

Pega Platform 8.6
Installation Guide
For WebSphere and IBM Db2 for z/OS



©2021 Pegasystems Inc., Cambridge, MA. All rights reserved.

Trademarks

For Pegasystems Inc. trademarks and registered trademarks, all rights reserved. All other trademarks or service marks are property of their respective holders.

For information about the third-party software that is delivered with the product, refer to the third-party license file on your installation media that is specific to your release.

Notices

This publication describes and/or represents products and services of Pegasystems Inc. It may contain trade secrets and proprietary information that are protected by various federal, state, and international laws, and distributed under licenses restricting their use, copying, modification, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

This publication is current as of the date of publication only. Changes to the publication may be made from time to time at the discretion of Pegasystems Inc. This publication remains the property of Pegasystems Inc. and must be returned to it upon request. This publication does not imply any commitment to offer or deliver the products or services described herein.

This publication may include references to Pegasystems Inc. product features that have not been licensed by you or your company. If you have questions about whether a particular capability is included in your installation, please consult your Pegasystems Inc. services consultant.

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors, as well as technical inaccuracies. Pegasystems Inc. shall not be liable for technical or editorial errors or omissions contained herein. Pegasystems Inc. may make improvements and/or changes to the publication at any time without notice.

Any references in this publication to non-Pegasystems websites are provided for convenience only and do not serve as an endorsement of these websites. The materials at these websites are not part of the material for Pegasystems products, and use of those websites is at your own risk.

Information concerning non-Pegasystems products was obtained from the suppliers of those products, their publications, or other publicly available sources. Address questions about non-Pegasystems products to the suppliers of those products.

This publication may contain examples used in daily business operations that include the names of people, companies, products, and other third-party publications. Such examples are fictitious and any similarity to the names or other data used by an actual business enterprise or individual is coincidental.

This document is the property of:

Pegasystems
One Rogers Street
Cambridge, MA 02142-1209, USA
Phone: 617-374-9600 Fax: 617-374-9620

www.pegasystems.com

Document: Pega Platform installation

Publication date: November 10, 2021

Feedback

If you have comments for how we can improve our materials, send an email to DocRequest@Pega.com.

Contents

Planning your installation.....	5
Understanding Pega Platform split-schema architecture.....	5
Determining your transport-layer encryption method.....	5
Reviewing system requirements.....	6
Installation and Upgrade Assistant requirements.....	6
Application server requirements.....	6
Database server requirements.....	7
Storage and logging requirements.....	7
Configuring Java.....	7
Verifying time zones, character encoding, and regional settings.....	7
Configuring access to an external Cassandra database.....	7
Configuring access to Cassandra ports.....	8
Preparing your IBM-Db2 for z/OS database.....	9
Enabling support for user-defined functions.....	9
Reviewing ZPARM settings.....	9
Optional: Preparing to deploy to a UNICODE database.....	10
Configuring database users.....	10
Configuring deployment user permissions.....	10
Configuring runtime user permissions.....	10
Configuring access to the IBM Db2 for z/OS database.....	11
Creating storage groups and buffer pools.....	13
Configuring the database size.....	13
Installing Pega Platform.....	14
Extracting and validating the distribution image.....	14
Installing by using the Installation and Upgrade Assistant (IUA).....	14
Installing from the command line.....	16
Customizing deployment script behavior.....	17
Database connection properties and script arguments.....	18
Optional: Enabling Kerberos authentication.....	19
Using JCL for IBM Db2 for z/OS databases.....	20
Configuring your application server.....	22
Preparing to configure the application server.....	22
WAR file and EAR file considerations.....	23
Data source resources, data source entries, and default schema entries.....	23
Validating database connections.....	24
Configuring Hazelcast to run on Java 11.....	24
Defining default schema names for WebSphere.....	24
Configuring IBM WebSphere for IBM Db2 for z/OS.....	25
Configuring the application server by using the IBM WebSphere Administrative Console.....	26
Setting JVM properties.....	26
Creating URL providers and setting the temporary directory.....	27
Creating a JDBC provider.....	28
Creating a data source.....	28

Defining database authentication credentials.....	29
Configuring IBM WebSphere connection pool properties.....	30
Configuring the IBM WebSphere WorkManager.....	30
Setting the HTTP transport channel custom property.....	31
Configuring pass-by-reference.....	32
For systems with multiple VMs, or multiple NICs - setting the public and interface addresses.....	32
Deploying the Pega Platform file.....	32
Deploying Pega Platform.....	33
Post-installation tasks.....	35
Starting Pega Platform.....	35
Maintaining security by logging in and changing the administrator password.....	35
Enabling granting one-time access to external users by configuring Directed Web Access.....	36
Configuring search index host node settings.....	36
Configuring logging.....	37
Installing applications.....	37
Enabling server-side screen captures for application documents.....	38
Configuring PhantomJS REST server security for including screen captures in an application document.....	38
Enabling operators.....	39
Appendix A — Properties files.....	41
Appendix B — Troubleshooting.....	42
Recovering from a failed deployment.....	42
PEGA0055 alert — clocks not synchronized between nodes.....	42
ClassNotFoundException error — session persistence.....	42
System hangs with no error message — insufficient memory.....	43
Database connection information for IBM Db2 for z/OS.....	43
Manually generating and applying the DDL.....	44
Generating the DDL file.....	44
Applying the DDL file.....	44
Bypassing applying existing schemas to avoid deployment errors.....	45
Optional: Manually installing user-defined functions (UDFs) if you did not opt to automatically install UDFs.....	46

Planning your installation

Pega Platform supports several configuration options that can affect the choices that you make during the installation. Before beginning, read this section thoroughly.

Plan your architecture and configuration.

- Choose whether to use Kerberos functionality. Kerberos is a computer network authentication protocol that allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner. If you enable Kerberos authentication, you must use the command line method to deploy Pega Platform. For more information, see your installation guide.
- Review the [Platform Support Guide](#) before you install Pega Platform to verify that your database and application servers are supported.

Consider the following precautions before you continue:

- Do not change your environment while you are deploying Pega Platform. For example, if you are making changes to your application server or database server, do so before you deploy Pega Platform.

Understanding Pega Platform split-schema architecture

As a best practice, Pega Platform recommends a split-schema configuration where the rules and data objects reside on separate schemas.

With a split-schema configuration, you can upgrade one environment, and then migrate the upgraded objects to other environments.

In a split-schema configuration, Pega Platform uses the Java Naming and Directory Interface (JNDI) standard to identify and access the appropriate schema. One of the benefits of using JNDI is that it allows Pega Platform to access different schemas while using only a single data source.

If you plan to use a Pegasystems-supplied application and want to store any non-Pega-specific data in a separate schema, you can optionally configure a separate customer data schema in addition to the default Pega data schema.

Determining your transport-layer encryption method

Pega recommends that you use a strong transport-layer encryption method (for example, Transport Layer Security 1.2) to secure Pega Platform web applications. This encryption requires that you create and install transport-layer security and secure socket layer digital certificates on your application server for Pega Platform.

Before you continue, determine what transport-layer encryption method you will use. For more information, see the documentation for your application server.

Reviewing system requirements

Before you proceed, ensure that your system meets the following minimum requirements.

Installation and Upgrade Assistant requirements

If you plan to use the UI-based Installation and Upgrade Assistant, ensure that the system meets these minimum system requirements in addition to all other requirements.

- 1.25 GB minimum available memory
- 10 GB minimum disk space plus at least 8 GB available space in the temporary directory of the root file system. The default temporary directory for the deployment is `java.io.tmpdir`.
- Java Platform, Standard Edition Development Kit (JDK)

Application server requirements

Deploy only the Pega Platform web application (`prweb.war`) on the application server. The application server must meet the minimum requirements listed in the [Platform Support Guide](#).

- Supported 64-bit JDK. See the [Platform Support Guide](#)
 - IBM WebSphere Network Deployment requires that the deployment manager, the node agent, and the application servers are all on the same JDK version.
- 1 GB minimum free disk space. You might need additional storage space for debugging and logging. The upgrade process checks for 6400000 KB of free disk space and will not proceed without it, so 8 GB of free disk space is recommended.
- Memory requirements: Pega Platform runs in memory (heap) on Java Virtual Machines (JVMs). In general, all activity is distributed over multiple JVMs (nodes) on the application server.
 - Standard suggested system heap size is 4 - 8 GB based on monitoring of memory usage and garbage collection frequency.
 - Larger heaps are advisable if your applications allow a high number of concurrent open tasks per session or cache a large collection of transaction or reference data.
 - Do not deploy Pega Platform in an environment where the heap size exceeds the vendor-specific effectiveness limit.
 - IBM JDKs use compression to minimize the cost of large heaps. The compression option is labeled `CompressedRefs` and is effective up to 28 GB. In current 64-bit JVMs, compression is enabled by default.
 - The host application server memory size must be at least 4 GB larger than the Pega Platform heap size to allow space for the operating system, monitoring tools, operating system network file buffering, and JVM memory size (`-XX` option). The minimum host application server memory size is 8 GB:

4 GB heap + 4 GB for native memory, operating system, and buffering

If the server does not have enough memory allocated to run Pega Platform, the system can hang without an error message. The correct memory settings depend on your server hardware, the number of other applications, and the number of users on the server, and might be larger than these recommendations. Set `MaxMetaspaceSize` to a minimum of 768m to avoid a kernel out of memory crash or Metaspace size errors by using `-XX:MaxMetaspaceSize=768m`.

Database server requirements

Your database server must meet the minimum requirements listed in the *Platform Support Guide* on the Pega Community.

Verify that the system also includes support for SQL-based stored procedures. Pega 8.6 uses native stored procedures, not external stored procedures. This is not necessary for MS-SQL environments.

Storage and logging requirements

Before you configure, configure your system to manage log storage space.

- Allocate enough storage to accommodate debugging and other logging requirements.
- Configure logging to avoid writing logs to the directory that contains the application server run-time components.

Configuring Java

Before you install, configure the JAVA_HOME environment variable.

1. Set JAVA_HOME to the root directory of the JDK.
2. Remove from the PATH any references to a Java shortcut.

Verifying time zones, character encoding, and regional settings

Verify that your database server, application server, and the system on which you are deploying Pega Platform use the same:

- Time zone
- Character encoding (UNICODE or EBCDIC)
- Regional settings/locale

Configuring access to an external Cassandra database

If you use Pega Platform decision management capabilities, Pega Platform uses Cassandra as the underlying storage system for the Decision Data Store (DDS), which manages the Cassandra cluster and stores decision management data in a Cassandra database. Cassandra is a distributed, NoSQL open source database that stores your data across multiple machines and nodes and is highly available and scalable.

Future versions of Pega Platform will no longer support deployments on embedded Cassandra. In Pega Platform version 8.6, deployments using embedded Cassandra are deprecated but still work. To ensure future compatibility, do not create any new installations using embedded Cassandra.

For information about how to configure Pega Platform to access an external database, see [Defining Pega Platform access to an external Cassandra database](#).

Configuring access to Cassandra ports

While using embedded Cassandra is deprecated for new environments, if you need to set up existing environments, several ports must be open on the nodes in the cluster where Cassandra is running so that the nodes can communicate with each other.

Configure the firewall that is running on nodes in your Cassandra cluster so that these ports are open.

1. Ensure that the following ports are open to other members of the Cassandra cluster for their internode (Gossip) exchange:
 - 7000 - Internode communication (not used if TLS enabled)
 - 7001 - TLS internode communication (used if TLS enabled)
2. Ensure that port 9042, the CQL native transport port, is open to all the nodes in the Pega cluster.

Preparing your IBM-Db2 for z/OS database

Before you continue, prepare your IBM-Db2 for z/OS database for the deployment.

Enabling support for user-defined functions

To support user-defined functions (UDF) installed by Pega Platform, provide a dedicated Java-enabled Work Load Manager (WLM).

- Define the WLM application for Pega Platform to the system WLM with the MVS WLM utility tool.
- Ensure that the new MVS WLM application is part of the MVS WLM Policy that is in effect on the z/OS LPAR.

You will specify this WLM as the #WLMUDF property in the **DB2SiteDependent.properties** file found in the directory **Pega-image/scripts/config/db2zos**. You must refresh this WLM application after the UDFs have been loaded. You can do this from the z/OS console or from a tool like SDSF.

Reviewing ZPARM settings

Review the following ZPARM settings with your z/OS system programmer before you continue:

- CACHEDYN=YES
- CMTSTAT MAY BE INACTIVE OR ACTIVE
- DESCSTAT=YES. (Mandatory)
- IDBACK=200 (Minimum)
- IDFORE=200 (Minimum)
- IDTHION=300 (Minimum if CMTSTAT=INACTIVE)
- LOBVALA=10240 (minimum)
- LOBVALS=2048 (minimum)
- NUMLKTS=5000 (Minimum)
- NUMLKUS=30000 (Minimum)
- OPTIOWGT=ENABLE
- OPTIXOPREF=OFF
- OPTXQB=OFF (TEMPORARY A/O 5/09 DUE TO IBM APARS)
- STATCLUS=ENHANCED
- SKIPUNCI=YES

Optional: Preparing to deploy to a UNICODE database

If you are deploying to a UNICODE database, ensure that the appropriate code page translation is configured across all tools used on the platform, including FTP, TSO, IBM Data Studio, and SPUFI.

Configuring database users

Depending on your configuration, you must configure specific database users.

- Deployment user — The user who runs the deployment. After the deployment, you can remove this user.
- Run-time users: These users perform actions on Pega Platform after the deployment. In a dual-user configuration, an Admin user is granted full privileges, and a Base user is granted a smaller subset. Pega recommends the dual-user configuration:
 - Base user — The user who will run Pega Platform. For most run-time operations, Pega Platform uses the Base user and associated data source.
 - Admin user — An optional user provided to Pega Platform that is preferentially used by certain features that require creating, modifying, or dropping database schema objects including System Management facilities and certain decisioning operations.

Pega recommends that you create the Admin user separate from the Base user, however, you can create a single Base user with both sets of privileges. If there is no separate Admin user, Pega Platform attempts to use the Base user.



Note: The user names must be in all capital letters in the `DB2SiteDependent.properties` file.

Configuring deployment user permissions

The Pega Platform requires IBM Db2 for z/OS credentials to create the database and to access it once created. You provide the Deployment user name in the `DB2SiteDependent.properties` file. You need both the Deployment user name and associated password to configure data access for Pega Platform in your application server.

These credentials need to have permissions to perform the following actions:

- Log onto IBM's Time Sharing Option (TSO)
- Run z/OS batch jobs
- Allocate Datasets (both z/OS and zFS)
- Log onto IBM's UNIX System Services (USS)
- Change the OMVS (USS) memory size to 2 GB (RACF example `ASSIZEMAX 2147483647`)
- Grant database CREATE, DROP and ALTER authority (at the database level and for all objects within the database)
- Grant database INSERT, UPDATE and DELETE authority.
- Grant database access to create and execute stored procedures and user-defined functions.

Configuring runtime user permissions

For split-user configurations, you have two users, Base and Admin, with different permissions.

- Admin user — The database owner, `#DBOWNER`, with permissions to issue SQL CREATE, DROP and ALTER commands against the Pega Platform database instance.

- Base user — The database user, #DBUSER, with permissions to issue SQL INSERT, DELETE, UPDATE, and SELECT statements.


Configuring access to the IBM Db2 for z/OS database

To configure access to your IBM Db2 for z/OS database, edit the **Pega-image /scripts/config/db2zos/DB2SiteDependent.properties** file.

Review the following properties and edit the values as needed:

- #DBNAME is the database name assigned by your database administrator.
The value you select here will be substituted everywhere in the target libraries that the database expects a valid database name. There are no additional special requirements for this name other than that it must be acceptable to IBM Db2. The entire database must be able to be dropped and re-created. Avoid placing any non-Pegasystems related database objects in this schema.
- #DBOWNER is the database owner for the library copy of the Pega Platform database.
The deployment assumes that #DBOWNER has DBADM on #DBNAME either directly through a grant or indirectly by processing a Secondary Authorization ID. The application servers can log on to this database using #DBOWNER. The deployment creates a PLAN_TABLE owned by #DBOWNER.
- #DBUSER is the ID of the user who accesses the Pega Platform database. If your site strictly enforces segregation of duties, this userid can be used to run the applications. #DBUSER can only issue DML (select, insert, delete, update) commands while #DBOWNER can issue DDL as well as DML. Enter the user name in all capital letters, or the deployment fails.

By default, the Pega Platform installer takes the following actions:

- If you prefer to enable lock escalation, configure #BLBLCM and #TSPLCM to a non-zero positive integer before tailoring the libraries. Refer to your IBM Db2 for z/OS manuals for details on setting these properties.
 **Note:** If you change the #BLBLCM or #TSPLCM parameter to anything other than a non-zero positive integer, there may be a conflict when you try to create the tablespace. For example, if you set these parameters to SYSTEM, certain tables are designated as table locked which causes a conflict in allocation.
- Assign separate storage groups for table spaces, index spaces, and LOB table spaces.
To keep all three object types in a single storage group, set #BLBSTG, #IDXSTG, and #TSPSTG to the same value.
- Segregate each object type into its own buffer pool.
For the LOB buffer pools, set Deferred Write Threshold DWQT to zero. By default, Pega Platform defines all LOB table spaces as LOG YES.
- Use the #DBOWNER userid to connect to the database.

Variable with default value	Your choice	Description
LOB parameters		
#BLBBFP=BP32K1		Blob 32 K Blob buffer pool. This must be different from the table space.
#BLBLCK=LOB		Blob lock size

Variable with default value	Your choice	Description
#BLBLCM=0		Blob lock maximum
#BLBLOG=YES		Blob logging
#BLBPRI=14400		Blob primary allocation in pages
#BLBSEC=7200		Blob secondary allocation in pages
#BLBSTG=Blob.Storage.Group.Nam		Blob database storage group
Index parameters		
#IDXBP=BP2		Index buffer pool
#IDXPRI=7200		Index primary allocation in pages
#IDXSEC=7200		Index secondary allocation in pages
#IDXSTG=Index.Storage.Group.Name		Index database storage group
Tablespace parameters		
#TSPLCK=PAGE		Table space lock size
#TSPLCM=0		Table space lock maximum
#TSPPRI=14000		Table space primary allocation in pages
#TSPSEC=7200		Table space secondary allocation in pages
#TSPSTG=Table.Storage.Group.Name		Table space storage group
#TS32BP=BP32K		Table space 32K buffer pool. This must be different from the Blob buffer pool.
Database access parameters		
#CCSID=EBCDIC		EBCDIC or UNICODE
#DBNAME=PEGDB		Database name
#DBOWNR=PEGDBDBO		Database owner
#DBUSER=PEGDB		Pega Platform database user

Variable with default value	Your choice	Description
#WLMUDF		Name of the dedicated Java-enabled Work Load Manager (WLM) for UDFs

Creating storage groups and buffer pools

Ensure that the Storage Groups and Buffer pools selected for use by Pega Platform are created, and permission is granted to the #DBOWNER userid to use them.

Configuring the database size

Allocate a minimum of 5 GB to the database and allow the database to grow. This minimum size allows you to load the initial Pega Platform rulebase and do simple development.

Installing Pega Platform

Select the best installation method based on your specific environment.



Note: Install only Pega Platform on the application server.

- UI tool — The Installation and Upgrade Assistant is a Java-based UI tool that sets up the Pega Platform rules schema in the database and loads the Pega Platform rules.
- Command-line script — A command-line script automates the installation of Pega Platform in headless environments.
- JCL — JCL member scripts that generate the schema and apply the schema on the z/OS system.

Extracting and validating the distribution image

Follow these steps to extract and validate the distribution image:

1. Copy the compressed distribution image to the computer that you will use to run the installation. Extract the contents of the compressed file into an empty directory. If you are installing the software from a DVD, copy the contents of the DVD to an empty directory.
2. Verify the contents of the extracted distribution image.
The **Pega-image\checksum** directory provides an MD5 checksum for each the file in the distribution image. To verify that the files downloaded and uncompressed correctly, calculate a checksum using the Jacksum tool at www.jonelo.de/java/jacksum/. For example, if you uncompressed the distribution image to **PEGA** enter the following command: `java -jar jacksum.jar -m -a md5 -r -p -O outputFile.md5PEGA`
3. Compare **outputFile.md5** to the md5 file located in **Pega-image\checksum**. The checksum values should be identical.

What to do next: Choose the installation method:

- To use the IUA, continue at [Installing by using the Installation and Upgrade Assistant \(IUA\)](#).
- To use the command line tool, continue at [Customizing deployment script behavior](#).
- To use JCL for IBM Db2 for z/OS databases, continue at [Using JCL for IBM Db2 for z/OS databases](#).

Installing by using the Installation and Upgrade Assistant (IUA)


Because of the large volume of data, run the IUA on the same network as the database server. If this is not possible, run the tool on a system with fast, direct access to the database server. The Deployment user performs these steps.

Pega Platform writes command-line output to a file in the **Pega-image\scripts\logs** directory.

The process can last for several hours and the time can vary widely based on network proximity to the database server.


To run the IUA:

1. Double-click the **PRPC_Setup.jar** file to start the IUA.

 **Note:** If JAR files are not associated with Java commands on your system, start the IUA from the command line. Navigate to the directory containing the `PRPC_Setup.jar` file, and type `java -jar PRPC_Setup.jar`.

The IUA loads and the Pega icon is displayed in your task bar.

2. Click **Next** to display the license agreement.
3. Review the license agreement and click **Accept**.
4. On the **Installer Mode** screen, choose **Installation** and click **Next**.
5. Choose your database type and click **Next**.
6. Choose **Standard Edition** and click **Next**.
7. Configure the database connection. The JDBC drivers allow the Pega Platform application to communicate with the database.

 **Note:** Some of the fields on the **Database Connection** screen are pre-populated based on the type of database you selected. If you edit these or any other fields on this screen, and then later decide to change the database type, the IUA might not populate the fields correctly. If this occurs, enter the correct field values as documented below, or exit and rerun the IUA to select the intended database type.

- **JDBC Driver Class Name** – Verify that the pre-populated value is accurate:
`com.ibm.db2.jcc.DB2Driver`
- **JDBC Driver JAR Files** – Click **Select Jar** to browse to the appropriate driver files for your database type and version. Ensure that you use the correct drivers for your system. For a list of supported drivers, see the latest [Pega Platform support guide](#).

IBM Db2 for z/OS requires two JDBC JAR files to establish a connection: `db2jcc4.jar` and `db2jcc_license_cisuz.jar`.

- **Site Dependent Properties File** – Enter the full path of the site-dependent properties file or click **Select File** to browse and select the file.
- **Database JDBC URL** – Verify that the pre-populated value is accurate.

For information about URLs, see [Obtaining database connection information](#). Replace items in *italics* with the values for your system:

```
jdbc:db2://server:50000/database:fullyMaterializeLobData=true;
fullyMaterializeInputStreams=true;progressiveStreaming=2;
useJDBC4ColumnNameAndLabelSemantics=2;resultSetHoldability=1;
resultSetHoldabilityForCatalogQueries=1;
```

- **Database Username and Password** – Enter the user name and password that you created for the Deployment user on your database.
 - **Rules Schema Name** – Enter the name of the rules schema in the database.
 - **Data Schema Name** – Enter the name of the data schema in the database.
 - **Customer Data Schema Name** – Optional: Enter the name of the customer data schema if it is separate from the data schema.
8. Click **Test Connection**. If the connection is not successful, review your connection information, correct any errors, and retest. When the connection is successful, click **Next**.
 9. Optional: Specify whether you will have your database administrator manually apply the DDL changes to the schema. These changes include the user-defined functions (UDF) supplied by Pegasystems. By default, the tool generates and applies the schema changes to your database.

- To generate and apply the DDL outside the UI tool, select **Bypass Automatic DDL Application** and continue the deployment. After you complete the deployment, manually generate and apply the DDL and UDF. For more information, see [Optional: Generating and applying DDL](#) and [Optional: Installing user-defined functions](#).
- To have the tool automatically apply the DDL changes and the UDF, clear **Bypass Automatic DDL Application**.



Note: If you select **Bypass Automatic DDL Application**, you must manually apply the changes or the deployment is not successful. The deployment resolves after the DDL changes and the UDF are applied.

10. Enter the system name and production level and click **Next**:

- **System Name** – Enter the name of your Pega Platform system. To find the system name, navigate to **System > Settings > System Name**.
- **Production Level** – Enter a production level. The production level affects many security features of your system. Both the system name and production level can be changed after the system is running. Depending on the type of installation, choose:
 - 5 for a system that will be used in production
 - 4 for a preproduction system
 - 3 for a test system
 - 2 for a development system
 - 1 for an experimental system

Edit the production level from the App Explorer. Enter `Data-Admin-System` in the search field and select **SysAdmin > Class > Data-Admin-System** to open your system.

11. Click **Start** to begin loading the rulebase.

Logs display in the log window and are also stored in the **Pega-image \scripts\logs** directory. During the deployment, the log window might appear inactive when the IUA is processing larger files.

12. Click **Back** to return to the previous screen, and then click **Exit** to close the IUA.

What to do next: Determine the next step:

- If you opted to have the IUA automatically apply the schema changes, and you will not enable Kerberos authentication, configure the application server.
- If your database administrator will apply DDL manually, or if you will enable Kerberos authentication, continue at [Customizing deployment script behavior](#).

Installing from the command line

Because of the large volume of data, run the command-line script on the same network as the database server. If this is not possible, run the script on a system with fast, direct access to the database server.

The `install.bat` and `install.sh` scripts use the properties in the `setupDatabase.properties` file. To overwrite any property, pass command line arguments.

1. If you have not done so already, edit the `setupDatabase.properties` file.
 - a. Open the `setupDatabase.properties` file in the scripts directory of your distribution image:
Directories.distributionDirectory\scripts\setupDatabase.properties.

- b. Configure the connection properties. For more information about parameter values, see [Properties file parameters](#).
 - c. Set the initial administrator password. If you do not set this password before you install, the installation fails. The administrator must change this password after the first time they log in. For more information, see [Maintaining security by logging in and changing the administrator password](#).
`pega.admin.password=initial-admin-password`
 - d. Save and close the file.
2. Open a command prompt and navigate to the scripts directory.
 3. Type `install.bat` or `./install.sh` to run the script.

Installing the rulebase can take several hours, depending on the proximity of the database to the system running the installation script. When the installation is complete, you see a **BUILD SUCCESSFUL** message.

Pega Platform writes command-line output to a file in the `Pega-image\scripts\logs` directory.

What to do next: Now configure the application server.

Customizing deployment script behavior

Edit the `setupDatabase.properties` file to configure deployment scripts.

Skip this section if your deployment meets all the following criteria:

- You will use the Installation and Upgrade Assistant.
- You will allow the Installation and Upgrade Assistant to automatically apply the schema changes and do not need to create a DDL file.
- You will not use Kerberos authentication.

If your deployment does not meet all these criteria, follow the steps in this section to edit the `setupDatabase.properties` file. The `setupDatabase.properties` file controls scripts which perform the following tasks:

- The `install.bat` (for Windows) or `install.sh` (for Linux) script installs Pega Platform.
- The `generateddl.bat` or `generateddl.sh` script generates a DDL file of your schema.
 You can use the `generateddl` script regardless of whether you use the IUA or the command-line script.
- The `generateudf.bat` or `generateudf.sh` script generates user-defined functions.

1. Open the `setupDatabase.properties` file in the scripts directory of your distribution image:
`Directories.distributionDirectory\scripts\setupDatabase.properties`.
2. Specify the properties for your system. For each property, add the appropriate value after the equal sign. See [Database connection properties and script arguments](#).
3. Save and close the file.

Database connection properties and script arguments

The database connection properties in the `setupDatabase.properties` file specify the settings needed to connect to the database. The script arguments specify the same settings when you use command-line scripts. Command-line settings override property file settings.

Script argument	Property	Description
<code>--pega.jdbc.driver.jar</code>	<code>pega.jdbc.driver.jar</code>	Path and file name of the JDBC driver. IBM Db2 for z/OS requires the following JAR files: db2jcc4.jar and db2jcc_license_cisuz.jar
<code>--pega.jdbc.driver.class</code>	<code>pega.jdbc.driver.class</code>	Class of the JDBC driver
<code>--pega.database.type</code>	<code>pega.database.type</code>	Database vendor type. Enter: <code>db2zos</code>
<code>--pega.jdbc.url</code>	<code>pega.jdbc.url</code>	The database JDBC URL. For more information for IBM Db2 for z/OS, see Obtaining database connection information .
<code>--pega.jdbc.username</code>	<code>pega.jdbc.username</code>	User name of the Deployment user.
<code>--pega.jdbc.password</code>	<code>pega.jdbc.password</code>	Password of the Deployment user. For encrypted passwords, leave this blank.
<code>--pega.admin.password</code>	<code>pega.admin.password</code>	For new installations only. The initial password for <code>administrator@pega.com</code> . If you do not set this password before you install, the installation fails.
<code>-- jdbc.custom.connection.properties</code>	<code>jdbc.custom.connection.properties</code>	Optional: Semicolon-delimited list of custom JDBC properties. (for example: <code>prop1=value;prop2=value;prop3=value</code>)
<code>--rules.schema.name</code>	<code>rules.schema.name</code>	Sets the rules schema name.
<code>--data.schema.name</code>	<code>data.schema.name</code>	For split-schema configurations only, sets the data schema name.

Script argument	Property	Description
--customerdata.schema.name	customerdata.schema.name	An optional customer data schema separate from the default Pega data schema.
--user.temp.dir	user.temp.dir	Optional: The location of the temp directory. Set this location to any accessible location.
--pega.zos.properties	pega.zos.properties	Specify the full path and file name to the IBM Db2 for z/OS DB2SiteSpecific.properties file.

Optional: Enabling Kerberos authentication

Kerberos is a computer network authentication protocol that allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Skip this section if you do not want to enable Kerberos authentication.

To enable Kerberos for authentication, you must use the command line to deploy Pega Platform:

1. Open the **setupDatabase.properties** file in the scripts directory of your distribution image: *Directories.distributionDirectory\scripts\setupDatabase.properties*.

2. Add custom JDBC properties:

- a. In the **Custom Connections Properties** of the file, uncomment the custom property:

```
jdbc.custom.connection.properties
```

- b. Provide the correct parameters as semicolon-delimited name/value pairs. The specific parameters depend on your security infrastructure, for example:

```
jdbc.custom.connection.properties=
parameter1=value1;
parameter2=value2;
parameter3=value3;
```

3. Add custom JVM arguments:

- a. In the **Custom JVM Arguments** section of the file, uncomment the custom property:

```
custom.jvm.args=-Xmx4g
```

- b. Provide the correct parameters as a space-delimited list in the following format. The specific parameters depend on your security infrastructure but the max heap size must be set to a minimum of 4g.

```
custom.jvm.args=-Xmx4g jvm1 jvm2
```

4. Comment out all the user name and password properties so that they appear as follows:

```
# pega.jdbc.username db username
# pega.jdbc.password db password
[lines removed here]
# pega.jdbc.username=ADMIN
```

```
# pega.jdbc.password=ADMIN
```

5. Save and close the file.
6. Configure your database to enable Kerberos functionality. This might include additional vendor-specific JDBC driver configuration, or other setup procedures. For more information, see your database documentation.

What to do next: Continue at [Installing from the command line](#).

Using JCL for IBM Db2 for z/OS databases

Skip this section if you plan to use the Installation and Upgrade Assistant or the command line scripts to deploy. Follow these instructions to use JCL to install Pega Platform in IBM Db2 for z/OS environments.

Pega Platform includes JCL scripts to configure the z/OS environment to run the shell scripts and java programs packaged with Pega Platform. The JCL scripts use IBM batch z/OS launchers to run UNIX System Services in z/OS and JZOS (Java on z/OS).

The JCL scripts are in the library:

Pega-image/scripts/config/db2zos/libraries/PEGA.DB2.JCLLIB

The library includes the following members:

- SETUPZOS - This member includes:
 - Shell script set statements to establish the fully-qualified directory paths for the Java – 64 Bit System
 - JDBC Jar members
 - IBM's JZOS system
 - JDBC connection properties
 - Path to the Pega Platform distribution image
- SITEDIRS — Contains the JCL set statements used to establish the fully qualified zFS working directory, the fully qualified zFS directory to the Pega root directory and the MVS fully qualified library name of the Pega Platform JCL library that contains these members.
- PU01EXP — This job expands the Pega Platform distribution image into the zFS directory specified by the SITEDIRS script.
- PI01DDL — This job creates the SQL DDL statements required to deploy the database objects.
- PI01ENV — This is the environment script for the job PI01DDL. Do not modify this script.
- PI02SQL — This job applies the SQL DDL statements generated by PI01DDL creating the necessary tables, triggers, views, indexes and stored procedures required to deploy Pega Platform.
- PI02ENV — This is the environment script for the job PI02SQL. Do not modify this script.

The JCL procedure uses a combination of both MVS libraries and zFS members.

To follow this procedure, you need a user id with OMVS access, and 5 GB of available storage.

1. If you have not already done so, edit the properties file:


```
Pega-image/scripts/config/db2zos/DB2SiteDependent.properties.
```
2. Upload the Pega 8.6 distribution image to the z/OS server.
3. Sign into the z/OS server from an OMVS terminal.

4. Use either USS services or the ISPF panel to expand the distribution image.
5. Make your current directory the directory that contains the distribution image and run **PU01EXP**.
6. Allocate an MVS library to hold the library members:

Pega-image/scripts/config/db2zos/libraries/PEGA.DB2.JCLLIB



Note: Name the library **PEGA.DB2.JCLLIB** with the DCB attributes of FB/80/4000. Then, use ISPF 3.3 to copy the file into the library as members. If you use another library name, change the library name in each member to the new name.

7. Using the ISPF editor, review and edit the members **SETUPZOS** and **SITEDIRS** one at a time, updating the file paths. The instructions for editing these members are included in the comments. Remember to save the member each time before you submit the change.



Note: **SETUPZOS** contains DSN statements for the JCL. **SITEDIRS** contains pointers to members that the shell scripts require. These are provided as a common place to set these values.

8. Using the ISPF editor, review and edit the members **PI01DDL** and **PI02SQL** one at a time, updating the comments by the JCL. Remember to save the member each time before you submit the change.

9. Run **PI01DDL**.



Note: This member generates the database DDL statements that create the database. Review the output for accuracy.

10. Run **PI02SQL** to apply the DDL.

Pega Platform writes command-line output to a file in the **Pega-image/scripts/logs** directory.

Configuring your application server

Follow the instructions in this section to configure your application server.

- Ensure that you are following best practices and have installed only Pega Platform on the application server.
- Ensure that your application server meets the prerequisites listed in [Application server requirements](#) and in the *Platform Support Guide* on the Pega Community.
- Prepare and configure the application server.
- Deploy the Pega Platform applications.


Preparing to configure the application server

Complete these steps before you configure the application server:

1. Ensure that your operating system references a common time standard such as the one available at www.time.gov.
 - On UNIX, this is the Network Time Protocol daemon, `ntpd`.
 - On Windows, you can set a similar service through the clock settings in the Windows Control Panel or task bar.


See the documentation for your specific hardware operating system for information about setting this critical service.

2. Ensure that the following ports are open and available:
 - Search — One TCP port in the range 9300-9399 (the default is 9300). This port is used for internal node-to-node communication only, and should not be externally accessible.
 - Cluster communication — Leave open the port range 5701-5800. By default, the system begins with port 5701, and then looks for the next port in the sequence (5702, followed by 5703 and so on). To override the default port range, set a different value for the `initialization/cluster/ports` setting in the `prconfig.xml` file.

 **Note:** The number of available ports in this range must be greater than or equal to the greatest number of JVMs on any one node in the cluster. For example, if three JVMs are on one node, and seven JVMs on another node, at least seven ports must be available.

3. Obtain the following information from your database administrator to determine the database connection URL:
 - Host name
 - Port number

What to do next: Determine whether to deploy the WAR file or the EAR file. See [WAR file and EAR file considerations](#).

 **Note:** In Pega Platform version 8.6, EAR deployments are deprecated. Pega Platform version 8.7 will no longer support rules that require EAR deployments. For information about alternative rules that you can use, see [Deprecation of EAR deployments in Pega Platform 8.6](#).

Continue at [Data source resources, data source entries, and default schema entries](#).

WAR file and EAR file considerations

Pega Platform is available both as a WAR file, `prweb.war`, and an EAR file. Using the WAR file is the best practice for all new deployments. Use the EAR file only if you need Java Transaction API (JTA): Two-phased commits to the database.



Note: In Pega Platform version 8.6, EAR deployments are deprecated. Pega Platform version 8.7 will no longer support rules that require EAR deployments. For information about alternative rules that you can use, see [Deprecation of EAR deployments in Pega Platform 8.6](#).

In addition, although you can use the following features when Pega Platform is deployed as a WAR file into a non-JEE container, the correct Java libraries must be installed into the runtime Pega classpath. You must determine which provider libraries to install. To avoid instability when the wrong Java libraries are installed into the Pega classpath, it is a best practice to use an EAR deployment for the following features:

- Java Messaging Service (JMS): Pega connectors and services
- Enterprise Java Beans (EJB): Pega connectors and services
- Java Connector Architecture (Connect JCA)

The specific EAR file name depends on your application server.

Before you continue, determine whether you will use the WAR file or the EAR file.

Continue at [Data source resources, data source entries, and default schema entries](#).

Data source resources, data source entries, and default schema entries

The application server configuration defines the required data source resources, data source entries, and default schema entries:

- Data source resources — Data source resources define the Pega Platform database connection information. The number of data source resources depends on whether you have a single-user or dual-user configuration:
 - All systems require one data source resource for the Base user.
 - Dual-user configurations also require a second data source resource for the Admin user.
- Data source entries — Data source entries specify which data source resource to use for database operations in each schema. For dual-user environments, you must explicitly define two additional data source entries for the Admin user:
 - Admin data source entry for the rules schema
 - Admin data source entry for the data schema
- Default schema entries — Every system requires two entries that define the default schema names:
 - Default rules schema, for example, PegaRULES.
 - Default data schema, for example, PegaDATA.

Continue at [For systems with multiple VMs, or multiple NICs - setting the public and interface addresses](#).

Validating database connections

To avoid stale connections or closed connections being returned to the pool, work with your database administrator to validate the connections in your connection pool and configure the connection pool settings appropriately.

Configuring Hazelcast to run on Java 11

Pega Platform uses Hazelcast for scaling and processing application data. If you are deploying Pega Platform on Java 11 or later, then you need provide JVM arguments to allow Hazelcast to run in a modular environment.

For more information about running Hazelcast in modular Java, refer to the documentation for your instance of Hazelcast provided on the [Hazelcast](#) website.

1. Open the configuration file on the Hazelcast server and enter the following JVM arguments.

```
--add-exports java.base/jdk.internal.ref=ALL-UNNAMED
--add-opens java.base/java.lang=ALL-UNNAMED
--add-opens java.base/java.nio=ALL-UNNAMED
--add-opens java.base/sun.nio.ch=ALL-UNNAMED
--add-opens java.management/sun.management=ALL-UNNAMED
--add-opens jdk.management/
com.ibm.lang.management.internal=ALL-UNNAMED
--add-opens jdk.management/
com.sun.management.internal=ALL-UNNAMED
```

2. Save and close the configuration file.

Defining default schema names for WebSphere

Create binding identifiers to define the default values for the rules schema and the data schema.

1. In the IBM WebSphere Administrative Console, select `Environment > Naming > Name Space Bindings` to display the **Name space bindings** page.
2. Create the rules schema binding identifier:
 - a. For the **Scope**, select **server**, and click **New**.
 - b. For the binding type, select **String** and click **Next**.
 - c. On the **Step 2: Specify basic properties** screen, enter the following values:
 - Binding identifier: `PegaRULESDefaultSchema`
 - Name in the name space relative to lookup name prefix: `prconfig/database/databases/PegaRULES/defaultSchema`
 - String Value: the schema name of your rules schema.
 - d. Click **Next**.
 - e. On the **Summary** panel, click **Finish**.
 - f. Click **Save** in the **Messages** box at the top of the **Name Space Bindings** screen.
3. Repeat step 2 to create the data schema binding identifier, but specify the following properties on the **Step 2: Specify basic properties** screen:

- Binding identifier: PegaDATADefaultSchema
 - Name in the name space relative to lookup name prefix: prconfig/database/databases/PegaDATA/defaultSchema
 - String Value: the schema name of your data schema
4. Optional: For dual-user configurations, repeat step 2 to add a binding identifier for the Admin user on the data schema. Specify the following properties on the **Step 2: Specify basic properties** screen:
 - Binding identifier: PegaDATADataSourceAdmin
 - Name in the name space relative to lookup name prefix: prconfig/database/databases/PegaDATA/dataSourceAdmin
 - String Value: the JNDI name of the Admin data source for your data schema
 5. Optional: For dual-user configurations, repeat step 2 to add a binding identifier for the Admin user on the rules schema. Specify the following properties on the **Step 2: Specify basic properties** screen:
 - Binding identifier: PegaRULESDataSourceAdmin
 - Name in the name space relative to lookup name prefix: prconfig/database/databases/PegaRULES/dataSourceAdmin
 - String Value: the JNDI name of the Admin data source for your rules schema
 6. Repeat step 2 to create the customer data schema binding identifier, but specify the following properties on the **Step 2: Specify basic properties** screen:
 - Binding identifier: PegaCustomerdataDefaultSchema
 - Name in the name space relative to lookup name prefix: prconfig/database/databases/CustomerData/defaultSchema
 - String Value: the JNDI name of the Admin data source for your customer data schema
 7. Click **Save** in the **Messages** box at the top of the **Name Space Bindings** screen.

Configuring IBM WebSphere for IBM Db2 for z/OS

Follow the IBM Washington System Centers process to implement your IBM WebSphere application server environments, Base Application Servers, Network Deployment Servers, Clustered Servers, or additional configurations available for IBM Db2 for z/OS.

Pegasystems follows the methods documented in the IBM WebSphere for z/OS Configuration Planning Spreadsheets for testing and development. The following recommendations are based on the IBM Washington Systems Center Technical Documentation website:

<http://www-03.ibm.com/support/techdocs/atmastr.nsf/Web/TechDocs>
<http://www-03.ibm.com/support/techdocs/atmastr.nsf/Web/TechDocs>

1. For your first configuration, build a Standalone Server. Use these IBM WebSphere for z/OS Configuration Planning Spreadsheets from IBM:
 - IBM WebSphere 7: PRS3341
 - IBM WebSphere 8: PRS4686
 - IBM WebSphere 8.5: PRS4944
2. For subsequent configurations, build Network Deployment cells. See these IBM white papers:
 - Top Down Configuration Approach to WAS on z/OS: WP101030
 - WAS z/OS v6-WSC Sample ND Configuration – WP100653.

- WAS z/OS v6-WSC Sample ND Configuration – WP100653.

Configuring the application server by using the IBM WebSphere Administrative Console

Start the server for the new profile, launch the IBM WebSphere Administrative Console, and log in. Use a browser to launch the IBM WebSphere Administrative Console, and log in.

Setting JVM properties

Follow these steps to set the JVM properties.

- 1.
2. In the IBM WebSphere Administrative Console, on the left side of the screen, click **Servers > Server Types > WebSphere application servers**.
3. Select the server on which the Pega Platform will run. The **Configuration** tab for the server opens.
4. In the **Server Infrastructure** section, expand **Java and Process Management** and click **Process Definition**. The Configuration tab opens.
5. Select **Servant**.
6. In the **Additional Properties** section, click **Java Virtual Machine** to display its Configuration tab.
7. Select **Verbose garbage collection**.
8. Set the JVM memory options to increase the amount of system memory allocated to the application server running the Pega Platform:
 - Initial Heap Size (Xms) — Between 4 GB - 8 GB, based on monitoring of memory usage and garbage collection frequency
 - Maximum Heap Size (Xmx) — Between 4 GB - 8 GB or larger, depending on your system configuration. For more information, see [Application server requirements](#).
 - MaxMetaspaceSize — Set to a minimum of 768m to avoid a kernel out of memory crash or Metaspace size errors.

If the server does not have enough memory allocated to run Pega Platform, the system can hang without an error message. The correct memory settings depend on your server hardware, the number of other applications, and the number of users on the server, and might be larger than these recommendations. Set MaxMetaspaceSize to a minimum of 768m to avoid a kernel out of memory crash or Metaspace size errors by using `-XX:MaxMetaspaceSize=768m`.
9. In the **Generic JVM Arguments** field, enter `-Xgcpolicy:optavgpause` to set the garbage collection policy to concurrent collector.
10. Enter the following argument to enable AWT for graphical reports:
`-Djava.awt.headless=true`
11. Enter the following arguments to set the temporary directory:
`-Dpega.tmpdir=/temporary directory path>`
12. If you use either UNIX or Linux, enter the following argument to set security to urandom:
`-Djava.security.egd=file:///dev/urandom`
13. If you want to enable CyberArk password vault support, enter the following argument.

```
-Dcom.pegas.prminiloader.avoiddbforgetresource=true
```

14. Set the node type with the JVM argument: `-DNodeType=<node type>`.

To support queue processing, Pega Platform requires at least one stream node (node type `Universal` or `Stream`). In a single node cluster, set the node type to `Universal`, otherwise set the node type to `Stream`; for example, `-DNodeType=Stream`.

If you have a high availability environment, configure at least two stream nodes by using the JVM argument: `-DNodeType=Stream`.

15. To enable editing of the default Kafka parameters, in the `prconfig.xml` file, enter the JVM argument: `-D pegarules.config=<USS_directory>/prconfig.xml`
16. Click **Apply**. A message is displayed at the top of the screen explaining that the changes were made.
17. Click **Save** in the confirmation message at the top of the page to save these changes to the master configuration.

Creating URL providers and setting the temporary directory

This procedure sets the required URL Pega reference and an explicit temporary directory for Pega Platform. The temporary directory stores static data. It is important that the directory be properly specified and accessible to the JVM user. In IBM WebSphere, specify this directory as a JNDI reference to a URL object.

1. In the IBM WebSphere Administrative Console, select **Resources > URL > URL Providers** in the left frame.
2. Set the Scope level to **server** and click `Default URL Provider`. The default URL configuration page opens.
3. Under the **Additional Properties** section, click the **URLs** link to display the URLs listing page.
4. Click **New** to display the **Configuration** page.
5. Enter the following values to define a URL for the NULL file that the Pega Platform uses to discard erroneous error messages:
 - Name — `PRPCnone`
 - JNDI name — `url/pega/none`
 - Specification — `file:/nul`
6. Click **OK**, and then click **Save** in the confirmation message.
7. From the URLs page, click **New** again to return to the Configuration page.
8. Complete this form to create a URL specification for a temporary directory to store static data.
 - **Name** — `PegaTempDir`
 - **JNDI** — `url/initialization/explicittempdir`
 - **Specification** — `file:/// full-path-to-temporary directory`



Note: The directory names are case sensitive. Be sure to enter the names of the directories exactly as they have been created on your system.



Note: If the directory you specify does not exist, Pega Platform attempts to allocate it. It is good practice to allocate the directory on the system before specifying it here. The user that owns the Java process must have the appropriate permissions to use this directory, including write access. In particular, if you have J2 security enabled, ensure that this directory is accessible under your security policy.



Note: In a clustered deployment, each instance must have its own temporary directory. You cannot share a temporary directory with more than one instance of the Pega Platform.

9. Click **OK**, and then click **Save** in the confirmation message.

Creating a JDBC provider

In the IBM WebSphere Administrative Console, click **Resources > JDBC > JDBC Providers** to display the JDBC Providers page.

1. In the Scopes list, select **server**.
2. Click **New** to display the JDBC provider wizard.
3. Complete the fields as follows:
 - **Database** — DB2
 - **Provider type** — DB2 Universal JDBC Driver Provider
 - **Implementation** — Connection pool data source
 - In the **Name** and **Description** fields, accept the defaults or enter more descriptive information.
4. Click **Next**.
5. In the directory location field, confirm the entry or enter the path to the JDBC driver JAR file, db2jcc4.jar, listed in the class path field. See the [Platform Support Guide](#) for more information about supported drivers.
6. Click **Next** to display the Summary screen.
7. Confirm that the settings are correct, and click **Finish** to return to the JDBC Providers page.
8. Click **Save** in the confirmation message.

Creating a data source

Follow these steps to create the data source for the Base user. If you are using the dual-user configuration, repeat these steps to create a data source for the Admin user.

1. In the IBM WebSphere Administrative Console, click **Resources > JDBC > JDBC Providers** to display the JDBC Providers page.
2. On the JDBC providers page, click on the name of the provider you just created to display the **General Properties** page.
3. Under the Additional Properties heading, click **Data Sources**.
4. Click **New** to display the data source wizard.
5. In the Step 1, **Enter basic datasource information** screen, enter the following information:
 - In the **Data Source name** field, enter the correct name for this data source:
 - For the Base user, enter `PegaRULES`.
 - For the Admin user, enter `AdminPegaRULES`.
 - In the **JNDI name** field, enter the correct name:
 - For the Base user, enter `jdbc/PegaRULES`.
 - For the Admin user, enter `jdbc/AdminPegaRULES`.



Note: JNDI settings are case-sensitive.

6. Click **Next** to display the Step 2, **Enter database specific properties for the data source** screen.
7. Enter the connection information for your database.

```
jdbc:db2://server:port/database
```
8. Clear **Use this data source in container managed persistence (CMP)**.
9. Click **Next** to display the **Setup security aliases** page.
10. Click **Next** to display the Summary page.
11. On the **Summary** page, confirm that the settings are correct and click **Finish** to return to the **JDBC Providers** page.
12. Click **Save** in the confirmation message.
13. On the **Data Sources** page, click the datasource link in the **Name** column to open the **Configuration** page for this data source. Then, under **Additional Properties**, click the **Custom Properties** link to display the Custom Properties page.
14. Click **New** to define additional properties for your database connection.

After creating each property, click **OK** to save the property, and click **New** again to create the next property. Set the **Type** field appropriately for the **Value** of the connection property.

The Pega Platform requires the following property; if the property already exists, modify the value if needed:

- Name: **currentFunctionPath** Value: **SYSIBM,SYSFUN**, *<data schema name>*



Note: SYSIBM and SYSFUN are system schemas provided by the database. They are required because the Pega Platform uses some functions made available in these schemas.

- Name: **currentSchema** Value: *<data schema name>*
- Name: **currentSQLID** Value: *<user name>*



Note: This value must be no more than 8 characters long.

- Name: **fullyMaterializeInputStreams** Value: **true**
- Name: **fullyMaterializeLobData** Value: **true**
- Name: **progressiveStreaming** Value: **2**
- Name: **useJDBC4ColumnNameAndLabelSemantics** Value= **2**
- Name: **webSphereDefaultIsolationLevel** Value: **2**

15. After you have set the necessary properties, click the **Save** link in the message at the top of the page.
16. Optional: For dual-user configurations, repeat these steps to create the AdminPegaRULES data source.

Defining database authentication credentials

Follow these steps to define the database authentication credentials for the Base user. For dual-user configurations, repeat these steps to define credentials for the Admin user.

1. In the link path at the top of the IBM WebSphere Administrative Console, click the name of the data source, either **PegaRULES** or **AdminPegaRULES** to return to the properties page.
2. Under the **Related Items** section, click the link **JAAS – J2C authentication data**.
3. Click **New** to specify the General Properties.
4. Complete this form as follows:
 - In the **Alias** field enter any name that uniquely identifies this J2C entry.

- In the **User ID and Password** fields, enter the user name and password:
 - For the PegaRULES data source, enter the credentials for the Base user.
 - For the AdminPegaRULES data source, enter the credentials for the Admin user.
- 5. Click **OK** to return to the authentication data entries page, and click **Save** in the Messages section at the top of the page.
- 6. In the link path at the top of the page, click the name of the data source to return to the properties page.
- 7. In the **Security Settings** section near the bottom of the page, use the **Component-managed authentication alias** menu to select the J2C alias you just created.
- 8. Click **OK**, and then **Save** in the confirmation message on the Data sources page.
- 9. On the Data sources page, select the check box for the data source and click **Test Connection** to confirm your data source configuration.
- 10. Optional: For dual-user configurations, repeat these steps for AdminPegaRULES.

Configuring IBM WebSphere connection pool properties

Follow these steps to configure the properties for the PegaRULES data source. For dual-user configurations, repeat these steps for the AdminPegaRULES data source.

Determine the best value of this setting based on your application architecture, usage profile and environment considerations. The database connection pool should be no smaller than the Work Manager pool. At a minimum, set the maximum data connections to 180.

To set the maximum connections:

1. In the IBM WebSphere Administrative Console, open the Data sources page: **Resources > JDBC > Data sources**.
2. Click the name of the data source, either **PegaRULES** or **AdminPegaRULES**.
3. Under **Additional Properties**, click **Connection pool properties**.
4. Set the **Connection timeout** value to 180.
5. Set the **Maximum connections** value to 180 or higher based on your environment needs. See the Pega Community article [How to configure a non-blocking UI using Asynchronous Declare Pages](#) for more information about connections.
6. Set the **Minimum connections** value to 20.
7. Set **Purge policy** to **Entire pool**.
8. Click **Apply**.
9. Click **Save** in the Messages pane at the top of the screen to save the configuration changes.
10. Optional: For dual-user configurations, repeat these steps for the AdminPegaRULES data source.

Configuring the IBM WebSphere WorkManager

To deploy the Pega Platform archive, complete the following procedures to define a WorkManager. The Pega Platform uses the WorkManager to run asynchronous tasks to support internal components such as services, daemons, and child-requesters.

Complete the following steps to define the WorkManager:

1. In the navigation menus on the left side of the IBM WebSphere Administrative Console, select `Resources > Asynchronous Beans > Work managers`.

2. Set the Scope to `server`.
3. Click **New**.
4. Complete the Configuration page for the Work Manager.
 - Name — Work Manager name, for example, `PegaWorkManager`
 - JNDI name — `wm/PegaWorkManager`
 - Service — `Security`
 - Maximum number of threads — `20`
 - Clear the Growable check box.
5. Click **OK**.
6. In the Messages box at the top of the page, click **Save**.

Setting the HTTP transport channel custom property

To support the ability to open files attached to work objects in the Pega Platform, set custom property **CookiesConfigureNoCache** to false on the transport chain in the Web Container settings for the Pega Platform server.

For earlier versions of IBM WebSphere, custom property **CookiesConfigureNoCache** was set to false by default so that cookies could be cached. The current version of IBM WebSphere has this property set to true by default. Because the Pega Platform requires this caching to allow users to open attached files directly from a work object, you must change the property setting.

1. In the IBM WebSphere Administrative Console, select **Servers > Server Types > WebSphere application servers** to display the Application servers page.
2. Click the name of the Pega Platform server to display the Configuration page.
3. Under Container Settings, expand **Web Container Settings**, and then click **Web Container transport chains**.
4. Click the name of the appropriate transport chain.
 - If your site is using the default transport configuration for this server, select **WCInboundDefault**.
 - If you have enabled SSL for the Pega Platform port, **WCInboundDefaultSecure**.
 - If your site has defined a custom transport chain for this server, select that chain name.
5. Under Transport Channels on this page, click **HTTP inbound channel**.
6. Under Additional Properties, click **Custom Properties**.
7. Click **New**.
8. Complete this form.
 - a. In the **Name** field, enter `CookiesConfigureNoCache`.
 - b. In the **Value** field, enter `False`.
9. Click **OK**, and then **Save** in the Messages box on the top of the Custom Properties listing page.

Configuring pass-by-reference

To improve performance, enable pass by reference in the WebSphere application server Object Request Broker services.

1. In the IBM WebSphere Administrative Console, select **Servers > Server Types > WebSphere application servers** to display the **Application servers** page.
2. Click the name of your Pega Platform server to display the **Configuration** page.
3. Under Container Settings, expand **Container Services**, and then click **ORB service**.
4. Select **Pass by Reference**.
5. Click **OK**.
6. Click **Save** in the Messages box at the top of the **Application servers** page to save the setting.

For systems with multiple VMs, or multiple NICs - setting the public and interface addresses

Pega Platform uses Hazelcast distributed clustering technology to share data and send events between server nodes. If the cluster uses separate virtual machines (VMs), or multiple network interfaces (NICs), set the public and interface addresses in the `prconfig.xml` file for each Pega Platform node.

1. Open the `prconfig.xml` configuration file in the `prweb/WEB-INF/classes` subdirectory of the application server directory. For more information, see [Changing node settings by modifying the prconfig.xml file](#).

2. Modify the `prconfig.xml` file. Add the following setting to set the public address:

```
<env name=" identification/cluster/public/address" value=" IP address " />
```

For example, if the IP address of the node on which you run the Pega Platform node is 10.254.34.210, add the following setting:

```
<env name=" identification/cluster/public/address" value="10.254.34.210" />
```

The new setting controls the address that is used by the Pega Platform node.

3. Specify the IP address of the node that Hazelcast uses to communicate with its cluster members. Use only one IP address. Add the following setting:

```
<env name=" prconfig/cluster/hazelcast/interface" value=" IP address " />
```

For example, if the IP address of the node is 10.254.34.210, add the following setting:

```
<env name=" prconfig/cluster/hazelcast/interface" value="10.254.34.210" />
```

4. Repeat steps 1 to 3 for the remaining nodes.
5. Save and close the `prconfig.xml` file.

Deploying the Pega Platform file

After you configure your application server, you must deploy the `prweb.war` or `prpc_j2ee14.ear` file.

For more information about deploying the Pega Platform, see the following topics.

Using the WAR file is the best practice for all new deployments. Use the EAR file only if you need one of the EAR-only features. For more information, see [WAR file and EAR file considerations](#).



Note: In Pega Platform version 8.6, EAR deployments are deprecated. Pega Platform version 8.7 will no longer support rules that require EAR deployments. For information about alternative rules that you can use, see [Deprecation of EAR deployments in Pega Platform 8.6](#).

If you deploy and start the application before creating the database, the application generates an error and fails to start. This error is not harmful, and you can restart the application successfully when the database is available.

Deploying Pega Platform

Deploy the Pega Platform application using the `prweb.war` or `prpc_j2ee14_ws.ear` file included in your distribution image.

The application server starts the application when it deploys. When the application starts, you might see error messages for missing resources and references. Ignore these messages; you supply these resources as you deploy. Stop the application after deploying.

1. Make sure the application server is running. Log in to the IBM WebSphere Administration Console.
2. From the left frame of the IBM WebSphere Administrative Console, select **Applications > New Application**.
3. Click **New Enterprise Application**.
4. Click **Browse** and select `prweb.war` from the archives directory.
5. Click **Open**, and then click **Next**.
6. Select **Detailed - Show me all installation options and parameters**.
This option allows you to review all the deployment options for the application, including the default bindings and resource mappings.
7. Click **+** to expand **Choose to generate default bindings and mappings**.
8. Complete this page.
 - Check **Generate Default Bindings**.
 - Check **Use default virtual host name for Web and SIP modules**.
 - Leave the other default settings unchanged, and click **Next**.
9. Scroll to the bottom on this page and click **Continue** to display a wizard where you can specify deployment options.
This security file allows the Pega Platform to run when Java EE Security Checking is enabled.
This section of the deployment is a series of steps under the general heading of **Install New Application**.
10. For Step One, accept the defaults and click **Next**.
11. Continue through the next steps, either accepting the defaults, or customizing for your organization, as needed.
12. In the **Map context roots for Web Modules** step, enter `prweb` as the context root, and click **Next**.
13. Locate the step where you **Map resource references to resources**.
14. In the **Map resource references to resources** step, there are three rows that include "explicittempdir" in the Resource Reference column. Use the find tool on your browser to find the correct rows for:
 - **EJB EngineCMT bean**
 - **EngineBMT beans**
 - **prweb.war module**

15. For each of the three rows, change the value in the **Target Resource JNDI Name** field to the temp directory, for example **url/initialization/explicittempdir**.

This maps the location you specified in the URL provider you created to the corresponding Resource Reference in the application, so that the application will use the location for the **PegaTempDir**. Use the Browse button and Apply to change each of the three values.

16. Click **Next**.

Depending on your configuration, you might see a set of warnings related to missing resource references. These warnings are informational. Review the warnings, and then continue.



Note: These are resource references that are defined in **web.xml**, the deployment configuration files for the application, but not mapped to resource definitions in your application. In the page, **Map resources to references**, they are mapped to the Target Resource JNDI Name **url/pega/none**, indicating that they are not used. Pega provides these references for Java EE compliance, but their use is optional. You can continue with the deployment.

17. At the bottom of the Warnings page, click **Continue**.
18. Click **Next** as needed to continue through the remaining steps, accepting the defaults, or setting them to the requirements of your organization.
19. On the **Summary** page, click **Finish**.

The system begins deploying the EAR file, which can take a few minutes. When the deployment completes successfully, WebSphere displays a success message similar to the following: "Application Pega Platform installed successfully."
20. Click **Save directly to the master configuration**.
21. Stop the application.

Post-installation tasks

This section describes the post-deployment activities that are performed in the system after you have completed the setup and configuration of your application server and deployed the archives. All post-installation tasks are required.

- [Starting applications](#)
- [Logging in to the Pega Platform and changing your password](#)
- Prior to Pega 7.3.1, you needed to configure the database name for schema import. Starting in Pega 7.3.1, installations and upgrades automatically set the database name for PegaDATA and PegaRULES. If you have an additional external database, see the [Database data instances](#) for instructions on setting the database name.
- [Configuring Directed Web Access](#)
- [Configuring search index host node settings](#)
- [Configuring logging](#)
- [Monitoring database size](#)
- [Installing custom applications](#)
- [Enabling server-side screen captures for application documents](#)
- [Enabling operators](#)

Starting Pega Platform

Ensure that the application server is running and start prweb.

Maintaining security by logging in and changing the administrator password

To test the deployment and index the rules, log in to Pega Platform web application. For security, you must change the administrator password.

1. Navigate to the PRServlet URL, replacing the *server* and *port* values with your specific values.

`http://server:port/prweb`

2. Use the following credentials to log in the first time:

- User ID — `administrator@pega.com`
- Password — the password you set when you installed Pega Platform.

After logging in, Pega Platform indexes the rules in the system to support full-text search. During the index process, there might be a delay in the responsiveness of Pega Platform user interface. The process usually takes from 10 to 15 minutes to complete depending on your system configuration.

If the index process ends without generating an error message, the deployment is successful.

3. Immediately after the index process completes, change the administrator password. The new password must be at least 10 characters long.

If the system does not prompt you to change your password, follow these steps:

- a. From the **Operator Menu**, select the **Profile**.

- b. Click **Change Password**.
- c. Verify the **Current Password**, and then enter and confirm the **New Password**.
- d. Click **Save**.

Enabling granting one-time access to external users by configuring Directed Web Access

A Directed Web Access (DWA) address allows you to grant one-time access to external users to enable them to process an assignment in your application. When you grant the access, the Pega Platform sends an email to the external user; this email includes a URL to access the application and can identify a proxy server.

Follow these instructions to configure the URL:


1. In the header of Dev Studio, click **Configure > System > Settings > URLs**.
2. In the **Public Link URL** field, enter the URL that you want to provide in emails in this format:
`http://host:port/prweb`
3. Click **Save**.
4. Log out and log back in to Dev Studio for these changes to take effect.

Configuring search index host node settings

The Pega Platform supports full-text search for rules, data instances, and work objects. By default, search indexing is enabled and indexing starts when you start the application server after deploying the Pega Platform. The first node that starts after the deployment becomes the default initial search node. The default index directory is PegaSearchIndex in your temporary directory.

After the search indexes are completely built, you can change the default settings. Do not stop or bring down the default node until the search indexes build completely. The Search Landing Page displays the status.

Follow these steps to configure the search index host node settings:

1. Check your directory sizes. Ensure that the directories for all Elasticsearch host nodes have sufficient free space to hold the Elasticsearch indexes.
2. In the header of Dev Studio, click **Configure > System > Settings > Search**.
3. Expand **Search Index Host Node Setting**.
4. Specify one node to set as the Host Node. If necessary, delete all but one node. This is the node on which Elasticsearch indexes will be built.
 **Note:** Do not include more than one node in the **Search Index Host Node Setting** list. Including more than one node in the list at this point might cause duplicate index builds and compromise system performance. You will create additional nodes later in this process.
5. Verify the **Search Index Host Node ID** and the **Search Index File Directory**.
6. Expand **Automated Search Alerts**, and enable **Automatically Monitor Files**.
7. Click **Submit** to save the settings.
8. After the first indexing is complete, add any needed additional host nodes. The system replicates the indexes on the new nodes.

**Note:**

- Configure a minimum of two Elasticsearch host nodes. Pegasystems recommends that you configure a minimum of three nodes for maximum fault tolerance; however, you might need more than three nodes, so estimate and configure the appropriate nodes based on the size of your data and your business needs.
 - Pegasystems recommends to use `-Dindex.directory JVM` argument for each node. For more information about configuring index host nodes, see [Configuring index host nodes](#).
9. To enable communication between Elasticsearch host nodes in the cluster, open a TCP port in the range 9300-9399 on each node. (By default, Elasticsearch uses port 9300.) These ports are used for internal node-to-node communication only, and should not be externally accessible. Ensure that these ports are not subject to an idle connection timeout policy in the software or hardware that runs between these host nodes.

Configuring logging


To customize what is logged for installations, upgrades, and the prpcUtils tool, customize the template `prlog4j2.xml` file. To customize the processes that use java logging, edit the `deploylogging.properties` file.

Work with your system administrator to configure logging.

1. To configure the details of what is logged in the Apache log files, open `scripts/config/prlog4j2.xml` and update the details of what is logged for installations, upgrades and prpcUtils actions.
For more information, see [Apache Log4j2](#).
2. To configure what is logged by the installation and upgrade processes that use Java logging, edit the `scripts/config/deploylogging.properties` file.
3. **Optional:** Increase the size of the log files. The specific size will depend on your environment and the size of your application.
The initial log file size is 250 MB.

Installing applications

Install any applications now. If you obtained your application from Pega, follow the instructions in the Installation Guide for your application.

-  **Caution:** Grant the database user permissions as described in [Configuring database users](#). Some applications use triggers. During startup, Pega Platform checks for triggers that reference the upgrade cache and rule view tables; if these triggers exist, Pega Platform attempts to drop them. If the user does not have the correct permissions, Pega Platform cannot drop the triggers and fails to start up.

If you installed the applications before you deployed Pega Platform, Pega Platform automatically drops the triggers and this error does not occur.

Enabling server-side screen captures for application documents

To avoid client-side limitations, such as browser incompatibilities or client software requirements, set up a Tomcat server to support taking and storing screen captures on a server rather than on a client. Set up this Tomcat server regardless of your application server platform.

As a best practice, virtually install Tomcat and deploy the `prScreenShot.war` file on the same server that is running Pega Platform. Otherwise, use a standalone Linux or Windows server. If you use a Linux server, you must include the following components:

- fontconfig
- freetype
- libfreetype.so.6
- libfontconfig.so.1
- libstdc++.so.6

You can include screen captures in an application document that is generated by the Document Application tool. Screen captures provide stakeholders with a realistic picture of an application's user interface. Install a PhantomJS REST server to include screen captures in an application document.

1. Download the following WAR file: `Pega_DistributionImage\Additional_Products\PhantomJS\prScreenShot.war`

2. Deploy the WAR file on a Tomcat server.

3. Edit the `tomcat-users.xml` file to add the following role and user. This file is located at `\apache-tomcat-XX\conf\tomcat-users.xml`.

```
<role rolename="pegascreencapture" /> <user username="restUser" password="rules"
roles="pegascreencapture" />
```

4. Start the Tomcat server. The service is hosted at `http://IPAddress:port/prScreenShot/rest/capture`, where `IPAddress` is the address of the system where Tomcat is hosted, and `port` is the port on which the service is deployed.

5. Log in to your Pega Platform application and make the following changes:

- a. Edit the Data-Admin-System-Setting instance *Pega-AppDefinition - CaptureScreenshotsResourcePath* with the URL of the service, for example, `http://10.224.232.91:8080/prScreenShot/rest/capture`.
- b. Add the user that you created in step 3 to the Data-Admin-Security-Authentication profile instance `CaptureScreenshotsAuthProfile`.

What to do next: Continue at [Configuring PhantomJS REST server security for including screen captures in an application document](#).

Configuring PhantomJS REST server security for including screen captures in an application document

To ensure a secure installation of Pega Platform, enable the PhantomJS REST server to take and store server-side screen captures. In application documents generated by the Document Application tool, screen captures provide stakeholders with a realistic picture of the application's user interface.

1. Obtain the SSL certificate from the Pega Platform administrator.
2. Add the SSL certificate to the list of trusted certificates:
 - a. Double-click the certificate.
 - b. Click **Install certificate** to start the **Certificate Import** wizard.
 - c. Click **Next**, and select **Place all certificates in the following store**.
 - d. Click **Browse**, select **Trusted Root certificate**, and click **OK**.
 - e. Click **Next**, and then click **Finish** to complete the wizard.
3. Add the certificate to the truststore of the JVM on which the REST server is installed:
 - a. Open a command prompt.
 - b. Change the root directory to the security folder in the Java installation folder.
 - c. Run the following command:

```
keytool -keystore cacerts -importcert -alias certificate alias -file certificate name
```
 - d. When prompted, enter the password for the cacerts keystore. The default password is `changeit`.

Enabling operators

Pega Platform deployment security requires an administrator to enable new operators shipped with Pega Platform and requires password changes after the first login.

The administrator and new operators shipped with Pega Platform must change their passwords when they first log in:

- Administrator@pega.com
 - AESRemoteUser
 - DatabaseAdmin@pega.com
 - External
 - ExternalInviteUser
 - IntSampleUser
 - PRPC_SOAPOper
 - PortalUser@pega.com
 - UIServiceManager
 - UVUser@pega.com
1. In the header of Dev Studio, click **Configure > Org & Security > Authentication > Operator Access**.
 2. In the **Disabled operators** list, click the link for the Pega-provided operator that you want to enable. The following standard operators are installed but disabled by default. When these standard operators first log on, they are required to change their passwords. Enable only those operators you plan to use.
 3. On the **Edit Operator ID** page, on the **Security** tab, select **Force password change on next login** and clear **Disable Operator**.
 4. Select **Update password**.
 5. Enter a password that conforms to your site standards and click **Submit**.
 6. Click **Save** and close the operator page.

7. Repeat steps 2 through 6 for the remaining operators.

Appendix A — Properties files

The Pega Platform properties files include several database-specific properties.

- JDBC driver JAR file — path to your current `db2jcc4.jar` and `db2jcc_license_cisuz.jar` files
- Database driver class — `com.ibm.db2.jcc.DB2Driver`
- Database vendor type — `db2zos`
- JDBC URL — `url="jdbc:db2:// host:port/dbname`

Appendix B — Troubleshooting

Use the information in this section to troubleshoot errors. The error logs are displayed in the Installation and Upgrade Assistant window and are also stored in the **Pega-image** \scripts\logs directory.

Recovering from a failed deployment

If the deployment fails, follow these steps to drop the schemas and start a new installation:

1. Review the log files in the \scripts\logs directory.
2. Make any necessary changes to your system. If the error was due to a data entry mistake, make note of the correct information.
3. Generate the DDL files and drop the schemas:
 - a. Verify the settings in the `setupDatabase.properties` file. For information about the properties, see [Editing the setupDatabase.properties file](#).
 - b. At a command prompt, navigate to the **Pega-image** \scripts directory.
 - c. Run the `generateddl.bat` or `generateddl.sh` script with the `--action=drops` option, for example:


```
generateddl.bat --action=drops
```
 - d. Review the DDL files in the **Pega-image** \schema\generated\output directory.
 - e. Have your database administrator apply the DDL to drop the schemas.
4. Repeat the installation steps.

PEGA0055 alert — clocks not synchronized between nodes

The Pega Platform validates time synchronization to ensure proper operations and displays a PEGA0055 alert if clocks are not synchronized between nodes.

For information about how to reference a common time standard, see the documentation for your operating system.

ClassNotFoundException error — session persistence

During application server shutdown, Tomcat persists session information into the `session.ser` file in the server file directory. When the application server restarts, it reloads the session information from the `session.ser` file and deletes the file. If serialized session objects refer to classes that are not visible to the container layer, you see a `ClassNotFoundException` error.

This is a sample error message:

```
May 19, 2016 2:37:46 PM org.apache.catalina.session.StandardManager
doLoad SEVERE: ClassNotFoundException while loading persisted sessions:
java.lang.ClassNotFoundException:com.pegap.pegarules.session.internal.authorization.ContextM
java.lang.ClassNotFoundException:
com.pegap.pegarules.session.internal.authorization.ContextMapDiagCallback
```

To suppress these errors, turn off Tomcat session persistence in the `context.xml` file.

System hangs with no error message — insufficient memory

If the server does not have enough memory allocated to run the Pega Platform, the system can hang without an error message. The correct memory settings depend on your server hardware, the number of other applications, and the number of users on the server, and might be larger than the minimum recommendations in [System requirements](#).

Database connection information for IBM Db2 for z/OS

When you configure the data source resources, you need the correct database connection URL. To determine the database connection URL, obtain the following information from your database administrator:

- Host name
- Port number

When you configure the application server, enter the connection string, `pega.jdbc.url`. Replace items in *italics* with the values for your system:

```
jdbc:db2://server:port/database
```

Manually generating and applying the DDL

If you chose not to automatically apply the DDL, then you must generate and apply the DDL manually.

The process for generating and applying DDL differs depending on whether you are performing an out-of-place upgrade or an in-place upgrade.

Generating the DDL file

Follow these steps to generate a DDL file for your database administrator to apply manually.

1. Edit the site dependent properties. See [Configuring access to the IBM Db2 for z/OS database](#).
2. Edit the `setupDatabase.properties` file.
 - a. Configure the connection properties. The customer data schema is optional.

```
# Connection Information
pega.jdbc.driver.jar=\path-to-the-database-JAR-file\DRIVER.jar
pega.jdbc.driver.class=database driver class
pega.database.type=database vendor type
pega.jdbc.url=URL of the database
pega.jdbc.username=Deployment username
pega.jdbc.password=password
rules.schema.name=rules-schema-name
data.schema.name=data-schema-name
customerdata.schema.name=optional-customer-data-schema
```

- b. Save and close the file.
3. At a command prompt, navigate to the **Pega-image** \scripts directory.
4. Run `generateddl.bat` or `generateddl.sh` and pass in the required `--action` argument:

```
#generateddl.bat --action install
```

If you do not specify an output directory, the script writes the output to the default directory: **Pega-image\schema\generated**



Note: The output directory is deleted and re-created each time the generateddl script runs. To save a copy of the DDL, rename the directory before you run the script.

Applying the DDL file

Before you continue, have your database administrator follow these general steps to apply the schema changes; these schema changes can include changes to user-defined functions:

Review the DDL file in the output directory and make any necessary changes.

The default directory is: **Pega-image\schema\generated\database\db2zos**

The output directory is deleted and re-created each time the generateddl script runs. To save a copy of the DDL, rename the directory before you rerun the script.

Bypassing applying existing schemas to avoid deployment errors

After your database administrator applies the changes to your database, configure the **setupDatabase.properties** file to bypass applying a schema that already exists. Reapplying an existing schema would cause the deployment to fail.

1. Open the **setupDatabase.properties** file in the scripts directory of your distribution image:
Directories.distributionDirectory\scripts\setupDatabase.properties.
2. Set the property `bypass.pegaschema=true`.
3. Save and close the file.

Optional: Manually installing user-defined functions (UDFs) if you did not opt to automatically install UDFs

The user-defined functions (UDFs) enable the Pega Platform to read data directly from the BLOB without creating and exposing columns. Skip this section if you installed the UDFs when you deployed Pega Platform.

There are several ways you might have bypassed generating and installing the UDFs when you deployed:

- Setting either `bypass.pegaschema=true` or `bypass.udfgeneration=true` in the `setupDatabase.properties` file
- Setting `pega.target.bypass.udf=true` in the `migrateSystem.properties` file
- Selecting **Bypass Automatic DDL Application** from the Installation and Upgrade Assistant

Before you install the UDFs, verify that you have the appropriate user permissions.

For more information about user permissions, see [Database users](#).

1. Edit the `setupDatabase.properties` file.
 - a. Configure the connection properties.

```
# Connection Information
pega.jdbc.driver.jar=\path-to-the-database-JAR-file\DRIVER.jar
pega.jdbc.driver.class=database driver class
pega.database.type=database vendor type
pega.jdbc.url=URL of the database
pega.jdbc.username=Deployment user name
pega.jdbc.password=password
rules.schema.name= rules-schema-name
data.schema.name=data-schema-name
```

- b. Save and close the file.

2. On the rules schema, run the following commands to remove any partially installed UDFs:

```
DROP FUNCTION rules-schema-name.pr_read_from_stream;
DROP FUNCTION rules-schema-name.pr_read_decimal_from_stream;
DROP FUNCTION rules-schema-name.pr_read_int_from_stream;
```

3. Optional: If you have a split-schema, on the data schema run the following commands::

```
DROP FUNCTION data-schema-name.pr_read_from_stream;
DROP FUNCTION data-schema-name.pr_read_decimal_from_stream;
DROP FUNCTION data-schema-name.pr_read_int_from_stream;
```

4. From the **Pega-image** \scripts directory, run the `generateudf.bat` (for Windows) or `generateudf.sh` (for Linux) script with the `--action install` argument.