Upgrading to ThingWorx 8.4

Version 1.1

Document Revision History

Revision Date	Version	Description of Change
February 2019	1.1	Updated matrix.
January 2019	1.0	Initial version for 8.4.0.

Upgrading ThingWorx

Upgrading to ThingWorx 8.4

This guide contains steps for upgrading to a newer version of ThingWorx. If you are installing ThingWorx for the first time, refer to the Installing ThingWorx guide.

Determine Upgrade Path: In-place vs. Migration

When upgrading to a newer version of ThingWorx, administrators have two options for getting new features and enhancements into existing landscapes: in-place upgrades and migrations. Generally, for in-place upgrades, you do not need to delete the **ThingworxStorage** and **ThingworxBackupStorage** folders or import data/entities after installing ThingWorx, and is applicable when upgrading to the same persistence provider. Migrating includes more steps, since you must export data and entities to the **ThingworxStorage** folder and then import those entities and data into the new version of ThingWorx.

Note: As of v.8.4.0. of ThingWorx platform, Neo4j and Neo4j+DSE are no longer supported configurations.

	Upgrading To ThingWorx					
Upgrading From ThingWorx		**8.4 on AzureSQL**	**8.4 on PostgreSQL**	**8.4 on PostgreSQL with DSE**	**8.4 on H2**	**8.4 on MS SQL Server**
	6.0, 6.5, 6.6, 7.0, 7.1, 7.2, 7.3, 7.4, 8.0, 8.1, 8.2, 8.3 on Neo4j	--	Migration	Migration + DSE Setup	Migration [4]	Not supported
	6.0, 6.5, 6.6, 7.0, 7.1, 7.2, 7.3, 7,4, 8.0, 8.1, 8.2, 8.3 on Neo4j with DSE[1]	--	--	Migration	--	Not supported
	6.5, 6.6, 7.0, 7.1, 7.2, 7.3, 7.4, 8.0, 8.1, 8.2, 8.3 on PostgreSQL with DSE[1]	--	--	In Place[2,3]	--	Not supported
	6.5, 6.6, 7.0, 7.1, 7.2, 7.3, 7.4, 8.0, 8.1, 8.2, 8.3 on PostgreSQL	Not supported	In Place[2,3]	In Place[2,3] + DSE Setup	--	Not supported
	7.2, 7.3, 7.4, 8.0, 8.1, 8.2, 8.3 on H2	Not supported	Migration	Migration	In-place	Not supported
	7.4, 8.0, 8.1, 8.2, 8.3 on MSSQL	Migration	Migration	Migration	Migration	In-place

[1] Also referred to as ThingWorx Enterprise Edition.

[2]: Can upgrade in place, upgrade optional for parallel implementation if desired.

[3]: Extensions need to be reimported.

4: Exported applications built on Neo4j architecture might experience performance issues on H2. Upgrading to PostgreSQL-based versions for larger applications that have high data volumes should be investigated instead.

Database Options: PostgreSQL, Microsoft SQL Server, InfluxDB, AzureSQL, or H2

With ThingWorx 8.4, you can use PostgreSQL (with an optional High Availability layer), Microsoft SQL Server, InfluxDB, AzureSQL, or H2 for your data solution. The following download package options are available when obtaining the **Thingworx.war** file from PTC Software Downloads:

- H2: **Thingworx-Platform-H2-8.4.0**
- PostgreSQL/HA/InfluxDB: **Thingworx-Platform-Postgres-8.4.0**
- AzureSQL: **Thingworx-Platform-Azuresql-8.4.0**
- Microsoft SQL Server/InfluxDB: **Thingworx-Platform-mssql-8.4.0**

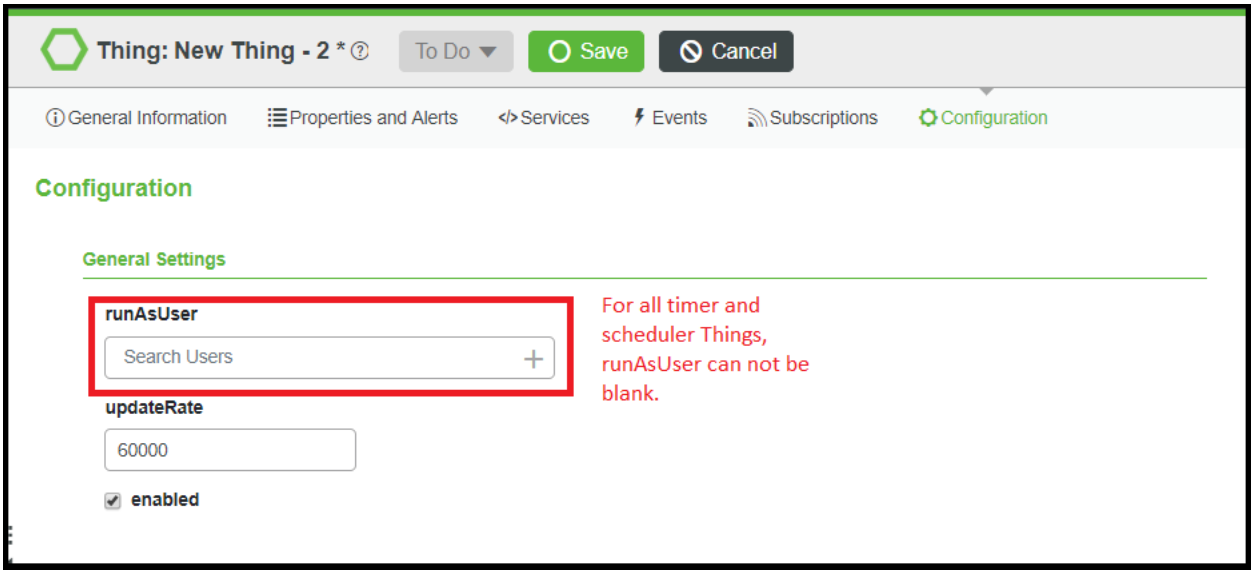
Note: As of v.8.4.0.x of ThingWorx platform, Neo4j and Neo4j+DSE are no longer supported configurations.

Action Required before you Upgrade to 8.4

Changes were made in 8.4 that may potentially affect any application that uses Timer or Scheduler Things. If your application/extension contains Timers or Schedulers, you must verify that the runAsUser setting is not blank before upgrading. If your Timers and Schedulers are set as extension entities with editable configuration tables, you must perform a platform restart after the RunAsUser has been set. See Thing Templates for additional information.

The following error will be seen in the Application Log if you attempt to import extensions that contain an empty runAsUser field:

```
Thing State is being set to ERROR because it failed during the
initialization phase: The runAsUser field was empty!
```



Extension Import Changes in 8.4

The ability to import extensions is disabled for all users in 8.4 by default. To enable extension imports, the following must be added and configured in the **platform-settings.json** file. See the Importing Extensions topic in the Help Center for more information.

```
"ExtensionPackageImportPolicy": {
            "importEnabled": <true or false>,
            "allowJarResources": <true or false>,
            "allowJavascriptResources": <true or false>,
            "allowCSSResources": <true or false>,
            "allowJSONResources": <true or false>,
            "allowWebAppResources": <true or false>,
            "allowEntities": <true or false>,
            "allowExtensibleEntities": <true or false>,
        },
```

H2 Database Password Changes in 8.4

If you are using H2 as a database with ThingWorx, a password is now required.

The following parameters must be added and configured in the **platform-settings.json** file. See platform-settings.json Configuration Details for more information.

```
},
"PersistenceProviderPackageConfigs":{
"H2PersistenceProviderPackage":{
"ConnectionInformation":
      { "password": "<changeme>",
      "username": "twadmin"}
},
```

Maintenance Release/Service Pack Upgrade Process

Upgrading to a maintenance release/service pack (for example, 8.4.**0** to 8.4.**1** or 8.4.**1** to 8.4.**2**) follows the following simplified in-place upgrade path. Upgrade scripts are not required for these upgrades. NOTE: This assumes you are upgrading to the same database.

1. Obtain the new **Thingworx.war**.
2. Stop Tomcat.
3. Go to the Tomcat installation at **\Apache Software Foundation\Tomcat 8.5\webapps** and delete the **Thingworx.war** file and the **Thingworx** folder.
4. Place the new **Thingworx.war** file in **\Apache Software Foundation\Tomcat 8.5\webapps**.
5. Start Tomcat.

Before You Begin

- Before upgrading, be sure to back up your model and runtime data.

- If you have localization tables with locales that are not in the `language-Script-REGION-variant` format, you should rename them before migrating by doing the following:
  1. From the ThingWorx Explorer, select **System** ▸ **Localization Tables**.
     A list of localization tables appears.
  2. Select a localization table with a non-conforming name (for example, *French*).
  3. Choose **Duplicate**.
  4. Enter the standard locale name (in this example for French, enter *fr*).
  5. Choose **Save**.
  6. Delete the original localization table with the non-conforming name.
  7. Repeat the above steps for all tables with names that do not conform to the standard.

  After importing the localization tables into your new system, do the following:
  1. From the ThingWorx Explorer, select **System** ▸ **Localization Tables**.
  2. Select a localization table to edit.
  3. Enter values in the **Language Name (Native)** field (for example, *français*) and **Language Name (Common)** field (for example, enter *French*).
  4. If you want to choose or change an avatar (such as a national flag), click Change next to the Avatar field.
  5. The avatar appears in the language preferences editor.
  6. Repeat these steps for all imported localization tables.

Migrating to ThingWorx 8.4 on Windows

Refer to the table above to determine your upgrade path. The steps below are for migration only. For an in-place upgrade, refer to In-place Upgrade to ThingWorx 8.4 on Windows.

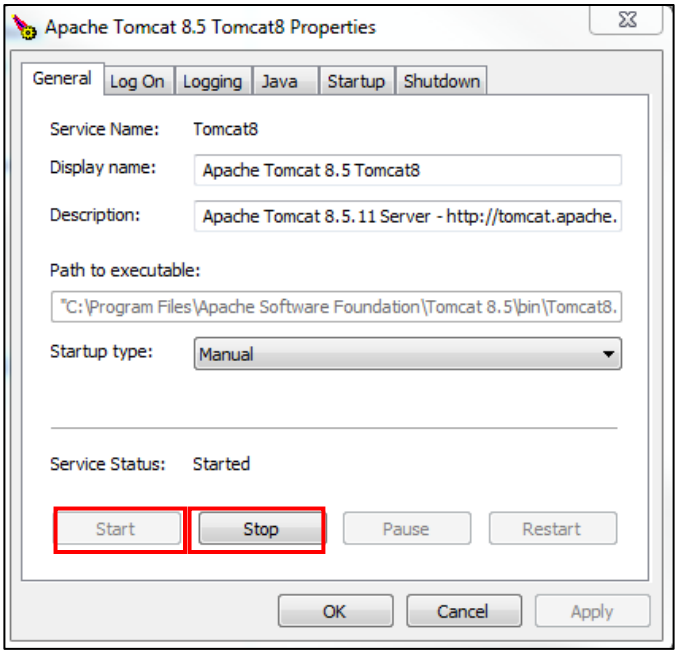
1. Obtain the latest version of ThingWorx.
   NOTE: ThingWorx downloads are available in PTC Software Downloads.

2. Verify that you are running the required versions of Tomcat and Java.

   NOTE: Refer to the System Requirements and Compatibility Matrix document for version requirements.

3. Tomcat Java option settings may have changed between versions. Refer to the Apache Tomcat Java Option Settings Appendix of the Installation Guide to verify that your settings are correct.

4. Restart Tomcat: In the Tomcat Properties, click **Stop**. Wait for Tomcat to stop.



5. It is highly recommended to back up the following two folders before continuing:
   - **Apache Software Foundation\Tomcat 8.5\webapps\Thingworx**
   - ***<drive>*:\\ThingworxStorage**

   Where *<drive>* is the drive Tomcat is installed on.

6. Click **Start**.

   Restarting Tomcat assures that the database is clear before exporting.

7. Export entities and data. In Composer, click **Import/Export>Export>To ThingworxStorage**.
8. If necessary, click **Include Data**.
9. Click **Export**.

   NOTE: Data and entities are exported to **ThingworxStorage\exports**

   NOTE: Data export progress can be monitored in the Application Log.

10. Copy these data and entity export files and move to a safe location. You will import these files in a later step.

11. Locate the **keystore.jks** file in the **ThingworxStorage** folder and move it to a safe place. You will add this file back to the **ThingworxStorage** folder later.

12. Note any extensions that are in use. They will be reimported in a later step.

13. Stop Tomcat.

14. Delete the contents of the **ThingworxStorage** and **ThingworxBackupStorage** folders.

15. Go to the Tomcat installation at **\Apache Software Foundation\Tomcat 8.5\webapps** and delete the **Thingworx.war** file.

16. Delete the **Thingworx** folder located at **Apache Software Foundation\Tomcat 8.5\webapps**

17. Copy the **Thingworx.war** file and place it in the following location of your Tomcat installation:
    \Apache Software Foundation\Tomcat 8.5\webapps

18. Verify licensing:
    a. Rename the existing **license.bin** file located in the **ThingworxPlatform** folder.
       NOTE: You can delete the file, but if login is unsuccessful, it will need to be recovered.
    b. Verify that your PTC support site **username,** and **password** are added to the **platform-settings.json** in the **PlatformSettingsConfig** section.

```
"LicensingConnectionSettings":{
      "username":"PTC Support site user name",
      "password":"PTC Support site password",
      “timeout”:”60”
      }
```

       NOTE: If the settings are filled out incorrectly or if the server can’t connect, a License Request text file (**licenseRequestFile.txt**) is created in the **ThingworxPlatform** folder. In this scenario, a license must be created manually. (If it is not created, ThingWorx will start in limited mode. Limited mode does not allow you to persist licensed entities to the database. Licensed entities are Things, Mashups, Masters, Gadgets, Users, and Persistence Providers).

       Further information on obtaining a ThingWorx disconnected site license through our License Management site can be found in the Licensing Guide for disconnected sites (no connection to PTC Support portal).

       Open a case with Technical Support if you are doing the manual disconnected mode of licensing and have any questions or need assistance with generating a license.

19. Configure the Administrator password. Add the following to the **platform-settings.json** file along with a password that is at least 10 characters long to the **PlatformSettingsConfig** section. Reference Appendix A: platform-settings.json Configuration Details for more information on placement.

```
{
    "PlatformSettingsConfig": {
        "AdministratorUserSettings": {
            "InitialPassword": "changeme"
        }
    }
}
```

20. Enable extension import.

    NOTE: By default, extension import is disabled for all users.

    Add the following to the **platform-settings.json** file. Add or update the following **ExtensionPackageImportPolicy** parameters to **true** to allow extensions to be imported. See the Help Center for best practices on configuration.

```
"ExtensionPackageImportPolicy": {
            "importEnabled": <true or false>,
            "allowJarResources": <true or false>,
            "allowJavascriptResources": <true or false>,
            "allowCSSResources": <true or false>,
            "allowJSONResources": <true or false>,
            "allowWebAppResources": <true or false>,
            "allowEntities": <true or false>,
            "allowExtensibleEntities": <true or false>
        },
```

21. If you are using H2 as a database with ThingWorx, a username and password must be added to the **platform-settings.json** file.

    NOTE: Skip this step if you are not using H2.

    Add and configure the following parameters to the **platform-settings.json** file. See platform-settings.json Configuration Details for more information.

```
},
"PersistenceProviderPackageConfigs":{
"H2PersistenceProviderPackage":{
"ConnectionInformation":
{
      "password": "<changeme>",
      "username": "twadmin"
}
},
```

22. Navigate to the **ThingWorxStorage** folder and delete the **keystore.jks** file. Locate the **keystore.jks** file that you previously moved from the **ThingworxStorage** folder and paste it in the folder.

23. Start Tomcat.

    NOTE: If Tomcat fails to start and reports the error message: “Check the InitialPassword setting in the AdministratorUserSettings section in platform-settings.json. Password must be a minimum of 10 characters”, check the following:
    - The password setting exists in platform-settings.json
    - The password is valid (10 or more characters)
    - The platform-settings.json file is formatted correctly - bad formatting could lead to errors. Reference Passwords for more information on formatting.

24. To launch ThingWorx, go to **<servername>\Thingworx** in a web browser.

    Use the following login information:

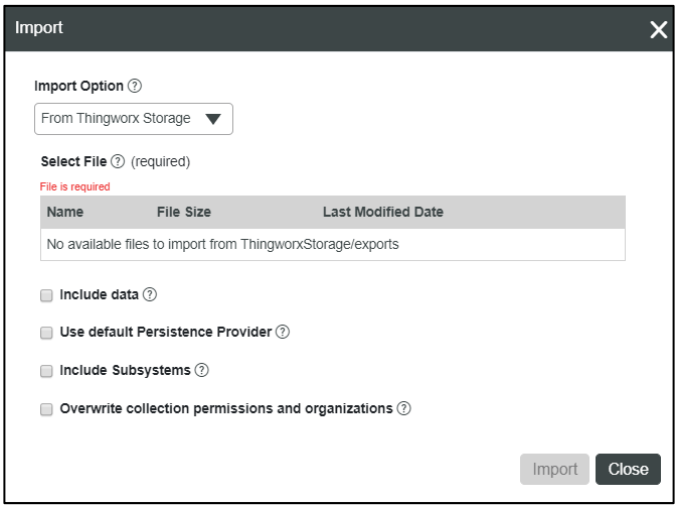
    Login Name: Administrator
    Password: *<password as defined in previous step>*

25. Move the export files back to **ThingworxStorage\exports.**

26. If necessary, import any extensions.

    NOTE: Obtain and import the latest versions of the extensions. If you are upgrading to a major version (for example, from 7.x to 8.x, you must import the 8.x versions of the extensions.) Extensions are available in the PTC Marketplace.

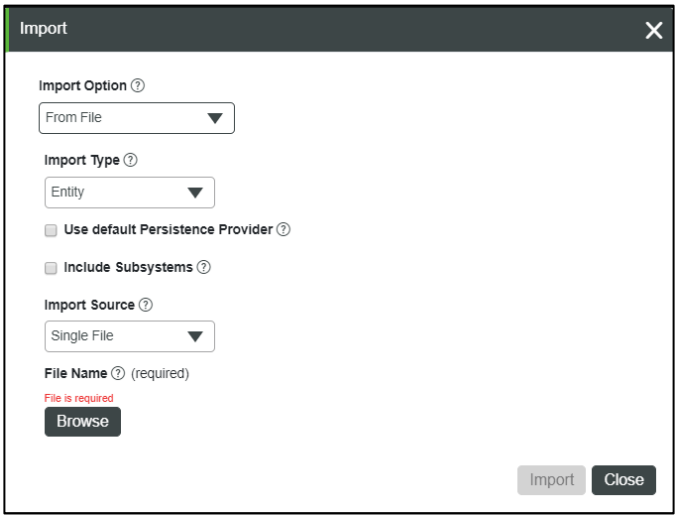
    NOTE: If you are importing **From ThingworxStorage**, you can select the **Overwrite Collection Permissions and Organizations** option, so that the collection permissions and organizations in the import will overwrite the settings on the server with the collection permissions and organizations contained in the import. If unchecked, the default behavior merges the collection permissions and organizations from the import into what is already defined on the server.



27. Import entities and data. In Composer, click **Import/Export> From File**.

28. Select the data and/or entities to import.
    a. Select the **Use Default Persistence Provider** check box if your data/entities were exported from 6.0.

    b. Select the **Include Subsystems** checkbox if you want to include the Subsystem settings of the imported entities (for example, if you are going from a test environment to production).



In-place Upgrade to ThingWorx 8.4 on Windows

Refer to the table at the beginning of this document to determine your upgrade path. The steps below are for in-place upgrades only. For migration steps, go to Migrating to ThingWorx 8.4 on Windows.

1. Obtain the latest version of ThingWorx.

   NOTE: ThingWorx downloads are available in PTC Software Downloads.

2. Verify that you are running the required versions of Tomcat and Java.

   NOTE: Refer to the System Requirements and Compatibility Matrix document for version requirements.

3. Tomcat Java option settings may have changed between versions. Refer to the Apache Tomcat Java Option Settings Appendix of the Installation Guide to verify that your settings are correct.

4. Stop Tomcat.

5. It is highly recommended to backup the following two folders before continuing:
   a. **Apache Software Foundation\Tomcat 8.5\webapps\Thingworx**
   b. **<*drive*>:\\ThingworxStorage**
   Where <*drive*> is the drive Tomcat is installed on.

6. Remove the **validation.properties** file from **\ThingworxStorage\esapi**

   NOTE: If this file is not deleted, exports will fail.

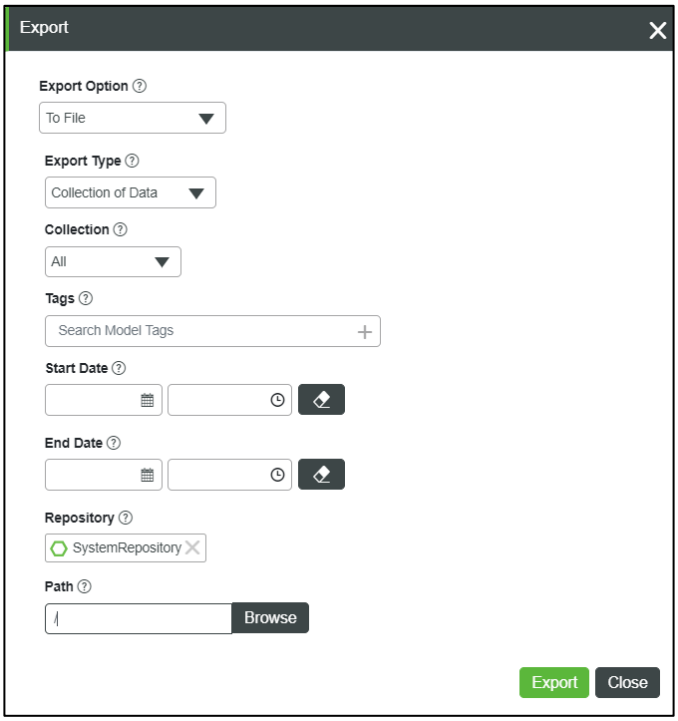
   NOTE: If you have custom configurations in the file, move the file for later reference.

   NOTE: The **validation.properties** file is created upon startup of ThingWorx. If you do not remove the file, the updated file with additional parameters will not overwrite the current version during upgrade. Reference the Help Center for additional information.

7. **This step is for DataStax Enterprise (DSE) ONLY. If you are not using DSE, skip and go to the next step.**

   If you have any data located in Neo4j and you are persisting any blog, wiki, stream, value stream, or data table data to DSE, do not export all data to the **ThingworxStorage** folder (to prevent duplicating blog, wiki, stream, value stream, or data table data when imported). Instead, you must explicitly export Neo4j data to file.

   NOTE: Data export progress can be monitored in the Application Log.



8. Export wiki data.

   NOTE: Perform this step only if you have wiki data**.**

9. Stop Tomcat.

10. Go to the Tomcat installation at **\Apache Software Foundation\Tomcat 8.5\webapps** and delete the **Thingworx.war** file and the **Thingworx** folder.

11. **THIS STEP IS FOR POSTGRESQL ONLY. SKIP AND PROCEED TO THE NEXT STEP IF YOU ARE NOT UPGRADING FROM POSTGRESQL.**

    Run the following scripts that are located in the **update** folder (starting with the version you are upgrading from):

    - **thingworxPostgresSchemaUpdate6.5-to-6.6.bat**
    - **thingworxPostgresSchemaUpdate6.6-to-7.0.bat**
    - **thingworxPostgresSchemaUpdate7.0-to-7.1.bat**
    - **thingworxPostgresSchemaUpdate7.1-to-7.2.bat**
    - **thingworxPostgresSchemaUpdate7.2-to-7.3.bat**
    - **thingworxPostgresSchemaUpdate7.3-to-7.4.bat**
    - **thingworxPostgresSchemaUpdate7.4-to-8.0.bat**
    - **thingworxPostgresSchemaUpdate8.0-to-8.1.bat**
    - **thingworxPostgresSchemaUpdate8.1-to-8.2.bat**
    - **thingworxPostgresSchemaUpdate8.2-to-8.3.bat**
    - **thingworxPostgresSchemaUpdate8.3-to-8.4.bat**
    - **thingworxPostgresValueStreamSchemaUpdate.bat**
    - **thingworxPostgresValueStreamDataUpdate.bat**

12. **THIS STEP IS FOR MS SQL ONLY. SKIP AND PROCEED TO THE NEXT STEP IF YOU ARE NOT UPGRADING FROM MS SQL.**

    Copy the entire **update** folder to the MSSQL server and run the following scripts that are located in the **update** folder (starting with the version you are upgrading from):

    - **thingworxMssqlSchemaUpdate7.4-to-8.0.bat**
    - **thingworxMssqlSchemaUpdate8.0-to-8.1.bat**
    - **thingworxMssqlSchemaUpdate8.1-to-8.2.bat**
    - **thingworxMssqlSchemaUpdate8.2-to-8.3.bat**
    - **thingworxMssqlSchemaUpdate8.3-to-8.4.bat**
    - **thingworxMssqlValueStreamSchemaUpdate.bat**
    - **thingworxMssqlValueStreamDataUpdate.bat**

    Give the instance name using the **-i** parameter while running the update script.

13. Copy the **Thingworx.war** file and place it in the following location of your Tomcat installation:

    \Apache Software Foundation\Tomcat 8.5\webapps

14. Verify licensing:
    a. Rename the existing **license.bin** file located in the **\ThingworxPlatform** folder.
       NOTE: You can delete the file, but if login is unsuccessful, it will need to be recovered.
    b. Verify that your PTC support site **username, password,** and **timeout** (optional) are added to the **platform-settings.json** in the **PlatformSettingsConfig** section.

```
"LicensingConnectionSettings":{
      "username":"PTC Support site user name",
      "password":"PTC Support site password",
      “timeout”:”60”,
}
```

NOTE: If the settings are filled out incorrectly or if the server can’t connect, a License Request text file (**licenseRequestFile.txt**) is created in the **ThingworxPlatform** folder. In this scenario, a license must be created manually. (If it is not created, ThingWorx will start in limited mode. Limited mode does not allow you to persist licensed entities to the database. Licensed entities are Things, Mashups, Masters, Gadgets, Users, and Persistence Providers).

Further information on obtaining a ThingWorx disconnected site license through our License Management site can be found in the Licensing Guide for disconnected sites (no connection to PTC Support portal).

Open a case with Technical Support if you are doing the manual disconnected mode of licensing and have any questions or need assistance with generating a license.

15. Enable extension import.
    NOTE: By default, extension import is disabled for all users.

Add the following to the **platform-settings.json** file. Add or update the following **ExtensionPackageImportPolicy** parameters to **true** to allow extensions to be imported. See the Help Center for best practices on configuration.

```
"ExtensionPackageImportPolicy": {
            "importEnabled": <true or false>,
            "allowJarResources": <true or false>,
            "allowJavascriptResources": <true or false>,
            "allowCSSResources": <true or false>,
            "allowJSONResources": <true or false>,
            "allowWebAppResources": <true or false>,
            "allowEntities": <true or false>,
            "allowExtensibleEntities": <true or false>
      },
```

16. If you are using H2 as a database with ThingWorx, a username and password must be added to the **platform-settings.json** file.

    NOTE: Skip this step if you are not using H2.

    Add and configure the following parameters to the **platform-settings.json** file. See platform-settings.json Configuration Details for more information.

```
},
"PersistenceProviderPackageConfigs":{
"H2PersistenceProviderPackage":{
"ConnectionInformation":
{
      "password": "<changeme>",
      "username": "twadmin"
}
},
```

17. Start Tomcat.

18. To launch ThingWorx, go to **<servername>\Thingworx** in a web browser and log in as Administrator.

19. Import previously exported data.
    a. FOR DSE, import the data that you previously exported.
    b. For all other persistence providers, import wiki data if necessary.

19. OPTIONAL STEP: If you are using Integration Connectors, you must obtain and install the latest version of the integration runtime. For more information, refer to Initial Setup of Integration Runtime Service for Integration Connectors.

Note: After starting the ThingWorx platform, check the Application log for the platform. If you are using MSSQL, PostgreSQL, or H2, you may see property conflict error messages. If so, see Appendix C: Troubleshooting an In-Place Migration to ThingWorx 8.4

Migrating to ThingWorx 8.4 on Ubuntu

The steps below are for migration only. For in-place upgrade, refer to In-place Upgrade to ThingWorx 8.4 on Ubuntu.

1. Obtain the latest version of ThingWorx.

   NOTE: ThingWorx downloads are available in PTC Software Downloads.

2. Export entities and data.* In Composer, click **Import/Export>Export>To ThingworxStorage.**

   *For in-place migration using DSE (PostgreSQL): If you have any data located in Neo4j and you are persisting any blog, wiki, stream, value stream, or data table data to DSE, do not export all data to **ThingworxStorage** (to prevent duplicating blog, wiki, stream, value stream, or data table data when imported). Instead, you must explicitly export Neo4j data to file.

   NOTE: Data export progress can be monitored in the Application Log.

3. If necessary, click **Include Data**.

4. Click **Export**.

   NOTE: Data and entities are exported to **ThingworxStorage/exports**.

5. Copy these data and entity export files and move them to a safe location. You will import these files in a later step.

6. It is highly recommended to back up the contents of the following folders before continuing:
   - **Apache Software Foundation/Tomcat8.5/webapps/Thingworx**
   - **/ThingworxStorage**

7. Backup the **validation.properties** file from **/ThingworxStorage/esapi**
   NOTE: If you have custom configurations in the file, move the file for later reference.

   NOTE: The **validation.properties** file is created upon startup of ThingWorx. If you do not remove the file, the updated file with additional parameters will not overwrite the current version during upgrade. Reference the Help Center for additional information.

8. Locate the **keystore.jks** file in the **ThingworxStorage** folder and move it to a safe place. You will add this file back to the **ThingworxStorage** folder later.

9. Note any extensions that are in use (located in **/ThingworxStorage/extensions**). They will be reimported in a later step.

10. Stop Tomcat.

11. Delete the contents of the **/ThingworxStorage** and **/ThingworxBackupStorage** folders.

12. Remove the contents of the following folder wherever Tomcat is installed:

    Tomcat8.5/webapps/Thingworx.

13. Verify licensing:
    Rename the existing **license.bin** file located in the **ThingworxPlatform** folder.

    NOTE: You can delete the file, but if login is unsuccessful, it will need to be recovered.
    Verify that your PTC support site username, password, and timeout (optional) are added to the **platform-settings.json** in the **PlatformSettingsConfig** section.

    "LicensingConnectionSettings":{
    "username":"PTC Support site user name",
    "password":"PTC Support site password",
    “timeout”:”60”
    }

    NOTE: If the settings are filled out incorrectly or if the server can’t connect, a License Request text file (**licenseRequestFile.txt**) is created in the **ThingworxPlatform** folder. In this scenario, a license must be created manually. (If it is not created, ThingWorx will start in limited mode. Limited mode does not allow you to persist licensed entities to the database. Licensed entities are Things, Mashups, Masters, Gadgets, Users, and Persistence Providers).

    Further information on obtaining a ThingWorx disconnected site license through our License Management site can be found in the Licensing Guide for disconnected sites (no connection to PTC Support portal).

    Open a case with Technical Support if you are doing the manual disconnected mode of licensing and have any questions or need assistance with generating a license.

14. Configure the Administrator password. Add the following to the **platform-settings.json** file along with a password that is at least 10 characters long to the **PlatformSettingsConfig** section. Reference Appendix A: platform-settings.json Configuration Details for more information on placement.

```
{
    "PlatformSettingsConfig": {
        "AdministratorUserSettings": {
            "InitialPassword": "changeme"
        }
    }
```

15. Enable extension import.
    NOTE: By default, extension import is disabled for all users.

    Add the following to the **platform-settings.json** file. Add or update the following **ExtensionPackageImportPolicy** parameters to true to allow extensions to be imported. See the Help Center for best practices on configuration.

```
"ExtensionPackageImportPolicy": {
            "importEnabled": <true or false>,
            "allowJarResources": <true or false>,
            "allowJavascriptResources": <true or false>,
            "allowCSSResources": <true or false>,
            "allowJSONResources": <true or false>,
            "allowWebAppResources": <true or false>,
            "allowEntities": <true or false>,
            "allowExtensibleEntities": <true or false>
        },
```

16. If you are using H2 as a database with ThingWorx, a username and password must be added to the **platform-settings.json** file.

    NOTE: Skip this step if you are not using H2.

    Add and configure the following parameters to the **platform-settings.json** file. See platform-settings.json Configuration Details for more information.

```
},
"PersistenceProviderPackageConfigs":{
"H2PersistenceProviderPackage":{
"ConnectionInformation":
{
      "password": "<changeme>",
      "username": "twadmin"
}
},
```

17. Navigate to the **ThingWorxStorage** folder and delete the **keystore.jks** file. Locate the **keystore.jks** file that you previously moved from the **ThingworxStorage** folder and paste it in the folder.

18. Unzip the ThingWorx zip archive to a temporary directory.

    Move the Thingworx.war file to **/usr/share/tomcat8.5/8.5.xx/webapps**:

    NOTE: The zip filepath below uses the PostgreSQL version of ThingWorx. If you are using another version, change as necessary.

    ```
    $ unzip MED-61111-CD-081_F000_ThingWorx-Platform-Postgres-8.4.0.zip
    $ sudo mv Thingworx.war $CATALINA_HOME/webapps
    $ sudo chown tomcat8.5:tomcat8.5 $CATALINA_HOME/webapps/Thingworx.war
    $ sudo chmod 775 $CATALINA_HOME/webapps/Thingworx.war
    ```

19. Start Tomcat to deploy the ThingWorx web application:

    ```
    $ sudo service tomcat8.5 start
    ```

    NOTE: If Tomcat fails to start and reports the error message: “Check the InitialPassword setting in the AdministratorUserSettings section in platform-settings.json. Password must be a minimum of 10 characters”, check the following:

    - The password setting exists in platform-settings.json
    - The password is valid (10 or more characters)
    - The platform-settings.json file is formatted correctly - bad formatting could lead to errors. Reference Passwords for more information on formatting.

20. Move the exports file back to the **ThingworxStorage/exports** folder.

21. To launch ThingWorx, go to **<servername>/Thingworx** in a web browser.

    Use the following login information:

    Login Name: Administrator
    Password: <*password as defined in previous step*>

22. Import extensions. In Compser, click **Import/Export>Import**.

    NOTE: Obtain and import the latest versions of the extensions. If you are upgrading to a major version (for example, from 7.x to 8.0, you must import the 8.x versions of the extensions.)

    Extensions are available in the PTC Marketplace.

    NOTE: For in-place upgrade from 6.5 to 8.0 for Neo4j with DataStax Enterprise (DSE), an additional Tomcat restart is required when you are installing the latest version of:

    - **DsePersistenceProvider_ExtensionPackage.zip**

    NOTE: This extension must be requested from Support.

NOTE: If you are importing **From ThingworxStorage**, you can select the **Overwrite Collection Permissions and Organizations** option, so that the collection permissions and organizations in the import will overwrite the settings on the server with the collection permissions and organizations contained in the import. If unchecked, the default behavior merges the collection permissions and organizations from the import into what is already defined on the server

23. Import entities and data. In Composer, click **Import/Export>From ThingworxStorage**.

24. OPTIONAL STEP: If you are using Integration Connectors, you must obtain and install the latest version of the integration runtime. For more information, refer to Initial Setup of Integration Runtime Service for Integration Connectors.

In-place Upgrade to ThingWorx 8.4 on Ubuntu

Refer to the table at the beginning of this document to determine your upgrade path. The steps below are for in-place upgrades only.

1. Obtain the latest version of ThingWorx.

   NOTE: ThingWorx downloads are available in PTC Software Downloads.

2. Verify that you are running the required versions of Tomcat and Java.

   NOTE: Refer to the System Requirements and Compatibility Matrix document for version requirements.

3. Tomcat Java option settings may have changed between versions. Refer to the Apache Tomcat Java Option Settings in the Appendix of the Installation Guide to verify that your settings are correct.

4. Stop Tomcat.

5. It is highly recommended to backup the following two folders before continuing:
   a. **Apache Software Foundation/Tomcat 8.5/webapps/Thingworx**
   b. **/ThingworxStorage**

6. Backup and delete the **validation.properties** file from **/ThingworxStorage/esapi**

   NOTE: If this file is not deleted, exports will fail.

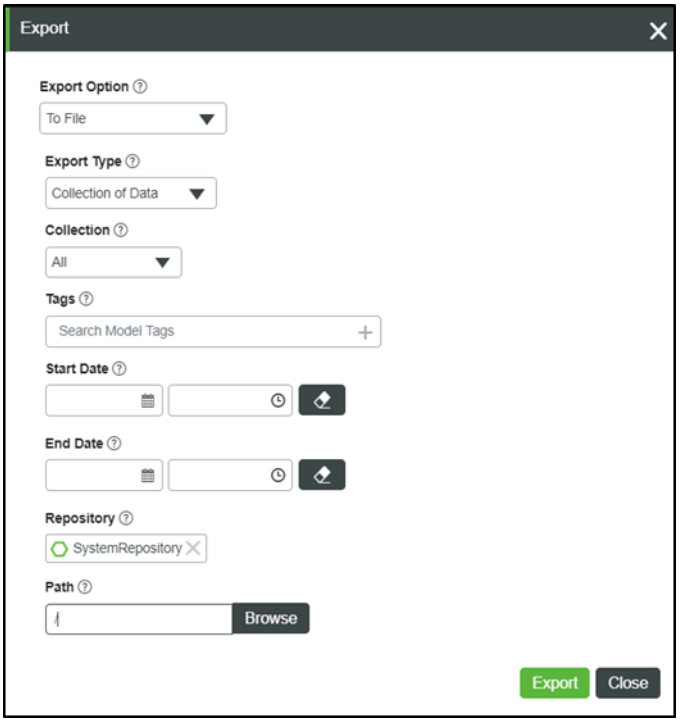
   NOTE: If you have custom configurations in the file, move the file for later reference.

   NOTE: The **validation.properties** file is created upon startup of ThingWorx. If you do not remove the file, the updated file with additional parameters will not overwrite the current version during upgrade. Reference the Help Center for additional information.

7. **This step is for DataStax Enterprise (DSE) ONLY. If you are not using DSE, skip and go to the next step.**

   If you have any data located in Neo4j and you are persisting any blog, wiki, stream, value stream, or data table data to DSE, do not export all data to ThingworxStorage (to prevent duplicating blog, wiki, stream, value stream, or data table data when imported). Instead, you must explicitly export Neo4j data to file.

   NOTE: Data export progress can be monitored in the Application Log.



8. Export wiki data.

   NOTE: Perform this step only if you have wiki data.

- **THIS STEP IS FOR POSTGRESQL ONLY. SKIP AND PROCEED TO THE NEXT STEP IF YOU ARE NOT UPGRADING FROM POSTGRESQL.**

  Run the following scripts that are located in the **update** folder (starting with the version you are upgrading from):

  - **thingworxPostgresSchemaUpdate6.5-to-6.6.sh**
  - **thingworxPostgresSchemaUpdate6.6-to-7.0.sh**
  - **thingworxPostgresSchemaUpdate7.0-to-7.1.sh**
  - **thingworxPostgresSchemaUpdate7.1-to-7.2.sh**
  - **thingworxPostgresSchemaUpdate7.2-to-7.3.sh**
  - **thingworxPostgresSchemaUpdate7.3-to-7.4.sh**
  - **thingworxPostgresSchemaUpdate7.4-to-8.0.sh**
  - **thingworxPostgresSchemaUpdate8.0-to-8.1.sh**
  - **thingworxPostgresSchemaUpdate8.1-to-8.2.sh**
  - **thingworxPostgresSchemaUpdate8.2-to-8.3.sh**
  - **thingworxPostgresSchemaUpdate8.3-to-8.4.sh**
  - **thingworxPostgresValueStreamSchemaUpdate.sh**
  - **thingworxPostgresValueStreamDataUpdate.sh**

9. Move the **Thingworx.war** file to **/usr/share/tomcat8.5/8.5.xx/webapps:**

   NOTE: Your zip file path may be different. The zip file path below uses the PostgreSQL version of ThingWorx. If you are using another version, change as necessary.

```
$ unzip MED-61111-CD-081_F000_ThingWorx-Platform-Postgres-8.4.0.zip
$ sudo mv Thingworx.war /usr/share/tomcat8.5/8.5.xx/webapps
$ sudo chown tomcat8.5:tomcat8.5
/usr/share/tomcat8.5/8.5.xx/webapps/Thingworx.war
$ sudo chmod 775 /usr/share/tomcat8.5/8.5.xx/webapps/Thingworx.war
```

10. Verify licensing:
    a. Rename the existing **license.bin** file located in the **ThingworxPlatform** folder.
       NOTE: You can delete the file, but if login is unsuccessful, it will need to be recovered.
    b. Verify that your PTC support site **username, password,** and **timeout** (optional) are added to the **platform-settings.json** in the **PlatformSettingsConfig** section.

```
"LicensingConnectionSettings":{
      "username":"PTC Support site user name",
      "password":"PTC Support site password",
      “timeout”:”60”
  }
```

   NOTE: If the settings are filled out incorrectly or if the server can’t connect, a License Request text file (**licenseRequestFile.txt**) is created in the **ThingworxPlatform** folder. In this scenario, a license must be created manually. (If it is not created, ThingWorx will start in limited mode. Limited mode does not allow you to persist licensed entities to the database. Licensed entities are Things, Mashups, Masters, Gadgets, Users, and Persistence Providers).

   Further information on obtaining a ThingWorx disconnected site license through our License Management site can be found in the Licensing Guide for disconnected sites (no connection to PTC Support portal).

   Open a case with Technical Support if you are doing the manual disconnected mode of licensing and have any questions or need assistance with generating a license.

11. Enable extension import.

    NOTE: By default, extension import is disabled for all users.

    Add the following to the **platform-settings.json** file. Add or update the following **ExtensionPackageImportPolicy** parameters to true to allow extensions to be imported. See the Help Center for best practices on configuration.

```
"ExtensionPackageImportPolicy": {
            "importEnabled": <true or false>,
            "allowJarResources": <true or false>,
            "allowJavascriptResources": <true or false>,
            "allowCSSResources": <true or false>,
            "allowJSONResources": <true or false>,
            "allowWebAppResources": <true or false>,
            "allowEntities": <true or false>,
            "allowExtensibleEntities": <true or false>
        },
```

12. If you are using H2 as a database with ThingWorx, a username and password must be added to the **platform-settings.json** file.

    NOTE: Skip this step if you are not using H2.

    Add and configure the following parameters to the **platform-settings.json** file. See platform-settings.json Configuration Details for more information.

```
},
"PersistenceProviderPackageConfigs":{
"H2PersistenceProviderPackage":{
"ConnectionInformation":
{
      "password": "<changeme>",
      "username": "twadmin"
}
},
```

13. Start Tomcat.

14. To launch ThingWorx, go to **<servername>/Thingworx** in a web browser and log in as Administrator.

15. Import previously exported data.
    a. FOR DSE, import the data that you previously exported.
    b. For all other persistence providers, import wiki data if necessary.

16. OPTIONAL STEP: If you are using Integration Connectors, you must obtain and install the latest version of the integration runtime. For more information, refer to Initial Setup of Integration Runtime Service for Integration Connectors.

Note: After starting the ThingWorx platform, check the Application log for the platform. If you are using MSSQL, PostgreSQL, or H2, you may see property conflict error messages. If so, see Appendix C: Troubleshooting an In-Place Migration to **Appendix C: Troubleshooting an In-Place Migration to ThingWorx 8.4**ThingWorx 8.4)**.**

Migrating to ThingWorx 8.4 on RHEL

The steps below are for migration only. For in-place upgrade, refer to In-place Upgrade to ThingWorx 8.4 on RHEL.

1. Obtain the latest version of ThingWorx.

   NOTE: ThingWorx downloads are available in PTC Software Downloads.

2. Export entities and data.* In Composer, click **Import/Export>Export>To ThingworxStorage.**

   NOTE: Data export progress can be monitored in the Application Log.

3. If necessary, click **Include Data**.

4. Click **Export**.
   NOTE: Data and entities are exported to **/ThingworxStorage/exports**

5. Copy these data and entity export files and move to a safe location. For example:

   ```
   $ sudo cp –R /ThingworxStorage/exports /tempDirectory
   ```

6. Stop Tomcat:

   ```
   $ sudo systemctl stop tomcat
   ```

7. It is highly recommended to back up the contents of the following folders before continuing:
   - **/usr/share/tomcat8.5/8.5.xx/webapps/Thingworx**
   - **/ThingworxStorage**
   - **/ThingworxBackupStorage**

8. Backup the **validation.properties** file from **/ThingworxStorage/esapi**
   NOTE: If you have custom configurations in the file, move the file for later reference.

   NOTE: The **validation.properties** file is created upon startup of ThingWorx. If you do not remove the file, the updated file with additional parameters will not overwrite the current version during upgrade. Reference the Help Center for additional information.

9. Locate the **keystore.jks** file in the **ThingworxStorage** folder and move it to a safe place. You will add this file back to the **ThingworxStorage** folder later.

10. Delete the contents of both **/ThingworxStorage** and **/ThingworxBackupStorage** folders.

```
$ sudo rm –r /ThingworxStorage/*
$ sudo rm –r /ThingworxBackupStorage/*
```

11. Remove the contents of the following folder:
    /usr/share/tomcat8.5/8.5.xx/webapps/Thingworx

```
$ sudo rm –r /usr/share/tomcat8.5/8.5.xx/webapps/Thingworx/*
```

12. Undeploy ThingWorx by deleting the **Thingworx.war** file from **/usr/share/tomcat8.5/8.5.xx/webapps**

```
$ sudo rm /usr/share/tomcat8.5/8.5.xx/webapps/Thingworx.war
```

13. Unzip the ThingWorx zip archive to a temporary directory.
14. Move the **Thingworx.war** file to **/usr/share/tomcat8.5/8.5.xx/webapps:**
    NOTE: Your zip file path may be different. The zip filepath below uses the PostgreSQL version of ThingWorx. If you are using another version, change as necessary.

```
$ unzip MED-61111-CD-081_F000_ThingWorx-Platform-Postgres-8.4.0.zip
$ sudo mv Thingworx.war /usr/share/tomcat8.5/8.5.xx/webapps
$ sudo chown tomcat8.5:tomcat8.5
/usr/share/tomcat8.5/8.5.xx/webapps/Thingworx.war
$ sudo chmod 775 /usr/share/tomcat8.5/8.5.xx/webapps/Thingworx.war
```

15. Verify licensing:
    a. Rename the existing **license.bin** file located in the **ThingworxPlatform** folder.
    NOTE: You can delete the file, but if login is unsuccessful, it will need to be recovered.
    b. Verify that your PTC support site **username, password,** and **timeout** (optional) are added to the **platform-settings.json** in the **PlatformSettingsConfig** section.

```
"LicensingConnectionSettings":{
      "username":"PTC Support site user name",
      "password":"PTC Support site password",
      “timeout”:”60”
  }
```

NOTE: If the settings are filled out incorrectly or if the server can’t connect, a License Request text file (**licenseRequestFile.txt**) is created in the **ThingworxPlatform** folder. In this scenario, a license must be created manually. (If it is not created, ThingWorx will start in limited mode. Limited mode does not allow you to persist licensed entities to the database. Licensed entities are Things, Mashups, Masters, Gadgets, Users, and Persistence Providers).

Further information on obtaining a ThingWorx disconnected site license through our License Management site can be found in the Licensing Guide for disconnected sites (no connection to PTC Support portal).

Open a case with Technical Support if you are doing the manual disconnected mode of licensing and have any questions or need assistance with generating a license.

16. Configure the Administrator password. Add the following to the **platform-settings.json** file along with a password that is at least 10 characters long to the **PlatformSettingsConfig** section. Reference Appendix A: platform-settings.json Configuration Details for more information on placement.

```
{
    "PlatformSettingsConfig": {
        "AdministratorUserSettings": {
            "InitialPassword": "changeme"
        }
    }
}
```

17. Enable extension import.
    NOTE: By default, extension import is disabled for all users.

    Add the following to the **platform-settings.json** file. Add or update the following **ExtensionPackageImportPolicy** parameters to true to allow extensions to be imported. See the Help Center for best practices on configuration.

```
"ExtensionPackageImportPolicy": {
            "importEnabled": <true or false>,
            "allowJarResources": <true or false>,
            "allowJavascriptResources": <true or false>,
            "allowCSSResources": <true or false>,
            "allowJSONResources": <true or false>,
            "allowWebAppResources": <true or false>,
            "allowEntities": <true or false>,
            "allowExtensibleEntities": <true or false>
        },
```

18. If you are using H2 as a database with ThingWorx, a username and password must be added to the **platform-settings.json** file.

    NOTE: Skip this step if you are not using H2.

    Add and configure the following parameters to the **platform-settings.json** file. See platform-settings.json Configuration Details for more information.

```
},
"PersistenceProviderPackageConfigs":{
"H2PersistenceProviderPackage":{
"ConnectionInformation":
{
      "password": "<changeme>",
      "username": "twadmin"
}
},
```

19. Navigate to the **ThingWorxStorage** folder and delete the **keystore.jks** file. Locate the **keystore.jks** file that you previously moved from the **ThingworxStorage** folder and paste it in the folder.
20. Copy the data and entity files that were previously exported. Move them back to **/ThingworxStorage/exports**. For example:

```
$ sudo cp -R /tempDirectory/* /ThingworxStorage/exports
$ sudo chown –R tomcat8.5:tomcat8.5 /ThingworxStorage/exports/*
```

21. Start Tomcat to deploy the ThingWorx web application:

```
$ sudo service tomcat8.5 start
```

NOTE: If Tomcat fails to start and reports the error message: “Check the InitialPassword setting in the AdministratorUserSettings section in platform-settings.json. Password must be a minimum of 10 characters”, check the following:

- The password setting exists in platform-settings.json
- The password is valid (10 or more characters)
- The platform-settings.json file is formatted correctly - bad formatting could lead to errors. Reference Passwords for more information on formatting.

22. To launch ThingWorx, go to **<servername>/Thingworx** in a web browser.
Use the following login information:

Login Name: Administrator
Password: <*password as defined in previous step*>

23. Import extensions. In Composer, click **Import/Export>Import**>Extension.

NOTE: Obtain and import the latest versions of the extensions. If you are upgrading to a major version (for example, from 7.x to 8.0, you must import the 8.x versions of the extensions.) Extensions are available in the PTC Marketplace.

NOTE: For in-place migration from 6.5 to 8.0 for Neo4j with DataStax Enterprise (DSE), an additional Tomcat restart is required when you are installing the latest version of: **DsePersistenceProvider_ExtensionPackage.zip**

NOTE: This extension must be requested from Support.

NOTE: If you are importing **From ThingworxStorage**, you can select the **Overwrite Collection Permissions and Organizations** option, so that the collection permissions and organizations in the import will overwrite the settings on the server with the collection permissions and organizations contained in the import. If unchecked, the default behavior merges the collection permissions and organizations from the import into what is already defined on the server.

24. Import entities and data. In Composer, click **Import/Export>Import>From ThingworxStorage**

25. OPTIONAL STEP: If you are using Integration Connectors, you must obtain and install the latest version of the integration runtime. For more information, refer to Initial Setup of Integration Runtime Service for Integration Connectors.

In-place Upgrade to ThingWorx 8.4 on RHEL

Refer to the table at the beginning of this document to determine your upgrade path. The steps below are for in-place upgrades only.

1. Obtain the latest version of ThingWorx.

   NOTE: ThingWorx downloads are available in PTC Software Downloads.

2. Verify that you are running the required versions of Tomcat and Java.

   NOTE: Refer to the System Requirements and Compatibility Matrix document for version requirements.

3. Tomcat Java option settings may have changed between versions. Refer to the Apache Tomcat Java Option Settings in the Appendix of the Installation Guide to verify that your settings are correct.

4. It is highly recommended to backup the following folders before continuing:
   a. **/usr/share/tomcat8.5/8.5.xx/webapps/Thingworx**
   b. **/ThingworxStorage**
   c. /**ThingworxBackupStorage**

5. Backup and delete the **validation.properties** file from **/ThingworxStorage/esapi**

   NOTE**:** If this file is not deleted, exports will fail.

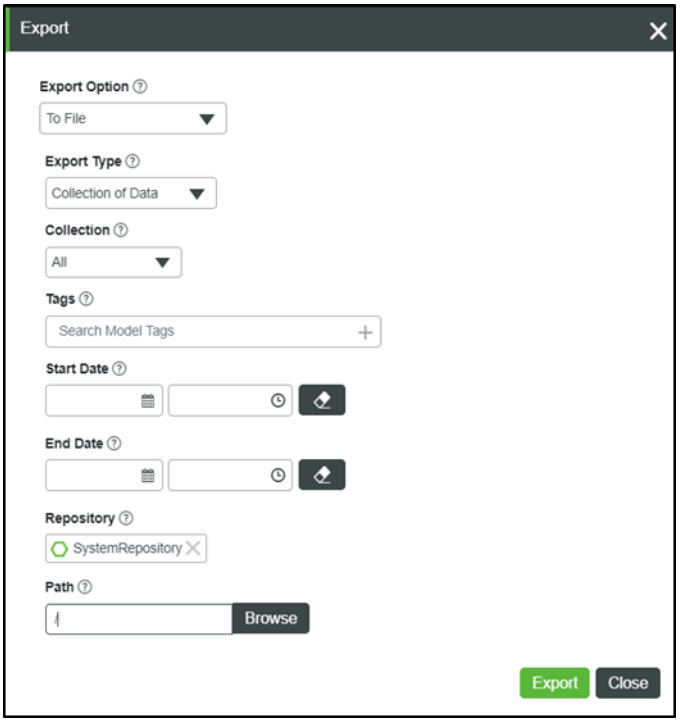
   NOTE: If you have custom configurations in the file, move the file for later reference.

   NOTE: The **validation.properties** file is created upon startup of ThingWorx. If you do not remove the file, the updated file with additional parameters will not overwrite the current version during upgrade. Reference the Help Center for additional information.

6. **This step is for DataStax Enterprise (DSE) ONLY. If you are not using DSE, skip and go to the next step.**

   If you have any data located in Neo4j and you are persisting any blog, wiki, stream, value stream, or data table data to DSE, do not export all data to **ThingworxStorage** (to prevent duplicating blog, wiki, stream, value stream, or data table data when imported). Instead, you must explicitly export Neo4j data to file.

   NOTE: Data export progress can be monitored in the Application Log.



7. Export wiki data.
   NOTE: Perform this step only if you have wiki data**.**

8. Stop Tomcat:

   ```
   $ sudo systemctl stop tomcat
   ```

9. Go to the Tomcat installation at **/usr/share/tomcat8.5/8.5.xx/webapps** and delete the **Thingworx.war** file:

   ```
   $ sudo rm /usr/share/tomcat8.5/8.5.xx/webapps/Thingworx.war
   ```

- **THIS STEP IS FOR POSTGRESQL ONLY. SKIP AND PROCEED TO THE NEXT STEP IF YOU ARE NOT UPGRADING FROM POSTGRESQL.**

  Run the following scripts that are located in the **update** folder (starting with the version you are upgrading from):

  - **thingworxPostgresSchemaUpdate6.5-to-6.6.sh**
  - **thingworxPostgresSchemaUpdate6.6-to-7.0.sh**
  - **thingworxPostgresSchemaUpdate7.0-to-7.1.sh**
  - **thingworxPostgresSchemaUpdate7.1-to-7.2.sh**
  - **thingworxPostgresSchemaUpdate7.2-to-7.3.sh**
  - **thingworxPostgresSchemaUpdate7.3-to-7.4.sh**
  - **thingworxPostgresSchemaUpdate7.4-to-8.0.sh**
  - **thingworxPostgresSchemaUpdate8.0-to-8.1.sh**
  - **thingworxPostgresSchemaUpdate8.1-to-8.2.sh**
  - **thingworxPostgresSchemaUpdate8.2-to-8.3.sh**
  - **thingworxPostgresSchemaUpdate8.3-to-8.4.sh**
  - **thingworxPostgresValueStreamSchemaUpdate.sh**
  - **thingworxPostgresValueStreamDataUpdate.sh**

10. Unzip the ThingWorx zip archive to a temporary directory. Move the **Thingworx.war** file to **/usr/share/tomcat8.5/8.5.xx/webapps:**

    NOTE: The zip filepath below uses the PostgreSQL version of ThingWorx. If you are using another version, change as necessary.

```
$ unzip MED-61111-CD-081_F000_ThingWorx-Platform-Postgres-8.4.0.zip
$ sudo mv Thingworx.war /usr/share/tomcat8.5/8.5.xx/webapps
$ sudo chown tomcat8.5:tomcat8.5
/usr/share/tomcat8.5/8.5.xx/webapps/Thingworx.war
$ sudo chmod 775 /usr/share/tomcat8.5/8.5.xx/webapps/Thingworx.war
```

11. Verify licensing:
    a. Rename the existing **license.bin** file located in the **ThingworxPlatform** folder.
    NOTE: You can delete the file, but if login is unsuccessful, it will need to be recovered.
    b. Verify that your PTC support site **username, password,** and **timeout** (optional) are added to the **platform-settings.json** in the **PlatformSettingsConfig** section.

```
"LicensingConnectionSettings":{
      "username":"PTC Support site user name",
      "password":"PTC Support site password",
      “timeout”:”60”
  }
```

NOTE: If the settings are filled out incorrectly or if the server can’t connect, a License Request text file (**licenseRequestFile.txt**) is created in the **ThingworxPlatform** folder. In this scenario, a license must be created manually. (If it is not created, ThingWorx will start in limited mode. Limited mode does not allow you to persist licensed entities to the database. Licensed entities are Things, Mashups, Masters, Gadgets, Users, and Persistence Providers).

Further information on obtaining a ThingWorx disconnected site license through our License Management site can be found in the Licensing Guide for disconnected sites (no connection to PTC Support portal).

Open a case with Technical Support if you are doing the manual disconnected mode of licensing and have any questions or need assistance with generating a license.

12. Enable extension import.
    NOTE: By default, extension import is disabled for all users.

Add the following to the **platform-settings.json** file. Add or update the following **ExtensionPackageImportPolicy** parameters to true to allow extensions to be imported. See the Help Center for best practices on configuration.

```
"ExtensionPackageImportPolicy": {
            "importEnabled": <true or false>,
            "allowJarResources": <true or false>,
            "allowJavascriptResources": <true or false>,
            "allowCSSResources": <true or false>,
            "allowJSONResources": <true or false>,
            "allowWebAppResources": <true or false>,
            "allowEntities": <true or false>,
            "allowExtensibleEntities": <true or false>
      },
```

13. If you are using H2 as a database with ThingWorx, a username and password must be added to the **platform-settings.json** file.

    NOTE: Skip this step if you are not using H2.

    Add and configure the following parameters to the **platform-settings.json** file. See platform-settings.json Configuration Details for more information.

```
},
"PersistenceProviderPackageConfigs":{
"H2PersistenceProviderPackage":{
"ConnectionInformation":
{
      "password": "<changeme>",
      "username": "twadmin"
}
},
```

14. Start Tomcat.

```
$ sudo systemctl start tomcat
```

15. To launch ThingWorx, go to **<servername>/Thingworx** in a web browser and log in as Administrator.

16. Import previously exported data.
    a. FOR DSE, import the data that you previously exported.
    b. For all other persistence providers, import wiki data if necessary.

17. OPTIONAL STEP: If you are using Integration Connectors, you must obtain and install the latest version of the integration runtime. For more information, refer to Initial Setup of Integration Runtime Service for Integration Connectors.

    Note: After starting the ThingWorx platform, check the Application log for the platform. If you are using MSSQL, PostGres, or H2, you may see property conflict error messages. If so, see Appendix C: Troubleshooting an In-Place Migration to ThingWorx 8.4 **.**

Appendix A: platform-settings.json Configuration Details

The platform-settings.json file is available for administrators to adjust settings for fine-tuning and is available in the software download. Reference the Help Center for descriptions on all options.

NOTE: The sample below contains all options. Only one persistence provider is required.

```
{
   "PlatformSettingsConfig":{
      "BasicSettings":{
         "BackupStorage":"/ThingworxBackupStorage",
         "DatabaseLogRetentionPolicy":7,
         "EnableBackup":true,
         "EnableHA":false,
         "EnableSystemLogging":false,
         "EnableSSO":false,
         "FileRepositoryRoot":"/ThingworxStorage",
         "HTTPRequestHeaderMaxLength":2000,
         "HTTPRequestParameterMaxLength":2000,
         "InternalAesCryptographicKeyLength":128,
         "Storage":"/ThingworxStorage"
      },
      "ExtensionPackageImportPolicy": {
         "importEnabled": false,
         "allowJarResources": false,
         "allowJavascriptResources": false,
         "allowCSSResources": false,
         "allowJSONResources": false,
         "allowWebAppResources": false,
         "allowEntities": false,
         "allowExtensibleEntities": false
     },
      "AdministratorUserSettings":{
         "InitialPassword":"changeme"
      },
      "HASettings":{
         "CoordinatorConnectionTimeout":15000,
         "CoordinatorHosts":"127.0.0.1:2181",
         "CoordinatorMaxRetries":3,
         "CoordinatorRetryTimeout":1000,
         "CoordinatorSessionTimeout":90000,
         "CoordinatorZNode":"/HALeadershipCoordinator",

"LoadBalancerBase64EncodedCredentials":"QWRtaW5pc3RyYXRvcjphZG1pbg=="
      },
      "LicensingConnectionSettings":{
         "username":"<username>",
         "password":"<password>",
         "timeout":"60"
      }
   },
   "PersistenceProviderPackageConfigs":{
      "NeoPersistenceProviderPackage":{
         "StreamProcessorSettings":{
            "maximumBlockSize":2500,
```

```
            "maximumQueueSize":250000,
            "maximumWaitTime":10000,
            "scanRate":5,
            "sizeThreshold":1000
         },
         "ValueStreamProcessorSettings":{
            "maximumBlockSize":2500,
            "maximumQueueSize":500000,
            "maximumWaitTime":10000,
            "scanRate":5,
            "sizeThreshold":1000
         },
         "PersistentPropertyProcessorSettings":{
            "maximumBlockSize":2500,
            "maximumWaitTime":1000,
            "maximumQueueSize":100000,
            "numberOfProcessingThreads":20,
            "scanRate":25,
            "sizeThreshold":1000
         }
      },
      "H2PersistenceProviderPackage":{
         "ConnectionInformation":{
            "acquireIncrement":5,
            "acquireRetryAttempts":30,
            "acquireRetryDelay":1000,
            "checkoutTimeout":2000,
            "idleConnectionTestPeriod":6,
            "initialPoolSize":10,
            "maxConnectionAge":0,
            "maxIdleTime":0,
            "maxIdleTimeExcessConnections":36000,
            "maxPoolSize":100,
            "maxStatements":0,
            "maxStatementsPerConnection":50,
            "minPoolSize":10,
            "numHelperThreads":6,
            “password”:”<securepassword>”,
            “username”:”twadmin”,
            "tableLockTimeout":10000,
            "testConnectionOnCheckout":false,
            "unreturnedConnectionTimeout":0
         },
         "StreamProcessorSettings":{
            "maximumBlockSize":2500,
            "maximumQueueSize":250000,
            "maximumWaitTime":10000,
            "numberOfProcessingThreads":5,
            "scanRate":5,
            "sizeThreshold":1000
         },
         "ValueStreamProcessorSettings":{
            "maximumBlockSize":2500,
            "maximumWaitTime":10000,
            "maximumQueueSize":500000,
            "numberOfProcessingThreads":5,
            "scanRate":5,
```

```
            "sizeThreshold":1000
         },
         "PersistentPropertyProcessorSettings":{
            "maximumBlockSize":2500,
            "maximumWaitTime":1000,
            "maximumQueueSize":100000,
            "numberOfProcessingThreads":20,
            "scanRate":25,
            "sizeThreshold":1000
         }
      },
      "PostgresPersistenceProviderPackage":{
         "ConnectionInformation":{
            "acquireIncrement":5,
            "acquireRetryAttempts":3,
            "acquireRetryDelay":10000,
            "checkoutTimeout":1000000,
            "driverClass":"org.postgresql.Driver",
            "fetchSize":5000,
            "idleConnectionTestPeriod":60,
            "initialPoolSize":5,
            "jdbcUrl":"jdbc:postgresql://localhost:5432/thingworx",
            "maxConnectionAge":0,
            "maxIdleTime":0,
            "maxIdleTimeExcessConnections":300,
            "maxPoolSize":100,
            "maxStatements":100,
            "minPoolSize":5,
            "numHelperThreads":8,
            "password":"password",
            "testConnectionOnCheckout":false,
            "unreturnedConnectionTimeout":0,
            "username":"twadmin"
         },
         "StreamProcessorSettings":{
            "maximumBlockSize":2500,
            "maximumQueueSize":250000,
            "maximumWaitTime":10000,
            "numberOfProcessingThreads":5,
            "scanRate":5,
            "sizeThreshold":1000
         },
         "ValueStreamProcessorSettings":{
            "maximumBlockSize":2500,
            "maximumQueueSize":500000,
            "maximumWaitTime":10000,
            "numberOfProcessingThreads":5,
            "scanRate":5,
            "sizeThreshold":1000
         },
         "PersistentPropertyProcessorSettings":{
            "maximumBlockSize":2500,
            "maximumWaitTime":1000,
            "maximumQueueSize":100000,
            "numberOfProcessingThreads":20,
            "scanRate":25,
            "sizeThreshold":1000
```

```
            }
         },
         "MssqlPersistenceProviderPackage":{
            "ConnectionInformation":{
               "acquireIncrement":5,
               "acquireRetryAttempts":3,
               "acquireRetryDelay":10000,
               "checkoutTimeout":1000000,

"driverClass":"com.microsoft.sqlserver.jdbc.SQLServerDriver",
               "fetchSize":5000,
               "idleConnectionTestPeriod":60,
               "initialPoolSize":5,

"jdbcUrl":"jdbc:sqlserver://localhost:1433;databaseName=thingworx;appli
cationName=Thingworx;",
               "maxConnectionAge":0,
               "maxIdleTime":0,
               "maxIdleTimeExcessConnections":300,
               "maxPoolSize":100,
               "maxStatements":100,
               "minPoolSize":5,
               "numHelperThreads":8,
               "password":"Password@123",
               "testConnectionOnCheckout":false,
               "unreturnedConnectionTimeout":0,
               "username":"msadmin"
            },
            "StreamProcessorSettings":{
               "maximumBlockSize":2500,
               "maximumQueueSize":250000,
               "maximumWaitTime":10000,
               "numberOfProcessingThreads":5,
               "scanRate":5,
               "sizeThreshold":1000
            },
            "ValueStreamProcessorSettings":{
               "maximumBlockSize":2500,
               "maximumWaitTime":10000,
               "maximumQueueSize":500000,
               "numberOfProcessingThreads":5,
               "scanRate":5,
               "sizeThreshold":1000
            },
            "PersistentPropertyProcessorSettings":{
               "maximumBlockSize":2500,
               "maximumWaitTime":1000,
               "maximumQueueSize":100000,
               "numberOfProcessingThreads":20,
               "scanRate":25,
               "sizeThreshold":1000
            }
         }
      }
   }
```

Appendix B: Licensing Troubleshooting

You must have a license file for ThingWorx 8.0 and later. Some possible situations that may require troubleshooting are described below:

Issue	Possible Resolution
The following error is received when deploying ThingWorx: `org.apache.catalina.core.ApplicationContext.log HTMLManager: FAIL - Deploy Upload Failed, Exception: org.apache.tomcat.util.http.fileupload.FileUploadBase$SizeLimitExceededException: the request was rejected because its size (90883556) exceeds the configured maximum (52437800) java.lang.IllegalStateException: org.apache.tomcat.util.http.fileupload.FileUploadBase$SizeLimitExceededException: the request was rejected because its size (90883556) exceeds the configured maximum (52437800) at org.apache.catalina.connector.Request.parseParts(Request.java:2871`	The max file size in the Tomcat web.xml file must be increased (default is 50MB). This file is located at : `<path to Tomcat>\Apache Software Foundation\Tomcat 8.5\webapps\manager\WEB-INF` 1. Open the **web.xml**. 2. Change the **max-file-size** and **max-request-size** to **104857600**. 3. Save and close the file. 4. Restart Tomcat.
The following error message is received when importing a PTC licensed extension: `is licensed but cannot find feature in license.bin file`	Visit the Manage Licenses section on the PTC Support site to confirm the correct license file that matches your entitlement. If you need further assistance with your licenses, please contact the License Management team.
The following error message is received when attempting to undeploy ThingWorx: `FAIL - Unable to delete [<path to Tomcat>\webapps\Thingworx]. The continued presence of this file may cause problems. Due to FlxCore64.dll (<path to Tomcat>\webapps\Thingworx\WEB-INF\extensions\FlxCore64.dll)`	Remove **-Djava.library.path** from Tomat's Java configuration before undeployment.

Issue	Possible Resolution
An error message similar to the following is seen in the *ConfigurationLog.log*: NOTE: The log message verifies if there is an issue with the license file and verifies if the Flexnet dlls are in the library path and have the correct permissions. `2017-03-10 05:56:07.097-0500 [L: ERROR] [O: ] [I: ] [U: SuperUser] [S: ] [T: localhost-startStop-1] **********LICENSING ERROR ANALYSIS` `2017-03-10 05:56:07.097-0500 [L: ERROR] [O:] [I:] [U: SuperUser] [S:] [T: localhost-startStop-1] /Library/flexs is listed as a java.library.path but it does not exist. /Library/blah is listed as a java.library.path but it does not exist. /Library/zzz is listed as a java.library.path but it does not exist. No flx dll files found. Is the java.library.path set?` `2017-03-10 05:56:07.097-0500 [L: ERROR] [O:] [I:] [U: SuperUser] [S:] [T: localhost-startStop-1] **********END LICENSING ERROR ANALYIS`	Set the java.library.path variable in the setenv.sh file or in the tomcat8.5 startup init.d script.

Issue	Possible Resolution

An error message similar to the following is thrown while the platform is starting:
 	
`2017-06-12 11:33:59.204+0530 [L: ERROR] [O: c.t.s.s.l.LicensingSubsystem] [I: ] [U: SuperUser] [S: ] [T: localhost-startStop-1] [message: The size of provided data is incorrect.]`
`2017-06-12 11:33:59.205+0530 [L: ERROR] [O: c.t.s.s.l.LicensingSubsystem] [I: ] [U: SuperUser] [S: ] [T: localhost-startStop-1]`
`==========`
`2017-06-12 11:33:59.205+0530 [L: ERROR] [O: c.t.s.s.l.LicensingSubsystem] [I: ] [U: SuperUser] [S: ] [T: localhost-startStop-1] Invalid License file: /ThingworxPlatform\license.bin`
`2017-06-12 11:33:59.205+0530 [L: ERROR] [O: c.t.s.s.l.LicensingSubsystem] [I: ] [U: SuperUser] [S: ] [T: localhost-startStop-1]`
`==========`
`2017-06-12 11:33:59.205+0530 [L: WARN] [O: c.t.s.ThingWorxServer] [I: ] [U: SuperUser] [S: ] [T: localhost-startStop-1] Shutting down the Platform.`
	The license file may have been opened/edited/saved in a browser. Download the license file again, rename it to **license_capability_response.bin**, and place in **ThingworxPlatform** folder without editing or saving it.

Appendix C: Troubleshooting an In-Place Migration to ThingWorx 8.4

As part of the Thing Presence feature added to ThingWorx platform 8.4, the following properties have been added to the Reportable thing shape and are used as part of presence evaluation on the things that implement this shape:

- isReporting
- reportingLastChange
- reportingLastEvaluation

Problem

If one of the property names above previously existed on a thing, thing template, or thing shape, the following errors will appear in the Application log when the platform starts up:
Thing Conflict:

```
[L: ERROR] [O: c.t.p.m.BaseReportingMigrator] [I: ] [U: SuperUser] [S: ]
[T: localhost-startStop-1] Thing: <Name of Thing>, has a property which
conflicts with one of the following system properties:
isReporting,reportingLastChange,reportingLastEvaluation. Please refer to
the ThingWorx Platform 8.4 documentation on how to resolve this problem.
```

Thing Template Conflict:

```
[L: ERROR] [O: c.t.p.m.BaseReportingMigrator] [I: ] [U: SuperUser] [S: ]
[T: localhost-startStop-1] ThingTempate: <Name of ThingTemplate>, has a
property which conflicts with one of the following system properties:
isReporting,reportingLastChange,reportingLastEvaluation. Please refer to
the ThingWorx Platform 8.4 documentation on how to resolve this problem.
```

Thing Shape Conflict:

```
[L: ERROR] [O: c.t.p.m.BaseReportingMigrator] [I: ] [U: SuperUser] [S: ]
[T: localhost-startStop-1] ThingShape: <Name of ThingShape>, has a
property which conflicts with one of the following system properties:
isReporting,reportingLastChange,reportingLastEvaluation. Please refer to
the ThingWorx Platform 8.4 documentation on how to resolve this problem.
```

Resolution

To resolve this problem, the property in conflict on each affected entity must be removed and any associated entities updated to accommodate this change (for example, Mashups or Services). Without this update, the associated things cannot display their reporting status properly and cannot be updated/saved. Once these entities are updated properly, the platform-specific reporting properties will be displayed and used in evaluating whether a device is connected and communicating. The following figure shows the properties that will be inherited by all things that are derived from the RemoteThing thing template (or another thing template derived from this thing template).