



thingworx®

Upgrading to ThingWorx 7.4

Version 1.2

Copyright © 2017 PTC Inc. and/or Its Subsidiary Companies. All Rights Reserved.

User and training guides and related documentation from PTC Inc. and its subsidiary companies (collectively "PTC") are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION. PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

Important Copyright, Trademark, Patent, and Licensing Information: See the About Box, or copyright notice, of your PTC software.

United States Governments Rights

PTC software products and software documentation are "commercial items" as that term is defined at 48 C.F.R. 2.101. Pursuant to Federal Acquisition Regulation (FAR) 12.212 (a)-(b) (Computer Software) (MAY 2014) for civilian agencies or the Defense Federal Acquisition Regulation Supplement (DFARS) at 227.7202-1 (a) (Policy) and 227.7202-3 (a) (Rights in commercial computer software or commercial computer software documentation) (FEB 2014) for the Department of Defense, PTC software products and software documentation are provided to the U.S. Government under the PTC commercial license agreement. Use, duplication or disclosure by the U.S. Government is subject solely to the terms and conditions set forth in the applicable PTC software license agreement.

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA

Document Revision History

Revision Date	Version	Description of Change
April 05, 2017	1.2	Added RHEL migrate and in-place steps.
March 23, 2017	1.1	Fixed error in licensing library path.
March 22, 2017	1.0	Initial version for 7.4



Installing ThingWorx

Document Revision History.....	1
Determine Upgrade Path: In-place vs. Migration	3
Database Options: Neo4j, PostgreSQL, Microsoft SQL Server, SAP HANA, or H2	4
Prerequisites	4
Licensing Changes in 7.4 (Windows/Ubuntu/RHEL)	4
Migrating to ThingWorx 7.4 (Neo4j/PostgreSQL/ MS SQL Server /H2) on Windows.....	7
In-place Upgrade to ThingWorx 7.4 (Neo4j/PostgreSQL/MS SQL Server/H2) on Windows.....	11
Migrating to ThingWorx 7.4 (Neo4j/PostgreSQL/MS SQL Server/H2) on Ubuntu	13
In-place Upgrade to ThingWorx 7.4 (Neo4j/PostgreSQL/MS SQL Server/H2) on Ubuntu	17
Migrating to ThingWorx 7.4 (Neo4j/PostgreSQL/MS SQL Server/H2) on RHEL	19
In-place Upgrade to ThingWorx 7.4 (Neo4j/PostgreSQL/MS SQL Server/H2) on RHEL	22
Appendix A: Licensing Troubleshooting.....	25

Upgrading to ThingWorx 7.4

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. Migrating includes more process 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.

		Upgrading To ThingWorx					
		7.4 on Neo4j	7.4 on Neo4j with DSE	7.4 on PostgreSQL	7.4 on PostgreSQL with DSE	7.4 on H2	7.4 on MS SQL Server
Upgrading From ThingWorx	5.X on Neo4j	Migration	Migration + DSE setup	Migration	Migration + DSE Setup	Migration ⁵	Migration ⁵
	6.0, 6.5, 6.6, 7.0, 7.1, 7.2, 7.3 on Neo4j	In Place ^{2,3,4}	In Place ^{2,3,4} + DSE Setup	Migration	Migration + DSE Setup	Migration ⁵	Migration ⁵
	6.0, 6.5, 6.6, 7.0, 7.1, 7.2, 7.3 on Neo4j with DSE ¹	--	In Place ^{2,3,4}	--	Migration	--	Migration
	6.5, 6.6, 7.0, 7.1, 7.2, 7.3 on PostgreSQL with DSE ¹	--	--	--	In Place ^{2,3}	--	Migration
	6.5, 6.6, 7.0, 7.1, 7.2, 7.3 on PostgreSQL	--	--	In Place ^{2,3}	In Place ^{2,3} + DSE Setup	--	Migration

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: In-place upgrades for Neo4j should be done only for service pack level upgrades, not major or minor releases.

5: 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: Neo4j, PostgreSQL, Microsoft SQL Server, SAP HANA, or H2

With ThingWorx 7.4, you can use PostgreSQL (with an optional High Availability layer), Neo4j, SAP HANA, Microsoft SQL Server, 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-H2-7.4.0**
- Neo4j: **Thingworx-Platform-Neo-7.4.0**
- PostgreSQL/HA: **Thingworx-Platform-Postgres-7.4.0**
- SAP HANA: **Thingworx-Platform-hana-7.4.0**
- Microsoft SQL Server: **Thingworx-Platform-mssql-7.4.0**

Prerequisites

- 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.

Licensing Changes in 7.4 (Windows/Ubuntu/RHEL)

Due to licensing changes in 7.4, if you are upgrading to 7.4 or later, you must perform the following steps for licensing before performing the upgrade.

1. Copy the **license.bin** file from the software download package to the **ThingworxPlatform** folder.

Ubuntu command:

```
$ sudo mv license.bin /ThingworxPlatform
```

RHEL command:

```
$ sudo mv license.bin /ThingworxPlatform/license.bin
```

NOTE: Failure to place the license.bin in the ThingworxPlatform folder will result in ThingWorx not starting after deployment.

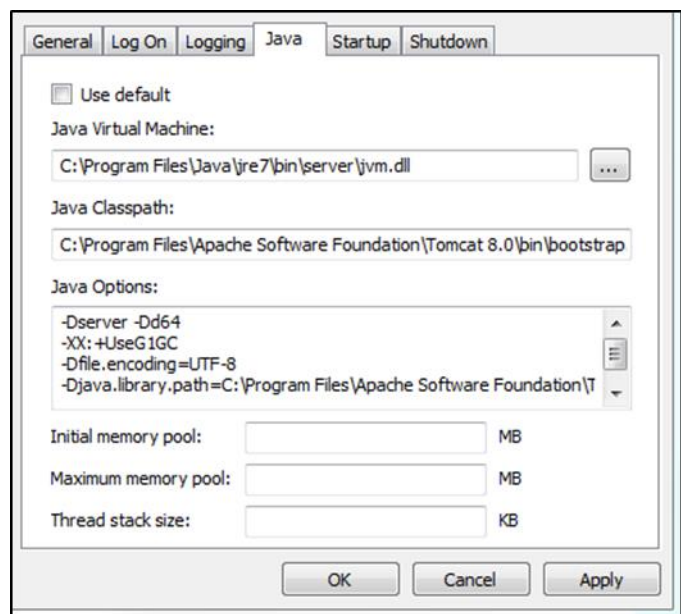
2. **FOR WINDOWS ONLY:**

In the Tomcat Java options In the **Java Options** field, add the following to the end of the options field:

```
-Djava.library.path=<path to  
Tomcat>\webapps\Thingworx\  
WEB-INF\extensions
```

Djava.library.path example:

```
-  
Djava.library.path=C:\Prog  
ram Files\Apache Software  
Foundation\Tomcat  
8.0\webapps\Thingworx\WEB-  
INF\extensions
```



3. **FOR UBUNTU ONLY:**

Create the Tomcat service file:

```
$ sudo touch /etc/systemd/system/tomcat8.service
```

Open `/etc/systemd/system/tomcat8.service` in a text editor (as root) and paste in the following:

```
[Unit]  
Description=Apache Tomcat Web Application Container  
After=network.target  
  
[Service]  
Type=forking  
PIDFile=/var/run/tomcat.pid  
Environment=CATALINA_PID=/var/run/tomcat.pid
```

Upgrading to ThingWorx 7.4

```
Environment=JAVA_HOME=/usr/lib/jvm/jdk1.8.0_92
Environment=CATALINA_HOME=/usr/share/tomcat8/8.0.33
Environment=CATALINA_BASE=/usr/share/tomcat8/8.0.33
Environment=CATALINA_OPTS=

ExecStart=/usr/share/tomcat8/8.0.33/bin/jsvc \
        -Dcatalina.home=${CATALINA_HOME} \
        -Dcatalina.base=${CATALINA_BASE} \
        -Djava.awt.headless=true -
Djava.net.preferIPv4Stack=true -Dserver -Dd64 -XX:+UseNUMA \
        -XX:+UseConcMarkSweepGC -
Dfile.encoding=UTF-8 \
        -
Djava.library.path=${CATALINA_BASE}/webapps/Thingworx/WEB-INF/extensions
\
        -cp ${CATALINA_HOME}/bin/commons-
daemon.jar:${CATALINA_HOME}/bin/bootstrap.jar:${CATALINA_HOME}/bin/tomcat
-juli.jar \
        -user tomcat8 \
        -java-home ${JAVA_HOME} \
        -pidfile /var/run/tomcat.pid \
        -errfile ${CATALINA_HOME}/logs/catalina.out
\
        -outfile ${CATALINA_HOME}/logs/catalina.out
\
        ${CATALINA_OPTS} \
        org.apache.catalina.startup.Bootstrap

[Install]
WantedBy=multi-user.target
```

4. FOR RHEL ONLY:

Create the Tomcat service file:

```
$ sudo touch /usr/lib/systemd/system/tomcat.service
```

Open /usr/lib/systemd/system/tomcat.service in a text editor (as root) and paste in the following:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking
PIDFile=/var/run/tomcat.pid
Environment=CATALINA_PID=/var/run/tomcat.pid
Environment=JAVA_HOME=/usr/lib/jvm/jdk1.8.0_92
Environment=CATALINA_HOME=/usr/share/tomcat8/8.0.33
Environment=CATALINA_BASE=/usr/share/tomcat8/8.0.33
Environment=CATALINA_OPTS=

ExecStart=/usr/share/tomcat8/8.0.33/bin/jsvc \
        -Dcatalina.home=${CATALINA_HOME} \
```


Upgrading to ThingWorx 7.4

```
-Dcatalina.base=${CATALINA_BASE} \  
-Djava.awt.headless=true - \  
Djava.net.preferIPv4Stack=true -Dserver -Dd64 -XX:+UseNUMA \  
-XX:+UseConcMarkSweepGC - \  
Dfile.encoding=UTF-8 \  
- \  
Djava.library.path=${CATALINA_BASE}/webapps/Thingworx/WEB- \  
INF/extensions \  
-cp ${CATALINA_HOME}/bin/commons- \  
daemon.jar:${CATALINA_HOME}/bin/bootstrap.jar:${CATALINA_HOME}/bin/tom- \  
cat-juli.jar \  
-user tomcat8 \  
-java-home ${JAVA_HOME} \  
-pidfile /var/run/tomcat.pid \  
-errfile \  
${CATALINA_HOME}/logs/catalina.out \  
-outfile \  
${CATALINA_HOME}/logs/catalina.out \  
$CATALINA_OPTS \  
org.apache.catalina.startup.Bootstrap  
  
[Install]  
WantedBy=multi-user.target
```

5. FOR UBUNTU AND RHEL ONLY:

Update the service:

```
sudo systemctl daemon-reload
```

6. Restart Tomcat.

Migrating to ThingWorx 7.4 (Neo4j/PostgreSQL/ MS SQL Server /H2) 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 7.4 \(Neo4j/PostgreSQL/H2\) 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.

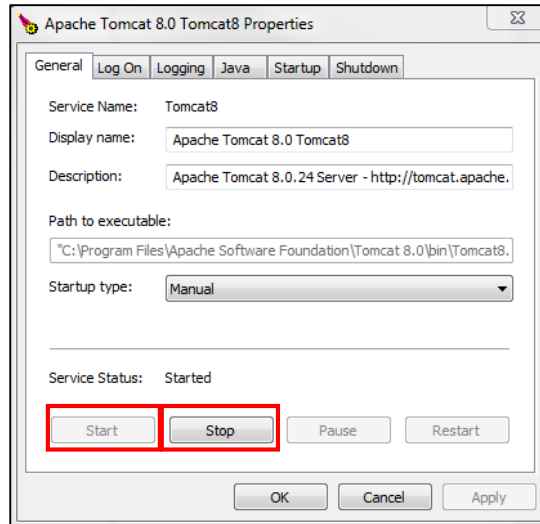
NOTE: Refer to the [System Requirements and Compatibility Matrix](#) document for version requirements.

Upgrading to ThingWorx 7.4

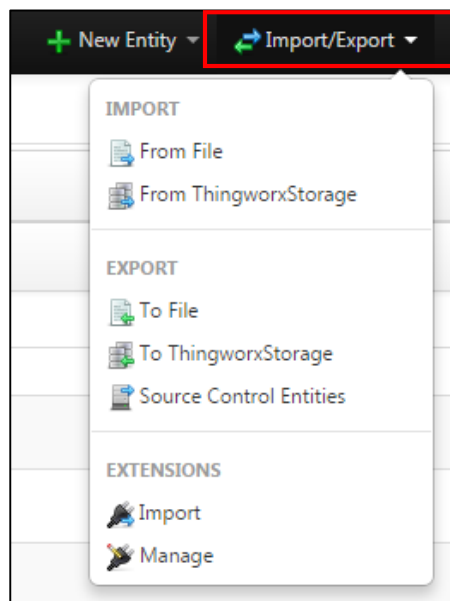
- Restart Tomcat: In the Tomcat Properties, click **Stop**. Wait for Tomcat to stop.
- It is highly recommended to back up the following two folders before continuing:
 - Apache Software Foundation/Tomcat 8.0/webapps/Thingworx**
 - <drive>:\ThingworxStorage**
Where <drive> is the drive Tomcat is installed on.
- If you are upgrading from 6.0 or above, remove the **validation.properties** file from **/ThingworxStorage/esapi**

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

- Click **Start**.
Restarting Tomcat assures that the database is clear before exporting.
- Export entities and data. In Composer, click **Import/Export>To ThingworxStorage**.



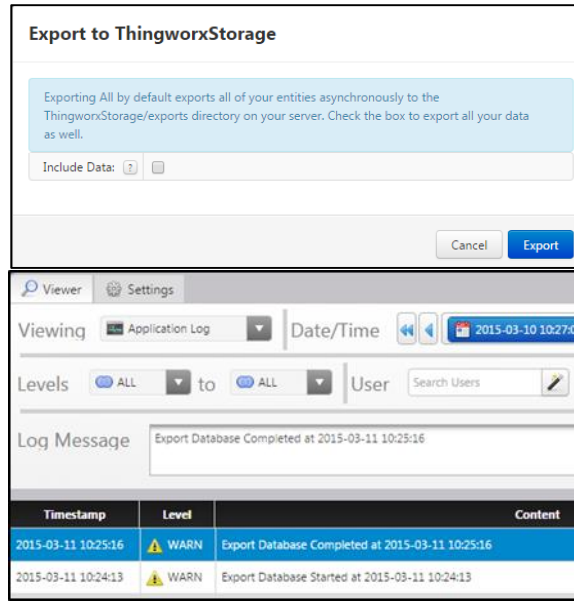
NOTE: The **validation.properties** file is created upon startup of ThingWorx. If you do not remove the file, the updated file with additional parameters will not overwrite the current version during upgrade. Reference the [Help Center](#) for additional information.



8. 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. Note any extensions that are in use.
12. Delete the **ThingworxStorage** and **ThingworxBackupStorage** folders.
13. Move the export files from step 10 back to the **ThingworxStorage\exports** folder.
14. Go to the Tomcat installation at **\Apache Software Foundation\Tomcat 8.0\webapps** and delete the **Thingworx.war** file.
15. Copy the **Thingworx.war** file and place it in the following location of your Tomcat installation:
\Apache Software Foundation\Tomcat 8.0\webapps
16. To launch ThingWorx, go to **<servername>/Thingworx** in a web browser.

NOTE: Use a strong password. The login information below is for the Administrator user only.

Use the following login information:
Login Name: Administrator
Password: admin

17. If necessary, import any extensions.

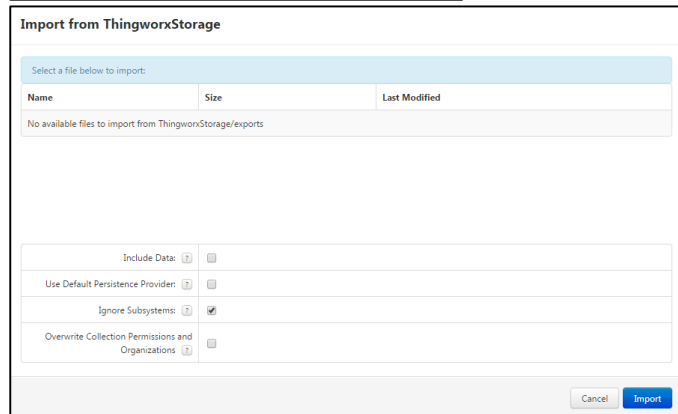
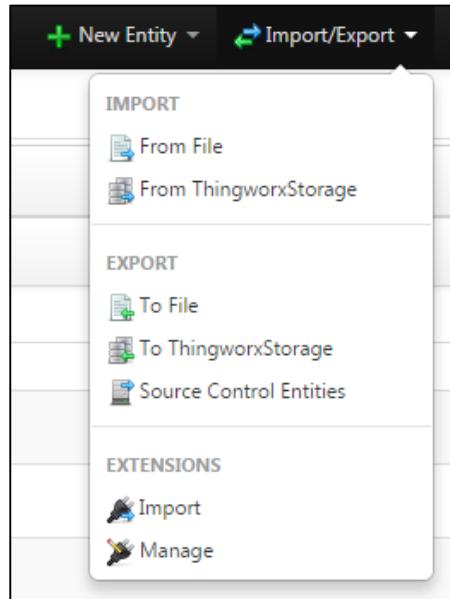
NOTE: Obtain and import the latest versions of the extensions. If you are upgrading to a major version (for example, from 6.x to 7.x, you must import the 7.x versions of the extensions.) Extensions are available in the [ThingWorx Marketplace](#).

18. If you have visibility permissions set in your model, and are importing **From File** you must delete the **Users** group from the **Everyone** organization in ThingWorx. See [“Removing the Users Group from the Everyone Organization”](#) for detailed steps.

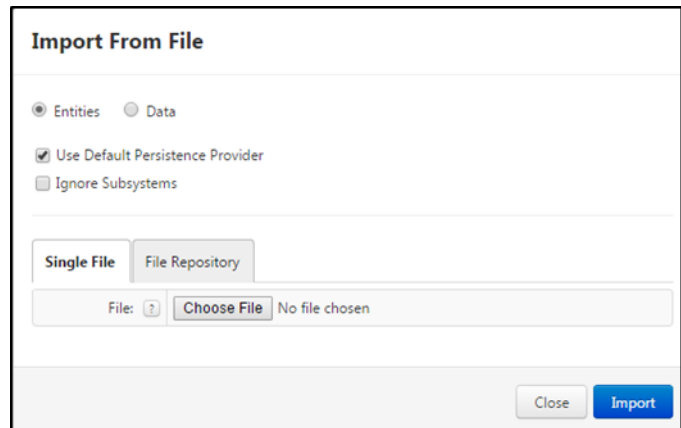
CAUTION: Failing to perform this step will result in your permissions being overwritten when performing the import in the next step.

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.

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



20. 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 **Ignore Subsystems** checkbox if you want to ignore the Subsystem settings of the imported entities (for example, if you are going from QA to production).



[In-place Upgrade to ThingWorx 7.4 \(Neo4j/PostgreSQL/MS SQL Server/H2\) 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 7.4 \(Neo4j/PostgreSQL/MS SQL Server/H2\) on Windows](#).

NOTE: In-place upgrades for Neo4j should only be performed for service pack level upgrades (for example, 7.4.0 to 7.4.1). Refer to [Migrating to ThingWorx 7.4](#) for major or minor release upgrades (for example, from 7.3.0 to 7.4.0).

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. It is highly recommended to backup the following two folders before continuing:
 - a. **Apache Software Foundation/Tomcat 8.0/webapps/Thingworx**
 - b. **<drive>:\ThingworxStorage**Where <drive> is the drive Tomcat is installed on.

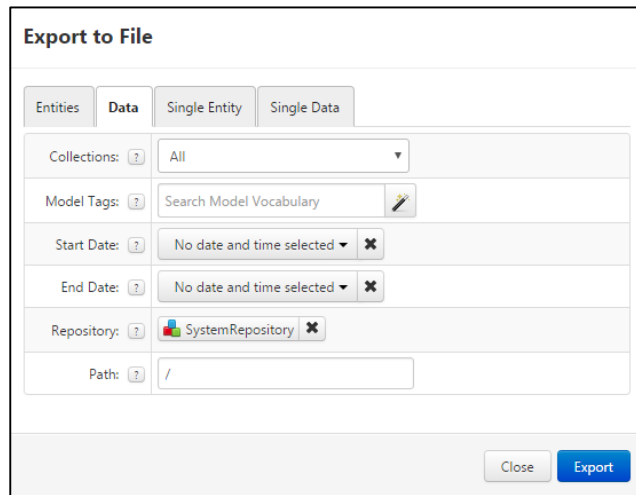
- If you are upgrading from 6.0 or above, remove the **validation.properties** file from **/ThingworxStorage/esapi**

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

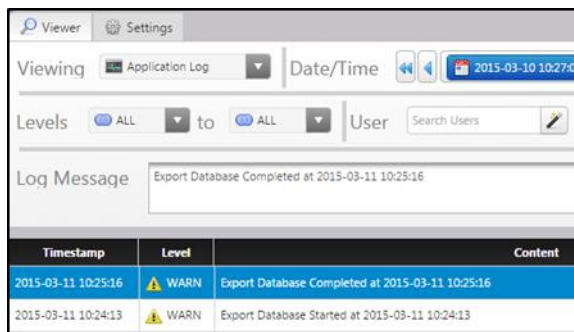
NOTE: The **validation.properties** file is created upon startup of ThingWorx. If you do not remove the file, the updated file with additional parameters will not overwrite the current version during upgrade. Reference the [Help Center](#) for additional information.

- 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.



- Export wiki data.
NOTE: Perform this step only if you have wiki data.
- Go to the Tomcat installation at **\Apache Software Foundation\Tomcat 8.0\webapps** and delete the **Thingworx.war** file.

8. **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**

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

\Apache Software Foundation\Tomcat 8.0\webapps

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

NOTE: Use a strong password. The login information below is for the Administrator user only.

Use the following login information:

Login Name: Administrator

Password: admin

11. 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.

[Migrating to ThingWorx 7.4 \(Neo4j/PostgreSQL/MS SQL Server/H2\) on Ubuntu](#)

The steps below are for migration only. For in-place upgrade, refer to [In-place Upgrade to ThingWorx 7.4 \(Neo4j/PostgreSQL/H2\) 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>To ThingworxStorage**.

*For in-place migration using DSE (Neo4j or 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 the **ThingworxStorage** and **ThingworxBackupStorage** folders to a temporary location for later retrieval.

6. It is highly recommended to back up the contents of the following folders before continuing:
 - **Apache Software Foundation/Tomcat8.0/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.

8. Note any extensions that are in use (located in **/ThingworxStorage/extensions**).

9. Delete the **/ThingworxStorage** and **/ThingworxBackupStorage** folders.

10. Move the exports file back to the **ThingworxStorage\exports** folder.

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

Tomcat8.0/webapps/Thingworx.

12. Verify that the **license.bin** file is copied to **ThingworxPlatform**. Reference [Licensing Changes in 7.4 \(Windows/Ubuntu/RHEL\)](#).

13. Start Tomcat.

```
$ sudo service tomcat8 start
```


14. Undeploy ThingWorx by deleting the **Thingworx.war** file from **/usr/share/tomcat8/8.0.33/webapps**

```
$ sudo rm /usr/share/tomcat8/8.0.33/webapps/Thingworx.war
```

15. **THIS STEP IS FOR POSTGRESQL ONLY. SKIP AND PROCEED TO THE NEXT STEP IF YOU ARE NOT UPGRADING FROM POSTGRESQL.** If you are performing an in-place migration from PostgreSQL 6.5 or later to 7.4, 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**

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

Move the **Thingworx.war** file to **/usr/share/tomcat8/8.0.33/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-074_F000_ThingWorx-Platform-Postgres-7.4.0.latest.zip
$ sudo mv Thingworx.war $CATALINA_HOME/webapps
$ sudo chown tomcat8:tomcat8 $CATALINA_HOME/webapps/Thingworx.war
$ sudo chmod 775 $CATALINA_HOME/webapps/Thingworx.war
```

17. Start Tomcat to deploy the ThingWorx web application:

```
$ sudo service tomcat8 start
```

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

NOTE: Use a strong password. The login information below is for the Administrator user only.

Use the following login information:

Login Name: Administrator

Password: admin

19. Import extensions. In Composer, 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 6.x to 7.4, you must import the 7.x versions of the extensions.) Extensions are available in the [ThingWorx Marketplace](#).

NOTE: For in-place migration from 6.5 to 7.4 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.

20. If you have visibility permissions set in your model, you must delete the **Users** group from the **Everyone** organization in ThingWorx. See [“Removing the Users Group from the Everyone Organization”](#) for detailed steps.

CAUTION: Failing to perform this step will result in your permissions being overwritten when performing the import in the next step.

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

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

In-place Upgrade to ThingWorx 7.4 (Neo4j/PostgreSQL/MS SQL Server/H2) 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.

NOTE: In-place upgrades for Neo4j should only be performed for service pack level upgrades (for example, 7.4.0 to 7.4.1). Refer to [Migrating to ThingWorx 7.4](#). for major or minor release upgrades (for example, from 7.3.0 to 7.4.0).

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. It is highly recommended to backup the following two folders before continuing:
 - a. **Apache Software Foundation/Tomcat 8.0/webapps/Thingworx**
 - b. **/ThingworxStorage**

4. If you are upgrading from 6.0 or above, 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.

5. **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.

Timestamp	Level	Content
2015-03-11 10:25:16	WARN	Export Database Completed at 2015-03-11 10:25:16
2015-03-11 10:24:13	WARN	Export Database Started at 2015-03-11 10:24:13

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

22. Move the **Thingworx.war** file to **/usr/share/tomcat8/8.0.33/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-074_F000_ThingWorx-Thingworx-Platform-Postgres-7.4.0.zip
$ sudo mv Thingworx.war /usr/share/tomcat8/8.0.33/webapps
$ sudo chown tomcat8:tomcat8 /usr/share/tomcat8/8.0.33/webapps/Thingworx.war
$ sudo chmod 775 /usr/share/tomcat8/8.0.33/webapps/Thingworx.war
```

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

NOTE: Use a strong password. The login information below is for the Administrator user only.

Use the following login information:

Login Name: Administrator

Password: admin

9. 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.

Migrating to ThingWorx 7.4 (Neo4j/PostgreSQL/MS SQL Server/H2) on RHEL

The steps below are for migration only. For in-place upgrade, refer to [In-place Upgrade to ThingWorx 7.4 \(Neo4j/PostgreSQL/H2\) 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>To ThingworxStorage**.

*For in-place migration using DSE (Neo4j or 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 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/tomcat8/8.0.33/webapps/Thingworx**
- **/ThingworxStorage**
- **/ThingworxBackupStorage**

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

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

9. Remove the contents of the following folder:

/usr/share/tomcat8/8.0.33/webapps/Thingworx

```
$ sudo rm -r /usr/share/tomcat8/8.0.33/webapps/Thingworx/*
```

10. Undeploy ThingWorx by deleting the **Thingworx.war** file from **/usr/share/tomcat8/8.0.33/webapps**

```
$ sudo rm /usr/share/tomcat8/8.0.33/webapps/Thingworx.war
```

11. **THIS STEP IS FOR POSTGRES ONLY. SKIP AND PROCEED TO THE NEXT STEP IF YOU ARE NOT UPGRADING FROM POSTGRES.** If you are performing an in-place migration from PostgreSQL 6.5 or later to 7.4, 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**

12. Unzip the ThingWorx zip archive to a temporary directory.
13. Move the **Thingworx.war** file to **/usr/share/tomcat8/8.0.33/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-074_F000_ThingWorx-Thingworx-Platform-Postgres-7.4.0.zip
$ sudo mv Thingworx.war /usr/share/tomcat8/8.0.33/webapps
$ sudo chown tomcat8:tomcat8 /usr/share/tomcat8/8.0.33/webapps/Thingworx.war
$ sudo chmod 775 /usr/share/tomcat8/8.0.33/webapps/Thingworx.war
```

14. Verify that the **license.bin** file has been placed in **/ThingworxPlatform**. Reference [Licensing Changes in 7.4 \(Windows/Ubuntu/RHEL\)](#).
15. Start Tomcat to deploy the ThingWorx web application:

```
$ sudo systemctl start tomcat
```

16. 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:tomcat8 /ThingworxStorage/exports/*
```

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

NOTE: Use a strong password. The login information below is for the Administrator user only. Use the following login information:
Login Name: Administrator
Password: admin

18. Import extensions. In Composer, 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 6.x to 7.4, you must import the 7.x versions of the extensions.) Extensions are available in the [ThingWorx Marketplace](#).

NOTE: For in-place migration from 6.5 to 7.4 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.

19. If you have visibility permissions set in your model, you must delete the **Users** group from the **Everyone** organization in ThingWorx. See "[Removing the Users Group from the Everyone Organization](#)" for detailed steps.

CAUTION: Failing to perform this step will result in your permissions being overwritten when performing the import in the next step.

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.

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

[In-place Upgrade to ThingWorx 7.4 \(Neo4j/PostgreSQL/MS SQL Server/H2\) 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.

NOTE: In-place upgrades for Neo4j should only be performed for service pack level upgrades (for example, 7.4.0 to 7.4.1). Refer to [Migrating to ThingWorx 7.4](#) for major or minor release upgrades (for example, from 7.3.0 to 7.4.0).

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. It is highly recommended to backup the following folders before continuing:
 - a. **/usr/share/tomcat8/8.0.33/webapps/Thingworx**
 - b. **/ThingworxStorage**
 - c. **/ThingworxBackupStorage**

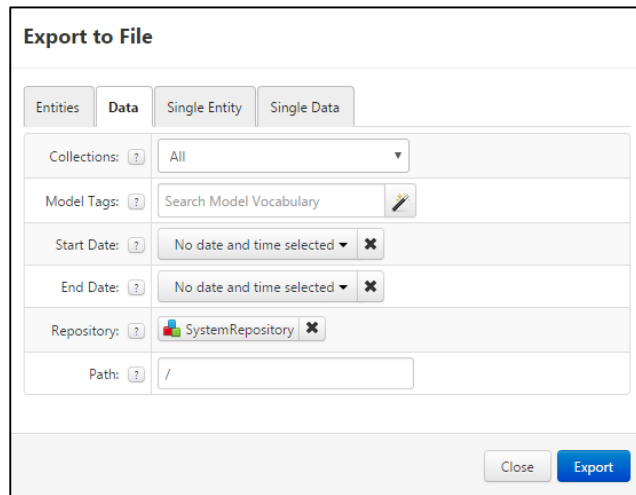
- If you are upgrading from 6.0 or above, backup the **validation.properties** file from **/ThingworxStorage/esapi**

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

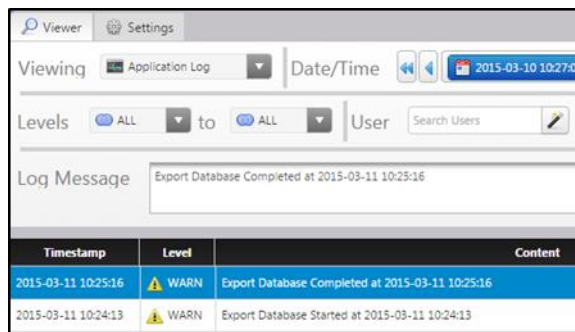
NOTE: The **validation.properties** file is created upon startup of ThingWorx. If you do not remove the file, the updated file with additional parameters will not overwrite the current version during upgrade. Reference the [Help Center](#) for additional information.

- 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.



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

- Stop Tomcat:

```
$ sudo systemctl stop tomcat
```

- Go to the Tomcat installation at **/usr/share/tomcat8/8.0.33/webapps** and delete the **Thingworx.war** file:

```
$ sudo rm /usr/share/tomcat8/8.0.33/webapps/Thingworx.war
```

9. 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**

10. Unzip the ThingWorx zip archive to a temporary directory. Move the **Thingworx.war file to **/usr/share/tomcat8/8.0.33/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-074_F000_ThingWorx-Platform-Postgres-7.4.0.zip
$ sudo mv Thingworx.war /usr/share/tomcat8/8.0.33/webapps
$ sudo chown tomcat8:tomcat8
/usr/share/tomcat8/8.0.33/webapps/Thingworx.war
$ sudo chmod 775 /usr/share/tomcat8/8.0.33/webapps/Thingworx.war
```

11. Verify that the **license.bin file has been placed in **/ThingworxPlatform**. Reference [Licensing Changes in 7.4 \(Windows/Ubuntu/RHEL\)](#).**

12. Start Tomcat

```
$ sudo systemctl start tomcat
```

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

NOTE: Use a strong password. The login information below is for the Administrator user only.

Use the following login information:

Login Name: Administrator

Password: admin

14. 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.

Appendix A: Licensing Troubleshooting

You must have a license file for ThingWorx 7.4 and later. Some possible situations that may require troubleshooting are described below:

Issue	Possible Resolution
<p>The following error is received when deploying ThingWorx:</p> <pre> org.apache.catalina.core.Applicati onContext.log HTMLManager: FAIL - Deploy Upload Failed, Exception: org.apache.tomcat.util.http.fileup load.FileUploadBase\$SizeLimitExcee dedException: the request was rejected because its size (90883556) exceeds the configured maximum (52437800) java.lang.IllegalStateException: org.apache.tomcat.util.http.fileup load.FileUploadBase\$SizeLimitExcee dedException: the request was rejected because its size (90883556) exceeds the configured maximum (52437800) at org.apache.catalina.connector.Requ est.parseParts(Request.java:2871 </pre>	<p>The max file size in the Tomcat web.xml file must be increased (default is 50MB). This file is located at :</p> <p><path to Tomcat>\Apache Software Foundation\Tomcat 8.5\webapps\manager\WEB-INF</p> <ol style="list-style-type: none"> 1. Open the web.xml. 2. Change the max-file-size and max-request-size to 104857600. 3. Save and close the file.
<p>The following error message is received when attempting to undeploy ThingWorx:</p> <pre> 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) </pre>	<p>Remove -Djava.library.path from Tomat’s Java configuration before undeployment.</p>

Issue	Possible Resolution
<p>An error message similar to the following is seen in the <i>ConfigurationLog.log</i>:</p> <p>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.</p> <pre> 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 ANALYSIS </pre>	<p>Set the <code>java.library.path</code> variable in the <code>setenv.sh</code> file or in the <code>tomcat8</code> startup <code>init.d</code> script.</p>