

Plato Infrastructure Services Installation Guide
Oracle FLEXCUBE Onboarding
Release 14.4.0.3.0
Part Number F39511-01
February 2021



Table of Contents

1. Preface	1-1
1.1 Introduction	1-1
1.2 Audience	1-1
1.3 Documentation Accessibility	1-1
1.4 Organization	1-1
2. Database Setup	2-1
2.1 Introduction	2-1
2.2 Prerequisite	2-1
3. Domain and Cluster Configuration	3-1
3.1 Plato Infrastructure Domain Configuration	3-1
3.1.1 Prerequisites	3-1
3.1.2 Domain Creation and Configuration	3-1
4. Data Sources Creation	4-1
4.1 Prerequisite	4-1
4.2 Data sources List	4-1
5. Security Configuration and Tools Installation	5-1
5.1 Prerequisite	5-1
5.2 Plato Security JWT	5-1
5.3 Plato Security Configuration (Online Web Application Authentication)	5-1
5.4 User Store	5-2
6. Deployments	6-1
6.1 Prerequisite	6-1
6.2 Deployment Order	6-1
7. Plato Infrastructure Software Deployment	7-1
7.1 Zookeeper Cluster Setup	7-1
7.1.1 Prerequisite	7-1
7.1.2 Installation	7-1
7.2 Kafka Cluster Setup	7-2
7.2.1 Prerequisite	7-2
7.2.2 Installation	7-2
7.3 Tesseract Installation	7-4
7.3.1 Prerequisite	7-4
7.3.2 Installation	7-5
7.4 Conductor Installation	7-9
7.4.1 Prerequisite	7-9
7.4.2 Installation	7-9
8. Plato Deployments	8-1
8.1 Prerequisite	8-1
8.2 Plato Applications Deployment Order	8-1
8.3 Steps to Deploy as Application	8-2
8.4 SSL Configuration	8-2
9. Restarts and Refresh	9-1
9.1 Restarting Servers	9-1
10. Logging Area	10-1
10.1 Introduction	10-1
10.2 Logging Area	10-1

1. Preface

1.1 Introduction

This guide helps you to install the Plato infrastructure services on designated environment. It is assumed that all the prior setup is already done related with WebLogic installation, WebLogic managed server creation and Oracle DB installation.

It is recommended to use dedicated managed server for each of the Plato infrastructure services.

Note: For the exact version to be installed, refer to **Software Prerequisites** section in **License Guide**.

1.2 Audience

This document is intended for WebLogic admin or ops-web team who are responsible for installing the OFSS banking products.

1.3 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>.

1.4 Organization

This installation user guide would allow you to install following services in same order:

- WebLogic system environment settings
- Plato Config Service
- Plato Discovery Service
- Plato API Gateway Service
- Plato UI Config Service
- Plato O (Conductor)
- Plato Orch Service
- Plato Feed Services
- Plato Batch Server
- Plato Alerts Management Services
- Security configuration and tool installation

2. Database Setup

2.1 Introduction

In this section you are going to setup database related configuration for PLATO Installation. Before you proceed ensure pre-installation setup is done.

2.2 Prerequisite

Before you proceed with the document, ensure Schemas are being created. It is recommended to have different schema for **Plato** and **Plato Security**. To configure Plato security refer Security Configuration chapter. Make sure that the schema user has the below rights:

	OPERATION					
DB OBJECT	CREATE	ALTER	DROP	INSERT	UPDATE	DELETE
TABLE	Y	Y	N	Y	Y	Y
VIEW	NA	NA	NA	NA	NA	NA
SEQUENCE	Y	Y	Y	NA	NA	NA
PACKAGE	NA	NA	NA	NA	NA	NA
PACKAGE BODY	NA	NA	NA	NA	NA	NA
INDEX	Y	Y	Y	NA	NA	NA
SYNONYM	NA	NA	NA	NA	NA	NA
FUNCTION	NA	NA	NA	NA	NA	NA
TRIGGER	NA	NA	NA	NA	NA	NA
TYPE	NA	NA	NA	NA	NA	NA

To know the server port number, refer to **How to check port number** section in ANNEXURE-1.

Ensure to configure Placeholder parameters in Weblogic server for Plato Config service, setDomain.env. To know more, refer to **Place Holder update for Plato-Config-Services** section in ANNEXURE-1.

3. Domain and Cluster Configuration

3.1 Plato Infrastructure Domain Configuration

3.1.1 Prerequisites

- Machine should have Java JDK to be installed.
- Oracle Fusion Middleware has to be installed on the machine.

Note: For the exact version to be installed, refer to **Software Prerequisites** section in **License Guide**.

3.1.2 Domain Creation and Configuration

It is recommended to have different managed server in one domain for each application. For Creating Domain and Configuration, refer to **How to create and Cluster Configuration** in ANNEXURE-1.

4. Data Sources Creation

4.1 Prerequisite

Before you proceed with Data source creation, make sure Domain and cluster configuration steps completed.

4.2 Data sources List

The table below lists the data sources to be created on each managed server prior to deployment of applications onto managed servers.

Data source Name	Data source JNDI	Targets
PLATO	jdbc/PLATO	Config Server, API Gateway Server, Plato Feed Server, Plato-Alerts-Management-Server, Plato-Batch-Server, Appshell Server
PLATOSEC	jdbc/PLATO_SECURITY	Config Server, API Gateway Server
PLATO_UI	jdbc/PLATO_UI_CONFIG	Plato UI Config Server, Appshell Server
CONDUCTOR	jdbc/PLATO-O	Plato-O, Plato Orch Server
PLATOFEED	jdbc/PLATOFEED	Plato-Feed-Server
PLATOALERTS	jdbc/PLATOALERTS	Plato-Alerts-Management-Server
PLATOBATCH	jdbc/PLATOBATCH	Plato-Batch-server

For creating data source, refer to **How to create Data sources section** in ANNEXURE-1.

5. Security Configuration and Tools Installation

5.1 Prerequisite

Before you proceed, do the following steps:

- In case you are planning to use LDAP for web application authentication with Weblogic as provider for LDAP. Please first go through the steps of Embedded Weblogic setup steps in ANNEXURE 1.
- In case you are planning to use OAuth without OAM(i.e. Spring OAuth), do the following **change in Weblogic configuration**:
In the config.xml file of the concerned domain in Weblogic add the following script at the end of **security-configuration** tag (Just before the line **</security-configuration>**)

```
<enforce-valid-basic-auth-credentials>false</enforce-valid-  
basic-auth-credentials>
```

To use the Standard LDAP directory authentication for Online Web Application authentication, make sure LDAP server details is provided to you:

Like LDAP_URL, USER_STORE, LDAP_SERVER_CREDENTIAL_SALT, LDAP_SERVER_USER, LDAP_SERVER_BASE, LDAP_SERVER_CREDENTIAL, LDAP_USER_SEARCH_BASE, LDAP_USER_PREFIX, CORS_ALLOWED_ORIGINS, LDAP_SERVER_CREDENTIAL_SALT etc.

5.2 Plato Security JWT

Plato security module enables securing API micro services with JWT (JSON Web Tokens). JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties. JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JSON object that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed.

5.3 Plato Security Configuration (Online Web Application Authentication)

Plato recommend to create new schema for security to keep the security related database objects at one place. If the environment is configured for multi-tenant, we require a security schema per tenant.

All the Plato security configurations are maintained at SECURITY_CONFIG table
Steps to configure in the table:

1. Change in case of **LDAP directory authentication** the below KEY with provided LDAP details:

KEY	VALUE
LDAP_SERVER_CREDENTIAL_SALT	Enter LDAP server Credential salt e.g. 0.9482628451234567
CORS_ALLOWED_ORIGINS	valid host names (comma delimited)
LDAP_URL	Enter LDAP Server URL Example: ldap://wxy00abc:9001
LDAP_SERVER_USER	Enter LDAP Server USERID Example: uid=admin
LDAP_SERVER_BASE	Enter LDAP server BASE Example: dc=oracle,dc=com
LDAP_SERVER_CREDENTIAL	Enter LDAP server encrypted password using provided jwt algorithm Example: m0o/F3UvlwvBSv5C/TSckA== (use plato encryption utility to generate encrypted password)
LDAP_USER_SEARCH_BASE	Enter LDAP User search Base Example: ou=people
LDAP_USER_PREFIX	Enter LDAP User Prefix Example: uid

2. Change in case of **SSO Agent** the below KEY with provided LDAP details:

KEY	VALUE
IS_SSO_CONFIGURED	True
CORS_ALLOWED_ORIGINS	valid host names(comma delimited)

5.4 User Store

Plato supports following user stores for authentication Users Maintained at table.

1. Plato security can authenticate the users maintained at table (APP_USER) in the security schema. However, we do not recommend to use this option.

6. Deployments

6.1 Prerequisite

Before you proceed with below, make sure previous steps are completed.

6.2 Deployment Order



7. Plato Infrastructure Software Deployment

Once everything is deployed, the managed servers. For each application, call path “/refresh” for refreshing the configuration properties.

7.1 Zookeeper Cluster Setup

To restart the server, refer to **How to restart** section in ANNEXURE-1.

7.1.1 Prerequisite

JDK should be installed in all node machines.

Download zookeeper and extract the binary in all node machines. Zookeeper can be found at <Unzip the file>/THIRD_PARTY_SOFTWARES/ZOOKEEPER/ARCHIVE

7.1.2 Installation

- Untar/unzip the zookeeper binary and move them into a folder which will be the zookeeper home directory.
- Create two directories named logs and data inside the zookeeper home directory folder in all the nodes with appropriate permission. If logs folder is already present, clear it.
- Inside the <zookeeper home directory>/data folder create a myid file. The myid file consists of a single line containing only the text of that machine's id. So myid of server 1 would contain the text "1" and nothing else. The id must be unique within the ensemble and should have a value between 1 and 255.
- Create a configuration file named zoo.cfg at <zookeeper **home** directory>/zookeeper_3.5.6/config
Add the following set of properties and values to that file:

```
dataDir= <zookeeper home directory>/data
tickTime=2000
clientPort= Zookeeper client Port value (2181)
initLimit=10
syncLimit=5

server.1=<hostname> :< peer port> :< leader port>
#1 is the id that we put in myid file.

server.2= <hostname> :< peer port> :< leader port>
#2 is the id that we will put in myid file of second
node.

server.3=<hostname> :< peer port> :< leader port>
#3 is the id that we will put in myid file of third
node.
```

NOTE: Any odd number of zookeeper servers can be configured under the cluster.

- Start the zookeeper on each node machine

Navigate to <zookeeper home directory>/zookeeper_3.5.6 and execute the below command

```
bin/zkServer.sh start
```

- To see who is the leader and followers in the cluster, run the below command on each node

```
echo stat | nc localhost 2181
```

- To check the zoo cluster functioning i.e dynamic leader election, kill the zookeeper process on the leader node and check again with the following commands on the remaining live zookeeper node.

```
echo stat | nc localhost 2181
```

7.2 Kafka Cluster Setup

7.2.1 Prerequisite

JDK should be installed in all node machines.

Download Kafka and extract the binary in all node machines. Kafka can be found at <Unzip the file>/THIRD_PARTY_SOFTWARES/KAFKA/ARCHIVE

7.2.2 Installation

- Untar/unzip the kafka binary and move them into a folder which will be the kafka home directory. If logs folder is already present, clear it.
- Create two directories named logs and data inside the kafka home directory folder in all the nodes with appropriate permission. If logs folder is already present, clear it.
- Edit the below lines in the
<kafka home directory>/kafka_2.12-2.3.1/config/server.properties.

```
broker.id= (Unique Integer which identifies the kafka broker in the
cluster.)
listeners=PLAINTEXT://<hostname>:<Kafka broker listen port(9092)>
log.dirs=<kafka home directory>/logs
log.retention.hours= <The number of hours to keep a log file before
deleting it (in hours),tertiary to log.retention.ms property>
log.retention.bytes= <The maximum size of the log before deleting it>
log.segment.bytes= <The maximum size of a single log file>
log.retention.check.interval.ms= <The frequency in milliseconds that
the log cleaner checks whether any log is eligible for deletion>
zookeeper.connect=<zookeeper_hostname_1>:<zookeeper_client_port>,<zook
eeper_hostname_2>:<zookeeper_client_port>,<zookeeper_hostname_3>:<zook
eeper_client_port>, ...
```

- To start the Kafka, navigate to <kafka home directory>/kafka_2.12-2.3.1/ folder and run the below command on each node.

```
export JMX_PORT=[PORT VALUE]

nohup bin/kafka-server-start.sh config/server.properties &
```

The Default value of JMX Port is 9999.

Tail the log for server status.

- To create topic, navigate to <kafka home directory>/kafka_2.12-2.3.1/ folder and run the below command:

```
/bin/kafka-topics.sh --create -zookeeper<hostname>:<client
port> --replication-factor 3 --partitions 3 --topic <topic
name>
```

- To list the available topic on kafka server, navigate to <kafka home directory>/kafka_2.12-2.3.1/ folder and run the below command:

```
./bin/kafka-topics.sh --list -zookeeper <hostname>:<client
port>
```

- To describe the topic, navigate to <kafka home directory>/kafka_2.12-2.3.1/ folder and run the below command:

```
./bin/kafka-topics.sh --describe --topic <topic name> --
zookeeper <hostname>:<client port>
```

- To start a producer, navigate to <kafka home directory>/kafka_2.12-2.3.1/ folder and run the below command:

```
export JMX_PORT=[PORT VALUE]//Different Value from the server
JMX port

./bin/kafka-console-producer.sh --broker-list
<hostname>:<port>, <hostname>:<port>, --topic <topic name>
```

By default, port is taken as 9092 for the producer.

- To start a consumer console for viewing the received messages sent by the producer, use the following command:

```
export JMX_PORT=[PORT VALUE]//Different Value from the server
JMX port

./bin/kafka-console-consumer.sh --bootstrap-server
<hostname>:<port>,<hostname>:<port>, --topic <topic_name> --
from-beginning
```

7.3 Tesseract Installation

7.3.1 Prerequisite

7.3.1.1 Build Tools

Ensure that the following build tools are available:

- GNU Autotools—autoconf, automake, libtool
- CMake (Optional, we will use this only if autoconf fails while building leptonica)

Both should be available inside Oracle yum.

7.3.1.2 Dependent Libraries

These libraries should be present in the server. By default, these libraries are available in Oracle Linux. If these libraries are not present, please install it through yum with the following command:

```
sudo yum install <LIBRARY_NAME>
```

Following are the library names:

1. libjpeg
2. libtiff
3. zlib
4. libjpeg-turbo
5. libwebp
6. libpng-devel
7. libtiff-devel
8. libwebp-devel

NOTE: If you are using any distribution other than Oracle Linux, please install the libraries from official Oracle repo or any other repo available for that distribution.

7.3.1.3 Installation Files

Download installation files required to install and set up tesseract. Files are available at the following location:

<Unzip the file>/THIRD_PARTY_SOFTWARES/Tesseract

Please find below the list of files present in the directory:

1. leptonica-1.76.0.tar.gz
2. tesseract-4.1.0.tar.gz
3. eng.traineddata
4. osd.traineddata

7.3.2 Installation

7.3.2.1 Leptonica Installation:

Tesseract uses leptonica internally for image processing. Leptonica can be build and installed either by autoconf or by CMake.

In this document, we will cover installation using both autoconf and CMake.

NOTE: If you already have all access to all installation directory then sudo is not required.

>sudo LINUX_COMMAND (In case the user does not have file access permissions)

>LINUX_COMMAND (In case the user has all access. Example: DBA user, Root user)

In this document, we will execute all the commands with sudo. You can omit it based upon your user's permission details.

7.3.2.1.1 Installation through Autoconf

- Copy the downloaded leptonica tarball (leptonica-1.76.0.tar.gz) in server (in the installation directory. Ex: /scratch)
- Execute below commands sequentially to install leptonica through autoconf

NOTE: In line 4, we used **sudo make -j4**. Here 4 is the number of CPU core. Generally, you can use **sudo make -jn** where n is the number of core. It will make the build process much faster.

In the document, we will use 4 as core number to build the software.

If the processor does not have multiple cores you can use normal make command **sudo make**.

```
sudo tar xvf leptonica-1.76.0.tar.gz
cd leptonica-1.76.0
sudo ./configure
sudo make -j4
sudo make install
```

If the installation is successful, then go to [7.3.2.2](#). Else, go to [7.3.2.1.2](#).

7.3.2.1.2 Installation through CMake

In case autoconf fails to generate the configure file or there is any other error, then proceed with the below steps, to build through CMake.

```
sudo tar xvf leptonica-1.76.0.tar.gz
cd leptonica-1.76.0
sudo mkdir build
cd build
sudo cmake ..
sudo make -j4
sudo make install
```

7.3.2.2 Leptonica Configuration

- Leptonica path should be configured such that tesseract can find the leptonica installation.
- Please add the leptonica installation directory (Ex: /usr/local/lib ,/usr/lib, /usr/lib64 etc) in library path.
- leptonica header path (Ex: /usr/local/include/leptonica) should be configured.
- Pkgconfig path also need to be set up.

Execute the below mentioned commands to set the path:

```
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig/  
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/lib64/pkgconfig/  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib  
export LIBLEPT_HEADERSDIR=/usr/local/include/leptonica
```

NOTE: Sometimes, tesseract will still be unable to find lept.pc file.

It will give configuration errors (ex: Leptonica 1.74 or higher is required). In that case locate the lept.pc file (usually present at /usr/local/lib/pkgconfig/) with the command **locate lept.pc** and copy the same in /usr/lib64 directory.

```
sudo cp /usr/local/lib/pkgconfig/lept.pc /usr/lib64/pkgconfig/
```

- Similarly, some services might not be able to get libleptonica shared object files (.so files, ex: liblept.so, libleptonica.so etc.)
- .so files are usually present in the server at /usr/local/lib. You can type **whereis libleptonica** or **locate libleptonica** to find the path. Then copy the .so files in /usr/lib64 path.

```
cd /usr/local/lib  
sudo cp -a *liblept* /usr/lib64
```

7.3.2.3 Tesseract Installation

- Copy the tesseract tarball tesseract-4.1.0.tar.gz in server (in the installation directory. Ex: /scratch)
- Copy the tesseract trained files eng.traineddata, osd.traineddata in the server
- Execute below commands sequentially to build and install tesseract

NOTE: /usr/bin is the directory where tesseract binary will be present if you pass prefix=/usr in configure. You can provide the path based upon where you want to install.

```
sudo tar xvf tesseract-4.1.0.tar.gz
cd tesseract-4.1.0
sudo ./autogen.sh
sudo ./configure --prefix=/usr
sudo make -j4
sudo make install
```

- Copy the traineddata files in tessdata directory. If you use prefix=/usr, tessdata directory will be present at /usr/share. If you use prefix=/usr/local, tessdata directory will be present at /usr/local/share.

```
sudo cp osd.traineddata /usr/share/tessdata
sudo cp eng.traineddata /usr/share/tessdata
```


7.3.2.4 Tesseract Configuration

- Set the tesseract library path by executing the below commands:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

- Sometimes services are unable to find libtesseract shared object files (.so files) present in system (Usually at /usr/lib). In that case copy the libtesseract files in /usr/lib64

```
cd /usr/lib
sudo cp -a *libtesseract* /usr/lib64
```

- Some of the programs search for the tessdata directory in a different path (/usr/share/tesseract/4/tessdata). Copy the existing tessdata directory (which will be present in either /usr/share or /usr/local/share based on your installation) in that path.

```
cd /usr/share
sudo mkdir tesseract (execute if tesseract
directory is not present)
cd tesseract
sudo mkdir 4
cd /usr/share
sudo cp -R tessdata /usr/share/tesseract/4
```

- Set tessdata prefix by running following command.

```
export TESSDATA_PREFIX=/usr/share/tesseract/4/tessdata
```

- Tesseract is now installed.
- You can verify the version with following command. It will give the tesseract version (4.1.0), leptonica version (1.76.0) along with other default libraries (libjpeg, libjpeg-turbo, libpng, libtiff, zlib).

```
tesseract --version
```

7.4 Conductor Installation

7.4.1 Prerequisite

Ensure that the datasource jdbc/PLATO-O is created.

The maximum capacity attribute of the datasource connection pool should be greater than 100.

Make sure that the Domain and cluster configuration steps completed.


NOTE: The conductor-server.war file needs to be deployed in a separate managed server because of its load and size.

7.4.2 Installation

Perform the following steps:

1. Required properties should be set in the config.properties file found in {**unzip the file**} THIRD_PARTY_SOFTWARES\CONDUCTOR_SERVER\CONFIG. Refer to the below table to find the description of properties in the config.properties. This file should be placed at <<CONFIG.PROPERTIES LOCATION >>.
2. An additional environment variable is required for setting up the conductor. Include the below mentioned -Dparam along with the existing environment variables.
-Dconductor.properties = << CONFIG.PROPERTIES LOCATION >>/config.properties
3. Deploy the conductor-server.war file in the weblogic. To deploy application, refer to **How to deploy section ANNEXURE-1**.

Property Name	Property Description
flyway.enabled	Set this to true to enable flyway and false to disable flyway.
flyway.setbaselineOnMigrate	Set this to true to enable flyway baselineOnMigrate and false to disable.

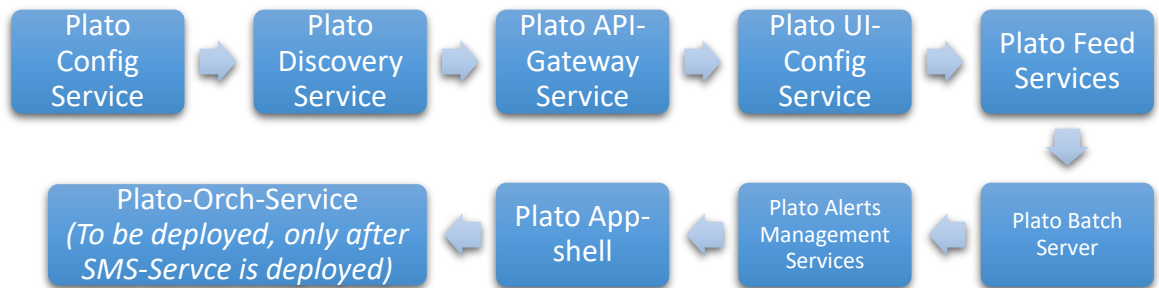
Property Name	Property Description
eureka.registration.enabled 	Should be set to true to enable discovery registration.
eureka.hostName	plato-o
eureka.instanceId	plato-o:<port-number>
eureka.serviceUrl.default	Discovery service URL (http://<hostname>:<port>/plato-discovery-service/eureka)
eureka.registerWithEureka	true
eureka.name	plato-o
eureka.vipAddress	plato-o
eureka.port	Port Number on which the conductor server war file is deployed.

8. Plato Deployments

8.1 Prerequisite

Before you proceed with below, make sure that the previous steps are completed.
Below table give details of the deployments required on each Server for the Plato application to run.

8.2 Plato Applications Deployment Order



Installation Summary for Plato Services:

Application	Archive name	OSDC path	Targets
Plato-config-service	plato-config-service-5.1.0.war	{unzip the file}PLATO\plato-config-service\	Config Server
Plato-discovery-service	plato-discovery-service-5.1.0.war	{unzip the file}PLATO\plato-discovery-service\	Discovery Server
Plato-api-gateway	plato-api-gateway-5.1.0.war	{unzip the file}PLATO\plato-api-gateway\	Api Gateway
Plato-ui-config-service	Plato-ui-config-service-5.1.0.war	{unzip the file}PLATO\plato-ui-config-service\	Plato UI Config
Plato-Orch-Service (To be deployed after sms-service is deployed)	Plato-Orch-Service-5.1.0.war	{unzip the file}PLATO\plato-orch-service\	Plato-Orch-Service
Plato-Feed-Services	Plato-Feed-Services-5.1.0.war	{unzip the file}PLATO\plato-feed-services\	Plato-Feed-Services
Plato-Batch-Server	Plato-Batch-Server-5.1.0.war	{unzip the file}PLATO\plato-batch-	Plato-Batch-Server

Application	Archive name	OSDC path	Targets
		server\	
Plato-Alerts-Management-Services	Plato-Alerts-Management-Services-5.1.0.war	{ unzip the file }PLATO\plato-alerts-management-services\	Plato-Alerts-Management-Server
Appshell	app-shell.war	{ unzip the file }UI\app-shell.war	Appshell Server

NOTE: Eventhub based applications should not to be deployed in admin server

8.3 Steps to Deploy as Application

To deploy application, refer to **How to deploy section** in ANNEXURE-1.

8.4 SSL Configuration

We recommend only https-based connections. Below, are the recommendations:

Appshell needs to be secured with SSL.

1. Api-Gateway needs to be secured with SSL.

Appshell to Api-gateway communication should happen over SSL. The api-gateway url mentioned as -D parameter for appshell should be ssl enabled(i.e. https-based).

9. Restarts and Refresh

Once everything is deployed, the managed servers. And for each application call path “/refresh” for refreshing the configuration properties.

9.1 Restarting Servers

To restart the server, refer to. **How to restart** section in ANNEXURE-1.

10. Logging Area

10.1 Introduction

This section describe about the logs area where after deployment of Plato Applications in WebLogic server.

10.2 Logging Area

Plato Application writes logs in the below area of the server:

<WEBLOGIC_DOMAIN_CONFIG_AREA>/ logs/plato-api-gateway.log

Let's assume a domain has been created **platoinfra_domain** in the following area of the server
"/scratch/oracle/middleware/user_projects/domains/platoinfra_domain".

Logging area for Plato
=<URL>



Plato Infrastructure Services Installation Guide

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Gurgaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax: +91 22 6718 3001

<https://www.oracle.com/industries/financial-services/index.html>

Copyright © 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.