Mid-Office Product Setup and Configuration Guide
Oracle Banking Digital Experience
Release 21.1.0.0.0

Part No. F40800-01

May 2021

ORACLE

Mid-Office Product Setup and Configuration Guide

May 2021

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001

www.oracle.com/financialservices/

Copyright © 2006, 2021, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

1. Preface

1.1 Intended Audience

This document is intended for the following audience:

- Customers
- Partners

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

1.3 Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1.4 Structure

This manual is organized into the following categories:

Preface gives information on the intended audience. It also describes the overall structure of the User Manual.

The subsequent chapters describes following details:

- Introduction
- Preferences & Database
- Configuration / Installation.

1.5 Related Information Sources

For more information on Oracle Banking Digital Experience Release 21.1.0.0.0, refer to the following documents:

- Oracle Banking Digital Experience Installation Manuals

2. Introduction

This document is intended for setting up OBDX 20.1.0.0.0 with different Mid-Office Products.

Home

3. Trade Finance

Following qualifications are available in Trade Finance

3.1 Oracle Banking Trade Finance (OBTF)

From release 20.1.0.1.0 onwards, OBDX Trade finance integration with OBTF is supported through OBRH only.

3.1.1 OBRH Mandatory Executions

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

../installables/modules/OBTF/DIGX_FW_CONFIG_ALL_O.sql

Note: '%ENTITY_ID%' should be replaced with entity identifier (For example 'OBDX_BU')

3.1.2 OBRH Configurations

Following day 1 system configuration should be available at entity level to connect OBDX with OBRH system.

select * from digx_fw_config_var_b where prop_id in ('OBRH_EXPIRYTIME','OBRH_HOST_IP','OBRH_HOST_PORT');

- OBRH Service Consumer

A separate service consumer should be available in OBRH for handling OBDX requests. (Example 'OBDX')

- OBRH Service Provider

A service consumer should contain a service provider for handling Trade Finance requests to OBTF host and for its different versions. (Example 'OBTF')

- OBRH Consumer Service

Service consumer should contain a consumer services for invoking particular host service and handling its input and output data.

Please find below the details of OBDX Trade Finance requests and their corresponding consumer services available in OBRH.

Trade Finance Request	Consumer Service
Trade applications list	TRADE_APPLICATIONS_LIST
Trade clarification list	TRADE_APPLICATION_CLARIFICATION_LIST
Trade application clarification update	TRADE_APPLICATION_CLARIFICATION_UPDATE
Download the document by given document ref id	TRADE_APPLICATION_DOCUMENT_DOWNLOAD
Link already uploaded documents to a contract	TRADE_APPLICATION_DOCUMENT_LINKAGE
Trade application document list	TRADE_APPLICATION_DOCUMENT_LIST
Trade application document upload	TRADE_APPLICATION_DOCUMENT_UPLOAD
Trade application list	TRADE_APPLICATION_LIST
Trade bank code list	TRADE_BANK_CODE_LIST
Trade bank details	TRADE_BANK_DETAILS
Trade base date list	TRADE_BASE_DATE_LIST
BC product currencies	TRADE_BC_PRODUCT_CURRENCIES
BC product list	TRADE_BC_PRODUCT_LIST
BC product read details	TRADE_BC_PRODUCT_READ_DETAILS
BG amend charges simulation	TRADE_BG_AMD_CHARGES_SIMULATION
BG amend create	TRADE_BG_AMEND_CREATE
BG amend list	TRADE_BG_AMEND_LIST
BG pending amendment listing	TRADE_BG_AMEND_PENDING_LIST
BG amend read	TRADE_BG_AMEND_READ
BG availment listing	TRADE_BG_AVAILAMENT_LIST
BG bank customer read	TRADE_BG_BANK_CUSTOMER_READ
BG charges simulation	TRADE_BG_CHARGES_SIMULATION
BG claim create	TRADE_BG_CLAIM_CREATE

Trade Finance Request	Consumer Service
BG create	TRADE_BG_CREATE
BG swift and advice message read details	TRADE_BG_LEVEL_ADVICE_SWIFT_READ
BG charges listing	TRADE_BG_LEVEL_CHARGES
BG listing	TRADE_BG_LIST
BG pending amendment acceptance	TRADE_BG_PENDING_AMEND_ACCEPT
BG pending amendment check	TRADE_BG_PENDING_AMEND_CHECK
BG pending amendment rejection	TRADE_BG_PENDING_AMEND_REJECT
BG product currency listing	TRADE_BG_PRODUCT_CURRENCY_LIST
BG product fft code listing	TRADE_BG_PRODUCT_FFT_LIST
BG product listing	TRADE_BG_PRODUCT_LIST
BG product read details	TRADE_BG_PRODUCT_READ
BG read details	TRADE_BG_READ
BG remarks	TRADE_BG_REMARK
BG summary details	TRADE_BG_SUMMARY_DETAILS
BG terms and conditions details	TRADE_BG_TERMSANDCONDITIONS_READ
BG type listing	TRADE_BG_TYPE_LIST
BG upload documents	TRADE_BG_UPLOAD_DOCUMENTS
BG version read	TRADE_BG_VERSION_READ
Bill charges list	TRADE_BILL_CHARGES_LIST
Bill charges simulation	TRADE_BILL_CHARGES_SIMULATION
Bill create	TRADE_BILL_CREATE
Bill discrepancy acceptance	TRADE_BILL_DISCREPANCY_CUST_ACCEPTANCE
Bill discrepancy listing	TRADE_BILL_DISCREPANCY_LIST
Bill discrepancy read details	TRADE_BILL_DISCREPANCY_READ

Trade Finance Request	Consumer Service
Bill swift and advice message read details	TRADE_BILL_LEVEL_SWIFT_ADVICE
Bill listing	TRADE_BILL_LIST
Bill product documents listing	TRADE_BILL_PRODUCT_DOCUMENTS
Bill product document clauses	TRADE_BILL_PRODUCT_DOCUMENTS_CLAUSES
Bill read detail	TRADE_BILL_READ
Bill settlement create	TRADE_BILL_SETTLEMENT_CREATE
Collection swift and advice message read details	TRADE_COLLECTION_ADVICE_SWIFT_READ
Collection charges list	TRADE_COLLECTION_CHARGES_LIST
Collection charges simulation	TRADE_COLLECTION_CHARGES_SIMULATION
Collection create	TRADE_COLLECTION_CREATE
Collection listing	TRADE_COLLECTION_LIST
Collection read details	TRADE_COLLECTION_READ
Contract fft code list	TRADE_CONTRACT_FFT_LIST
Trade Customer clarification list	TRADE_CUSTOMER_CLARIFICATION_LIST
Trade detail summary snapshot	TRADE_DETAIL_SUMMARY_SNAPSHOT
Document category listing	TRADE_DOCUMENT_CATEGORY_LIST
Document content listing	TRADE_DOCUMENT_CONTENT_LIST
Document level clauses	TRADE_DOCUMENT_LEVEL_CLAUSES
Document listing	TRADE_DOCUMENT_LIST
Document category level doc type listing	TRADE_DOC_CATEGORY_LEVEL_DOC_TYPE
Trade equivalent amount	TRADE_EQUIVALENT_AMOUNT
Goods listing	TRADE_GOODS_LIST
Incoterm listing	TRADE_INCOTERM_LIST
LC additional conditions listing	TRADE_LC_ADDITIONAL_COND_LIST

Trade Finance Request	Consumer Service
LC advice and swift read details	TRADE_LC_ADVICE_SWIFT_READ
LC amend charges simulation	TRADE_LC_AMD_CHARGES_SIMULATION
LC amendment create	TRADE_LC_AMENDMENT_CREATE
LC amendment listing	TRADE_LC_AMENDMENT_LIST
LC amendment read details	TRADE_LC_AMENDMENT_READ
LC amend swift simulation	TRADE_LC_AMEND_SWIFT_SIMULATION
LC charges	TRADE_LC_CHARGES
LC charges simulation	TRADE_LC_CHARGES_SIMULATION
LC create	TRADE_LC_CREATE
LC availment listing	TRADE_LC_LEVEL_AVAILMENTS
LC listing	TRADE_LC_LIST
LC pending amendment listing	TRADE_LC_PENDING_AMENDMENT_LIST
LC pending amendment acceptance	TRADE_LC_PENDING_AMEND_ACCEPT
LC pending amendment rejection	TRADE_LC_PENDING_AMEND_REJECT
LC product currencies	TRADE_LC_PRODUCT_CURRENCIES
LC product documents	TRADE_LC_PRODUCT_DOCUMENTS
LC product document clauses	TRADE_LC_PRODUCT_DOCUMENT_CLAUSES
LC product listing	TRADE_LC_PRODUCT_LIST
LC product read details	TRADE_LC_PRODUCT_READ
LC read details	TRADE_LC_READ
LC swift simulation	TRADE_LC_SWIFT_SIMULATION
LC update	TRADE_LC_UPDATE
LC version read details	TRADE_LC_VERSION_READ
Trade read customer for bank code	TRADE_READ_CUST_FOR_BANK_CODE
SG advice and swift message details	TRADE_SG_ADVICE_SWIFT_READ

Trade Finance Request	Consumer Service
SG charges simulation	TRADE_SG_CHARGES_SIMULATION
SG create	TRADE_SG_CREATE
SG level charges	TRADE_SG_LEVEL_CHARGES
SG listing	TRADE_SG_LIST
SG product currency listing	TRADE_SG_PRODUCT_CURRENCY_LIST
SG product level documents	TRADE_SG_PRODUCT_LEVEL_DOCUMENTS
SG product listing	TRADE_SG_PRODUCT_LIST
SG product read details	TRADE_SG_PRODUCT__READ
SG read details	TRADE_SG_READ
Trade summary snapshot	TRADE_SUMMARY_SNAPSHOT

3.2 Oracle Banking Trade Finance Process Management (OBTFPM)

OBDX Trade Finance integration with OBTFPM consists of integrations with OBTFPM application and OBTF application both. From release 20.1.0.2.0 onwards, OBDX Trade finance integration with OBTFPM is supported through OBRH only.

3.2.1 OBTF Dependency

Please follow the step 3.1 (Note- No need to restart the manage server)

3.2.2 Mandatory Executions

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

../installables/modules/OBTFPM/DIGX_FW_CONFIG_ALL_O.sql

Note: '%ENTITY_ID%' should be replaced with entity identifier (For example 'OBDX_BU')

Home

4. Corporate Lending

4.1 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';

and also update values for OBCL_HOST_IP , OBCL_HOST_PORT , OBCL_WS_VERSION in system configuration

OBCL_HOST_IP - 10.184.159.212

OBCL_HOST_PORT - 8555

OBCL_WS_VERSION- 141

Home

5. Supply Chain Finance

5.1 Core as Third Party

During Bank Implementation when there is case that the core system is Third Party and Mid office is OBSCF then in that case the entry in DIGX_FW_CONFIG_ALL_O will be something like

Determinant value for Third Party Entity | TP1.0, OBSCF14.4

For example, if determinant value for Third Party Entity is OBDXBU1 then

OBDXBU1 | TP1.0, OBSCF14.4

Now for all core services, the request will go to Third Party Adapters. But for all SCF services also, the request will go to third party adapters of SCF.

To avoid that, following scripts need to be executed

```
Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CRE
ATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.<Determinant
Value>.com.ofss.digx.extxface.scf.adapter.program.IProgramAdapter','ExtxfaceAdapterPreferenc
e','com.ofss.digx.extxface.obscf144.impl.program.ProgramAdapter','01','ofssuser',sysdate,'ofssus
er',sysdate);

Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CRE
ATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.<Determinant
Value>.com.ofss.digx.extxface.scf.adapter.programproduct.IProgramProductAdapter','ExtxfaceAd
apterPreference','com.ofss.digx.extxface.obscf144.impl.programproduct.ProgramProductAdapter'
,'01','ofssuser',sysdate,'ofssuser',sysdate);

Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CRE
ATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.<Determinant
Value>.com.ofss.digx.extxface.scf.adapter.finance.IFinanceAdapter','ExtxfaceAdapterPreference'
,'com.ofss.digx.extxface.obscf144.impl.finance.FinanceAdapter','01','ofssuser',sysdate,'ofssuser',
sysdate);
```

For example, if determinant value for Third Party Entity is OBDXBU1 then

```
Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CRE
ATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.OBDXBU1.com.ofss.digx.extxface.scf.adapter.program.IProgramAdapter','Extxfa
ceAdapterPreference','com.ofss.digx.extxface.obscf144.impl.program.ProgramAdapter','01','ofssu
ser',sysdate,'ofssuser',sysdate);
```

Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.OBDXBU1.com.ofss.digx.extxface.scf.adapter.programproduct.IProgramProductAdapter','ExtxfaceAdapterPreference','com.ofss.digx.extxface.obscf144.impl.programproduct.ProgramProductAdapter','01','ofssuser',sysdate,'ofssuser',sysdate);

Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.OBDXBU1.com.ofss.digx.extxface.scf.adapter.finance.IFinanceAdapter','ExtxfaceAdapterPreference','com.ofss.digx.extxface.obscf144.impl.finance.FinanceAdapter','01','ofssuser',sysdate,'ofssuser',sysdate);

5.2 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 in 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 3 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.**

For detailed explanation and scripts please refer section **6.1 Core as Third Party under Associated Party Management.**

5.3 OBRH Integration

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

For OBDX and mid Office 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 mid-office 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 OBDX using OBRH is as follows:

Interface Id	OBDX Screen	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.

Interface Id	OBDX Screen	Description
SCF_PURCHASE_ORDER_REJECT	Accept/Reject Purchase Order	This API is used to reject purchase orders.
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_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.

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

Home

6. Associated Party Management

6.1 Core as Third Party

During Bank Implementation when there is case that the core system is Third Party and Mid office is ASP (Associated Party) then in that case the entry in DIGX_FW_CONFIG_ALL_O will be -

Determinant value for Third Party Entity | TP1.0, ASP14.4

For example, if determinant value for Third Party Entity is OBDXBU1 then

OBDXBU1 | TP1.0, ASP14.4

Now for all core services, the request will go to Third Party Adapters. But, for all ASP services also, the request will go to third party adapters of ASP.

To avoid that, following scripts need to be executed:

Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.<Determinant Value>.com.ofss.digx.extxface.associatedparty.adapter.IAssociatedPartyAdapter','ExtxfaceAdapterPreference','com.ofss.digx.extxface.associatedparty144.impl.associatedparty.AssociatedPartyAdapter','01','ofssuser',sysdate,'ofssuser',sysdate);

For example, if determinant value for Third Party Entity is OBDXBU1 then

Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.OBDXBU1.com.ofss.digx.extxface.associatedparty.adapter.IAssociatedPartyAdapter','ExtxfaceAdapterPreference','com.ofss.digx.extxface.associatedparty144.impl.associatedparty.AssociatedPartyAdapter','01','ofssuser',sysdate,'ofssuser',sysdate);

6.2 OBRH Integration

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

For OBDX and mid Office OBASP 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 OBASP (Oracle Banking Associated Party Management) end-points configured in OBRH is **"**ASP**"**
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/OBASP/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 OBDX using OBRH is as follows:

Interface Id	OBDX Screen	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.

This completes the entire configuration needed for consuming OBASP APIs in OBDX through OBRH.

Home

7. Invoice Management

7.1 Core as Third Party

During Bank Implementation when there is case that the core system is Third Party and Mid office is INV (Invoice) then in that case the entry in DIGX_FW_CONFIG_ALL_O will be -

Determinant value for Third Party Entity | TP1.0, INV14.4

For example, if determinant value for Third Party Entity is OBDXBU1 then

OBDXBU1 | TP1.0, INV14.4

Now for all core services, the request will go to Third Party Adapters. But, for all ASP services also, the request will go to third party adapters of ASP.

To avoid that, following scripts need to be executed:

```
Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CRE
ATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.<Determinant
Value>.com.ofss.digx.extxface.invoice.adapter.IInvoiceAdapter','ExtxfaceAdapterPreference','co
m.ofss.digx.extxface.invoice144.impl.invoice.InvoiceAdapter','01','ofssuser',sysdate,'ofssuser',sys
date);
```

For example, if determinant value for Third Party Entity is OBDXBU1 then

```
Insert into DIGX_FW_CONFIG_ALL_O
(PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CRE
ATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values
('corporateuser.OBDXBU1.com.ofss.digx.extxface.invoice.adapter.IInvoiceAdapter','ExtxfaceAda
pterPreference','com.ofss.digx.extxface.invoice144.impl.invoice.InvoiceAdapter','01','ofssuser',sys
date,'ofssuser',sysdate);
```

7.2 OBRH Integration

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

For OBDX and mid Office OBINV 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 OBINV (Oracle Banking Invoice Management) end-points configured in OBRH is **"**INV**"**
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/OBINV/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 OBINV APIs that are integrated with OBDX using OBRH is as follows:

Interface Id	OBDX Screen	Description
INV_INVOICE_CREATE	Create Invoice	This API is used to create invoices.
INV_INVOICE_LIST	View/Edit Invoice	Fetches the list of invoices.
INV_INVOICE_READ	View/Edit Invoice	Fetches invoice details.
INV_INVOICE_UPDATE	View/Edit Invoice	This API is used to modify invoice details.

This completes the entire configuration needed for consuming OBINV APIs in OBDX through OBRH.

Home

8. Virtual Account Management

8.1 OBRH Integration

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

For OBDX and mid Office 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 mid-office product OBVAM (Oracle Banking Virtual Account Management) end-points configured in OBRH is **"OBVAM"**
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/OBVAM/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 OBVAM APIs that are integrated with OBDX using OBRH is as follows:

Interface Id	OBDX Screen	Description
getVirtualAccountClosureStatusDetails	Virtual Account Closure Status Details	This API is used to fetch the closure status of Virtual Accounts that have been initiated for closure or are already closed.
getAccruedAmountForVirtualAccount	Virtual Account Accrued Interest Detail during Virtual Account Closure Initiation flow	Fetches the accrued interest amounts for the requested Virtual Accounts
transferVirtualAccount	Fetch Sibling Virtual Account during Virtual Account Closure Initiation flow	When attempting to close a virtual account, if the virtual account has a non-zero balance then this balance needs to be transferred to another virtual account. This API used to fetch the sibling accounts from where the funds can be debited or credited as per the need

Interface Id	OBDX Screen	Description
retryVirtualAccountClosure	Retry Virtual Account Closure	If for any reason Virtual Account Closure request has failed; this gives the user a chance to retry the closure without changing any of the request parameters
abortVirtualAccountClosure	Abort Virtual Account Closure	If the user wishes to abort a Virtual Account closure request; then this API gives the user a chance to
readVirtualEntity	View Virtual Entity Details	API to fetch details of a Virtual Entity
createVirtualAccount	Create Virtual Account	API to create new Virtual Account
fetchVirtualAccountBalances	Virtual Account Balance overlay	API to fetch balances of a Virtual Account
createGeneralRates	Add General Rate	API to create/add General Rate
vamFetchAdhocCAMTReport	Adhoc Report	API to download Adhoc CAMT report
vamFetchAdhocMTReport	Adhoc Report	API to download Adhoc MT report
vamFetchAdhocPDFReport	Adhoc Report	API to download Adhoc PDF report
vamFetchPreGenReport	Pre-Generated Report	API to fetch Pre-Generated Report (CAMT/MT/PDF)
getChildAccountsForVirtualStructure	View/Edit Virtual Account Structure	API to fetch immediate child accounts of a virtual account in a Structure
fetchDefaultInterestRates	Add General/Special Rate	API to fetch IC Products and IC Rates mapped to a given OBVAM Product + Customer combination
modifyVirtualAccountStructure	Edit Virtual Account Structure	API to modify Virtual Account Structure

Interface Id	OBDX Screen	Description
downloadVASChildAccounts	View Virtual Account Structure	API to download the Virtual Account Structure
fetchInterestHistory	View Virtual Account	API to fetch Interest History of the Virtual Account
reopeningClosedVirtualAccount	Reopen Virtual Account	API to re-open a closed Virtual Account
listVirtualAccWithStructureCode	Create Virtual Account	API to fetch the Virtual Accounts already part of a Structure

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

8.2 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 = '*';

Ensure correct values are maintained against the above properties.

This maintenance should be done from the “System Configuration” admin screen.

Do ensure to follow the **User Credential Configuration** section in this document for configuring Connector Credential store. The OBVAM credentials required to generate token for making outbound REST API calls from OBDX to OBVAM needs to be stored here.

8.3 Generic REST Configurations

Few VAM APIs are not yet concretized i.e. they still follow the PassThru Framework of OBDX (Eg: Fetching countries, currencies etc on VAM screens).

For those APIs to work, kindly execute the below scripts.

Do ensure to replace the values of “branchCode”, “userId” and “sourceCode” values in the **HEADERS** column as per the details of OBVAM environment being pointed to. The “userId” here should be same as the one used to generate token (configured in connector credential store).

Scripts:-

delete from DIGX_AB_API where SERVICE_ID ='fetchCountries' and HOST_NAME='OBVAM' and HOST_VERSION='14.4';

INSERT INTO DIGX_AB_API (SERVICE_ID,SERVICE_NAME,JPATH_ACCTID,JPATH_CUSTID,JPATH_CURR,JPATH_AMT, JPATH_ERRORCODE,HOST_NAME,HOST_VERSION,HEADERS,OBDXURL_PATTERN,METHOD,RESPONSE_REDACTION_TYPE,REQUEST_REDACTION_TYPE,HOST_RESPONSE_REDACTION_TYPE,HOST_REQUEST_REDACTION_TYPE,TASK_ASPECTS,MODULE_NAME, CATEGORY_NAME,ACTION_TYPE,TRANSACTION_TYPE,TASKCODE,OBJECT_VERSION_NUMBER,LAST_UPDATED_DATE,LAST_UPDATED_BY,CREATION_DATE,CREATED_BY,IS_FACTORY_SHIPPED) VALUES ('fetchCountries','List Countries','','','','messages.codes[0].Code','OBVAM','14.4','appId:CMNCORE,branchCode:006,userId:OBVAMUSER1,content-type:application/json,sourceCode:EXTSYS','countries','GET','NONE','NONE','NONE','NONE','','OBVAM_C','OBVAM_SC','PRM','INQUIRY','VAM_I_LCC',1,sysdate,'ofssuser',sysdate,'ofssuser','Y');

delete from DIGX_AB_API where SERVICE_ID ='fetchCurrencies' and HOST_NAME='OBVAM' and HOST_VERSION='14.4';

INSERT INTO DIGX_AB_API (SERVICE_ID,SERVICE_NAME,JPATH_ACCTID,JPATH_CUSTID,JPATH_CURR,JPATH_AMT, JPATH_ERRORCODE,HOST_NAME,HOST_VERSION,HEADERS,OBDXURL_PATTERN,METHOD,RESPONSE_REDACTION_TYPE,REQUEST_REDACTION_TYPE,HOST_RESPONSE_REDACTION_TYPE,HOST_REQUEST_REDACTION_TYPE,TASK_ASPECTS,MODULE_NAME, CATEGORY_NAME,ACTION_TYPE,TRANSACTION_TYPE,TASKCODE,OBJECT_VERSION_NUMBER,LAST_UPDATED_DATE,LAST_UPDATED_BY,CREATION_DATE,CREATED_BY,IS_FACTORY_SHIPPED) VALUES ('fetchCurrencies','List Currencies','','','','messages.codes[0].Code','OBVAM','14.4','appId:CMNCORE,branchCode:006,userId:OBVAMUSER1,content-type:application/json,sourceCode:EXTSYS','currencies','GET','NONE','NONE','NONE','NONE','','OBVAM_C','OBVAM_SC','PRM','INQUIRY','VAM_I_CUR',1,sysdate,'ofssuser',sysdate,'ofssuser','Y');

delete from DIGX_AB_API where SERVICE_ID ='getIdentificationTypes' and HOST_NAME='OBVAM' and HOST_VERSION='14.4';

INSERT INTO DIGX_AB_API (SERVICE_ID,SERVICE_NAME,JPATH_ACCTID,JPATH_CUSTID,JPATH_CURR,JPATH_AMT, JPATH_ERRORCODE,HOST_NAME,HOST_VERSION,HEADERS,OBDXURL_PATTERN,METHOD,RESPONSE_REDACTION_TYPE,REQUEST_REDACTION_TYPE,HOST_RESPONSE_REDACTION_TYPE,HOST_REQUEST_REDACTION_TYPE,TASK_ASPECTS,MODULE_NAME, CATEGORY_NAME,ACTION_TYPE,TRANSACTION_TYPE,TASKCODE,OBJECT_VERSION_NUMBER,LAST_UPDATED_DATE,LAST_UPDATED_BY,CREATION_DATE,CREATED_BY,IS_FACTORY_SHIPPED) VALUES ('getIdentificationTypes','List Identification Types','','','','messages.codes[0].Code','OBVAM','14.4','appId:VAE,branchCode:006,userId:OBVAMUSER1,content-type:application/json,sourceCode:EXTSYS','identificationTypes','GET','NONE','NONE','NONE','NONE','','OBVAM_C','OBVAM_SC','PRM','INQUIRY','VAM_I_GIT',1,sysdate,'ofssuser',sysdate,'ofssuser','Y');

Home

9. Cash Management System

9.1 Core as Third Party

During Bank Implementation when there is case that the core system is Third Party and Mid office is CMS (Cash Management) then in that case the entry in DIGX_FW_CONFIG_ALL_O will be -

Determinant value for Third Party Entity | TP1.0, CMS14.4

For example, if determinant value for Third Party Entity is OBDXBU1 then

OBDXBU1 | TP1.0, CMS14.4

Now for all core services, the request will go to Third Party Adapters. But, for all CMS services also, the request will go to third party adapters of CMS.

To avoid that, following scripts need to be executed:

Insert into DIGX_FW_CONFIG_ALL_O (PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values ('corporateuser.<Determinant Value>.com.ofss.digx.extxface.cms.adapter.cashflow.ICashflowAdapter','ExtxfaceAdapterPreference','com.ofss.digx.extxface.cms144.impl.cashflow.CashflowAdapter','01','ofssuser',sysdate,'ofssuser',sysdate);

For example, if determinant value for Third Party Entity is OBDXBU1 then

Insert into DIGX_FW_CONFIG_ALL_O (PROP_ID,PREFERENCE_NAME,PROP_VALUE,DETERMINANT_VALUE,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE) values ('corporateuser.OBDXBU1.com.ofss.digx.extxface.cms.adapter.cashflow.ICashflowAdapter','ExtxfaceAdapterPreference','com.ofss.digx.extxface.cms144.impl.cashflow.CashflowAdapter','01','ofssuser',sysdate,'ofssuser',sysdate);

9.2 OBCMS Datasource Configuration

We need to create a Datasource in Weblogic for integration with CMS14.4 adapter.

Datasource Details:

Type: Generic Data Source

**Name: CMS

**JNDI Name: CMS

Database Type: Oracle

Scope: Global

*Database Name: <Database_Name>

Hostname: <Hostname>

Port: <Port>

Username: <Username>

Password: <Password>

** Name, JNDI Name should be set to **CMS**; Database Name is the SID/Service Name for the database.

9.3 OBRH Integration

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

For OBDX 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/Modules/OBCM/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 OBCM APIs that are integrated with OBDX using OBRH is as follows:

Interface Id	OBDX Screen	Description
CMS_CASHFLOW_TRANSACTION_CREATE	Create Expected Cash Flow	This API is used to create expected cashflows.
CMS_CASHFLOW_TRANSACTION_LIST	View/Edit Expected Cash Flow Details	Fetches the list of cashflows.
CMS_CASHFLOW_TRANSACTION_READ	View/Edit Expected Cash Flow Details	Fetches Cashflow Details

Interface Id	OBDX Screen	Description
CMS_CASHFLOW_TRANSACTION_UPDATE	View/Edit Expected Cash Flow Details	This API is used to modify expected cashflow details.
CMS_PAYMENTS_LIST	View Payments	This API is used to fetch the list of payments.
CMS_PAYMENT_READ	View Payment Details	This API is used to fetch payment details.
CMS_MANUAL_RECONCILIATION	Manual Reconciliation	This API is used to manually reconcile cashflows/invoices with payments
CMS_LIST_RECONCILED_TRANSACTIONS	De-Reconciliation	This API is used to fetch the list of reconciled cashflows/invoices.
CMS_DERECONCILE	De-Reconciliation	This API is used to de-reconcile already reconciled cashflows/invoices.

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

Home

10. Credit Facility Management

10.1 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

10.2 OBRH Integration

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

For OBDX 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/Modules/ OBCFPM/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 OBCFPM and ELCM APIs that are integrated with OBDX using OBRH is as follows:

Interface Id	OBDX Screen	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.

Interface Id	OBDX Screen	Description
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_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 document.
CF_FETCH_APPLICATION_STATUS	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.

Interface Id	OBDX Screen	Description
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_REVALUATE_COLLATERAL	Collateral Reevaluation	This API is used to revaluate collateral.

Home

11. Liquidity Management

11.1 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 = '*';

Ensure correct values are maintained against the above properties.

This maintenance should be done from the “System Configuration” admin screen.

Do ensure to follow the **User Credential Configuration** section in this document for configuring Connector Credential store. The OBLM credentials required to generate token for making outbound REST API calls from OBDX to OBLM needs to be stored here.

11.2 Generic REST Configurations

Out of the box, OBDX – OBLM Integration is done using JWT token for authentication.

The API call to fetch JWT token is still via PassThru (GenericRest) framework.

Now an additional header for **userId** needs to be sent in this API. The value of this header should be same user id used for fetching token.

Since this user id is dependent on each setup/environment, kindly execute the following script (replacing **%OBLM_USER_ID%** with correct OBLM user id as used for token generation):-

DELETE FROM DIGX_AB_API WHERE service_id = 'tokenOBLM144';

INSERT INTO DIGX_AB_API (SERVICE_ID, SERVICE_NAME, HOST_NAME,
HOST_VERSION,
HEADERS, OBDXURL_PATTERN, METHOD, RESPONSE_REDACTION_TYPE,
REQUEST_REDACTION_TYPE, HOST_RESPONSE_REDACTION_TYPE,
HOST_REQUEST_REDACTION_TYPE, TASK_ASPECTS, MODULE_NAME,
CATEGORY_NAME,
ACTION_TYPE, TRANSACTION_TYPE, TASKCODE, IS_FACTORY_SHIPPED,
OBJECT_VERSION_NUMBER, LAST_UPDATED_DATE, LAST_UPDATED_BY,
CREATION_DATE,
CREATED_BY)
VALUES ('tokenOBLM144', 'Create OBLM JWT', 'OBLM', '14.4',
*'appId:SECSRV001,content-type:application/json,userId:**%OBLM_USER_ID%**',*
'OBLM/token', 'POST', 'NONE', 'NONE', 'NONE', 'NONE', 'audit', 'OBLM_C',
'OBLM_SC', 'PRM', 'MAINTENANCE', 'LM_M_CJWT', 'Y', '1', SYSDATE, 'ofssuser',
SYSDATE, 'ofssuser');

Home

12. User Credential Configuration

For some of the Mid-Office Products (OBVAM, OBTFPM, OBSCF.) 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 -

i. 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';
```

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

Home

13. OBRH Integration Configuration

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

1. To integrate OBRH with OBDX, first some generic configurations and scripts needs to be executed. The Details for the same can be referred from **Oracle Banking Digital Experience Host Integration Guide; Section 7: Configurations for OBRH Integration.**

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

2. For Consuming Mid-Office Product services via OBRH, where OBDX will act as a consumer for OBRH, OBDX 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/<json File>.*

   Refer **OBRH User Manual, Import Service Consumer** section for how to import a consumer JSON in OBRH

3. Once the import is done successfully, you need to update the mid-office service provider's default implementation for IP, Port, Token Username and Token Password. Refer **OBRH User Manual, Add/Edit Implementation** section 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 OBDX Request and Response
b. Transformations needs to be changed in OBRH. Refer **OBRH User Manual, Request and Response Transformation Section.**

Home