

opentext™

OpenText™ Project and Portfolio Management (PPM)

Software version: 25.2

Installation and Upgrade Guide

Go to Help Center online

<https://admhelp.microfocus.com/ppm/>



Document release date: April 2025

Send Us Feedback



Let us know how we can improve your experience with the Installation and Upgrade Guide.

Send your email to: admdocteam@opentext.com

Legal Notices

© Copyright 2025 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Disclaimer

Certain versions of software accessible here may contain branding from Hewlett-Packard Company (now HP Inc.) and Hewlett Packard Enterprise Company. This software was acquired on September 1, 2017 by Micro Focus and is now offered by OpenText, a separately owned and operated company. Any reference to the HP and Hewlett Packard Enterprise/HPE marks is historical in nature, and the HP and Hewlett Packard Enterprise/HPE marks are the property of their respective owners.

Contents

Installation and upgrade	5
Install PPM with Oracle	6
Before you install	6
Install the Java Development Kit (JDK)	8
Overview	8
Install the JDK	8
(Optional) Configure JDK to use the unlimited strength Java Cryptography Jars	9
Create a system account for OpenText PPM	10
Create database schemas	11
Default Permissions for OpenText PPM Schemas	14
Other Permissions Needed or Not Needed for OpenText PPM Schemas	
Accounts	15
Verify port availability	17
Collect required information	19
Generate Autopass license	23
Create a shared folder for the server.conf File	23
Prepare a Shared Folder for server.conf on a Windows system	24
Prepare a Shared Folder for server.conf on a UNIX system	25
Install PPM on a Windows system	25
Install PPM on a UNIX system	31
Install PPM with special command	33
Configure the FTP server on Windows	35
Contact support for troubleshooting	36
Optional installations	37
Install Deployment Management extensions	43
Verify OpenText PPM installation	43
Install PPM with PostgreSQL	45
Supported and unsupported features	45
Before you install	47
Install PPM	48
Install PPM on the cloud	54
Best practices of installing PPM on AWS	54
Recommended architecture	54
Install PPM on AWS	55
Best practices of installing PPM on Azure	56

Recommended architecture	56
Install PPM on Azure	57
Docker install and upgrade	58
Before the installation and upgrade	59
Install	59
Upgrade	61
Start up	62
Upgrade PPM	64
Upgrade paths	64
Upgrade guide	64
Upgrade PPM - Oracle	65
Upgrade PPM with Oracle database	65
Best practices for upgrade	67
Troubleshooting	70
Upgrade PPM - PostgreSQL	70
Web service configuration update for upgrades to 23.4 and above	73

Installation and upgrade

This guide provides information about installing and upgrading OpenText™ PPM, which includes the following tasks:

- ["Install PPM with Oracle" on page 6](#)
- ["Install PPM with PostgreSQL " on page 45](#)
- ["Install PPM on the cloud" on page 54](#)
- ["Upgrade PPM" on page 64](#)

Install PPM with Oracle

This section provides the detailed steps used to install the database objects and data that the PPM Server uses (with an Oracle database). You can perform these steps on any computer (running UNIX or Windows) that has SQL*Net connected to the database on which the OpenText PPM database objects are to be installed.

- Before you install
- Install on a Windows System
- Install on UNIX Systems
- Install PPM with Command
- Optional installations
- Verify the installation

Before you install

This section provides information that will help you prepare for the PPM Center installation.

1. Check the *System Requirements and Compatibility Matrix* to make sure your system meets all of the minimum requirements.
2. Make sure that you have at least 300 MB temporary space and 0.5~1 GB swap space on your operating system.

Caution: Limits on physical memory for 32-bit Windows operating systems depend in part on whether the Physical Address Extension (PAE) is enabled. The PAE allows some 32-bit Windows systems (Windows Server 2008 Datacenter and Windows Server 2008 Enterprise) to use more than 4 GB of physical memory. (To enable PAE, use the /PAE switch in the `Boot.ini` file.)

The total available physical RAM on this system is limited to 4 GB. For detailed information about memory support and memory limitations on Windows operating systems, see Microsoft Support online.

3. Enable the Oracle Java Virtual Machine (Oracle JVM)

OpenText PPM uses Java Stored Procedures in Oracle. Java Stored Procedures enable you to call Java code from PL/SQL. To use this feature, you must enable the Oracle Java virtual machine (OracleJVM). For information about how to install and configure the Oracle Java virtual machine (JVM), see the *Oracle® Database Java Developer's Guide* for your Oracle software version.

Note: We strongly recommend that you automate memory so that the size of the JAVA pool (Oracle `JAVA_POOL_SIZE` parameter setting) is allocated automatically. To automate memory allocation for Oracle 11g databases, use Automatic Memory Management (AMM).

4. Make sure your organization has obtained the installation bundle.

5. Make sure you have generated Autopass license files for version of the product you purchased and plan to install. See ["Generate Autopass license" on page 23](#).

6. Install the JDK. See the "Install the Java Development Kit (JDK)" topic in this guide.

7. (Windows systems only) On each Windows server with which OpenText PPM is to interact, download and install Cygwin.

For information about this UNIX emulator and how to install it, go to the Cygwin web site. For PPM-supported Cygwin versions, see *System Requirements and Compatibility Matrix*.

8. Install Oracle client on the PPM Server.

9. Create a system account for PPM. See the Create a system account for OpenText PPM topic in this guide.

10. Set up the Oracle tablespaces required to create the schemas and database objects.

You can create the database schemas either before the installation or during the installation. See the *Create database schemas* section in this guide if you create schemas before the installation.

11. Set Oracle database parameters.

- Set Oracle database parameter `NLS_LENGTH_SEMANTICS` to `CHAR`.
- Set Oracle database parameter `NLS_CHARACTERSET` to `AL32UTF8`.

Note: For new PPM installations, make sure you always use `AL32UTF8`.

12. Verify the required ports are open through firewall and that other applications are not using them. See the "Verify port availability" section in this guide.
13. Collect the information required for installation. See [Collecting Required Information](#).

Install the Java Development Kit (JDK)

This topic provides details on how to install and configure the Java Development Kit (JDK).

Overview

Because the PPM Server is based on Java, the machine that hosts it must also host a Java Virtual Machine (JVM), which is part of the Java Development Kit (JDK). JDKs native to the operating systems supported by OpenText PPM are available from either Oracle or from the operating system vendor.

Note: You must install the complete JDK. The Java Runtime Environment (JRE) alone is not supported.

Install the JDK

This section provides detailed steps of installing the JDK.

To install the JDK:

1. Download the JDK for your operating system from Oracle or from your operating system vendor's Web site.
2. Install the JDK according to the instructions provided by the vendor.

Many operating systems require that you apply operating system-specific patches before you install the JDK. Make sure that you follow all instructions that the vendor provides.

Some vendors provide custom installation packages that you can install automatically using a command such as `pkgadd`. Other vendors provide a TAR file that you must extract.

Note: The directory path name must not contain spaces.

3. Verify that your user name has the Java executable in its path by logging on and running the following the command:

```
java -version
```

This returns the Java version. If an error message is displayed, modify the path environment variable, as required.

4. Verify that the `JAVA_HOME` environment variable is set correctly. If the path set for `JAVA_HOME` is not correct, set it to the correct value.

(Optional) Configure JDK to use the unlimited strength Java Cryptography Jars

OpenText PPM supports control over the encryption suites used by its SSL (TLS) sockets. This can be specified by the server configuration parameter `SSL_ENCRYPTION_SUITES`.

The value for this parameter should contain a comma-separated list of the encryption suites to be made available to PPM Centre. These should be specified using the standard SSL/TLS cipher suite names.

For example, to specify that OpenText PPM should only establish connections using the `TLS_DHE_RSA_WITH_AES_256_CBC_SHA` cipher suite:

```
com.kintana.core.server.SSL_ENCRYPTION_SUITES=TLS_DHE_RSA_WITH_AES_256_CBC_SHA
```

If using AES256 or similarly strong encryption, the JDK used by both OpenText PPM and the client must be configured to use the unlimited strength Java

cryptography jars, if this is permissible in your jurisdiction and under US export laws.

Note: The **SSL_ENCRYPTION_SUITES** parameter only impacts the encryption algorithm used for RMIS traffic. There is no impact on HTTPS (SSL) encryption, nor on how the passwords and sensitive data are encrypted in OpenText PPM.

To configure your JDK to use the unlimited strength Java cryptography jars:

1. Go to the Oracle Java Downloads web site.
2. Download the unlimited strength Java cryptography jars that match your JDK version.

For JDK 1.7.0, download Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 7 ([UnlimitedJCEPolicyJDK7.zip](#)).

3. Extract the downloaded zip package.
4. Copy the **local_policy.jar** and **US_export_policy.jar** files to the **<JDK_HOME>/jre/lib/security** directory on both your server side and client side to replace the existing files.

Note: If you enabled secure RMI and are using a high strength encryption suite, such as AES256, make sure to follow the steps above to install the unlimited jars on machines which will run workbench.

Create a system account for OpenText PPM

To install OpenText PPM and maintain the system after installation, you must create a system account. After you do, always log on to this account on the server machine to perform any PPM Server maintenance—for example, stopping and restarting the PPM Server. This helps to avoid file system permission issues, which can be difficult to track.

Configuring a System Account for OpenText PPM in Windows

In Windows, configure the user to be a member of the Administrators and Domain Users groups, at a minimum. Provide the user with full access to the installation directory for OpenText PPM and all of its subdirectories. Provide the Administrators screen group with at least read access to these directories.

Configuring a OpenText PPM User for OpenText PPM in UNIX

In UNIX, OpenText PPM does not require root access for installation. Do not install the server as the root user.

Configure your OpenText PPM user with the following:

- In the `.profile` file,
 - Set the `JAVA_HOME` environment variable.
 - Set the `ORACLE_HOME` environment variable.
- Set the `term` to `dumb` option.

Create database schemas

To create the empty database schemas (with tables to be populated during installation):

1. Set up the required data, index, and CLOB tablespaces for the OpenText PPM database schema.

Tip: Even though a fresh installation of OpenText PPM typically requires less than 1 GB of database space, its size could sharply increase with time, especially if you choose to store attachments in the database. For an accurate estimation of your DB space requirement on the long term, contact Software Support.

Use locally-managed SYSTEM tablespaces with automatic segment-space management.

Note: Locally-managed tablespaces eliminate extent fragmentation and provide better performance than dictionary-managed tablespaces.

2. Create each tablespace as shown in the following example for a data tablespace.

```
CREATE TABLESPACE <PPM_Data>
datafile <'<u00/oracle/oradata/G1120/ppm_data01.dbf'>
size <1024m>
AUTOEXTEND ON MAXSIZE <4096m>
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;
```

Note: Oracle has the default TEMP tablespace, which you can resize to improve performance.

The PPM Server requires two separate database schemas to store application data. A DBA can create these schemas before installation. Creating database schemas requires privileges that a DBA might not want to grant to a OpenText PPM administrator. Either create the database schemas before installation or make sure that a DBA is available during installation.

To create the database schemas and grant the permissions between them:

1. Unpack the OpenText PPM installation bundle.

The **<PPM_Extract>/ppm<version>** directory is created. The **<PPM_Extract>/ppm<version>/sys** and **<PPM_Extract>/ppm<version>/system** directories contain the scripts required to create the database schemas.

2. From the **<PPM_Extract>/ppm<version>/system** directory, run the **CreateKintanaUser.sql** script against the database into which you plan to install PPM. The script prompts for a user name and password, and the tablespaces that the OpenText PPM database schema are to use, as follows:

```
sh> sqlplus system/<Password>@<SID> \
@CreateKintanaUser.sql \
<PPM_Username> \
```

```

<Password> \
<Data_Tablespace> \
<Index_Tablespace> \
<CLOB_Tablespace>

```

3. To enable the OpenText PPM database user to create views and synonyms in the RML schema, connect to the database that contains the RML schema, and then run the following SQL statements:

```

grant create any synonym to &KNTA_USERNAME;
grant create any view to &KNTA_USERNAME;
grant drop any synonym to &KNTA_USERNAME;
grant drop any view to &KNTA_USERNAME;
grant comment any table to &KNTA_USERNAME;

```

4. Run the **CreateRMLUser.sql** script located in the **/system** directory.

The script prompts for a user name and password for the Reporting Meta Layer (RML) schema, tablespace information, and the OpenText PPM database schema user name. The script creates the RML schema and establishes the permissions between the RML and the OpenText PPM database schema.

Note: Because the RML schema contains only views (and no physical objects), it does not require a separate tablespace.

```

sh> sqlplus system/<Password>@<SID> \
@CreateRMLUser.sql \
<RML_Username> \
<RML_Password> \
<Data_Tablespace> \

```

5. As the SYS DBA user, run the **GrantSysPrivs.sql** script located in the **<PPM_Extract>/ppm<version>/sys** directory.

This script grants the privileges that the PPM Server requires.

If you created the schemas before installation, select **Please use existing schemas** when prompted during installation. Supply the same values as those

used in this procedure (that is, the values **<PPM_Username>** and **<RML_Username>**).

Default Permissions for OpenText PPM Schemas

By default, the OpenText PPM database schema and RML database schema accounts (PPM_USER and RML_USER) are granted Oracle database privileges from an overall OpenText PPM perspective, which works for all customer scenarios and environments. Some of these privileges are required, but some are not, and can be revoked without affecting the OpenText PPM system.

The OpenText PPM database schema account is granted Oracle CONNECT role privileges. If this presents a problem for your organization, you can have your DBA revoke the CONNECT role privilege for the OpenText PPM database schema account.

Although revoking the CONNECT role privilege does not affect the OpenText PPM system, the OpenText PPM schema does require the following grants:

- grant create session
- grant create database link
- grant create procedure
- grant create sequence
- grant create synonym
- grant create table
- grant create view
- grant create trigger
- grant create job
- grant execute on ctxsys.ctx_ddl

The RML database schema account is granted Oracle RESOURCE role privileges. Because the RML schema requires the RESOURCE role privilege, it cannot be revoked. You can, however, revoke the following privileges, which are also granted to the RML database schema account:

- CREATE CLUSTER
- CREATE INDEXTYPE
- CREATE OPERATOR

Other Permissions Needed or Not Needed for OpenText PPM Schemas Accounts

This section provides more information about other permissions needed or not needed for OpenText PPM database schema and RML database schema accounts. You can decide whether you want to have them revoked after reading this section.

PPM has not tested every access grant and its impact, because every access grant has a different impact on each customer, as every customer is using PPM differently and for different scenarios. If you are looking for specific access grant impact, you may need to test that out.

SELECT_CATALOG_ROLE
SELECT ANY DICTIONARY

If only SELECT_CATALOG_ROLE is enabled then it provides access to all SYS views only.

If only SELECT ANY DICTIONARY privilege is enabled then it provides access to SYS schema objects only.

If both SELECT ANY TABLE privilege and SELECT ANY DICTIONARY privilege are enabled then it allow access to all SYS and non-SYS objects.

OpenText PPM needs to access many SYS/DBA views and objects, thus needs SELECT_CATALOG_ROLE role and SELECT ANY DICTIONARY privilege.

To generate AWR/ADDM/ASH reports, OpenText PPM needs SELECT_CATALOG_ROLE role too.

OpenText PPM needs access to the following packages:	<ul style="list-style-type: none"> • DBMS_JAVA • DBMS_JAVA_TEST • DBMS_LOB • DBMS_SCHEDULER • DBMS_SQL
OpenText PPM does not need access to the following packages:	<ul style="list-style-type: none"> • UTL_FILE • UTL_HTTP • UTL_TCP
For other packages, read the information below to decide whether your OpenText PPM still needs access to them:	<ul style="list-style-type: none"> • DBMS_JAVA — OpenText PPM uses a stored procedure written in Java, to generate a hash that is used as REFERENCE_CODEs for various OpenText PPM entities. Generating this hash using pure PL/SQL is cumbersome and unreliable, if not impossible. So, OpenText PPM used Java for it and the application code needs access to this package. • DBMS_JAVA_TEST — Allows you to test Java Stored Procedures. OpenText PPM might not need to access this package. So far there seems no harm or implications of revoking the access. • DBMS_LOB — OpenText PPM stores a lot of data in BLOB and CLOB columns and the application code might need to parse or modify the contents. • DBMS_SCHEDULER — Not needed for core OpenText PPM. This is required for Operational Reporting. If your organization does not use Operational Reporting, you may revoke access to this package. • DBMS_SQL — OpenText PPM generates dynamic SQL and executes it during installation and upgrade. This package is also used to create triggers as part of the application functionality.

EXECUTE ANY PROCEDURE EXECUTE ANY PROGRAM	<p>OpenText PPM does not need the EXECUTE ANY PROCEDURE privilege.</p> <p>OpenText PPM needs the EXECUTE ANY PROGRAM privilege. The definition of EXECUTE ANY PROGRAM is: use any program in a job in the grantee's schema.</p>
CREATE ANY VIEW, CREATE ANY TABLE, SELECT ANY TABLE	<p>RML_USER needs these three privileges:</p> <ul style="list-style-type: none"> • CREATE ANY TABLE — Create tables in any schema. The owner of the schema containing the table must have space quota on the tablespace to contain the table. • SELECT ANY TABLE — Query tables, views, or materialized views in any schema. • CREATE ANY VIEW — Create views in any schema.
Privileges required for RML database schema	<pre>grant create session to &RML_USERNAME; grant create table to &RML_USERNAME; grant create view to &RML_USERNAME; grant create synonym to &RML_USERNAME; grant resource to &RML_USERNAME;</pre>

For information about other Oracle database privileges, see Oracle documentation.

Verify port availability

To successfully install and configure OpenText PPM, specific ports must be available through the firewall. To expedite installation, make sure that the ports are available before you start to install the product. The following table contains summary information about the ports and protocols that OpenText PPM system components use to communicate.

Note: If you are using an external Web server, you must assign it a port number other than the one assigned to the internal Web server.

Table 3-6. PPM Center ports and protocols

Communication Channel	Protocols	Ports
Web Browser and Web Server	HTTP/HTTPS	80/443 (configurable)
	<p>If you do not use the default port, you must specify the port number in the URL.</p> <p>Example:</p> <p><code>http://xyz.com:<Port></code></p> <p>You may also be required to open the firewall for ports other than the defaults.</p> <p>On UNIX systems, only processes started by the root user can be assigned a port number that is less than 1024.</p>	
PPM Workbench and App Server	RMI / RMIS	1099 (configurable)
External Web Server and App Server	AJP13	8009 (configurable)
App Server and Database	JDBC	1521 (configurable)
App Server and Mail Server	SMTP	25
App Server and LDAP Server	LDAP	389
App Server and LDAP Server	LDAP over SSL	636
App Server and External System	SSH	22
App Server and External System	FTP (control)	21
App Server and External System	FTP Data	Dynamic
App Server and External System	SCP (Secure Copy)	22

Table 3-6. PPM Center ports and protocols, continued

Communication Channel	Protocols	Ports
App Server nodes	TCP/UDP	Customized in <code><PPM_Home>/conf/integrity.conf</code> , <code>cache.conf</code> , and <code><PPM_Home>/conf/jboss/clusterservices.xls</code> . It depends on your requirement.

Assigning Ports Outside of the Ephemeral Port Range

OpenText PPM uses many client sockets for its internal communications. These are allocated randomly from the operating system's ephemeral port range.

To prevent conflicts between internal client socket use and your PPM Server port settings, you must check to make sure that the ephemeral port range set for your operating system does not conflict with any of the ports assigned in your `server.conf` file. For information about the default ephemeral port range on your operating system and how to adjust it, see the documentation for your specific operating system.

Collect required information

The OpenText PPM installer prompts you to enter information that it uses to create and configure the PPM Server. The installer validates each value you enter before it continues the installation. The following table lists the information required for installing OpenText PPM on either a single PPM Server or the primary node for a server cluster.

Note: For additional information that you must provide if you are installing the primary PPM Server for a server cluster, see the *Configure a server cluster* section in the *System Administration Guide*.

Table 3-4. Required installation information

Prompt	Description
--------	-------------

Table 3-4. Required installation information, continued

CLASSPATH	<p>Environment variable that specifies the directory in which Java class files reside.</p> <p>Note: The directory path must not contain spaces.</p>
Software installation location	<p>Directory in which the PPM Server is to be installed and configured. If the directory does not exist, the installer creates it. The directory path cannot contain spaces.</p> <p>Note: Do not map the <PPM_Home> directory so that it is accessible from an external Web server. This introduces a potential security risk. We recommend that you not share this directory.</p>
Path to the Autopass license key file	<p>The Autopass license key file contains valid OpenText PPM license keys that you generated from the OpenText Licensing for Software portal.</p> <p>If you do not have a valid Autopass license key file, see "Generate Autopass license" on page 23.</p>
Path to JAVA_HOME	<p>The directory in which Java is installed. On UNIX systems, this environment variable is set in the profile file (a *.profile or *.cshrc file) of the user who is installing OpenText PPM.</p> <p>Windows example</p> <p>C:\jdk1.7</p> <p>Note: Make sure that the value specified for JAVA_HOME contains no spaces.</p>
<p>Database Access page</p> <p>(Displayed if you chose to have the installer create the database schemas)</p>	
System access username	System database user name to give the installer access to the database.
System password	<p>System database password to give the installer access to the database.</p> <p>Note: Do not use a string like "hpswDemo\$09\$" as a password during the installation, because the Installer treats "\$09\$" as a parameter and consequently turns "hpswDemo\$09\$" into "hpswDemo".</p>

Table 3-4. Required installation information, continued

JDBC URL	<p>JDBC URL the PPM Server uses to connect the Oracle database.</p> <p>Short format (non-RAC):</p> <pre>jdbc:oracle:thin:@<Host_Name><Port><SID></pre> <p>where</p> <ul style="list-style-type: none"> • <i><Host_Name></i> is the host name or IP address of the computer running the database • <i><Port></i> is the port number that SQL*Net uses to connect to the database. To get the actual value, look at the corresponding entry in <code>tnsnames.ora</code> • <i><SID></i> is the security identifier of the database. This is usually identical to the database connect string. If it is different, an extra parameter is required. <p>RAC format:</p> <pre>RAC (description=(address_list=(address= (protocol=TCP) (host=<Host_Name1>) (port=<Port>)) (address=(protocol=TCP) (host=<Host_Name2>) (port=<Port>)))(load_balance=YES))(connect_ data=(server=DEDICATED)(service_name= <Service_Name>))</pre> <p>Example of database access information used to enable the PPM Server to communicate with databases on two servers named Jaguar1 and Jaguar2:</p> <pre>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS= (PROTOCOL=TCP) (HOST=jaguar1) (PORT=1521)) (ADDRESS= (PROTOCOL=TCP) (HOST=jaguar2)(PORT=1521))) (CONNECT_DATA= (SERVICE_NAME=J920)))</pre>
ORACLE_HOME	Home directory for the Oracle client tools on the PPM Server machine. The directory path cannot contain spaces.
PATH	Environment variable that specifies the directories to be searched to find a command.

Table 3-4. Required installation information, continued

SQL*PLUS	<p>Location of the SQL*Plus utility.</p> <p>SQL*Plus is not required for installation, but is required for the PPM Server.</p> <p>Example</p> <p>C:\Oracle\bin\sqlplus.exe</p> <p>If the ORACLE_HOME environment variable is set, then this parameter is detected automatically.</p>
OpenText PPM Schema page	
(Displayed if you created the database schemas before installation)	
Username	OpenText PPM database schema user name.
Password	OpenText PPM database schema password.
RML Schema page	
Username	Username for the OpenText PPM Reporting Meta Layer (RML) schema.
Password	Password for the OpenText PPM Reporting Meta Layer (RML) schema.
Tablespaces page	
Table	Data type tablespace in the Oracle database
Index	Index type tablespace in the Oracle database
CLOB data	Character large object data type (CLOB) tablespace in the Oracle database
NT Service	
Service name	Name of the Windows service for the PPM Server.
Regional Settings	
Holiday schedule	Holiday schedule on which to base the OpenText PPM regional calendar. If your holiday schedule is not listed, you can select None . In that case, a new calendar with no holidays is set as the system default regional calendar.

Table 3-4. Required installation information, continued

Currency code	Three-letter code for the default currency. The system default is in US dollars (USD). For information on currency codes for other countries, see the <i>Financial Management User Guide</i> . Caution: Once you choose your default currency during installation, you cannot change it unless you reinstall PPM.
Region name	Name of the region for the installation, which is defined by a combination of calendar and currency. If your organization operates in only one region, use "Enterprise" or your company name.

Generate Autopass license

To activate and generate your Autopass license for the version of OpenText PPM you purchased:

1. Go to the Software Licenses and Downloads Portal at <https://sld.microfocus.com/mysoftware/index>
2. On the Manage Entitlements page, select the **Available for Activation** option. Select a product and click **Activate**.
3. Complete the activation process to generate an Autopass license.

Create a shared folder for the server.conf File

In order to implement a server cluster (recommended) you must have a shared folder for the server configuration file (`server.conf`). In addition to giving all nodes in a cluster access to the same `server.conf` file, the shared folder simplifies maintenance of the `server.conf` file.

This section provides instructions on how to prepare the shared folder on both Windows and UNIX systems.

Note: The shared folder described in this section is also required to give users access to the Administration Console interface after you deploy your PPM Center instance. For information about the Administration Console, see the

"Tools in the Administration Console" in the *System Administration Guide*.

Prepare a Shared Folder for server.conf on a Windows system

1. Create a shared folder on a file server.

Note: If you plan to configure the server cluster configurations on multiple machines, keep in mind that the nodes in the cluster must all run on the same operating system. Shared access to the `server.conf` file does not support mixed operating systems.

2. Attach the shared folder to each machine that is to host OpenText PPM.
3. If you plan to host multiple PPM Server clusters (instances) under the same account on a single machine, do the following. Otherwise, proceed to step 4.
 - a. Using a text editor, create a file named `ppm_server_conf.env`, and add to it the following text:

```
export PPM_SERVER_CONF_DIR=//<IP_Address>/<Local_Drive_Letter>$/<Shared_Folder>/
```
 - b. Save the `ppm_server_conf.env` file in the `<PPM_Home>` directory and close the file.
4. Open the Control Panel and define an environment variable named `PPM_SERVER_CONF_DIR` for an account that is to run OpenText PPM nodes on Windows. The value of the environment variable is the location of the shared folder.

Caution: Make sure that you use Universal Naming Convention (UNC) notation

(`//<IP_Address>/<Local_Drive_Letter>$/<Shared_Folder>/` or `<File_Server_Name>$/<Shared_Folder>`) to specify the location of your shared folder.

Prepare a Shared Folder for server.conf on a UNIX system

1. Create a shared folder on a file server.

Note: If you plan to configure the server cluster configurations on multiple machines, keep in mind that the nodes in the cluster must all run on the same operating system. Shared access to the `server.conf` file does not support mixed operating systems.

2. Mount the shared folder to each machine that is to host OpenText PPM.
3. If you plan to host multiple PPM Server clusters under the same account on a single machine, do the following. Otherwise, proceed to step 4.
 - a. Using a text editor, create a file named "ppm_server_conf.env", and add to it the following text:
4. In the `$HOME/.profile` file of the account that is to run OpenText PPM, add the following line:

```
export PPM_SERVER_CONF_DIR=//<IP_Address>/<Shared_Folder>
```

- b. Save the file to the `<PPM_Home>` directory and close the file.

```
export PPM_SERVER_CONF_DIR=<Mount_Point>/<Shared_Folder>
```

Install PPM on a Windows system

The installation utility for a Windows server is an executable file that performs the steps required for a basic server installation. The executable and supporting files are contained in a Zip file. The typical installation automatically installs the following components on the server:

- PPM program files
- PPM database objects
- **Start** menu item
- Windows service

Caution: You cannot install PPM unless you have SYSDBA privileges or a DBA has already created the required schemas.

Prerequisite:

You as the system user should have grants to the following sys package before the installation:

- `grant EXECUTE on SYS.DBMS_CRYPTO to <system_user> with grant option;`
- `grant SELECT on SYS.USER_OBJECTS to system with grant option;`

The **<system_user>** is the user you input in the installation bundle UI. Make sure this user has the privileges to execute the following SQL statements if you do not create database schemas before the installation:

- `CreateKintanaUser.sql`
- `CreateRMLUser.sql`

To install the PPM Server on Windows:

1. Make sure that you have a UNIX emulator (such as Cygwin) installed.

Note: To configure private key authentication with secure shell (see the Configure Private Key Authentication with Secure Shell section in the *System Administration Guide*), you use the `ssh-keygen` utility, which is part of the Cygwin installation. To get this utility, you must enable the Open SSH components during Cygwin installation.

2. Extract all files from `ppm-<ppm_version>-install.zip` to the file system.

The extraction creates the `<PPM_Extract>/ppm<ppm_version>` directory, which includes the `install.sh` file.

3. From the `ppm<ppm_version>` directory, use Windows Command or Cygwin to run the `install.sh` script.

The OpenText PPM installer starts.

Note: If you install PPM on Windows 2016 with JDK 7, you have to modify the properties of the ppm_install.exe file for the installer to run properly. To do so:

- a. Locate and right click the ppm_install.exe file in the <PPM_Extract>/ppm<ppm_version>/Disk1/InstData/Windows/NoVM directory.
- b. Select **Properties** from the popup menu.
- c. In the ppm_install Properties window, go to the **Compatibility** tab.
- d. In the **Compatibility mode** section, select the **Run this program in compatibility mode for:** checkbox and select a desired option from the drop-down list.
- e. Then in the **Privilege level** section, select the **Run this program as an administrator** checkbox.
- f. Click **Apply**.
- g. Click **OK**.
- h. Double click the ppm_install.exe file to launch the OpenText PPM installer.

4. From the **Select the language for this installation** list, select the language that you want the installer to use to display the installation steps.

Depending on the operating system language of the host machine, the languages available for displaying the installation wizard steps are limited, as shown in the following table.

Operating System Language	Languages Available for Displaying Installation Wizard Steps
English	English
Spanish	Spanish
Italian	Italian
French	French
Dutch	Dutch
German	German
Portuguese (Brazilian)	Portuguese (Brazilian)
Swedish	Swedish

Operating System Language	Languages Available for Displaying Installation Wizard Steps
Chinese	Chinese English
Korean	Korean English
Japanese	Japanese English
Turkish	Turkish English
Arabic	Arabic English
Polish (available in 23.3 and later versions)	Polish English

So, for example, if your operating system is Chinese, you can view the installer steps in either Chinese or English. This option in no way affects the language packs that you can install.

Limitation: If the Windows system language is Chinese, Japanese, or Korean, and the system has JDK 8 installed, PPM installer wizard does not display language list.

Workaround: This is the limitation of the third-party software InstallAnywhere. To work around this issue, run the following command to set the locale for the PPM installer wizard:

```
sh install.sh -l <language_code>
```

where language codes for Chinese, Japanese, and Korean are zh_CN, ja, and ko respectively.

Note: The language you select here is not the same as the system language, which you select at a later step.

5. On the License Agreement page, read the agreement carefully, and then select **I accept the terms of the License Agreement.**
6. On the Choose License page, select **Yes, I have a license and want to install it now** to install an Autopass license file that is sent to you. Or, select **No, thanks. I will install the license later.** to generate and activate a 60-day trial license.
7. On the next several pages, provide the information you collected to prepare for the installation (see [Collect required information](#)).
8. From the **Select the system language** list, select the OpenText PPM system language to use.

The system language is the language used to generate system-level information such as server start-up and shut-down messages. The system language is also used to display attributes of system data that does not support multiple translations. For more information, see the *Multilingual User Interface Guide*.

Limitation: (Available in 23.3 and later versions) If you select Polish as the system language, the regional calendar for Poland is not available for selection.

Workaround: You need to create a Polish regional calendar. For details, see the Set up a regional calendar section in the *Resource Management Guide*.

9. Select the check boxes for any languages you want to deploy in addition to the system language on your instance.

Note: You can always install languages later, as needed, by running the `kDeploy.sh` script. For information on how to deploy languages after you install OpenText PPM, see the *Multilingual User Interface Guide*.

10. On the server mode step, do one of the following.

Note: In this context, the term "server mode" refers to a stand-alone or a clustered type of installation, and is different than the server mode addressed in the Set the Server Mode section in the *System Administration Guide*.

- If you are installing a primary PPM Server for a production instance, select **Cluster**.
- If you are installing a OpenText PPM testing instance or development instance that will consist of a single PPM Server, leave **Stand-alone** selected.

The Configuration Option page opens next.

11. On the Configuration Option page, indicate whether you want to configure the PPM Server during installation, or later, after installation.

Note: For instructions on configuring the server later, see the Configure the System section in the *System Administration Guide*.

12. If you choose to configure the server, the installer displays several pages of server configuration parameters. Provide values for all of the required parameters, which are displayed in red text, and for any optional parameters you want to set.

For descriptions of and valid values for all listed parameters, see the OpenText PPM Configuration Parameters in the *System Administration Guide*.

After you provide all required information, the Pre-Install Summary page displays summary information about the installation to be performed.

13. To proceed with the installation, click **Install**.

Note: The installer displays a progress bar that enables you to monitor installation.

To optimize system performance, the installation script rebuilds statistics for the Oracle optimizer. For the installation procedure to perform this step, you must have the following grants to the schema in place:

- `grant select on v_$parameter to <PPM_Schema>`
- `grant select on v_$mystat to <PPM_Schema>`
- `grant select on v_$process to <PPM_Schema>`
- `grant select on v_$session to <PPM_Schema>`

- `grant execute on dbms_stats to <PPM_Schema>`

The `GrantSysPrivs.sql` script (located in the `<PPM_Extract>/ppm960/sys` directory) performs these required grants.

If you did not run the `GrantSysPrivs.sql` script before you started installation, do it now (with SYS DBA access).

14. After successful installation, OpenText PPM is installed as a Windows service. You can view the properties for this service through the Services Control Panel item. To complete the service setup, start the OpenText PPM service from the Control Panel (**Start > Administrative Tools > Services.**)

We recommend that you set the startup type to **Automatic** so that the PPM Server restarts automatically after the computer is restarted. If you have generated a custom OpenText PPM user, specify this user name for the "Log On As" value.

Note: OpenText PPM comes with an Administrator user with the default username/password combination admin/admin. We recommend that you change the password for the administrator user after you install OpenText PPM.

An item that corresponds to the Windows service name that you specified during installation is added to the **Start** menu. The menu provides links to OpenText PPM documentation and an uninstall program.

If you did not configure the PPM Server during installation, see the Configure or Reconfigure the PPM Server section in the *System Administration Guide*.

Install PPM on a UNIX system

Note: You as the system user should have grants to the following sys package before the installation:

- `grant execute on SYS.DBMS_CRYPT0 to <system_user> with grant option;`
- `grant SELECT on SYS.USER_OBJECTS to system with grant option;`

The `<system_user>` is the user you input in the installation bundle UI. Make sure this user has the privileges to execute the following SQL statements if you do not create database schemas before the installation:

- `CreateKintanaUser.sql`
- `CreateRMLUser.sql`

1. Download the bundle and copy it to a directory, referred to as `<PPM_Extract>` in the following steps.
2. To extract the files into the `<PPM_Extract>` directory from the download bundle, at a command prompt, type one of the following:

```
unzip ppm-<ppm_version>-install.zip
```

Alternatively,

```
jar xvf ppm-<ppm_version>-install.zip
```

All the files and scripts required for OpenText PPM installation are extracted (to `<PPM_Extract>`). The installer prompts for the software install directory. You can specify any directory for installation.

The `<PPM_Extract>/ppm<ppm_version>` directory that results from the extraction contains the `install.sh` shell script.

3. To start the installation, run the installation script (as the system account for OpenText PPM) and specify the installation mode.

Example

```
sh ./install.sh -i[swing|console]
```

where

- `-i swing`: Swing mode is an interactive, GUI-based installation mode that requires an X Window session. A wizard guides you through the installation steps.
- `-i console`: Console mode is the interactive command-line mode. The installation script runs within the terminal session and you respond to the prompts.

- `-i swing`: Swing mode is an interactive, GUI-based installation mode that requires an X Window session. A wizard guides you through the installation steps.
- `-i console`: Console mode is the interactive command-line mode. The installation script runs within the terminal session and you respond to the prompts.

To optimize system performance, the installation script rebuilds statistics for the Oracle optimizer. For the installation procedure to perform this step, you must have the following grants to the schema in place:

- `grant select on v_$PARAMETER to <PPM_Schema>`
- `grant execute on DBMS_STATS to <PPM_Schema>`
- `grant select on v_$MYSTAT to <PPM_Schema>`
- `grant select on v_$PROCESS to <PPM_Schema>`
- `grant select on v_$SESSION to <PPM_Schema>`

The `GrantSysPrivs.sql` script (located in the `<PPM_Extract>/ppm964/sys` directory) performs these required grants.

4. If you did not run the `GrantSysPrivs.sql` script before you started installation, do it now (with SYS DBA access).

Note: OpenText PPM comes with an Administrator user with the default username/password combination `admin/admin`. We recommend that you change the password for the administrator user after you install OpenText PPM.

Install PPM with special command

You can use the `install_auto.sh` command to install PPM.

How to install PPM with `install_auto.sh`:

1. Extract the `ppm-1000-install.zip` package.

The extraction creates the `ppm1000` directory, which includes the `install_auto.sh` and `template_installation_file.properties` file.

2. In the `ppm1000` directory, run the `install_auto.sh` command. You can run this command either in interactive mode or from file.

Run `install_auto.sh` in interactive mode

Run `install_auto.sh` without any parameter attached. And follow the on-screen instructions to provide required information to complete the installation.

For example, you will be asked to provide the system username and password, and to select the currency and region.

Run `install_auto.sh` from file

To run `install_auto.sh` from file, you should first set the properties in the `template_installation_file.properties` file:

- a. Open the `template_installation_file.properties` file from the `ppm1000` directory. Or, you can copy the file and save it somewhere else.
- b. In the `template_installation_file.properties` file, follow the instructions in the file to set the properties.

You can set the `sysPassword`, `DB_PASSWORD`, and `RML_PASSWORD` in the environment variable `DB_PASSWORD` to avoid security problems.

Run `install_auto.sh` with specific parameters attached.

Parameter	Description
<code>-f</code> , or <code>--install_file</code>	Specifies the path of the <code>template_installation_file.properties</code> file. Used when the file is stored in a local place. You can find the details in the <code>template_installation_file.properties</code> file.

Parameter	Description
-u, or --install_url	Specifies the URL of the <code>template_installation_file.properties</code> file. Used when the file is stored in a URL. You can find the details in the <code>template_installation_file.properties</code> file.
-o, or --install_filesystem_only	Installs filesystem only.
-h, or --help	Shows the usage of the parameters.

Configure the FTP server on Windows

OpenText PPM uses FTP to move files between machines. To transfer files between machines on a network, each source and destination machine must be running an FTP server. On UNIX platforms, this is standard functionality, but machines running Windows require additional FTP server configuration to function with OpenText PPM.

Before you configure the FTP server on a machine, make sure that the Windows user account (which OpenText PPM uses to open a connection) has access to the directories to which files are to be moved. Some FTP servers require that you map these directories to FTP aliases, and a configuration utility is usually provided for this (for example, for Microsoft IIS, the utility is Internet Services Manager).

Note: On Windows, most FTP servers, including Microsoft IIS, do not support drive letters. If you use FTP in OpenText PPM, the drive letter is removed from the base path. If your base path is `d:\ppm940`, then FTP tries to start from the ftp root directory and FTP fails.

To work around this, you must create an FTP alias. (For example, map `/ppm940` to `D:\ppm940`.)

Configure the FTP server according to directions that the vendor has provided. For the File and Directory Chooser components to work, you must set the FTP server directory listing style to UNIX, and not to MS-DOS.

To set the directory listing style to UNIX:

1. In Windows, open the Internet Services Manager.
2. In the left pane, under **Console Root**, open the Internet Information Server.
3. Select the machine name.
4. Right-click the Default FTP site displayed in the right pane, and then click **Properties** on the shortcut menu.

The Default FTP Site window opens.

5. Click the **Home Directory** tab.
6. Under **Directory Listing Style**, click **UNIX**.
7. Test the connection by trying to open a session manually. If you can open an FTP session and navigate from one directory to another, then OpenText PPM can do this too.

Contact support for troubleshooting

If you encounter problems with your installation or if have questions, contact Software Support (<https://softwaresupport.softwaregrp.com/>). Before you contact Software Support, have the following information ready:

1. Open the `ppm_install.log` file (located in the `<PPM_Home>/install_950/logs` directory) in a text editor.

This file provides information about what part of the installation failed.

2. Search the `ppm_install.log` file for an error message that is specific to installation failure.
3. Place all of the files in the `<PPM_Home>/install_950/logs` directory in a compressed file.

The installation utility creates a separate log directory for each installation attempt. In the most recent directory, examine each file to determine exactly

where the PPM Server has failed. The log file contains information about which failed action it attempted.

Optional installations

In addition to installing the foundation product, the database, and the application server (PPM Server), you can install Best Practices and Deployment Management Extensions on your system. Best Practices helps optimize your use of Program Management and Portfolio Management. Deployment Management Extensions provides entities unique to Oracle and SAP development environments.

Install Best Practices

This section provides details on installing Best Practices.

Requirements for installing Best Practices

To install Best Practices, you must have the following ready:

- The OpenText PPM database schema name
- The OpenText PPM database username and password
- The OpenText PPM administrator username and password
- Licenses for both the Portfolio Management and Program Management.

Install Best Practices

Follow the steps below to install Best Practices:

1. Start the PPM Server from the command line.
2. Set your server to **RESTRICTED** mode.

Note: Although setting your server to RESTRICTED mode is optional, we recommend that you do so. In RESTRICTED mode, the PPM Server enables only users with Administrator access granted to log on. You can use the **setServerMode.sh** script to set the server to RESTRICTED mode.

3. Run the **kDeploy.sh** script, as follows:

sh ./kDeploy.sh -best-practices

4. To verify that Best Practices is successfully installed, run the **sh ./kDeploy.sh -I** script. This returns a list of the deployed bundles in an instance.

Language support for Best Practices

OpenText PPM Best Practices content is available for all supported language packs. We recommend that you deploy language packs soon after you install Best Practices so that the its content matches in all languages.

Note: For detailed information about the language packs and how to install them, see the *Multilingual User Interface Guide*.

Best Practices content

The Best Practices content includes the request types and workflows.

Caution: In order for OpenText PPM software to function correctly, Best Practices request types must be installed on your system and correctly associated with the menu items and project types.

Best Practices request types

Request Type and Description	Menu Selections	Field Groups
PFM - Proposal Represents a project proposal within the Portfolio Management module.	<ul style="list-style-type: none">• Create > Proposal• Open > Portfolio Management > Create Proposal	PFM Proposal

Request Type and Description	Menu Selections	Field Groups
<p>PFM - Asset</p> <p>Represents the ongoing costs and maintenance of the result of a project in the Portfolio Management system.</p>	<ul style="list-style-type: none"> • Create > Asset • Open > Portfolio Management > Create Asset 	<p>PFM Asset</p>
<p>PFM - Project</p> <p>Represents data and process associated with project. This request type is connected to a project type.</p>	<ul style="list-style-type: none"> • Create > Project • Open > Portfolio Management > Create Project 	<p>PFM Project</p>
<p>Program Details</p>	<ul style="list-style-type: none"> • Create > Program • Open > Program Management > Create Program 	<p>PFM Program</p>
<p>Project Issue</p> <p>Represents issues associated with a project.</p>	<ul style="list-style-type: none"> • Create > Project Issue • Open > Project Management > Project Controls > Submit Project Issue • Search > Project Issues • Open > Project Management > Project Controls > Search Project Issues <p>Also associated with a project type to enable creating from within the Project Overview page.</p>	<ul style="list-style-type: none"> • Project Reference • Project Issue

Request Type and Description	Menu Selections	Field Groups
<p>Project Risk</p> <p>Represents risks associated with a project.</p>	<ul style="list-style-type: none"> • Create > Project Risk • Open > Project Management > Project Controls > Submit Risk • Search > Project Risks • Open > Project Management > Project Controls > Search Project Risks <p>Also associated with a project type to enable creating from within the Project Overview page.</p>	<ul style="list-style-type: none"> • Project Reference • Project Risk
<p>Project Scope Change Request</p> <p>Represents scope changes associated with a project.</p>	<ul style="list-style-type: none"> • Create > Project Scope Change • Open > Project Management > Project Controls > Submit Scope Change • Search > Project Scope Changes • Open > Project Management > Project Controls > Search Scope Changes <p>Also associated with a project type to enable creating from within the Project Overview page.</p>	<ul style="list-style-type: none"> • Project Reference • Project Scope Change
<p>Program Issue</p> <p>Represents issues associated with a program.</p>	<ul style="list-style-type: none"> • Create > Program Issue • Open > Program Management > Issues > Submit Program Issues • Search > Program Issues • Open > Program Management > Issues > Search Program Issues 	<ul style="list-style-type: none"> • Program Issue • Program Reference

Request Type and Description	Menu Selections	Field Groups
Program Risk Represents risks associated with a program	<ul style="list-style-type: none"> • Search > Program Risks • Create > Program Risk • Open > Program Management > Risks > Search Program Risks • Open > Program Management > Risks > Submit Program Risk 	<ul style="list-style-type: none"> • Program Reference • Program Risk
DEM - Application Enhancement Used to request new functionality in IT current applications.	<ul style="list-style-type: none"> • Create > Request • Open > Demand Management > Create Request • Search > Requests • Open > Demand Management > Search Requests 	<ul style="list-style-type: none"> • Demand Management SLA Fields • Demand Scheduling Fields
DEM - Database Refresh Database refresh requests can be made for all IT Ops applications in the testing phase.	<ul style="list-style-type: none"> • Create > Request • Open > Demand Management > Create Request 	<ul style="list-style-type: none"> • Demand Management SLA Fields • Demand Scheduling Fields
DEM - Application Bug Used to report problems in IT applications.	<ul style="list-style-type: none"> • Create > Request • Open > Demand Management > Create Request 	<ul style="list-style-type: none"> • Demand Management SLA Fields • Demand Scheduling Fields
DEM - Initiative Used to request key projects for future quarters.	<ul style="list-style-type: none"> • Create > Request • Open > Demand Management > Create Request 	<ul style="list-style-type: none"> • Demand Management SLA Fields • Demand Scheduling Fields

You can create your own versions of the Best Practices request types by adding the appropriate field group, and then either editing the menu XML files or associating the request type with the project type (for Project Issue, Project Risk, and Project Scope Change).

Best Practices workflows

Best Practices Workflow	Description
PFM - Proposal	Portfolio Management process for requesting a new project.
PFM - Asset	Portfolio Management process for an asset life cycle.
PFM - Project	Portfolio Management process for a project life cycle.
Program Process	Portfolio Management process for a program life cycle.
DEM - Enhancement Request Process	Demand Management process for enhancement requests for new functionality in applications.
DEM - Database Refresh	Demand Management process for database refresh requests.
DEM - Bug Request Workflow	Demand Management process for application bug requests.
DEM - Project Initiative Process	Demand Management process for initiative requests for key projects.
Program Risk Management Process	Automated process for program risk management.
Risk Management Process	Automated process for project risk management.
Scope Change Request Process	Automated scope change request process with three levels of severity.
Issue Management Process	Automated process for issue management.

Install Deployment Management extensions

If you plan to install an Deployment Management Extension, you must do so after you install and configure OpenText PPM, and before you use OpenText PPM for processing.

You are not required to stop the PPM Server(s) before you install an Extension. However, we recommend that you install the Extension when no users are logged on to the system. Consider placing the server in **RESTRICTED** mode before you install.

Note: Although setting your server to RESTRICTED mode is optional, we recommend that you do so. In RESTRICTED mode, the PPM Server enables only users with Administrator access granted to log on. You can use the **setServerMode.sh** script to set the server to RESTRICTED mode.

For specific information on how to install an Deployment Management Extension, see the documentation for the Extension you purchased.

Verify OpenText PPM installation

To verify your PPM installation:

1. Check the logs produced during installation. (Located in the *<PPM_Home>/install_940/logs* directory)
2. Start the PPM Server.
3. Log on to OpenText PPM.

Note: All OpenText PPM clients use the same base URL, which is the Web location (top directory name) of the PPM Server. To obtain the URL, open the `server.conf` file, which is located in the *<PPM_Home>* directory. The URL is the value specified for the `BASE_URL` parameter.

4. Start the PPM Workbench.

5. Run a report. For instructions, see the "Run Server Reports from the Admin Tools" section in the *System Administration Guide*.
6. Create a request and test the graphical view of the request. (For instructions, see the *Demand Management User Guide*.)
7. Add a portlet to a PPM Dashboard page and export the page in PDF format. For instructions, see the "Use PPM Dashboard" section in the *Create Portlets and Modules Guide*.

Note: Before you can export a page in PDF format, you must enable that functionality. For information, see the "Enable Export to PDF" section the *System Administration Guide*.

8. Create a project and a work plan. (For instructions, see the *Project Management User Guide*.)
9. Create a staffing profile. (For instructions, see the *Resource Management User Guide*.)
10. Create a time sheet. (For instructions, see the *Time Management User Guide*.)

Install PPM with PostgreSQL

You can install PPM with a PostgreSQL database.

Supported and unsupported features

The section lists the modules, special commands, and reports that are supported and not yet supported when running PPM with a PostgreSQL database.

Supported features

Area	Details
Modules	<ul style="list-style-type: none">• Demand Management• Project Management• Program Management• Time Management• Portfolio Management• Resource Management• Financial Management• Team Management• User Administration• Dashboard• Integration with Jira• Integration with OpenText Software Delivery Management• PPM for Mobile

Area	Details
Special commands	<ul style="list-style-type: none"> • ksc_comment • ksc_create_baseline • ksc_create_project • ksc_run_sql • ksc_set • ksc_set_staffing_profile_status • ksc_take_snapshot • ksc_update_program_status • ksc_update_project_status • ksc_update_workplan_status • ksc_create_baseline_staffing_profile • ksc_copy_request • ksc_create_workplan • ksc_store
Reports	<ul style="list-style-type: none"> • Import Users • Request Detail Report • Notification History Report • Run License Usage Report • Time Sheet Summary Report • Request Type Detail Report • Time Sheet Details Report • Work Allocation Details Report • Actual Time Summary Report • Project Status List Report • Project Cost Breakdown Report • Delinquent Time Sheet Report • Run Field Security Denormalization Report
DMS solution	Only support PPM Center File System

Unsupported features

Area	Details
Modules	<ul style="list-style-type: none">• Application Portfolio Management• Deployment Management• Operational Reporting• Chatbot• MSP Plugin• Integration with Quality Center / Application Lifecycle Management• Integration with Universal CMDB• Integration with Service Manager• Integration with CARally• Integration with Digital.ai Agility• Integration with ServiceNow• Integration with Microsoft Azure DevOps
Preconfigured portlets	<ul style="list-style-type: none">• Project Current Cost Metrics• Financial Summary to Financial Data Comparison• Financial Comparison• Operational Report

Before you install

Before you install PPM with a Postgre SQL database, you must have the following ready:

- You have installed a PostgreSQL database. For instructions, see the PostgreSQL documentation.

The PostgreSQL 13.8 and later versions are supported.

Caution: In a server cluster configuration, each cluster must have its own database. Do not share a PostgreSQL database among clusters. However, you can create multiple databases in one PostgreSQL server.

- To adapt the PPM function, change the authentication mode of the PostgreSQL server.

Modify the **pg_hba.conf** file and change the encryption mode of IPv4 connections from `trust` to `md5` or `scram-sha-256`, as shown below:

```
# IPv4 local connections:  
host all all 0.0.0.0/0 md5
```

- You have installed orafce 3_24_4. For instructions, see <https://github.com/orafce/orafce>.
- (Windows systems only) Make sure you have a UNIX emulator (such as Cygwin) installed. For information about the UNIX emulator and how to install it, see the Cygwin website.

Install PPM

Follow the steps below to install PPM with a PostgreSQL database:

1. Obtain the installation bundle for PostgreSQL: **ppm_<PPM version>_install_pg.zip** from the [Software Licenses and Downloads Portal](#).
2. Extract the **ppm_<PPM version>_install_pg.zip** package.

The extraction creates the following:

- **conf** directory
- **resources** directory
- **bundle.zip** file
- **ppm.env** file
- **kinstall.sh** command
- **create_template.sh** command
- **create_user.sh** command
- **ppm-<PPM version>-multi-lang.jar** file (applicable to 24.2 and later versions)

3.

Operate as a DBA to create a PostgreSQL template database. Later during the PPM installation, configure this template database in the **ppm.env** file.

Run the **create_template.sh** command with the following required parameters attached. For example:

```
sh create_template.sh -h <pg_host> -p <pg_port> -t <your_template_dbname> -s
<pg_superuser> -w <pg_superuser_password>
```

Required parameter	Description
-h, or --host	The host of the server where the PostgreSQL database is installed.
-p, or --port	The port of the server where the PostgreSQL database is installed.
-t, or --template	The name of the template database you create.
-s, or --system	Superuser username for the PostgreSQL database.
-w, or --password	Superuser password for the PostgreSQL database.

4.

Operate as a DBA to create a PPM database user. Later during the PPM installation, configure this use in the **ppm.env** file.

Run the **create_user.sh** command with the following required parameters attached. For example:

```
sh create_user.sh -h <pg_host> -p <pg_port> -s <pg_superuser> -w <pg_superuser_
password> -u <ppm_db_user> -W <ppm_db_password>
```

Required parameter	Description
-h, or --host	The host of the server where the PostgreSQL database is installed.
-p, or --port	The port of the server where the PostgreSQL database is installed.

Required parameter	Description
-s, or --system	Superuser username for the PostgreSQL database.
-w, or --password	Superuser password for the PostgreSQL database.
-u, or --user	User name for the PPM Center database.
-W, or --user-password	Password for the PPM Center database.

5. Do either of the following to configure the required parameters.

- Edit parameter values directly in the **ppm.env** file.
- Run the **export** command to set the value for each parameter. For example,

```
export PPM_HOME=<path_to_the_folder_where_your_PPM_system_is_to_be_created>
```

Note: The values set by the `export` command override the values set in the **ppm.env** file.

Required parameters

Parameter	Description
PPM_HOME	The folder where your PPM file system is to be installed.
PGHOST	The host of the server where the PostgreSQL database is installed.
PGPORT	The port of the server where the PostgreSQL database is installed.
PPM_DB_NAME	Name of the PostgreSQL database that PPM uses. It is created automatically during the installation progress.
PPM_TEMPLATE_DB	The template database created in step 3 .

Parameter	Description
PPM_DB_USER	The PPM database user created in step 4 .
PPM_DB_PASSWORD	The PPM database user password created in step 4 .
LOCAL_IP	The IP or hostname of the server where PPM is to be installed.
HTTP_PORT	PPM server HTTP port.
RMI_PORT	PPM server RMI port.
JMS_PORT	PPM server JMS port.

6. Run the **sh kInstall.sh** command. You can run this command either in interactive mode or silent mode.

Run kInstall.sh in interactive mode

Run kInstall.sh without any parameter attached and follow the on-screen instructions to provide required information to complete the installation.

For example, you will be asked to accept the terms of License Agreement, and to select the currency and region.

Run kInstall.sh in silent mode


Run kInstall.sh with specific parameters attached, for example:

```
sh kInstall.sh -a -H 1 -b USD -r Enterprise
```

Parameter(*Required)	Description
*-a, or --accept-license	Accepts the terms of the License Agreement.
*-b or --base-currency	Sets the system base currency. If no value is provided, the default value USD (US dollars) is used. For information on currency codes for other countries, see the <i>Financial Management Guide</i> .

Parameter(*Required)	Description
-c, or --calendar-name	Sets the system default regional calendar. If no value is provided, the default value NONE is used. In that case, a new calendar with no holidays is set as the system default regional calendar.
-d, or --database	Installs the PPM database only.
-f, or --filesystem	Installs the PPM file system only.
*-H, or --holiday-schedule	Sets the holiday schedule on which to base the PPM regional calendar. If no value is provided, the default value 1 (US) is used. For information on holiday schedule codes for other countries, see the <i>Resource Management Guide</i> .
*-r, or --region-name	Sets the region for the installation. If no value is provided, the default value Enterprise is used.
-t, or --trial	Installs a trial license. For details about product license types, see the "Overview of the Autopass Licensing Solution" section in the the <i>System Administration Guide</i> .
-F, or --force	Available in 24.2 and later versions Deletes the database. Required when re-installing PPM. Note: Use this parameter with caution.
-h, or --help	Available in 24.2 and later versions Displays the description and usage of each parameter.

7. (Optional) Configure the **server.conf** parameters in PPM. For details, see the "Server Configuration Parameters" part in the *System Administration Guide*.
8. In the **<PPM_Home>/bin** directory, run the **kUpdateHtml.sh** command to deploy the configurations. For details, see the "kUpdateHtml.sh" command in the *System Administration Guide*.
9. Install the license file by running the **kLicenseInstall.sh** command. For details, see the "kLicenseInstall.sh" command in the *System Administration Guide*.
10. Start the PPM server by running the **kStart.sh** script. For details, see the **kStart.sh** command in the *System Administration Guide*.

 See also:

- ["Before you install" on page 6](#)
- ["Upgrade PPM - PostgreSQL" on page 70](#)

Install PPM on the cloud

PPM supports installation on the cloud. This section provides the best practices for installing PPM on Amazon AWS and Microsoft Azure.

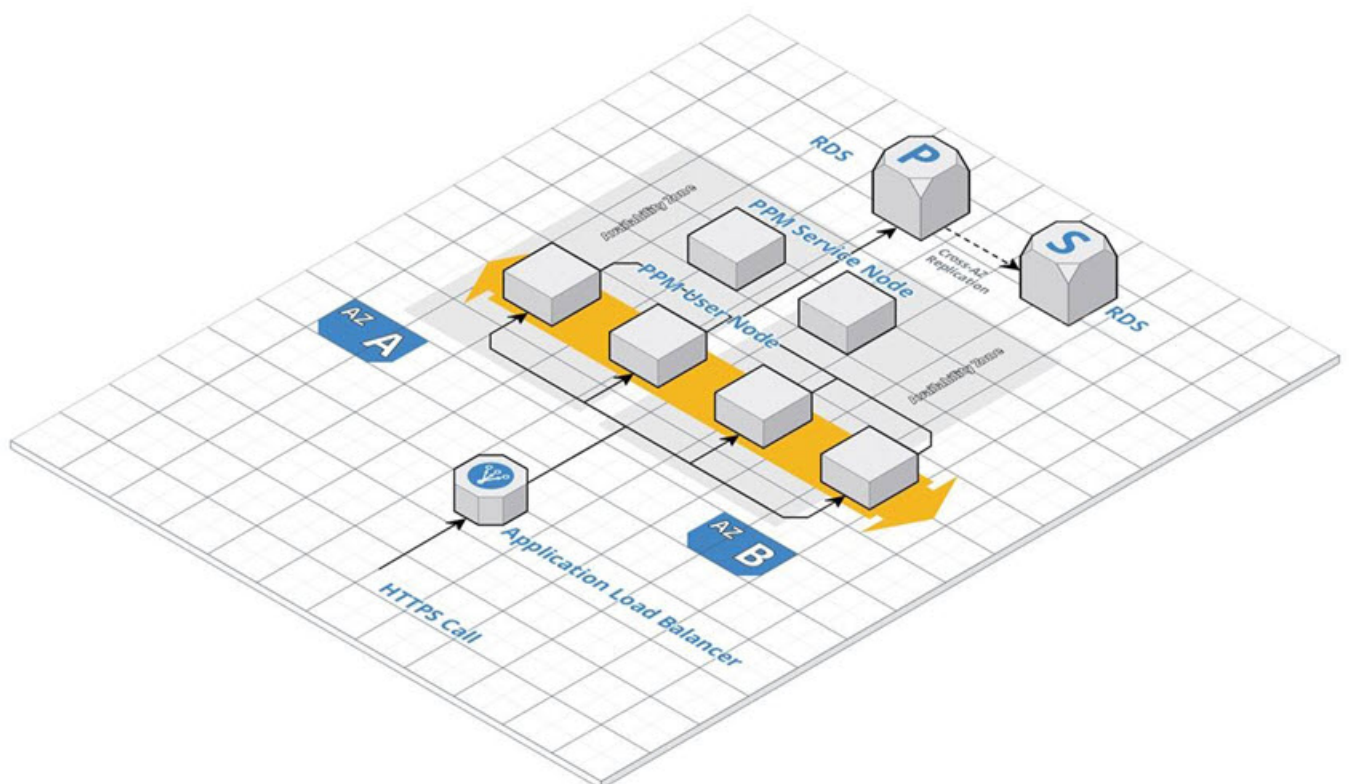
- ["Best practices of installing PPM on AWS" below](#)
- ["Best practices of installing PPM on Azure" on page 56](#)
- [Deploy PPM on AWS EKS \(Beta\)](#)

Best practices of installing PPM on AWS

You can install PPM 9.43 and later versions on the AWS cloud. This section provides the best practices of installing PPM on the AWS cloud.

Recommended architecture

The diagram below shows the recommended architecture on AWS.



- For PPM 9.43 - 9.53, a patch is required if you use RDS.
- You can also install Oracle in the EC2 instances by yourself instead of using RDS.

Install PPM on AWS

This section provides details on how to install PPM on AWS.

To install PPM on AWS:

1. Make sure you have a VPC and created EC2 servers in the private subnet. If you do not have them ready, [create a VPC](#) and [set up with Amazon EC2](#).
2. Install PPM on an EC2 server. See "[Install PPM with Oracle](#)" on page 6 for details.

Install one node per server. You can install PPM on the first EC2 server, and create an AMI from this server. Then you can launch new instances of this AMI to create the rest nodes in the cluster.

For more information on PPM server clustering, see the "Configure a Server Cluster" section in the *System Administration Guide*.

3. Create a shared folder for the `server.conf` file for all the nodes in the cluster. You can use the AWS EFS filesystem or NFS to create the shared folder.

For details, see the "Create a shared folder for the `server.conf` file" section in the *System Administration Guide*.

4. In the `server.conf` file, for each node, set values for the parameters listed in the "Server parameters required for server clustering" section in the *System Administration Guide*. Follow the guidelines below:

- Make sure you set the following parameters as below:

Parameter name	Value
EXTERNAL_WEB_PROTOCOL	http
BASE_URL	The load balancer address

- The values must be same for all the nodes in the cluster.
5. From <PPM_Home>/bin, run `kUpdateHtml.sh` to apply the changes to the `server.conf` file.
 6. Create an application load balancer as described in [Create an Application Load Balancer](#). Follow these guidelines:
 - Enable sticky sessions.
 - In the **Configure a target group** section, configure the target group as follows:

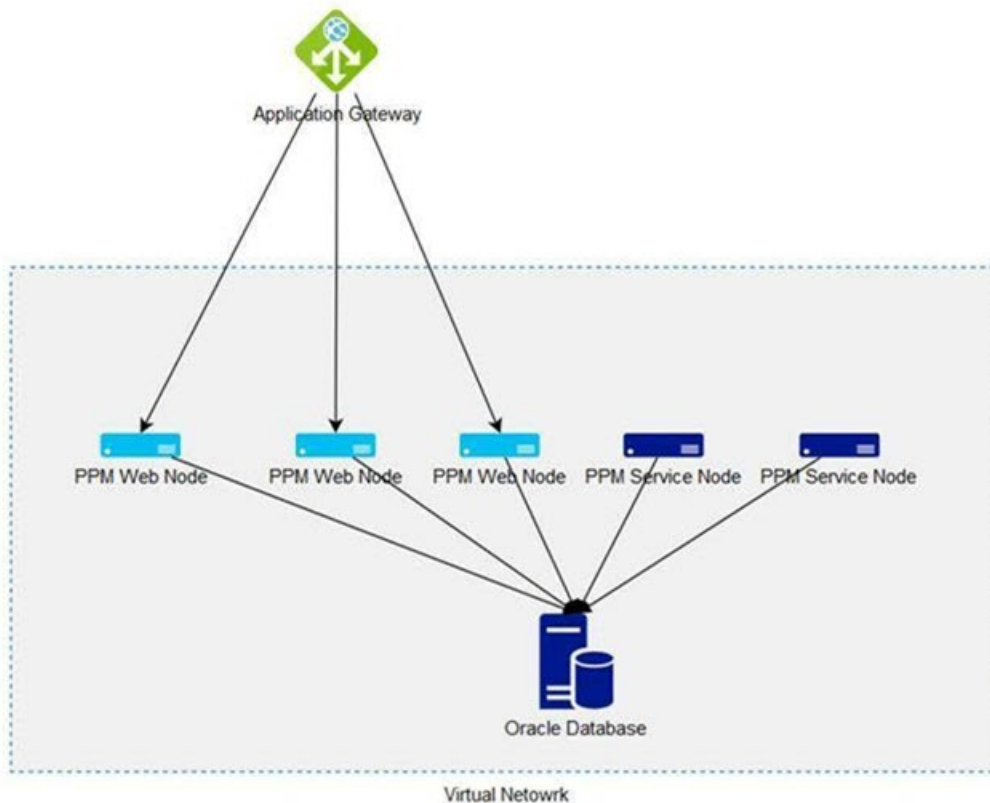
Parameter	Value
target group port	<code>com.kintana.core.server.EXTERNAL_WEB_PORT</code> in the <code>server.conf</code> file
target group protocol	http

Best practices of installing PPM on Azure

You can install PPM 9.43 and later versions on the Microsoft Azure cloud. This section provides the best practices of installing PPM on the Azure cloud.

Recommended architecture

The diagram below shows the recommended architecture on Azure.



Note: You can create additional virtual network for Oracle Database.

Install PPM on Azure

This section provides details on how to install PPM on Azure.

To install PPM on Azure:

1. Follow the instructions on [configuring an application gateway](#) to create an application gateway and virtual machines as backend servers. Follow these guidelines:

- Configure the **HTTP setting** of the application gateway as follows:

Parameter	Value
Backend protocol	http
Backend port	com.kintana.core.server.EXTERNAL_WEB_PORT in the server.conf file
Cookie-based affinity	Enable

- After you create virtual machines as backend servers, install PPM instead of IIS. Install one node per machine. See ["Installation and upgrade" on page 5](#) for details.
 - Add the PPM user nodes to the backend pool.
2. Create a shared folder for the `server.conf` file for all the nodes in the cluster. You can use Azure file share, Azure NetApp, or NFS to create the shared folder. For details, see the "Create a shared folder for the server.conf file" section in the *System Administration Guide*.
 3. In the `server.conf` file, for each node, set values for the parameters listed in the "Server parameters required for server clustering" section in the *System Administration Guide*. Follow these guidelines:
 - Make sure you set the following parameters as below:

Parameter name	Value
EXTERNAL_WEB_PROTOCOL	http
BASE_URL	The load balancer address

- The values must be the same for all the nodes in the cluster.
4. From `<PPM_Home>/bin`, run `kUpdateHtml.sh` to apply the changes to the `server.conf` file.

Docker install and upgrade

This topic describes the steps to install and upgrade PPM in a Docker container.

Before the installation and upgrade

Before installing using Docker images, install the Docker engine and client as described in the Docker documentation.

Install

Follow the steps below to install PPM from the Docker image.

1. Open a command window on the host machine.
2. Run the appropriate command based on the database you use:
 - If you use an Oracle database, run this command:

```
docker run -v <shared_folder>:/home/ppm/shared_dir -it
-e JDBC_URL=jdbc:oracle:thin:@xxxx:1521/ORCL
-e DB_SYSTEM_USER=DB_SYSTEM_USER
-e DB_SYSTEM_PASSWORD=DB_SYSTEM_PASSWORD
-e DB_USER=DB_USER
-e DB_PASSWORD=DB_PASSWORD
-e RML_DB_USER=RML_DB_USER
-e RML_DB_PASSWORD=RML_DB_PASSWORD
-e REINSTALL=true
<PPM_DOCKER_HUB>/ppm/ppmora:<ppm_version> sh install.sh
```

- If you use a PostgreSQL database, run this command:

```
docker run -v <shared_folder>:/home/ppm/shared_dir -it
-e JDBC_URL=jdbc:postgresql://xxxxxx:5432/<DB_NAME>
-e DB_SYSTEM_USER=DB_SYSTEM_USER
-e DB_SYSTEM_PASSWORD=DB_SYSTEM_PASSWORD
-e DB_USER=DB_USER
-e DB_PASSWORD=DB_PASSWORD
```

```
-e REINSTALL=true
-e TEMPLATE_DB=templateppm
<PPM_DOCKER_HUB>/ppm/ppmpg:<ppm_version> sh install.sh
```

Replace the following variables with your own values:

Variable	Description
<shared_folder>	The folder on the host machine that you want to share with the Docker container.
JDBC_URL	The Java Database Connectivity URL used to connect to a database.
DB_NAME	Name of the database. Applicable only when installing with a PostgreSQL database.
DB_SYSTEM_USER	The system account required for authentication when connecting to the database at the system level.
DB_SYSTEM_PASSWORD	The password associated with DB_SYSTEM_USER, required for authentication when connecting to the database at the system level.
DB_USER	The user name required for authentication when connecting to the database.
DB_PASSWORD	The password required for authentication when connecting to the database.
RML_DB_USER	The user name required for authentication when connecting to the RML database. Applicable only when installing with an Oracle database.
RML_DB_PASSWORD	The password required for authentication when connecting to the RML database. Applicable only when installing with an Oracle database.
REINSTALL	Specifies whether to reinstall PPM. Set to true to force re-installation, which will drop the existing schema.
SERVICES_ENABLED	Specifies whether a service node is installed or not.

Variable	Description
TEMPLATE_DB	The name of the template database. It is automatically created after the configuration
<ppm_version>	The specific version of the Docker image.

View install log

View the installation log from this directory: **<shared_folder>/install_logs**.

Upgrade

Follow the steps below to update PPM from the Docker image.

1. Open a command window on the host machine.
2. Run the appropriate command based on the database you use:
 - If you use an Oracle database, run this command:

```
docker run -v <shared_folder>:/home/ppm/shared_dir -it
-e DB_PASSWORD=DB_PASSWORD
<PPM_DOCKER_HUB>/ppm/ppmora:<ppm_version> sh upgrade.sh
```

- If you use a PostgreSQL database, run this command:

```
docker run -v <shared_folder>:/home/ppm/shared_dir -it
-e DB_PASSWORD=DB_PASSWORD
<PPM_DOCKER_HUB>/ppm/ppmpg:<ppm_version> sh upgrade.sh
```

Replace the following variables with your own values:

Variable	Description
<shared_folder>	The folder on the host machine that you want to share with the Docker container.
<ppm_version>	The specific version of the Docker image.

View upgrade log

View the upgrade log from this directory: **<shared_folder>/upgrade_logs**.

Start up

Follow the steps below to start up PPM in the Docker container.

1. Copy your license file to **<shared_folder>/license/**, where **<shared_folder>** is the folder mapped during installation.

Note:

- The IP address bound to the license file should match your database sever IP.
 - Make sure the license file has the **.dat** extension.
2. Open a command window on the host machine, and run the appropriate command based on the database you use:
 - If you use an Oracle database, run this command:

```
docker run -p <host_port>:8080 -v <shared_folder>:/home/ppm/shared_dir -it  
-e SERVICES_ENABLED=true  
-e BASE_URL=<your_base_url>:<host_port>  
<PPM_DOCKER_HUB>/ppm/ppmora:<ppm_version> sh startup.sh
```

- If you use a PostgreSQL database, run this command:

```
run -p <host_port>:8080 -v <shared_folder>:/home/ppm/shared_dir -it  
-e SERVICES_ENABLED=true  
-e BASE_URL=<your_base_url>:<host_port>  
<PPM_DOCKER_HUB>/ppm/ppmpg:<ppm_version> sh startup.sh
```

Replace the following variables with your own values:

Variable	Description
<host_port>	The port on the host machine through which to access PPM.
<shared_folder>	The folder on the host machine that you want to share with the Docker container.
SERVICES_ENABLED	Specifies whether a service node is enabled or not. The default value is false.
<ppm_version>	The specific version of the Docker image.

Upgrade PPM

This topic provides details about PPM upgrade process.

Upgrade paths

The following table outlines the upgrade paths for upgrading PPM from an earlier version to the latest.

Note: We strongly recommend upgrading to the latest version of PPM Center. In general, each new OpenText PPM version includes important product improvements that are not provided in earlier versions.

From versions	Upgrade paths
9.6x	<ol style="list-style-type: none">1. Upgrade to 10.02. Upgrade from 10.0 to 20233. Upgrade to the latest version
10.0.x	<ol style="list-style-type: none">1. Upgrade to 20232. Upgrade to the latest version
2023	Direct upgrade to the latest version
23.x	For PostgreSQL only:
24.x	To upgrade to version 24.3 or 24.4, first upgrade to version 24.2.
25.1	

Upgrade guide

Refer to the following table for the upgrade guides.

Note: If your PPM is upgraded from 10.0.3 or an earlier version, and you have applied the FIPS 140-2 compliant encryption algorithm before the upgrade, you need to reconfigure the algorithm after the upgrade. For details, see the "Apply FIPS 140-2 compliant encryption algorithm for PPM" section in the *System*

Administration Guide. .

Oracle	"Upgrade PPM - Oracle" below
PostgreSQL	"Upgrade PPM - PostgreSQL" on page 70

Upgrade PPM - Oracle

This topic provides detailed steps to upgrade PPM with an Oracle database to the latest version.

Upgrade PPM with Oracle database

You should use the `sh upgrade.sh` command to upgrade PPM.

To upgrade PPM with an Oracle database:

1. Extract the **ppm-<PPM version>-upgrade.zip** package.
2. Navigate to the **<PPM_Home>upgrade_<PPM version>** directory and run the **sh upgrade.sh** command.

Parameter	Description
<code>-o</code> , or <code>--on_current_home</code>	<p>Upgrades PPM on your current PPM system. The current PPM must be at version 9.61 or later. This parameter overrides the "-n" parameter. Do not use the "-o" and "-n" parameters together.</p> <p>Tip: We recommend that you upgrade PPM to a new <PPM_Home> folder, and use the current PPM system as a backup. For details, see "-n, or --new_PPM_Home" on the next page.</p>

-n, or --new_PPM_Home	<p>Upgrades your current PPM system to a new specified <PPM_Home> folder. For details, see "Standard procedures to upgrade PPM to 25.1".</p> <p>You cannot use this parameter if you already use the "-o" parameter.</p> <p>We recommend you run this command because doing this can back up the existing file system and makes it easier to restore.</p> <p>To define which files or folders to copy from the existing <PPM_Home> folder to the new folder, edit the <code>upgrade.properties</code> file under the <code>upgrade_<PPM version></code> folder, for example:</p> <pre>ppm.upgrade.copypath=server.conf;ppm_server_ conf.env;security;customization;content</pre> <p>Note: If you upgrade your instance to a new PPM home, in some cases, the attachments, report logs, and execution logs that are placed under the old PPM home might not be accessed by the upgraded instance. For details, see "Verify the paths of attachments, report logs, and executions logs are correct."</p>
-v, or --prevalidate	<p>Validates if your database is ready to upgrade. You can run this command anytime.</p> <p>Note: This command can only catch some most obvious issues. It cannot guarantee that no database related issues occur during the upgrade.</p>
-p, or --noprevalidate	<p>Skips the database validation.</p>
-b, or --no-backup	<p>Skips the file system backup.</p> <p>Note: This parameter is effective only when the -o parameter is used. For details, see "-o, or --on_current_home" on the previous page.</p>

-f, or --upgrade_filesystem	Upgrades the PPM code file system only. If you have multiple PPM nodes across multiple physical servers, you need to upgrade PPM code file system on each server, while upgrading the PPM database only once.
-d, or --upgrade_database	Upgrades the PPM database only.
-w, or --db-password	Enters the PPM schema password. Optional. You can enter it later.
-l, or --install_old_langs	Installs all the languages that have been installed in the current system to the upgraded instance.
-i, or --installation_locale	Installs the system language. You can use it together with the "-a" parameter, but cannot use it if you already use the "-l" parameter.
-a, or --add_langs	Installs extra languages in addition to the system language on the upgraded instance.
-m, or --name	Specifies the name of a non-primary node server. Optional. Use it when you upgrade on a non-primary node server.

Best practices for upgrade

The following are the recommended practices for upgrade.

Standard procedures to upgrade PPM to 25.1

Note: To upgrade to 25.1, upgrade the database first, then upgrade the file system. Alternatively, upgrade the database and file system together.

1. Before the upgrade, run **sh upgrade.sh -v** to validate your database.
2. Back up your database, and under any of the nodes, run **sh upgrade.sh -d** to upgrade the database.
3. For each node, upgrade your PPM code file system.

- a. Run **sh upgrade.sh -n <NEW_PPM_Home> -f** to upgrade your PPM code file system to a new <PPM_Home> directory.
 - b. Rename the old <PPM_Home> to a different name, for example, PPM_24.1_Backup.
 - c. Rename the new <PPM_Home> to the initial <PPM_Home> value.
4. Start PPM in the new <PPM_Home> folder.

Verify the paths of attachments, report logs, and executions logs are correct.

As the PPM code file system is upgraded to a new PPM home, to make sure the attachments, report logs, and execution logs can still be accessed by the upgraded instance, you should check the following parameters to verify the paths of these files.

- com.kintana.core.server.ATTACHMENT_DIRNAME
- com.kintana.core.server.BASE_LOG_DIR
- com.kintana.core.server.DISTRIBUTION_LOG_DIR
- com.kintana.core.server.PACKAGE_LOG_DIR
- com.kintana.core.server.REPORT_DIR
- com.kintana.core.server.REPORT_LOG_DIR
- com.kintana.core.server.REQUEST_LOG_DIR
- com.kintana.core.server.SCRIPT_DIR
- com.kintana.core.server.TEMP_DIR
- com.kintana.core.server.TRANSFER_PATH
- com.kintana.core.server.VALIDATION_LOG_DIR
- com.kintana.core.server.WEB_CACHE_DIR

If these parameters are set to the folders under the old PPM home that are mapped with shared folders where these files are actually stored, change the parameter values to mapped folders under the new PPM home.

If these parameters are set to the folders under the old PPM home where these files are actually stored, do either of the following.

- **Recommended way:** Do not host all your nodes on a single server. Separate them across multiple servers. Store attachments, reports, and execution logs in shared places and configure the above parameters to point to the shared places.
- Keep attachments/reports/execution logs in the old PPM home, and in the new PPM home, change the configurations to point to the old places.

For report and execution logs, in the `server.conf` file of the upgraded PPM instance, change the above parameters to point to the old locations (e.g. locations under the old PPM home).

For attachments, change `server.conf`: `com.kintana.core.server.ATTACHMENT_DIRNAME=/<old_PPM_Home>/attachments`, and in the Administration Console > DMS Configuration, set the **dms.filesys.attachmentDir** field to the old location.

Install languages during upgrade

You can install the previously deployed languages after or during the upgrade.

To install the languages after the upgrade, run the `kDeploy.sh` command.

To install the languages during the upgrade, do the following:

1. Before the upgrade, run **sh upgrade.sh -v** to validate your database.
2. Before the upgrade, leave one node untouched, and for each of the rest nodes, run **sh upgrade.sh -n <NEW_PPM_Home> -f -I** to upgrade your PPM code file system to a new <PPM_Home> directory and install the languages.
3. On the upgrade day, back up your database, and under the not-upgraded node, run **sh upgrade.sh -n <NEW_PPM_Home> -I**.
4. Start PPM in the new <PPM_Home> folder.

If you have frequent customizations, we have additional recommendations.

1. Commit your customizations to a source control system.
2. When you want to deploy customizations, download them from the source control system, and put them in a shared place.
3. Use **kSync.sh** to merge customizations from the shared place to the PPM code file system.

Troubleshooting

If your PPM is installed with the Oracle database 19c running in CDB mode, the following error may occur during the upgrade:

"ORA-38824: A CREATE OR REPLACE command may not change the EDITIONABLE property of an existing object."

To resolve this issue:

1. In the SQL Plus window, run the following query:

```
select 'alter ' || object_type || ' ' || object_name || ' editionable;' from
user_objects where editionable = 'N'
```

2. If no result is returned, close the SQL Plus window.
3. If a query is returned, for example: `alter procedure <procedure-name> editionable`, then run the returned query in the SQL Plus window.

 See also:

- ["Upgrade PPM - PostgreSQL" below](#)

Upgrade PPM - PostgreSQL

You can use the `sh kUpgrade.sh` command to upgrade PPM with a PostgreSQL database. This topic provides detailed steps to upgrade PPM with a PostgreSQL database to the latest version.

To upgrade PPM with a PostgreSQL database:

Note: To upgrade to version 24.3 or 24.4, you must first upgrade to version 24.2.

1. (Optional but highly recommended) Back up your database and license.conf file.

2. Extract the **ppm-*<PPM version>-upgrade-pg.zip*** package.

The extraction creates the **upgrade_*<PPM version>*** folder, which includes the following files:

- upgrade directory
- **bundle.zip** file
- **version.txt** file
- **kUpgrade.sh** file
- **ppm-*<PPM version>-multi-lang.jar*** file

3. Stop the PPM server.

4. Run the **sh kUpgrade.sh** command with the following commands to upgrade the PPM file system and database.

Parameter(*Required)	Description
*-p, or --ppmhome	Available in 24.2 and later versions Required. The folder where the PPM file system is installed. Make it sure it is consistent with the <code>PPM_HOME</code> parameter in the ppm.env file during the installation.
-f, or --filesystem	Attach this parameter if you want to upgrade the file system only.
-w, or --db-password	Available in 24.4 and later versions Password for the PPM database user. Optional. You can enter it later.

Tip: If you have multiple PPM nodes across multiple physical servers, you need to upgrade the file system on each server but need to upgrade the database only once.

During the upgrade process, PPM generates the logs in the **ppm-<PPM version>-upgrade-pg/logs** directory. If any problems occur during the upgrade, you can check the log file for troubleshooting.

5. Start PPM in your upgraded <PPM_Home> folder.
6. To make sure the attachments, report logs, and execution logs can still be accessed by the upgraded instance, set the paths of the following parameters to the corresponding folders under the upgraded <PPM_Home> directory:
 - com.kintana.core.server.ATTACHMENT_DIRNAME
 - com.kintana.core.server.BASE_LOG_DIR
 - com.kintana.core.server.DISTRIBUTION_LOG_DIR
 - com.kintana.core.server.PACKAGE_LOG_DIR
 - com.kintana.core.server.REPORT_DIR
 - com.kintana.core.server.REPORT_LOG_DIR
 - com.kintana.core.server.REQUEST_LOG_DIR
 - com.kintana.core.server.SCRIPT_DIR
 - com.kintana.core.server.TEMP_DIR
 - com.kintana.core.server.TRANSFER_PATH
 - com.kintana.core.server.VALIDATION_LOG_DIR
 - com.kintana.core.server.WEB_CACHE_DIR

 See also:

- ["Install PPM with PostgreSQL " on page 45](#)

Web service configuration update for upgrades to 23.4 and above

If you use PPM web services, after upgrading to 23.4 or a later version, additional client-server configurations are required to ensure continued web services functionality.

Tip: This update is a one-time action when first upgrading to 23.4 or a later version, and it is not needed for future upgrades.

Follow the steps below:

1. Download the latest **webservice_toolkit.zip** package, which includes sample code and updated configuration files. You can access the file from the following location:
`<Host>:<Port>/itg/download/webservice/webservice_toolkit.zip`
2. Update the **client-axis2.xml** configuration file. If your client setup includes a **client-axis2.xml** file, replace it with the updated version provided in **webservice_toolkit.zip** to ensure compatibility with version 23.4 and above.