OBLM/22.2.5.0.0_DIGX_FW_CONFIG_ALL_O.sql` (Oct'24 PS)

#### Note:

'%ENTITY\_ID%' in the above scripts should be replaced with the entity identifier (For Example if the entity during implementation is OBDX\_BU then '%ENTITY\_ID%' should be replaced by 'OBDX\_BU').

All the OBLM APIs consumed from OBAPI are via OBRH. List is as follows:

| OBRH Consumer Service Name         | Transaction Name                                                                       |
|------------------------------------|----------------------------------------------------------------------------------------|
| createLMStructure                  | Create Structure                                                                       |
| editLMStructure                    | Edit Structure                                                                         |
| executeLMPoolStructure             | Execute Pool Structure                                                                 |
| executeLMSweepStructure            | Execute Sweep Structure                                                                |
| fetchLMAccounts                    | Fetch Liquidity enabled Accounts for Primary as well as Linked Customers               |
| fetchLMAccountsByPartyId           | Fetch Liquidity enabled Accounts for Customer                                          |
| fetchLMAccountsWithStructure       | Fetch Accounts participating in multiple Structures                                    |
| fetchLMAccountsWithlinkedStructure | Fetch Structure details for list of Accounts                                           |
| fetchLMBranches                    | Fetch Branches                                                                         |
| fetchLMChargeDefinitionList        | Fetch Charge Definitions                                                               |
| fetchLMChargeList                  | Fetch Charges                                                                          |
| fetchLMCurrency                    | Fetch Currencies                                                                       |
| fetchLMFrequencies                 | Fetch Frequencies                                                                      |
| fetchLMInstruction                 | Fetch Instructions                                                                     |
| fetchPoolLogs                      | Fetch Pool Logs                                                                        |
| fetchSweepLogs                     | Fetch Sweep Logs                                                                       |
| fetchUpcomingSweepLogs             | Fetch Upcoming Sweep Logs                                                              |
| listStructurePriorities            | Fetch Structure Priorities                                                             |
| partyHierarchyList                 | Fetch Linked Customers Hierarchy                                                       |
| readLMStructure                    | View Structure details                                                                 |
| validateLMStructure                | Validate Structure                                                                     |
| fetchLMStructures                  | Fetch Structures                                                                       |
| fetchLMSimulationAccounts          | Fetch Accounts eligible for Simulation                                                 |
| fetchLMSimulation                  | Fetch Simulation Structures                                                            |
| readLMSimulation                   | View Simulation Structure details                                                      |
| createLMSimulation                 | Create Simulation Structure                                                            |
| downloadLMSimulationPDFReport      | Download Simulation Advice                                                             |
| editLMSimulation                   | Edit Simulation Structure                                                              |
| createlendlimit                    | Create Lend Limit                                                                      |
| readLMLendLimit                    | View Lend Limit details                                                                |
| editLendLimit                      | Edit Lend Limit                                                                        |
| closeLendLimit                     | Close Lend Limit                                                                       |
| getLMGroupCustomerId               | Fetch Root customer in customer hierarchy                                              |
| listLMInterCompanyLoans            | Fetch Intercompany Loans                                                               |
| readLMInterCompanyLoan             | View Intercompany Loan details                                                         |
| fetchICLTransactions               | Fetch Intercompany Loan transactions                                                   |
| initiateICLSettlement              | Initiate Intercompany Loan settlement                                                  |
| listIntercompanyLoanSummary        | Fetch Intercompany Loans summary                                                       |
| getChildAccountsForLMStructure     | Fetch immediate child accounts for a selected account in Structure in staggered manner |

| OBRH Consumer Service Name   | Transaction Name                       |
|------------------------------|----------------------------------------|
| readLMStructurePdfDownload   | Download Structure details             |
| fetchReallocationMonitorLogs | Fetch Reallocation Logs                |
| executeSweepAccountPairs     | Adhoc execution of sweep account pairs |

This completes the entire configuration needed for consuming OBLM APIs in OBAPI through OBRH.

## 9.2 Verify System Configurations

Following script helps in listing the LM specific System Configurations:

```
SELECT prop_id as PROPERTY_IN_DATABASE, NVL(SUBSTR(t.UI_definition,
INSTR(t.UI_definition, '"title"')+9, INSTR(t.UI_definition, '"')-11),
t.UI_definition) AS TITLE_ON_SCREEN, t.prop_value FROM digx_fw_config_var_b t
where prop_id
like '%LM%' and module = 'OTHERMODULE' and determinant_value = '*';
```

-- Please enter correct determinant value

Ensure correct values are maintained against the above properties.

This maintenance can be done from the “System Configuration” admin screen or directly in DB schema.

## 9.3 Enumerations

Following LM related enumerations are used in OBAPI . They are used to fetch the values on the OBAPI UI.

- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMSweepStatus';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMStructureTypes';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMCurrencyHolidayRates';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMHolidayTreatment';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMBackwardTreatment';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMInterestMethod';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMRellocationMethod';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMStructureStatus';`
- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMStructureHostApprovalStatus';`



- `select * from DIGX_FW_ENUM_REPRESENTATIONS where enum_fqn='getLMChargeCollectionStatus';`

## 9.4 Adapter Properties

Certain fields (Eg: Enumerations, Status etc) can have different values in OBAPI as compared to OBLM.

The mapping of all such values between OBAPI and OBLM can be found/maintained using the below script:-

```
select * from DIGX_FW_CONFIG_ADAPTER_PROP_B where host_id = 'OBLM';
```

## 9.5 Simulation IC Group maintenance

While creating Simulation from OBDX, we need to send IC Group values to OBLM as per the below matrix:

| Structure Type | Interest Method | Scenario                                     | PROP_ID                      |
|----------------|-----------------|----------------------------------------------|------------------------------|
| <b>Sweep</b>   | Interest        | All Participating accounts                   | SIM_SWEEP_INTEREST_ACCOUNT   |
| <b>Hybrid</b>  | Interest        | Notional Header                              | SIM_HYBRID_INTEREST_NOTIONAL |
| <b>Pool</b>    | Interest        | Notional Header                              | SIM_POOL_INTEREST_NOTIONAL   |
|                | Advance         | Notional Header                              | SIM_POOL_ADVANCE_NOTIONAL    |
|                |                 | All Participating Accounts (Except Notional) | SIM_POOL_ADVANCE_ACCOUNT     |
|                | Ratio           | All Participating Accounts (Except Notional) | SIM_POOL_RATIO_ACCOUNT       |

The values of these IC Groups can be different in each environment based on the IC Groups created in the respective OBLM system.

Certain properties have been created in OBDX, from where the values of these IC Groups shall be dynamically picked and sent to OBLM during Simulation creation from OBDX.

Below are the sample scripts to update those property values:-

```
UPDATE DIGX_FW_CONFIG_ADAPTER_PROP_B SET
PROP_VALUE='<SIM_SWEEP_INTEREST_ACCOUNT>' WHERE HOST_ID='OBLM' AND
TRANSACTION_TYPE='INTEREST_CALCULATION_GROUP' AND
PROP_ID='SIM_SWEEP_INTEREST_ACCOUNT';
```

```
UPDATE DIGX_FW_CONFIG_ADAPTER_PROP_B SET
PROP_VALUE='<SIM_HYBRID_INTEREST_NOTIONAL>' WHERE HOST_ID='OBLM' AND
TRANSACTION_TYPE='INTEREST_CALCULATION_GROUP' AND
PROP_ID='SIM_HYBRID_INTEREST_NOTIONAL';
```

```
UPDATE DIGX_FW_CONFIG_ADAPTER_PROP_B SET
PROP_VALUE='<SIM_POOL_INTEREST_NOTIONAL>' WHERE HOST_ID='OBLM' AND
TRANSACTION_TYPE='INTEREST_CALCULATION_GROUP' AND
PROP_ID='SIM_POOL_INTEREST_NOTIONAL';
```

```
UPDATE DIGX_FW_CONFIG_ADAPTER_PROP_B SET
PROP_VALUE='<SIM_POOL_ADVANCE_NOTIONAL>' WHERE HOST_ID='OBLM' AND
TRANSACTION_TYPE='INTEREST_CALCULATION_GROUP' AND
PROP_ID='SIM_POOL_ADVANCE_NOTIONAL';
```

```
UPDATE DIGX_FW_CONFIG_ADAPTER_PROP_B SET
PROP_VALUE='<SIM_POOL_ADVANCE_ACCOUNT>' WHERE HOST_ID='OBLM' AND
TRANSACTION_TYPE='INTEREST_CALCULATION_GROUP' AND
PROP_ID='SIM_POOL_ADVANCE_ACCOUNT';
```

```
UPDATE DIGX_FW_CONFIG_ADAPTER_PROP_B SET
PROP_VALUE='<SIM_POOL_RATIO_ACCOUNT>' WHERE HOST_ID='OBLM' AND
TRANSACTION_TYPE='INTEREST_CALCULATION_GROUP' AND
PROP_ID='SIM_POOL_RATIO_ACCOUNT';
```

## 9.6 Cloud Specific Configurations

Following additional configurations are required if OBDX and OBLM are being hosted on cloud:

1. In OBRH, enable Eureka instance for OBLM service provider.

## 9.7 Host Notifications

In order to listen to any Host events and trigger subsequent alerts in OBAPI for the same, please follow the below steps as part of extensibility:

### **Out-of-Box Notification Alert Support:**

Update the output of the following script:

```
SELECT * FROM digx_fw_config_all_b WHERE prop_id LIKE 'structure-
createdAndAuthorized@%' AND category_id='KAFKA_CONFIG';
```

### **New Notification Alert Support:**

1. Get the Avro schema format for the notification to be consumed from the host. Ensure that the deserialized objects based on the Avro are present in the class-path.
2. Create a new consumer class that implements the **IKafkaConsumable** interface. Consumers implementing this interface will always consume messages from Kafka topics.

#### **Override Methods:**

- **topicName():** Override this method to specify the name of the topic the consumer should listen to. Returns String.  
Example: structure-createdAndAuthorized
- **consumerGroup():** Override this method to specify the consumer group name. Returns String.
- **enableSeparateConsumerGroupsPerServer():**
  - a. When true, each instance of the consumer on each server creates its own consumer group.
  - b. When false, all instances of this consumer across all servers share the same consumer group. Default is false.
- **run():** Responsible for initiating the message consumption process. Within this method, the consume method is called with an instance of **IMessageProcessor** (created as part of point 4) to handle the processing of each consumed message.

- **OOTB Reference:**  
com.ofss.digx.kafka.liquiditymanagement.consumer.structure.StructureMessageConsumer
- 3. Create a file named com.ofss.digx.infra.events.kafka.consumer.**ICustomer** in resources/META-INF/services and provide the entry for the consumer class.
- 4. Create a new class implementing com.ofss.digx.infra.events.processor.**IMessageProcessor** for writing business logic. This class will be used from the consumer and listener classes and should be included in the service jar of the module.  
**Override Methods:**
  - **process(K key, V data):** Processes messages from the consumer. The out-of-box host alert service (Eg: com.ofss.digx.app.liquiditymanagement.service.hostalerts.HostAlertService) should be invoked from this method.  
key: The key object associated with the message.  
data: The data to be processed
- OOTB Reference:**  
com.ofss.digx.app.liquiditymanagement.processor.structure.StructureMessageProcessor
- 5. Kafka consumer configurations can be maintained in DIGX\_FW\_CONFIG\_ALL\_B with category\_id set to KAFKA\_CONFIG. For configurations specific to a topic, prop\_id can be specified as TOPIC\_NAME@CONFIGURATION.  
Example: structure-createdAndAuthorized@bootstrap.servers
- 6. Configure a subscription based OBAPI alert specific for the new notification configured. Post maintaining subscription for the new OBAPI alert, subscribed users will receive OBAPI alerts specific to the notification.

## 9.8 OBLM to OBAPI Error code mapping

1. OBLM to OBAPI error code mappings are present in the database table DIGX\_FW\_ERR\_COD\_MAP where MODULE\_ID is "LIQUIDITY\_MANAGEMENT"
2. Out of the box, the value in column EXT\_SYSTEM\_ID for all such rows would be UBS14.5.
3. The value in column EXT\_SYSTEM\_ID for all such rows will have to be modified during implementation, based on the value derived from below query:
  - a. 

```
SELECT
      CONCAT(prop_value, (select prop_value from digx_fw_config_var_b
      where prop_id =
      'HOST_VERSION' and determinant_value = '$entity_name$'))
      EXT_SYSTEM_ID from
      digx_fw_config_var_b where prop_id = 'HOST_NAME' and
      determinant_value = '$entity_name$'; --
      replace $entity_name$ with correct determinant_value.
```

# 10

## User Credential Configuration

For some of the Mid-Office Products (OBVAM, OBTFPM, OBSCF, OBCM, INV, ASP, OBCFPM, MO\_IPM) by default user credential configuration is DB-Based. However, it should be changed to use connector based configuration.

Following are the steps to change user credential configuration from DB-Based to connector based

1. Update `CREDENTIAL_STORE_TYPE` property in table `DIGX_FW_CONFIG_OUT_RS_CFG_B` to "credential\_impl" for the particular service ID.

**Sample Script**

```
UPDATE DIGX_FW_CONFIG_OUT_RS_CFG_B
set CREDENTIAL_STORE_TYPE='credential_impl' where
SERVICE_ID='tokenOBTFPM142';
```

2. Create/Update required Connector Credentials mapping in weblogic console for particular Host (Outbound Connection) by referring to **Oracle Banking APIs Connector Credential Store Guide**

# OBRH Integration Configuration

To consume Mid-Office APIs in OBAPI using OBRH, following configurations need to be completed:

1. To integrate OBRH with OBAPI, first some generic configurations and scripts needs to be executed. The Details for the same can be referred from section **Configurations for OBRH Integration** from **Oracle Banking APIs Host Integration Guide**

 **Note:**

Please skip adding entries to call OBRH end-point from adapters for already provided out of the box integrations from OBAPI.

2. For Consuming Mid-Office Product services via OBRH, where OBAPI will act as a consumer for OBRH, OBAPI Consumer configurations required by OBRH needs to be imported in OBRH. The File to be imported would be present at the following location:

```
/installables/obrh/OBAPI_Consumer.json
```

Refer section **Import Service Consumer** from **OBRH** user manual for how to import a consumer JSON in OBRH

3. Once the import is done successfully, you need to update each of the mid-office service provider's default implementation as well as other implementations for IP, Port, Token Username and Token Password. Refer section **Add/Edit Implementation** from **OBRH** user manual for achieving the same.

 **Note:**

\* When using OBRH there is no specific host implementation adapters. We use the third party adapter implementation for all services. The request and response specifications sent and received from OBRH for an end-point can be referred from the following: **externalinterface-api.zip**

\* Any other assistance required regarding OBRH, you could refer the **OBRH** user manual.

\* Also if anymore custom fields need to be sent to host or more fields are need to be configured in response; the following changes needs to be done

a. Fields needs to be added in OBAPI Request and Response

b. Transformations needs to be changed in OBRH. Refer section **Request and Response Transformation** from **OBRH** user manual.

# Index

## A

---

Adapter Properties, [6-4](#), [9-4](#)

## C

---

Cloud specific Configurations, [6-4](#)  
Cloud Specific Configurations, [9-5](#)  
Core as Third Party, [8-1](#)

## D

---

Day One Executions, [3-1](#)

## E

---

Enumerations, [6-4](#), [9-3](#)

## I

---

Introduction, [1-1](#)

## M

---

Mandatory Executions, [2-1](#)

## N

---

Non Customer Onboarding Using Chaining, [4-1](#)

## O

---

OBLM to OBAPI Error code mapping, [9-6](#)  
OBRH Configurations, [8-1](#), [9-1](#)  
OBRH Integration, [4-3](#), [5-1](#), [6-1](#), [7-1](#)  
OBRH Integration Configuration, [11-1](#)  
OBVAM to OBAPI Error code mapping, [6-4](#)  
Oracle Banking Trade Finance Process  
Management (OBTfPM), [2-1](#)

## S

---

Simulation IC Group maintenance, [9-4](#)

## U

---

User Credential Configuration, [10-1](#)

## V

---

Verify System Configurations, [6-4](#), [9-3](#)