

# Oracle® Banking APIs

## UK Open Banking Configuration Guide



Innovation Release 25.1.2.0.0

G51539-01

April 2026

ORACLE®

Copyright © 2006, 2026, Oracle and/or its affiliates.

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Purpose	i
Audience	i
Documentation Accessibility	i
Critical Patches	i
Diversity and Inclusion	ii
Conventions	ii
Related Resources	ii
Screenshot Disclaimer	ii
Acronyms and Abbreviations	ii

## 1 Objective and Scope

---

## 2 Technology Stack

---

## 3 Pre-requisites

---

## 4 Headers Configuration

---

## 5 Properties

---

## 6 OAuth Configuration

---

6.1 UI configuration	1
----------------------	---

## 7 Extensibility and Code Conventions

---

7.1 Key Providers support	3
---------------------------	---

## 8 Keystore and Certificate for UK Open Banking Directory

---

### Index

---

# Preface

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

## Purpose

This guide is designed to help acquaint you with the Oracle Banking application. This guide provides answers to specific features and procedures that the user need to be aware of the module to function successfully.

## Audience

This document is intended for the following audience:

- Customers
- Partners

## Documentation Accessibility

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

### Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

## Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at [Critical Patches, Security Alerts and](#)

[Bulletins](#). All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by [Oracle Software Security Assurance](#).

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<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

## Objective and Scope

### **Background**

Open Banking Configuration Document provides the various configurations required to enable UK Open Banking in OBAPI

### **Scope**

- Headers Configuration
- Properties
- SAML Integration
- OAuth Configuration
- Code Convention and Extensibility



# 2

## Technology Stack

### Technology

Software	Version
Java	Java JDK or JRE version 8
OBDX/OBAPI	25.1.0.0.0
OAuth	OBAPI Internal OAuth

### Abbreviations

OOTB	Out of the Box
TPP	Third Party Providers
ASPSP	Account Servicing Payment Service Provider

# 3

## Pre-requisites

- Java JDK or JRE version 11 or higher must be installed. For installation of Java please refer **Oracle Banking APIs Installation Guide**.
- OAuth Setup

# 4

## Headers Configuration

There are two types of headers configuration available for UK Open Banking.

- System Headers (i.e. Mandatory Headers and its respective value validation)
- Configuration Headers (i.e. Mandatory Headers).

Below are the configuration steps and Out of the box header already configured in the system.

**System Headers:-** As of now in OOTB one header has been added as mandatory “x-fapi-financial-id” with value as “491308330388688” (This is a random value and can be changed. This value is issued by OBIE and corresponds to the Organization Id of the ASPSP in the Open Banking Directory). This value needs to be configured by Bank or ASPSP. This header needs to be sent by the TPP to the ASPSP mandatorily with the same value. Both Header name and Header value are validated for System Headers.

For configuring more system headers, below script is to be executed in the OBAPI Admin schema.

```
Insert into DIGX_FW_CONFIG_ALL_B (PROP_ID, CATEGORY_ID, PROP_VALUE,
FACTORY_SHIPPED_FLAG,
PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY,
LAST_UPDATED_DATE,
OBJECT_STATUS,OBJECT_VERSION_NUMBER) values ('uk%%HEADER NAME%
%', 'OpenbankingSystemHeaders',
'%%HEADERVERVALUE%%', 'N', null, 'Open
Banking', 'ofssuser', sysdate, 'ofssuser', sysdate, 'Y', 1);
```

Below Query is used to check the System Headers in the system

```
select * from digx_fw_config_all_b where category_id =
'OpenbankingSystemHeaders';
```

**Configuration Headers :** As of now in OOTB one header has been added as mandatory - “x-fapi-interaction-id”. This header is required to be sent by the TPP to the ASPSP mandatorily with any value.

Only header name is validated in case of Configuration Headers.

For configuring more config headers, below script is to be executed in the OBDX/OBAPI Admin schema.

```
Insert into DIGX_FW_CONFIG_ALL_B (PROP_ID, CATEGORY_ID, PROP_VALUE,
FACTORY_SHIPPED_FLAG,
PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY,
LAST_UPDATED_DATE,
OBJECT_STATUS, OBJECT_VERSION_NUMBER) values ('uk%%HEADER NAME%%', '
OpenbankingConfigHeaders', null,
'N', null, 'Open Banking', 'ofssuser', sysdate, 'ofssuser', sysdate, 'Y', 1);
```

Below Query is used to check the System Headers in the system

```
select * from digx_fw_config_all_b where category_id =  
'OpenbankingConfigHeaders';
```

# 5

## Properties

Below are the properties required to be updated in the UK Open Banking. Please find the below properties, its purpose and OOTB values.

Table: DIGX\_FW\_CONFIG\_ALL\_B

Category-Id : OpenBankingConfig

Property Id	Property Value (Out of the Box)	Purpose
CONSENT_EXPIRYDAYS	90	This value is used to check if expiry date send by TPP for the Account Access Consent is not more than 90 days and if it is more than 90 days then ASPSP will reject this value

### Token Settings

Table: AUTH\_CONFIG

Category-Id : AuthServerConfig

Property Id	Property Value (Out of the Box)	Purpose
SIGNER	MAC/no row – MAC Signer	The algorithm used to generate JWT token.
	X509RS256 – x509 signed token with RS256 algorithm	
	X509PS256 - x509 signed token with PS256 algorithm	
OAUTH_REDIRECT_HOST_PORT	http://{(HOST)}:{(PORT)}	'HOST' refers to the hostname/IP of the application  'PORT' refers to the application's port

### Sort Code and Branch Mapping for UK.OBIE.SortCodeAccountNumber Scheme

For Sort Code, Account branch mapping following entry needs to be done in DIGX\_FW\_CONFIG\_ALL\_B in openBankingConfig preferences. This mapping used in account identification deserializer to replace sort code with appropriate branch code.

```
Insert into DIGX_FW_CONFIG_ALL_B
( PROP_ID, CATEGORY_ID, PROP_VALUE, FACTORY_SHIPPED_FLAG,
  PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS,
  OBJECT_VERSION_NUMBER, EDITABLE, CATEGORY_DESCRIPTION) values ( 'SORT_CODE_<6
Digit SortCode>',
'openBankingConfig', '<Branch Code>', 'N', null, 'Sort Code Branch Mapping for UK
```

```
Openbanking  
for Sort Code Scheme','ofssuser',sysdate,'ofssuser',sysdate,'A',1,'N',null);
```

# 6

## OAuth Configuration

- [UI configuration](#)

### 6.1 UI configuration

OAuth Identity Domain Maintenance will require below maintenance to configure UI Component for Authorizing consent.

The value of Consent Page URL ( Menu → OAuth → Identity Domain Maintenance) is configured as `http://host:port?existingDashboard=true&homeComponent=authorize-consent&homeModule=open-banking&applicationType=auth.`

# Extensibility and Code Conventions

## **Code Convention of Account API's**

Accounts related API should use below arguments and return type for working with UK Open Banking

### **Arguments**

```
SessionContext sessionContextcom.ofss.digx.app.openbanking.dto.
accounts.uk.AccountRequestDTO accountRequestDTO
```

### **Return Type**

```
BaseResponseDTO<T>Where T extends DataTransferObject
```

Any service implemented with the above type of argument will be compatible with UK Open Banking.

## **Code Convention of Payment API's**

Payment related API should use below arguments and return type for working with UK Open Banking

### **Arguments**

```
Create and Read Method
SessionContext sessionContext
```

Any DTO Object which extends  
com.ofss.digx.app.openbanking.dto.consent.uk.UKPaymentDTO

Any service implemented with the above type of argument will be compatible with UK Open Banking.

## **Error Message Framework**

The Error Message Framework helps convert the OBAPI error response according to the UK Open Banking Specifications.

The error response structure for Open Banking Read/Write APIs is as follows:

```
{  "Code": "...",  "Id": "...",  "Message": "...",
  "Errors": [
    {
      "ErrorCode": "...",
      "Message": "...",      "Path": "...",
      "Url": "..."
    }
  ]
}
```



The UK Open Banking specified error response is handled using DIGX\_OB\_UK\_OBAPI\_ERROR\_MAP table.

The contents of the table are as follows:

Column Name	Description
DIGX_ERROR_CODE	Represents the OBAPI error codes. This is a Primary and Unique Key
UK_ERROR_CODE	Represents the Open Banking specified error code
PATH	Represents the reference to the JSON Path of the field with error.  Can be null.
URL	Represents the URL to help remediate the problem, or provide more information etc.  Can be null.

For mapping OBAPI error codes with UK Open Banking specified codes below script can be used:

```
Insert into DIGX_OB_UK_OBAPI_ERROR_MAP
(DIGX_ERROR_CODE,UK_ERROR_CODE,PATH,URL)
values('%%OBAPI Error Code%%','%%Open Banking specified error code%%', '%%Path%%', '%%URL%%');
```

For example

```
Insert into DIGX_OB_UK_OBAPI_ERROR_MAP
(DIGX_ERROR_CODE,UK_ERROR_CODE,PATH,URL)
values ('DIGX_OB_0010','UK.OBIE.Field.Missing', 'Data.Initiation ',null);
```

Below Query is used to check the OBAPI errors mapped with UK Open Banking specified error codes in the system

```
select * from DIGX_OB_UK_OBAPI_ERROR_MAP;
```

For configuring HTTP status codes with custom message, below script can be used:

```
Insert into DIGX_FW_CONFIG_ALL_B (PROP_ID, CATEGORY_ID, PROP_VALUE,
FACTORY_SHIPPED_FLAG,
PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY,
LAST_UPDATED_DATE,
OBJECT_STATUS,OBJECT_VERSION_NUMBER)
values ('%%HTTP Status code%%','OpenBankingErrorConfig', '%%Error Message%%', 'N',null,
'OpenBanking Error Message','ofssuser',sysdate,'ofssuser',sysdate,'Y',1);
```

Below Query is used to check the Open Banking HTTP status codes in the system

```
select * from digx_fw_config_all_b where category_id = '
OpenBankingErrorConfig';
```

### **Permission Response Handler**

Permissions is used in only Account API's. Based on Permissions, Response is generated based on permissions.

OBAPI consists of Permission Handler against each type of permissions. This configuration is available in the table **DIGX\_OB\_UK\_PERMISSIONS\_PRIMARY**.

The contents of the table are as follows:

Column Name	Description
SERVICEID	Represents the OBAPI Service Id for which the permission and its handler is available
PERMISSION	Represents Permission
RESPONSEHANDLER	Represent Permission Handler

Permission Handler can be overridden or can be newly introduced. This will be required for additional fields mapping which is not available OOTB. Steps for the same are as follows

Introducing Permission Handler

New Permission Handler should implement interface `IResponseHandler`

New Permission Handler should have below methods

**public static** <T implements `IResponseHandler`> getInstance()

**public** <T extends `DataTransferObject`> assembleResponse(`DataTransferObject` object, `List<String>` permissions) – This method assembles response from object to the required response object which needs to be shown in the API response. Object is the response got from base service and T will be the response object required by API specifications. Assembling of the values will be done in this method

**public int** getPriority() – This defines the high priority of the handler to be applied for assembling response in case of permissions and its handler has been consented by the user i.e. Basic and Detail permission will have different handlers but if the consent is both the permission the priority of the handler will decide which needs to be executed on high priority.

- [Key Providers support](#)

## 7.1 Key Providers support

### **Key Providers Overview**

Whenever TPP initiates a DCR request, the payload is signed with the TPP's private key and same needs to be verified with the TPP's public key at the Bank's side. There could be different ways to get the TPP's public key which can vary as per open banking directory services and the geographical regions.

To accommodate those varying approaches of getting the public key, OBDX has provided factory pattern to get a 'Key Provider'. The main job of the key provider is to get the public key of the TPP, to verify the DCR payload, based on the Software Statement Issuer Name.

To implement the above, one IKeyProvider interface is added. This contains the methods which may differ based on the parameters mentioned above.

```

1 package com.ofss.digx.oauth2.spi;
2
3 import java.security.interfaces.RSAPublicKey;
4
5
6 public interface IKeyProvider {
7
8     public String getPublicKey(String clientId, String kid);
9
10    public Map<String, String> fetchPublicKey(String dcr_request_token);
11
12    public Map<String, String> getPublicKeyClaims(String x509Certificate, String keyId);
13
14    /**
15     * Derives the RSA public key from the Base64 public key/certificate
16     *
17     * @param encodedKeyOrCert
18     * @return
19     */
20    public RSAPublicKey getRSAPublicKey(String encodedKeyOrCert);
21 }

```

There are 4 methods to be implemented.

1. **public Map<String, String> fetchPublicKey(String dcr\_request\_token);** - to fetch the TPP's public key when the TPP is being onboarded with the bank with the help of DRC Request Token data.
2. **public String getPublicKey(String clientId, String kid);** - to fetch the TPP's public key based on the client id and the key id for further requests processing as and when required when the TPP is already onboarded with the bank.
3. **public Map<String, String> getPublicKeyClaims(String x509Certificate, String keyId);** - to get the various types of claims like certificate type, validity, expiry, revocation etc.
4. **public RSAPublicKey getRSAPublicKey(String encodedKeyOrCert);** - to get the decrypted RSA public key from the encoded key or extracted from the certificate.

In addition to above methods, to make the key provider class singleton, provider class must implement to return the singleton instance of the class

```
publicstatic IKeyProvider getInstance();
```

### **Key Provider Implementation & Configuration**

To create a key provider, one needs to create a KeyProvider class by extending the com.ofss.digx.oauth2.spi.IKeyProvider interface and making the provider class entry in the DIGX\_FW\_CONFIG\_ALL\_B table.

For example, we have a SSA Issuer called 'XYZ Ltd'.

We will need to follow below two steps to configure the XYZ key provider

1. Need to create a new key provider implementation class - com.ofss.digx.openid.service.XYZKeyProvider which must implement the IKeyProvider interface. Name and the package of the key provider class could be anything, those are not compelled to be same as the mentioned above, but it must implement the IKeyProvider interface.
2. Need to make the provider class entry in the DIGX\_FW\_CONFIG\_ALL\_B with prop\_id = 'XYZ Ltd\_KEY\_PROVIDER'. In this entry, the naming convention should strictly be followed as <SSA\_Issuer>\_KEY\_PROVIDE and the CATEGORY\_ID must be 'openBankingConfig'.

To configure new key provider in DB, refer below insert query and its values are described as below:

```
Insert into DIGX_FW_CONFIG_ALL_B

(PROP_ID,CATEGORY_ID,PROP_VALUE,FACTORY_SHIPPED_FLAG,PROP_COMMENTS,SUMMARY_
TEXT,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE,OBJECT_STAT
US,OBJECT_VERSION_NUMBER,EDITABLE,CATEGORY_DESCRIPTION)
values ('XYZ

Ltd_KEY_PROVIDER','openBankingConfig','com.ofss.digx.openid.service.XYZKeyP
rovider','N',null,'XYZ
Ltd Key Provider
Class','ofssuser',sysdate,'ofssuser',sysdate,'A',1,'N',null);
```

As per the current standards, there are mainly two open banking authorities in European Continent:

1. Open Banking Directory (OBD)
2. European Banking Authority (EBA)

A Third-Party Provider (TPP) gets registered with any of the above two authorities and obtains the Software Statement (SSA) before getting onboarded with the bank.

In this release, OBDX has provided the out of the box implementation of key providers for both directory services.

1. com.ofss.digx.openid.service.OBDKeyProvider – for Open Banking Directory
2. com.ofss.digx.openid.service.EBAKeyProvider – for European Banking Authority

To get the public key, OBD has provided 'software\_jwks\_endpoint'. This endpoint provides a JSON Web Key Set (JWKS), which is a set of keys containing the public keys used to verify any JSON Web Token (JWT). Based on the key id, TPP's public key is extracted from the JWKS to verify the payload.

Both the key providers currently communicate with the Open Banking Directory to fetch the TPP's public key currently as per the implementation.

We have below two configurations:

1. OpenBanking Ltd\_KEY\_PROVIDER – to fetch the public keys of TPP's whose SSA Issuer is the 'OpenBanking Ltd'.
2. DEFAULT\_KEY\_PROVIDER - to fetch the public keys of TPP's whose SSA Issuer is NOT the 'OpenBanking Ltd'.  
Besides above two configured providers, we have a mock key provider (for which, no configuration is needed in the DB):
3. MOCK\_KEY\_PROVIDER- "com.ofss.digx.oauth2.service.DBBasedKeyProvider" – this is only a dummy DB based key provider. If none of the above two providers are configured in the DB, KeyProviderFactory would return the mock key provider. It stores only single public-private key pair in the DB itself and uses the same pair for all the TPP payload verifications.

Below is a sample code snippet to get the key provider for reference:

```
IKeyProvider keyProvider = KeyProviderFactory.getInstance().getProvider(issuer);
Map<String, String> publicKeyMap = keyProvider.fetchPublicKey(dcr_request_token);
```

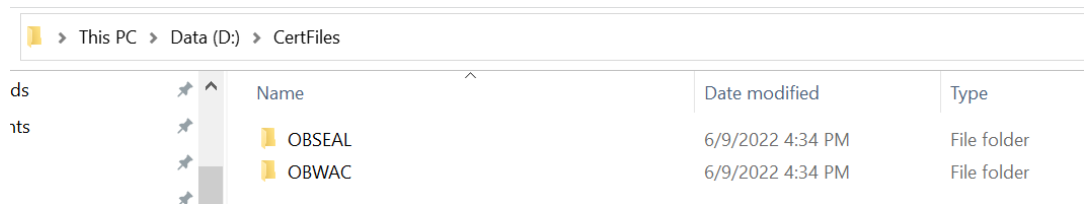
# 8

## Keystore and Certificate for UK Open Banking Directory

This section describes the steps to generate the 'jks' files and configure the same in OBDX for Open Banking Directory integration.

### Steps to create 'identity' & 'trust' JKS files

1. Create two different folders OBWAC and OBSEAL and perform the below steps in the respective folders.



This PC > Data (D:) > CertFiles				
ds	↑	Name	Date modified	Type
nts	↑	OBSEAL	6/9/2022 4:34 PM	File folder
	↑	OBWAC	6/9/2022 4:34 PM	File folder

2. One should have the bank's OBWAC and OBSEAL configuration files(.cnf) to proceed further. Copy the files in the respective folders created above.
3. To generate CSR and key files for OBWAC and OBSEAL certificate with the help of .cnf file, execute below openssl commands  
OBWAC> **openssl req -new -config obwac.cnf -out obwac.csr -keyout obwac.key**  
OSEAL> **openssl req -new -config obseal.cnf -out obseal.csr -keyout obseal.key**

#### Note

Enter the same pass phrase(pass1234 for example) for both obwac and obseal and make a note of it.

```
C:\Windows\System32\cmd.exe

D:\CertFiles\OBWAC>openssl req -new -config obwac.cnf -out obwac.csr -keyout obwac.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'obwac.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
D:\CertFiles\OBWAC>
```

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

D:\CertFiles\OBSEAL>openssl req -new -config obseal.cnf -out obseal.csr -keyout obseal.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'obseal.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
D:\CertFiles\OBSEAL>

```

.csr and .key files have been generated with the above commands

This PC > Data (D:) > CertFiles > OBWAC				
Name	Date modified	Type	Size	
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB	
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB	
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB	

This PC > Data (D:) > CertFiles > OBSEAL				
Name	Date modified	Type	Size	
obseal.cnf	4/19/2022 3:59 PM	CNF File	7 KB	
obseal.csr	6/9/2022 4:34 PM	CSR File	2 KB	
obseal.key	6/9/2022 4:34 PM	KEY File	2 KB	

4. Upload the above generated .csr files in Open Banking Directory Account to get OBWAC and OBSEAL pem files.

Let's assume, below output on uploading .csr files in the OB directory account

Your OB WAC certificate xT-9\_jWfAME1feTKZGaf8Dd\_x1s was successfully created

Your OB Seal certificate l6cfLYUSt91fOw13kdO0HYdIVTc was successfully created

Below are the steps to generate the OB WAC and OB Seal certificates in the Open Banking Directory Account(**Note: Below screenshots are from the Sandbox account, kindly use Production Open Banking Directory Account details for the production setup**)

- a. Login with Open Banking Directory account credentials and select the desired Directory Participant(Your Organization).

Search the Directory

Search by name or ID

Your Organisations

PS02 ORACLE CORPORATION UK LIMITED

Directory participants > ORACLE CORPORATION UK LIMITED

## ORACLE CORPORATION UK LIMITED

Organisation Details | Software Statements | Certificates

### Business information

Name	Organisation Id	Creation Date
ORACLE CORPORATION UK LIMITED	0014H00001IFE77QAG	Friday, 3rd September 2021, 1:28 am

### Address

PrimaryType	Name	Address
✓ Registered	Registered Office Address	Oracle Parkway, Thames Valley Park, Reading, RG6 1RA, United Kingdom

b. Go to 'Certificates' tab

Search the Directory

Search by name or ID

Your Organisations

PS02 ORACLE CORPORATION UK LIMITED

Directory participants > ORACLE CORPORATION UK LIMITED

## ORACLE CORPORATION UK LIMITED

Organisation Details | Software Statements | **Certificates**

[Add new Organisation Certificate](#)

### Organisation level certificates i

**⚠** Since 01 Jan 2021, EU and UK TPPs have different restrictions applied when minting/uploading certain certificates in the ecosystem.

### Transport certificates

c. Click on 'Add new Organisation Certificate' button

Search the Directory

Search by name or ID

Your Organisations

PS02 ORACLE CORPORATION UK LIMITED

Directory participants > ORACLE CORPORATION UK LIMITED > Organisation Level Certificates

> Add new Organisation Level Certificate

## Add Certificate

Before adding a certificate, please ensure you have read our information page

Please select the type of certificate being created \*

☐ Signing ☐ Transport

d. Select 'Signing' radio button to upload OB Seal .crs file

Search the Directory

Search by name or ID

Your Organisations

PS02 ORACLE CORPORATION UK LIMITED

Directory participants > ORACLE CORPORATION UK LIMITED > Organisation Level Certificates > Add new Organisation Level Certificate

## Add Certificate

Before adding a certificate, please ensure you have read our information page

Please select the type of certificate being created \*

☒ Signing ☐ Transport

Please select the file type for certificate generation \*

☒ Upload a .csr file for OB Seal ☐ Upload a .pem file for GSeal

Upload a CSR

Click to select file

Upload

- e. Select and upload the OB Seal .csr file

Search the Directory

Search by name or ID

Your Organisations

PS02 ORACLE CORPORATION UK LIMITED

Directory participants > ORACLE CORPORATION UK LIMITED > Organisation Level Certificates > Add new Organisation Level Certificate

## Add Certificate

Before adding a certificate, please ensure you have read our information page

Please select the type of certificate being created \*

☒ Signing ☐ Transport

Please select the file type for certificate generation \*

☒ Upload a .csr file for OB Seal ☐ Upload a .pem file for GSeal

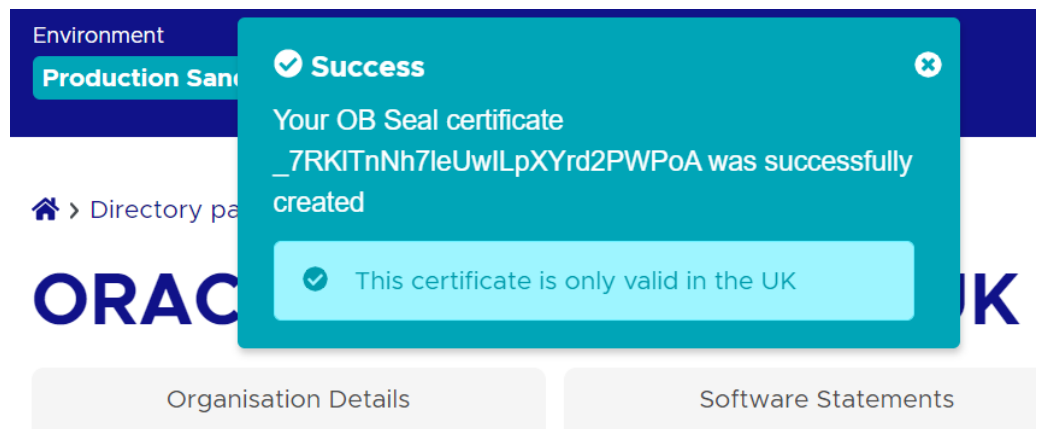
Upload a CSR

Click to select file

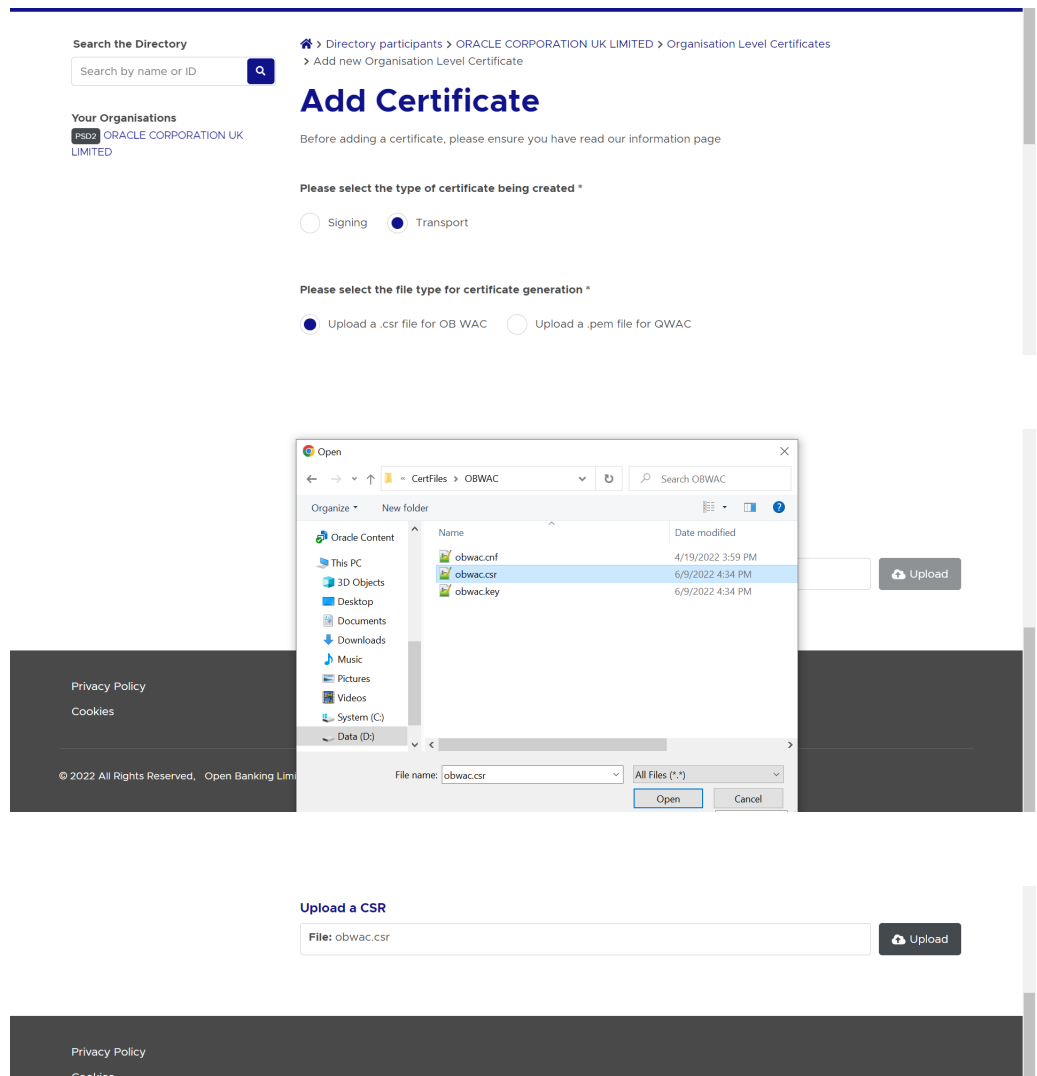
Upload

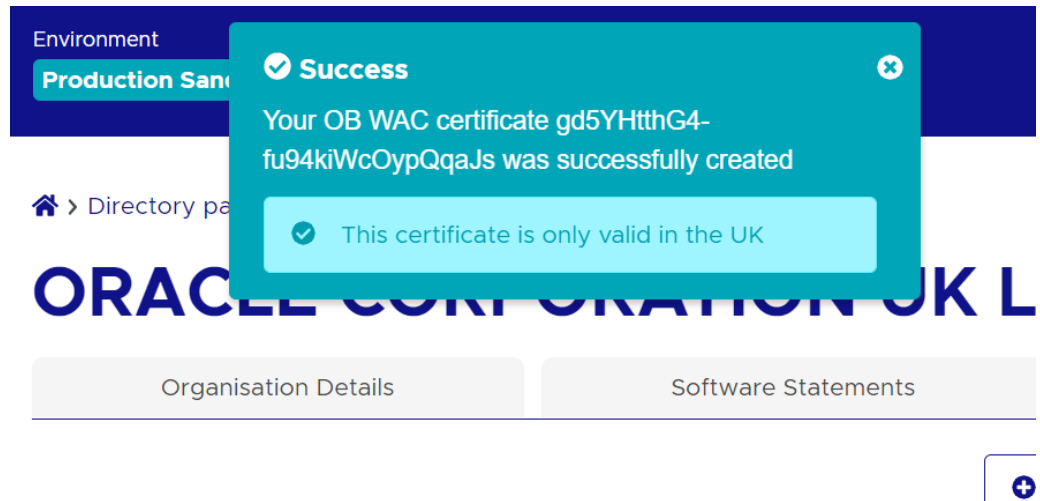
- f. Clicking on the 'Upload' button will upload and display success popup



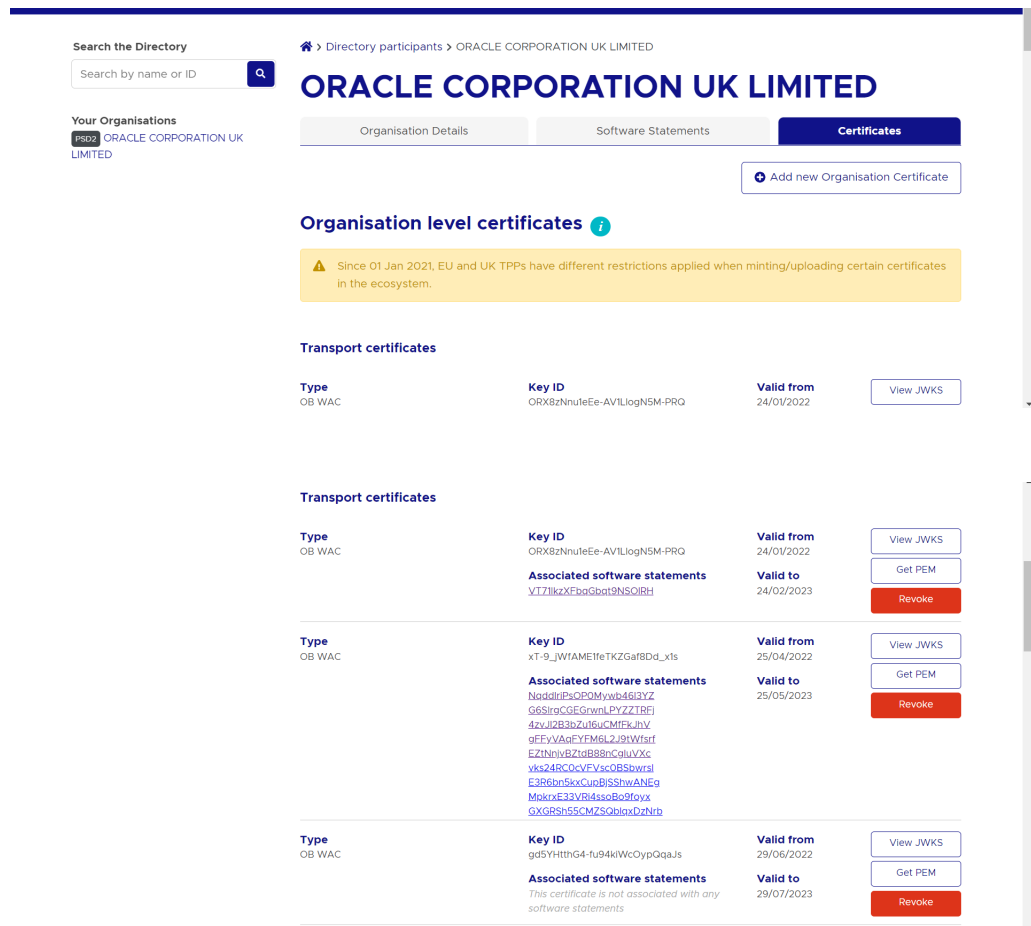


- g. Repeat the above steps for OB WAC certificate generation. Select the 'Transport' radio button for OB WAC.





- h. Generated certificates would be visible on the certificates listing page. Certificate .pem files can be downloaded with the help of 'Get PEM' button displayed next to the respective certificates



Signing certificates				
<b>Type</b> OB Seal	<b>Key ID</b> BDHknaKeeNnP_XHjSwJLJu4Xs	<b>Valid from</b> 24/01/2022	<a href="#">View JWKS</a>	
	<b>Associated software statements</b> <a href="#">Y17JlkzXFbaGbat9NSOIRH</a>	<b>Valid to</b> 24/02/2023	<a href="#">Get PEM</a>	
			<a href="#">Revoke</a>	
<b>Type</b> OB Seal	<b>Key ID</b> I6cFLYUSt9fOw13kdO0HYdlVTc	<b>Valid from</b> 25/04/2022	<a href="#">View JWKS</a>	
	<b>Associated software statements</b> <a href="#">NaddlrPsOPOMywb46i3YZ</a> <a href="#">G6SrgCGEGwnLPYZTRFJ</a> <a href="#">4zvJl2B3bZu16uCMFkjhV</a> <a href="#">qEFyVAgEYEM6L2J9tWtsrf</a> <a href="#">EZTnnvBZtdB88nCGluVXc</a> <a href="#">yxs24RCOcVFVscOBsbwral</a> <a href="#">E3R6bn5kxCupBjSshwANEq</a> <a href="#">MoknE33VRd4ssoBo9foyx</a> <a href="#">GXGRSh55CMZScabenaDzhNtp</a>	<b>Valid to</b> 25/05/2023	<a href="#">Get PEM</a>	
			<a href="#">Revoke</a>	
<b>Type</b> OB Seal	<b>Key ID</b> _7RKITnN7leUwLpXYrd2PWPoA	<b>Valid from</b> 29/06/2022	<a href="#">View JWKS</a>	
	<b>Associated software statements</b> <i>This certificate is not associated with any software statements</i>	<b>Valid to</b> 29/07/2023	<a href="#">Get PEM</a>	
			<a href="#">Revoke</a>	

- Download the generated OBWAC and OBSEAL files and copy in the respective folders which have created locally. Change the extension from '.cer' to '.pem' of the downloaded files if required.

This PC > Data (D:) > CertFiles > OBWAC				
Name	Date modified	Type	Size	
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB	
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB	
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB	
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	6/9/2022 4:40 PM	PEM File	3 KB	

This PC > Data (D:) > CertFiles > OBSEAL				
Name	Date modified	Type	Size	
I6cFLYUSt9fOw13kdO0HYdlVTc.pem	6/9/2022 4:41 PM	PEM File	3 KB	
obseal.cnf	4/19/2022 3:59 PM	CNF File	7 KB	
obseal.csr	6/9/2022 4:34 PM	CSR File	2 KB	
obseal.key	6/9/2022 4:34 PM	KEY File	2 KB	

- Generate decrypted keys by executing below commands  
 OBWAC> **openssl rsa -in obwac.key -out obwac\_dec.key**  
 OBSEAL> **openssl rsa -in obseal.key -out obseal\_dec.key**

Enter the pass phrase 'pass1234' when provided, which had been entered at the time of the .key files.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

D:\CertFiles\OBWAC>openssl rsa -in obwac.key -out obwac_dec.key
Enter pass phrase for obwac.key:
writing RSA key

D:\CertFiles\OBWAC>
```

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

D:\CertFiles\OBSEAL>openssl rsa -in obseal.key -out obseal_dec.key
Enter pass phrase for obseal.key:
writing RSA key

D:\CertFiles\OBSEAL>

```

Data (D:) > CertFiles > OBWAC

Name	Date modified	Type	Size
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB
obwac_dec.key	6/9/2022 4:43 PM	KEY File	2 KB
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	6/9/2022 4:40 PM	PEM File	3 KB

Data (D:) > CertFiles > OBSEAL

Name	Date modified	Type	Size
I6cfLYUSt91fOw13kdO0HYdlVTc.pem	6/9/2022 4:41 PM	PEM File	3 KB
obseal.cnf	4/19/2022 3:59 PM	CNF File	7 KB
obseal.csr	6/9/2022 4:34 PM	CSR File	2 KB
obseal.key	6/9/2022 4:34 PM	KEY File	2 KB
obseal_dec.key	6/9/2022 4:44 PM	KEY File	2 KB

## 7. Download OB Root and Issuing Certificates from the Open Banking directory

- URL for sandbox certificates: <https://openbanking.atlassian.net/wiki/spaces/DZ/pages/252018873/OB+Root+and+Issuing+Certificates+for+Sandbox>
- URL for production certificates: <https://openbanking.atlassian.net/wiki/spaces/DZ/pages/80544075/OB+Root+and+Issuing+Certificates+for+Production>

## 8. Create a copy of both the downloaded certificate files and change the extension from .cer to .pem and copy in the OBWAC folder. Keep the file names same

- OB\_SandBox\_PP\_IssuingCA.cer to OB\_SandBox\_PP\_IssuingCA.pem
- OB\_SandBox\_PP\_RootCA.cer to OB\_SandBox\_PP\_RootCA.pem

### Note

Remove the spaces from the pem file names if there are any.

Data (D:) > CertFiles > OBWAC

Name	Date modified	Type	Size
OB_SandBox_PP_IssuingCA.cer	6/9/2022 4:48 PM	Security Certificate	2 KB
OB_SandBox_PP_IssuingCA.pem	6/9/2022 4:48 PM	PEM File	2 KB
OB_SandBox_PP_RootCA.cer	6/9/2022 4:48 PM	Security Certificate	2 KB
OB_SandBox_PP_RootCA.pem	6/9/2022 4:48 PM	PEM File	2 KB
obwac_dec.key	6/9/2022 4:43 PM	KEY File	2 KB
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	6/9/2022 4:40 PM	PEM File	3 KB
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB

9. Use 'cat' command on linux or 'type' command in Windows machine to build the certificate chain from the above three .pem files

- a. `cat xT-9_jWfAME1feTKZGaf8Dd_x1s.pem OB_SandBox_PP_IssuingCA.pem OB_SandBox_PP_RootCA.pem > chain.pem`

OR

- b. `type xT-9_jWfAME1feTKZGaf8Dd_x1s.pem OB_SandBox_PP_IssuingCA.pem OB_SandBox_PP_RootCA.pem > chain.pem`

```
C:\Windows\System32\cmd.exe
D:\CertFiles\OBWAC>type xT-9_jWfAME1feTKZGaf8Dd_x1s.pem OB_SandBox_PP_IssuingCA.pem OB_SandBox_PP_RootCA.pem > chain.pem
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem
OB_SandBox_PP_IssuingCA.pem
OB_SandBox_PP_RootCA.pem
D:\CertFiles\OBWAC>
```

Data (D:) > CertFiles > OBWAC

Name	Date modified	Type	Size
chain.pem	6/9/2022 5:42 PM	PEM File	5 KB
OB_SandBox_PP_IssuingCA.cer	6/9/2022 4:48 PM	Security Certificate	2 KB
OB_SandBox_PP_IssuingCA.pem	6/9/2022 4:48 PM	PEM File	2 KB
OB_SandBox_PP_RootCA.cer	6/9/2022 4:48 PM	Security Certificate	2 KB
OB_SandBox_PP_RootCA.pem	6/9/2022 4:48 PM	PEM File	2 KB
obwac_dec.key	6/9/2022 4:43 PM	KEY File	2 KB
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	6/9/2022 4:40 PM	PEM File	3 KB
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB

## 10. Creating Custom Keystore and importing chain

WebLogic Server Java Utilities is used to create the custom keystore and importing private key & the certificates chains.

Resource URL for reference: [https://docs.oracle.com/cd/E13222\\_01/wls/docs81/admin\\_ref/utlis20.html](https://docs.oracle.com/cd/E13222_01/wls/docs81/admin_ref/utlis20.html)

Execute the below command with files in the OBWAC directory










```
java -cp /home/devops/Oracle/Middleware/Oracle_Home/wlserver/server/lib/weblogic.jar utils.ImportPrivateKey -certfile chain.pem -keyfile obwac_dec.key -keystore openbanking_custom_identity.jks -storepass pass1234 -alias openbanking_obtrans
```

### Note

“/home/devops/Oracle/Middleware/Oracle\_Home/wlserver/server/lib/” this path is to locate the weblogic.jar file, this may differ as per the setup.

```
@obdxwls/scratch/obdx/OpenBanking/CertFiles/OBWAC
[devops@obdxwls OBWAC]$ ls -l
total 40
-rwxrwxrwx 1 54323 54323 1559 Jun 9 16:48 OB_SandBox_PP_IssuingCA.pem
-rwxrwxrwx 1 54323 54323 1380 Jun 9 16:48 OB_SandBox_PP_RootCA.pem
-rwxrwxrwx 1 54323 54323 5048 Jun 9 17:42 chain.pem
-rwxrwxrwx 1 54323 54323 7712 Apr 19 15:59 obwac.cnf
-rwxrwxrwx 1 54323 54323 1392 Jun 9 16:34 obwac.csr
-rwxrwxrwx 1 54323 54323 1884 Jun 9 16:34 obwac.key
-rwxrwxrwx 1 54323 54323 1706 Jun 9 16:43 obwac_dec.key
-rwxrwxrwx 1 54323 54323 2109 Jun 9 16:40 xT-9_jWfAME1feTKZGaf8Dd_x1s.pem
[devops@obdxwls OBWAC]$ java -cp /home/devops/Oracle/Middleware/Oracle_Home/wlserver/server/lib/weblogic.jar utils.ImportPrivateKey -certfile chain.pem -keyfile obwac_dec.key -keystore openbanking_custom_identity.jks -storepass pass1234 -alias openbanking_obtrans
No password was specified for the key entry
Keystore password will be used
<Jun 9, 2022 6:23:49 PM IST> <Info> <Security> <BEA-090905> <Disabling the CryptoJ JCE Provider self-integrity check for better startup performance. To enable this check, specify -Dweblogic.security.allowCryptoJDefaultJCEVerification=true.>
<Jun 9, 2022 6:23:49 PM IST> <Info> <Security> <BEA-090906> <Changing the default Random Number Generator in RSA CryptoJ from ECDRBG128 to HMACDRBG. To disable this change, specify -Dweblogic.security.allowCryptoJDefaultPRNG=true.>
Imported private key obwac_dec.key and certificate chain.pem
into a new keystore openbanking_custom_identity.jks of type jks under alias openbanking_obtrans
[devops@obdxwls OBWAC]$
```

A new .jks file with the filename ‘openbanking\_custom\_identity.jks’ is created.

/scratch/trunk_docker/obdx/OpenBanking/CertFiles/OBWAC/					
Name	Size	Changed	Rights	Owner	
 openbanking_custom_identity.jks	3 KB	6/9/2022 6:02:42 PM	rwxrwxrwx	obdxdevops	
 chain.pem	5 KB	6/9/2022 5:42:07 PM	rw-rw-r--	1000	
 OB_SandBox_PP_IssuingCA.pem	2 KB	6/9/2022 4:48:14 PM	rwxrwxrwx	obdxdevops	
 OB_SandBox_PP_RootCA.pem	2 KB	6/9/2022 4:48:08 PM	rwxrwxrwx	obdxdevops	
 obwac_dec.key	2 KB	6/9/2022 4:43:12 PM	rwxrwxrwx	obdxdevops	
 xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	3 KB	6/9/2022 4:40:39 PM	rwxrwxrwx	obdxdevops	
 obwac.key	2 KB	6/9/2022 4:34:17 PM	rwxrwxrwx	obdxdevops	
 obwac.csr	2 KB	6/9/2022 4:34:17 PM	rwxrwxrwx	obdxdevops	
 obwac.cnf	8 KB	4/19/2022 3:59:26 PM	rwxrwxrwx	obdxdevops	

## 11. Creating Custom Identity Trust

Execute below two commands.

Enter 'yes' and press enter when prompted “Trust this certificate? [no]:”.

### Note

“/home/devops/jdk18/bin/” this path is to locate the java keytool, this may differ as per the setup.

a.

```

@obdwls/scratch/obdw/OpenBanking/CertFiles/OBWAC
[devops@obdwls OBWAC]$ ls -l
total 52
-rw-rw-r-- 1 54323 54323 1559 Jun  9 16:48 OB_SandBox_PP_IssuingCA.cer
-rw-rw-rw- 1 54323 54323 1559 Jun  9 16:48 OB_SandBox_PP_IssuingCA.pem
-rw-rw-r-- 1 54323 54323 1380 Jun  9 16:48 OB_SandBox_PP_RootCA.cer
-rw-rw-rw- 1 54323 54323 1380 Jun  9 16:48 OB_SandBox_PP_RootCA.pem
-rw-rw-rw- 1 54323 54323 5048 Jun  9 17:42 chain.pem
-rw-rw-rw- 1 54323 54323 7712 Apr 19 15:59 obwac.cnf
-rw-rw-rw- 1 54323 54323 1392 Jun  9 16:34 obwac.csr
-rw-rw-rw- 1 54323 54323 1884 Jun  9 16:34 obwac.key
-rw-rw-rw- 1 54323 54323 1706 Jun  9 16:43 obwac.dec.key
-rw-rw-r-- 1 devops devops 2884 Jun  9 18:23 openbanking_custom_identity.jks
-rw-rw-rw- 1 54323 54323 2109 Jun  9 16:40 XT-9_WIAMElifeTKZGaf8dd_xls.pem
[devops@obdwls OBWAC]$ /home/devops/jdk18/bin/keytool -importcert -file OB_SandBox_PP_RootCA.cer -alias openbankin
g_rootca -storepass pass1234
Owner: CN=OpenBanking Pre-Production Root CA, O=OpenBanking, C=GB
Issuer: CN=OpenBanking Pre-Production Root CA, O=OpenBanking, C=GB
Serial number: 58cfc6
Valid from: Fri Sep 22 17:09:42 IST 2017 until: Tue Sep 22 17:39:42 IST 2037
Certificate fingerprints:
    MD5: 38:BC:2F:F0:7F:34:A0:E0:42:DB:65:81:51:F9:6C:D7
    SHA1: 3C:97:AD:3F:63:9B:21:EF:00:F3:39:93:90:61:6C:8A:7D:0D:5F:03
    SHA256: 73:24:4E:0D:1F:5B:01:C5:F6:E5:A1:A0:2A:18:AC:67:10:01:4F:2C:AF:A3:0A:53:52:87:FE:37:A3:70:74:2F
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3
Extensions:
#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
    CA:true
    PathLen:2147483647
]
#2: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
    Key_CertSign
    Crl_Sign
]
#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
    keyIdentifier [

```

- b. **/home/devops/jdk18/bin/keytool -keystore openbanking\_custom\_trust.jks -importcert -file OB\_SandBox\_PP\_IssuingCA.cer -alias openbanking\_issueca -storepass pass1234**

```

@obdwls/scratch/obdw/OpenBanking/CertFiles/OBWAC
[devops@obdwls OBWAC]$ /home/devops/jdk18/bin/keytool -importcert -file OB_SandBox_PP_RootCA.cer -alias openbankin
g_rootca -storepass pass1234
Owner: CN=OpenBanking Pre-Production Root CA, O=OpenBanking, C=GB
Issuer: CN=OpenBanking Pre-Production Root CA, O=OpenBanking, C=GB
Serial number: 58cfc6
Valid from: Fri Sep 22 17:09:42 IST 2017 until: Tue Sep 22 17:39:42 IST 2037
Certificate fingerprints:
    MD5: 38:BC:2F:F0:7F:34:A0:E0:42:DB:65:81:51:F9:6C:D7
    SHA1: 3C:97:AD:3F:63:9B:21:EF:00:F3:39:93:90:61:6C:8A:7D:0D:5F:03
    SHA256: 73:24:4E:0D:1F:5B:01:C5:F6:E5:A1:A0:2A:18:AC:67:10:01:4F:2C:AF:A3:0A:53:52:87:FE:37:A3:70:74:2F
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3
Extensions:
#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
    CA:true
    PathLen:2147483647
]
#2: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
    Key_CertSign
    Crl_Sign
]
#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
    keyIdentifier [
0000: EC 38 8E 0B DA F3 F9 37 3E 90 DE 7D 5F 6A E6 60 .0.....7....J.'
0010: CD 79 42 83 .yB.
    ]
]
Trust this certificate? [no]: yes
Certificate was added to keystore

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srcke
ystore openbanking_custom_identity.jks -destkeystore openbanking_custom_identity.jks -deststoretype pkcs12".
[devops@obdwls OBWAC]$

```

Another .jks file with filename 'openbanking\_custom\_trust.jks' is created.

```
/scratch/trunk docker/obdx/OpenBanking/CertFiles/OBWAC/
```

Name	Size	Changed	Rights	Owner
..		6/9/2022 6:02:42 PM	rxwxrwxrwx	obdxdevops
openbanking_custom_trust.jks	2 KB	6/9/2022 6:47:47 PM	rw-rw-r--	1000
openbanking_custom_identity.jks	5 KB	6/9/2022 6:43:54 PM	rw-rw-r--	1000
chain.pem	5 KB	6/9/2022 5:42:07 PM	rxwxrwxrwx	obdxdevops
OB_SandBox_PP_IssuingCA.pem	2 KB	6/9/2022 4:48:14 PM	rxwxrwxrwx	obdxdevops
OB_SandBox_PP_IssuingCA.cer	2 KB	6/9/2022 4:48:14 PM	rw-rw-r--	obdxdevops
OB_SandBox_PP_RootCA.pem	2 KB	6/9/2022 4:48:08 PM	rxwxrwxrwx	obdxdevops
OB_SandBox_PP_RootCA.cer	2 KB	6/9/2022 4:48:08 PM	rw-rw-r--	obdxdevops
obwac_dec.key	2 KB	6/9/2022 4:43:12 PM	rxwxrwxrwx	obdxdevops
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	3 KB	6/9/2022 4:40:39 PM	rxwxrwxrwx	obdxdevops
obwac.key	2 KB	6/9/2022 4:34:17 PM	rxwxrwxrwx	obdxdevops
obwac.csr	2 KB	6/9/2022 4:34:17 PM	rxwxrwxrwx	obdxdevops
obwac.cnf	8 KB	4/19/2022 3:59:26 PM	rxwxrwxrwx	obdxdevops

OpenSSL 1.1.1n 15 Mar 2022 is used to perform above steps.

UK Open Banking Configuration Guide  
G51539-01  
Copyright © 2006, 2026, Oracle and/or its affiliates.



# Index

## E

---

Extensibility and Code Conventions, [1](#)

## H

---

Headers Configuration, [1](#)

## K

---

Key Providers support, [3](#)

Keystore and Certificate for UK Open Banking  
Directory, [1](#)

## O

---

Objective and Scope, [1](#)

## P

---

Pre-requisites, [1](#)

Properties, [1](#)

## T

---

Technology Stack, [1](#)

## U

---

UI configuration, [1](#)