NewgenONE OmniDocs

Mobile

Deployment Guide

Version: 11.3

Disclaimer

This document contains information proprietary to Newgen Software Technologies Ltd. User may not disclose or use any proprietary information or use any part of this document without written permission from Newgen Software Technologies Ltd.

Newgen Software Technologies Ltd. makes no representations or warranties regarding any software or to the contents or use of this guide. It also specifically disclaims any express or implied warranties of merchantability, title, or fitness for any particular purpose. Even though Newgen Software Technologies Ltd. has tested the hardware and software and reviewed the documentation, it does not guarantee or imply that this document is error free or accurate regarding any particular specification. As a result, this product is sold as it is and user, the purchaser, is assuming the entire risk as to its quality and performance. Further, Newgen Software Technologies Ltd. reserves the right to revise this publication and make changes in its content without any obligation to notify any person, of such revisions or changes. Newgen Software Technologies Ltd. authorizes no Newgen agent, dealer or employee to make any modification, extension, or addition to the above statements.

Newgen Software Technologies Ltd. has attempted to supply trademark information about company names, products, and services mentioned in this document. Trademarks indicated below were derived from various sources.

Newgen Software, Registered Office, New Delhi
E-44/13
Okhla Phase - II
New Delhi 110020
India
Phone: +91 1146 533 200
info@newgensoft.com

Contents

Preface

This guide describes the deployment process to successfully implement OmniDocs Mobile 11.3 in your environment.

Revision history

Revision date	Description
July 2024	Initial publication

Intended audience

This guide is intended for the following sets of users:

- Client machine — for users who set up the client project and run it for development. The reader must have knowledge of various development integrated development environments (IDEs) like Android Studio and Xcode. The reader must have access to an Apple Developer Account for certificate creation and a Mac machine for iOS app development.
- Server machine — for users who manually set up the OmniDocs Mobile server, without running the installer. The reader must have knowledge of different application servers such as JBoss, WebLogic, and WebSphere, and database servers such as Oracle, PostgreSQL , and Microsoft (MS) SQL Server. The reader must have administrative rights.

Documentation feedback

To provide feedback or any improvement suggestions on technical documentation, write an email to docs.feedback@newgensoft.com.
To help capture your feedback effectively, share the following information in your email:

- Document name
- Version
- Chapter, topic, or section
- Feedback or suggestions

Introduction to NewgenONE OmniDocs Mobile

NewgenONE OmniDocs Mobile is a valuable tool for businesses seeking to streamline their document management processes, improve collaboration, and enhance productivity in an increasingly mobile and digital work environment. OmniDocs Mobile server is built on a microservices architecture and consists of the following JAR and WAR files:

- Api-gateway
- Authenticate-service
- Folder-service
- Document-service
- Omni process-service

Deployment steps

To deploy OmniDocs Mobile 11.3, perform the following steps:

1. To initialize the OmniDocs Mobile Server components, run the Api Gateway WAR file by using the below command:

   ```
   Java –jar api-gateway.war
   ```

2. Configure your OmniDocs Server IP, Port, and database details. Locate the *application.properties* file inside the *authenticate-service.war* file.
   Path to the file — *authenticate-service\target\authenticate-service.war\WEB-INF\classes\application.properties*
   Update the following properties in the *application.properties* file:
   - od-ip=http:<Application Server IP>
   - od-port=<http connector port of the application server>

     Where,

     <Application server IP> is the IP of the machine where the application server is running.

     <http connector port of the application server> is the Port of the machine where the application server is running. The default port for JBoss is **8080**, for WebLogic, it is **7001**, and for WebSphere, it is **9080**.

3. To configure the database settings, uncomment the details of the desired database in the *application.properties* file and comment out the details of other databases:

   To activate the database details, remove the # symbol at the beginning of each line for the desired database configuration.

   ```
   #ForPostgreSQL
   #spring.datasource.url=jdbc:postgresql://aurora-posxxxxxx.cluster-
   cvxxxx.apxxxxzoxxs.com/odpoxxxresxxxx4
   #spring.datasource.username=abc
   #spring. datasource.password=abc123$
   #spring.jpa.hibernate.ddl-auto=update
   #spring.jpa.show-sql=false
   #spring.jpa.properties.hibernate.format_sql=false
   #spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQL81Dialect
   ```

```
#ForOracleSQL
#spring.datasource.url=jdbc:oracle:thin:@192.16x.1xx.xx:15xx/orclpdb
#spring.datasource.username=abc
#spring.datasource.password=abc123$
#spring.jpa.hibernate.ddl-auto=update
#spring.jpa.show-sql=false
#spring.jpa.properties.hibernate.format_sql=false
#spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
```

Also, update the following properties in the *application.properties* file inside the *authenticate-service.war* file.
Path — *authenticate-service\target\authenticate-service.war\WEB-INF\classes*

- spring.datasource.url
- spring.datasource.username
- spring.datasource.password

Below is the uncommented database configuration example (MSSQL):

```
#ForMSSQL
#spring.datasource.jndi-name = java:jboss/datasources/testDB
spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver
spring.datasource.url=jdbc:sqlserver://
192.168.1xx.xx:14xx;databaseName=od11sp2patch1fsql22m;encrypt =
true;trustServerCertificate=true;
spring.datasource.username= abc
spring.datasource.password= abc123
spring.jpa.show-sql=false
spring.jpa.properties.hibernate.format_sql = false

## Hibernate Properties
# The SQL dialect makes Hibernate generate better SQL for the chosen database
spring.jpa.properties.hibernate.dialect =
org.hibernate.dialect.SQLServer2012Dialect
# Hibernate ddl auto (create, create-drop, validate, update)
spring.jpa.hibernate.ddl-auto = update
```

4. Before starting the authenticate-service, copy the *odcablist.ini* file from the Jboss, WebSphere, or, Weblogic location and put it inside the *authenticate-service.war\WEB-INF\classes*.

   Whenever you register a new cabinet on the web remember to update *odcablist.ini* (the INI file must be in sync).

   Below are paths for the application server:

   - JBoss — *jboss-eap-7.4\bin\Newgen\NGConfig\ngdbini\odwebini*

- WebSphere — *WEBSPHERE_PROFILE\bin\Newgen\NGConfig\ngdbini\odwebini*
- WebLogic — *WEBLOGIC_DOMAIN\bin\Newgen\NGConfig\ngdbini\odwebini*

- For password encryption below tag default value is Y, If you want to remove the encryption set the value to N
  od-encryptPa$$=Y
- Use 2 for SHA-256 and 3 For SHA-512
  od-pa$$AlgoValue=2
- By default, the landing page on the first appearance is set to the repository (od-initialTabValue=3). To change the landing page to another option, use the corresponding code from the list:
  - Repository= 3
  - Favourites = 4
  - Omniprocess = 1
  - Dashboard = 0
- OmniDocs WebHelp and support section can be accessed using the URL specified in 'od-supportFilePath': *http://192.168.149.xx:80xx/omnidocs/estyle/web_help/Responsive_HTML5/OmniDocsWeb.htm#t=Introduction.htm. Users can also specify their own help and support path if needed.*
- Provide the URL in the *od-webUrl* tag inside the *application.properties* file. For example: *http://192.168.1xx.xx:80xx*.

5. After making the above changes, execute the following command to start OmniDocs Mobile Server Components:

```
Java –jar authenticate-service.war
```

6. Mention the IP and port of your deployed OmniDocs server inside the *application.properties* file for *folder-service.war*, *document-service.war*, and *omniprocess-service.war*.
   - od-ip=http:<Application Server IP>
   - od-port=<http connector port of the application server>
     Where,
     <Application server IP> is the IP of the machine where the application server is running.
     <http connector port of the application server> is the Port of the machine where the application server is running. The default port for JBoss is **8080**, for WebLogic, it is **7001**, and for WebSphere, it is **9080**.

7. Configuring files for Folder Service:
   - Before starting, ensure folder-service is synced with Application Server by copying and updating the *eworkstyle.ini* and *uploadmime.conf* files specific to the cabinet from the JBoss, WebLogic, and WebSphere location to the

CabinetConfiguration folder inside the *folder-service.war* file. Below are the paths for the application server:

- Path - *folder-service.war\WEB-INF\classes\CabinetConfiguration\od11sp2march13*

  Here, *od11sp2march13* is used as an example for the cabinet folder.

- JBoss — *jboss-eap-7.4\bin\Newgen\NGConfig\ngdbini\Custom*
- WebLogic — *WEBLOGIC_DOMAIN\bin\Newgen\NGConfig\ngdbini\Custom*
- WebSphere — *WEBSPHERE_PROFILE\bin\Newgen\NGConfig\ngdbini\Custom*

- Also, ensure that both files (eworkstyle.ini and uploadmime.conf) are synchronized with their respective servers. Whenever you update any of the above mentioned files on JBoss, WebLogic, or WebSphere, ensure to mirror these changes in the corresponding paths of OmniDocs Mobile Server.

8. Run Folder service using the below command:

```
Java –jar folder -service.war
```

9. Run Document and Omniprocess services WAR file using CMD:

```
Java –jar document-service.war
Java –jar omniprocess -service.war
```

After configuring the WAR files, start them individually to activate the changes. Alternatively, update the WAR file paths in the *mwars.bat* file, save it, and then run *mwars.bat* to start all WAR files simultaneously.

Registering OAuth

Provide the following details as required for registering the app:

- od-appRegistrationCabinetName=odxxxxxxxcabinetmarchxxx
- od-appRegistrationUserName=supervisor
- od-appRegistrationUserPassword=systxxxx#
- od-appName=ghk124686432454556
- od-expiryTimeInMins=30
- od-versionForAppRegistration=11(mandatory)

To Register the OAuth, perform the following steps:

1. Place the cabinet name in the *od-appRegistrationCabinetName* tag inside *application.properties* of *authenticate-service.war*.
2. Use an S-type username that is not part of the group Second Factor Immune in *od-appRegistrationUserName* inside *application.properties* of *authenticate-service.war*.
3. Provide the password for the same user in the *od-appRegistrationUserPassword* tag inside *application.properties* of *authenticate-service.war*.
4. Enter the app name in the *od-appName* tag inside *application.properties* of *authenticate-service.war*.
5. Specify the expiry time in minutes in the *od-expiryTimeInMins* tag inside *application.properties* of *authenticate-service.war*.
6. Specify the OmniDocs version in the *od-versionForAppRegistration* tag inside *application.properties* of *authenticate-service.war*.

   Till OmniDocs 11.0 SP2, mention the version as 11.0, and for OmniDocs 11.3, mention 11SP3 as the version.

7. Start the server with the command *java –jar authenticate-service.war*. If the app registers successfully, the appId is generated in the *ServerKeyManagement* table corresponding to the cabinet in the primary database.

To register the app on multiple cabinets, provide the above details as comma-separated values. The *ServerKeyManagement* table auto-generates for RDBMS

vendors Postgres and MSSQL. For Oracle, run the following script if it is the primary database:

```
CREATE SEQUENCE app_sequence
  START WITH 1
  INCREMENT BY 1
  NOCACHE
  NOCYCLE;
CREATE TABLE ServerKeyManagement (
    Id NUMBER PRIMARY KEY,
    MobileServerPublicKey VARCHAR2(1000),
    MobileServerPrivateKey VARCHAR2(4000),
    ODServerPublicKey VARCHAR2(1000),
    AppId VARCHAR2(255),
    AppName VARCHAR2(255),
    ExpiryTime VARCHAR2(255),
    SecretKey VARCHAR2(255),
    UserIndex VARCHAR2(255),
    ClientSideAlgorithm VARCHAR2(255),
    ServerSideAlgorithm VARCHAR2(255),
    CabinetName VARCHAR2(255)
);
```

Configuring ports

The following table consists of the default ports for our services.

Service	Default Port
api-gateway.war	8765
authenticate-service.war	8091
folder-service.war	8092
document-service.war	8093
omniprocess-service.war	8094

If clients want to update these default ports, then the respective deployment team needs to update the *server.port* property in *application.properties* or *application.yml*, whichever is present in each WAR file. If both are present, they need to be updated in both files.
For example: server.port = 8091

Also, if any port of any WAR file is updated, then the *application.yml* file inside the *api-gateway.war* needs to be updated. Along with the above change, we need to change the value of the port for the URI tag with the id same as the microservice name.
For example, if we update the port of the *folder-service.war*, then we need to change the port in the URI tag where the id is folder-service in the *application.yml* file inside *api-gateway.war*

Feint URLs

Need to update the feint URL corresponding to the port change inside the *application.properties* of each WAR file.

For example, if you change the *authenticate-service.war* file port, then inside *folder-service.war*, *document-service.war*, and *omniprocess-service.war*, UserFeint_URL needs to changed.

UserFeint_URL=http://localhost:8091/authenticate

FolderFeint URLs

If you update the *folder-service.war*, then remember to update FolderFeint_URL in *document-service.war* and *omniprocess-service.war*.

FolderFeint_URL=http://localhost:8093/folder

DocumentFeint URLs

If you update the *document-service.war*, then remember to update DocumentFeint_URL in *folder-service.war*, *document-service.war*, and *omniprocess-service.war*.

DocumentFeint_URL=http://localhost:8092/document

Configuring Single Sign-On

Configure Single Sign-On (SSO) for enhanced user authentication and seamless access management.

To enable SSO, change the below mentioned property in the *application.properties* file inside *authenticate-service.war* file:

Path — *authenticate-service\target\authenticate-service.war\WEB-INF\classes*

od-ssoEnabled=Y

By default, its value is *N*

Enabling the shared file feature

Configure the below properties in the *application.properties* file inside the *document-service.war* file:

Path — *document-service\target\document-service.war\WEB-INF\classes*

```
od-mailPort=587
od-mailHost=smtp.office365.com
od-mailUserName=vixxxxxxxh@nxxxxxt.com
od-mailUserpassword=lcrfnrxxxxxxxxnp
od-sslEnabled=false
od-tlsEnabled=true
```

For creating the value of od-mailUserpassword, refer to manage app passwords for two-step verification on the Microsoft support page.

MultiCabinet support

Run the below SQL statement in the database with which the application is connected according to its Relational Database Management System (RDBMS) vendor.

Execute the below dynamic script on the primary database, which is the main database connected to the application.

For MS SQL

```
CREATE TABLE DynamicDatasourceDetails(
  Id int PRIMARY KEY NOT NULL,
  CabinetName varchar(255) NOT NULL,
  DriverURL varchar(max) NULL,
  DatabaseType varchar(255) NULL,
  Username varchar(255) NULL,
  Password varchar(255) NULL
)
```

For PostgreSQL

```
CREATE TABLE DynamicDatasourceDetails(
  Id int PRIMARY KEY NOT NULL,
  CabinetName varchar(255) NOT NULL,
  DriverURL text NULL,
  DatabaseType varchar(255) NULL,
  Username varchar(255) NULL,
  Password varchar(255) NULL
)
```

For Oracle

```
CREATE TABLE DynamicDatasourceDetails(
  Id number(10) PRIMARY KEY NOT NULL,
  CabinetName varchar2(225) NOT NULL,
  DriverURL varchar2(225) NULL,
  DatabaseType varchar2(225) NULL,
  Username varchar2(225) NULL,
  Password varchar2(225) NULL
)
```

Add the below key-value pair in the *application.properties* file of the *authenticate-service.war* file.

Path — *authenticate-service\target\authenticate-service.war\WEB-INF\classes*

- dynamic-datasource.cabinateName=
- dynamic-datasource.username=
- dynamic-datasource.password=
- dynamic-datasource.url=
- dynamic-datasource.type=

For each of the keys above, provide the required cabinet names separated by commas, and provide corresponding username and password details in the same order.

The below script is intended for secondary databases created by you. Ensure to run the below scripts on the appropriate databases based on your deployment architecture.

Run the following MS SQL, PostgreSQL, or Oracle commands in other Cabinet databases to create tables in advance.

Execute the following commands in sequence: first, create the ODMobileUsers table, then create the ODMobileDevice table.

For MS SQL

- For mobile user

```
CREATE TABLE ODMobileUsers(
 UserId int PRIMARY KEY,
 AuthenticationToken varchar(255) NULL,
 LatestLoggedInUserDate varchar(255) NULL,
 LoginUserIndex varchar(255) NULL,
 UserName varchar(255) NULL,
 FolderIndex int
)
```

- For mobile device

```
CREATE TABLE ODMobileDevice(
  DeviceId int PRIMARY KEY,
  Authenticated bit NULL,
```

```
  DeviceName varchar(255) NULL,
  DeviceOS varchar(255) NULL,
  DeviceSaveDate varchar(255) NULL,
  Manufacturer varchar(255) NULL,
  PrivateKey varchar(max) NULL,
  PublicKey varchar(max) NULL,
  ThirdPartyToken varchar(255) NULL,
  Udid varchar(255) UNIQUE NOT NULL,
  UserId int NULL
)
ALTER TABLE ODMobileDevice ADD CONSTRAINT FKODMobileDevice_UserId FOREIGN
KEY(UserId)

REFERENCES ODMobileUsers (UserId)
```

For PostgreSQL

- For mobile user

```
CREATE TABLE ODMobileUsers(
 UserId int PRIMARY KEY,
 AuthenticationToken varchar(255) NULL,
 LatestLoggedInUserDate varchar(255) NULL,
 LoginUserIndex varchar(255) NULL,
 UserName varchar(255) NULL,
 FolderIndex int
)
```

- For mobile device

```
CREATE TABLE ODMobileDevice (

DeviceId int PRIMARY KEY,
Authenticated bit NULL,
DeviceName varchar(255) NULL,
DeviceOS varchar(255) NULL,
DeviceSaveDate varchar(255) NULL,
Manufacturer varchar(255) NULL,
PrivateKey text NULL,
PublicKey text NULL,
ThirdPartyToken varchar(255) NULL,
Udid varchar(255) UNIQUE NOT NULL,
UserId int NULL
```

```
)
ALTER TABLE
  ODMobileDevice ADD CONSTRAINT FKODMobileDevice_UserId FOREIGN KEY(UserId)

REFERENCES ODMobileUsers (UserId)
```

For Oracle

- For mobile user

```
CREATE TABLE ODMobileUsers(
 UserId number(10) PRIMARY KEY,
 AuthenticationToken varchar2(255) NULL,
 LatestLoggedInUserDate varchar2(255) NULL,
 LoginUserIndex varchar2(255) NULL,
 UserName varchar2(255) NULL,
 FolderIndex number(10)
)
```

- For mobile device

```
CREATE TABLE ODMobileDevice(
  DeviceId number(10) PRIMARY KEY,
  Authenticated char(1) NULL,
  DeviceName varchar2(255) NULL,
  DeviceOS varchar2(255) NULL,
  DeviceSaveDate varchar2(255) NULL,
  Manufacturer varchar2(255) NULL,
  PrivateKey varchar2(4000) NULL,
  PublicKey varchar2(4000) NULL,
  ThirdPartyToken varchar2(255) NULL,
  Udid varchar2(255) UNIQUE NOT NULL,
  UserId number(10) NULL
)
ALTER TABLE
  ODMobileDevice ADD CONSTRAINT FKODMobileDevice_UserId FOREIGN KEY(UserId)

REFERENCES ODMobileUsers (UserId)
```