Standalone Dashboard upgrade

Summary: This section describes the upgrade process for standalone Dashboards.

If you are using the Dashboards embedded in AIP Console, this step is **not required**. Instead, please see **CAST RESTAPI integrated upgrade** for more information.

Information and prerequisites

✓	**Upgrade process**	The upgrade process is a "**side-by-side upgrade**". This means that the new dashboard WAR/ZIP file should be deployed alongside the existing WAR/ZIP/JAR file and any settings transferred from the existing to the new.
✓	**Where to run the upgrade?**	The process must be run on the host server.
✓	**Obtain the installation media**	Download the latest release of the required CAST Dashboard from **CAST Extend**. This media contains: • a **WAR file** for deployment on Apache Tomcat (all releases) • a **ZIP file** for deployment without Apache Tomcat (2.0 - 2.4 releases) • a **JAR file** for deployment without Apache Tomcat (2.5 releases)
✓	**Access to a CAST Storage Service /PostgreSQL instance (2.1)**	When upgrading to **2.1** a **graphical user interface** has been implemented for managing the assignment of **role and data authorizations** to users and groups of users. This interface replaces the existing mechanism provided by the **roles.xml** and the **authorizations.xml** files. This new management interface relies on a **schema hosted on a CAST Storage Service /PostgreSQL instance** to store the roles/data authorizations. This schema is called **cast_dashboards** by default and will be created on first startup of the web application.

✅	**Stop all services**	Before starting any upgrade process, you must ensure ALL services are **stopped**, for example stop **Apache Tomcat** or any existing Dashboards running in SpringBoot mode.

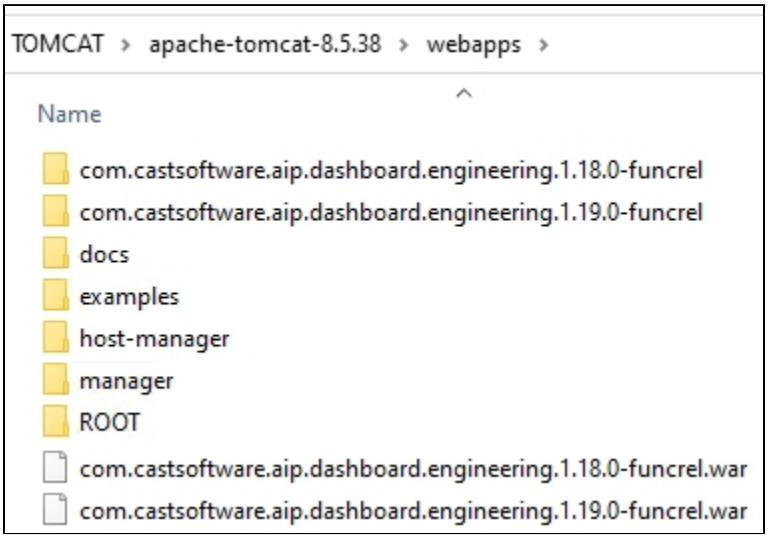
Instructions for transitioning between 1.x WAR files with Apache Tomcat

Step 1 - Deploy the new .WAR file

Obtain the new **.WAR** file and unzip it to the following location:

```
CATALINA_HOME\webapps
```

This should give you the following file hierarchy, for example when upgrading the CAST Engineering Dashboard from 1.18 to 1.19:



Step 2 - Reconfigure your Dashboard

In a "side-by-side" deployment, you will need to transfer the details of your environment (stored in the configuration files such as **web.xml/context.xml /other .properties/.xml/.json files**) from the old Dashboard to the new. Please be aware that the **location, structure and content** of configuration files **may have changed between releases**, so you should **not** perform a merge or overwrite configuration files (unless specifically requested to do so in the instructions below) - instead you must manually update the new configuration files with the details from the old configuration files.

Relevant config files

The following files may contain configuration that you will need to copy to the equivalent file in the newly deployed Dashboard:

```
CATALINA_HOME\webapps\<dashboard>\META-INF\context.xml
CATALINA_HOME\webapps\<dashboard>\WEB-INF\authorizations.xml
CATALINA_HOME\webapps\<dashboard>\WEB-INF\domains.properties
CATALINA_HOME\webapps\<dashboard>\WEB-INF\license.key >>> only for the Engineering Dashboard. This file will
not exist in the new Dashboard and can therefore be copied over as is.
CATALINA_HOME\webapps\<dashboard>\WEB-INF\license.xml
CATALINA_HOME\webapps\<dashboard>\WEB-INF\log4j2.xml
CATALINA_HOME\webapps\<dashboard>\WEB-INF\report.properties
CATALINA_HOME\webapps\<dashboard>\WEB-INF\roles.xml
CATALINA_HOME\webapps\<dashboard>\WEB-INF\security.properties
CATALINA_HOME\webapps\<dashboard>\WEB-INF\users.properties

If you have added custom languages, these can be found in the following folders:

CATALINA_HOME\webapps\<dashboard>\portal\locales
CATALINA_HOME\webapps\<dashboard>\engineering\locales
```

.json layout files

If you have made configuration changes to .JSON files to create custom tiles or layout and you want to retain those changes in the new version of the Dashboard, you can re-use old .JSON files with the new Dashboard, however, please note that doing so will mean that you will not benefit from any improvements or features added to the current release of the Dashboard. Files that may contain customizations are listed below:

```
Engineering/Security Dashboard
CATALINA_HOME\webapps\<dashboard>\engineering\resources\ed.json

Health Dashboard
CATALINA_HOME\webapps\<dashboard>\portal\resources\app.json
CATALINA_HOME\webapps\<dashboard>\portal\resources\cmp.json
CATALINA_HOME\webapps\<dashboard>\portal\resources\internal\app-navigation.json
```

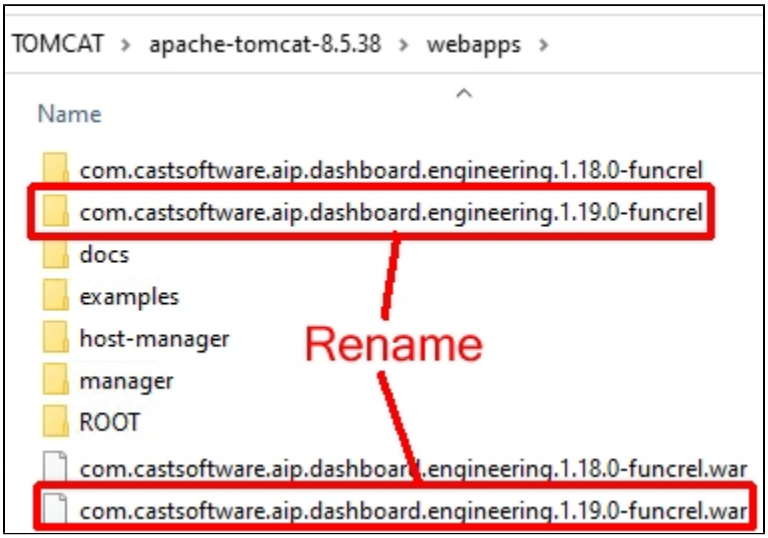
⚠ In **1.23**, the **app-navigation.json** file has been removed. It is no longer used.

Step 3 - Undeploy the existing WAR file (optional)

Undeploy the existing **war** file from Apache Tomcat - you can use Apache Tomcat manager to do this, or simply delete the relevant folder and file in `CATALINA_HOME\webapps`. This step is optional and you may wish to leave this in place until the new dashboard is fully tested and working.

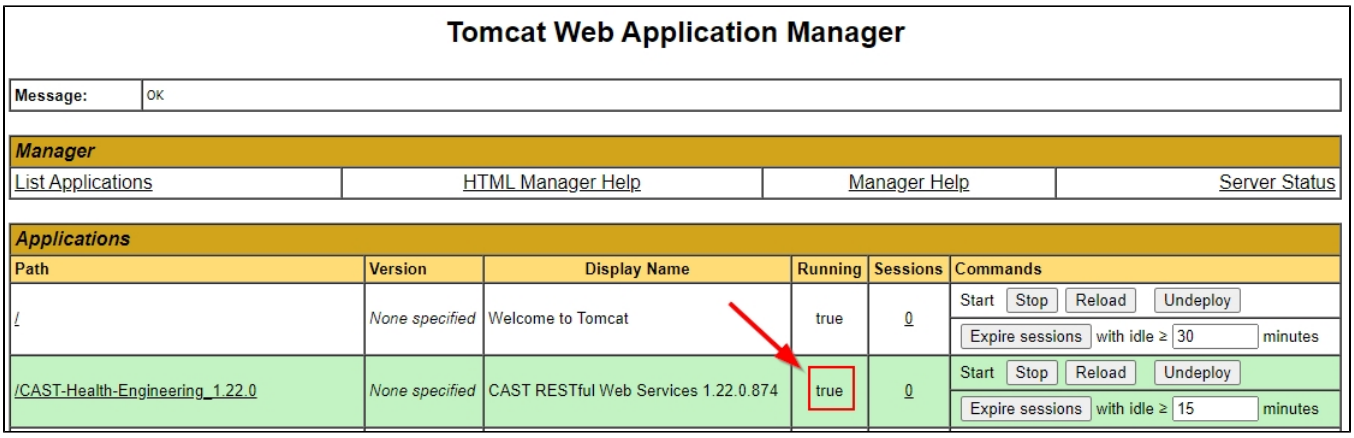
Step 4 - Rename the new WAR file and corresponding folder

If you use a specific naming convention to access the dashboards, you may need to rename the .WAR file and its corresponding folder (the two must match). The name of these two items forms the URL to access the dashboards:



Step 5 - Restart Apache Tomcat

Restart Apache Tomcat and ensure that the **webapp** has started correctly using **Apache Tomcat manager**:



Step 6 - Check access to CAST Dashboards

Browse to your Dashboard URL, login and check that you can access the data you require.

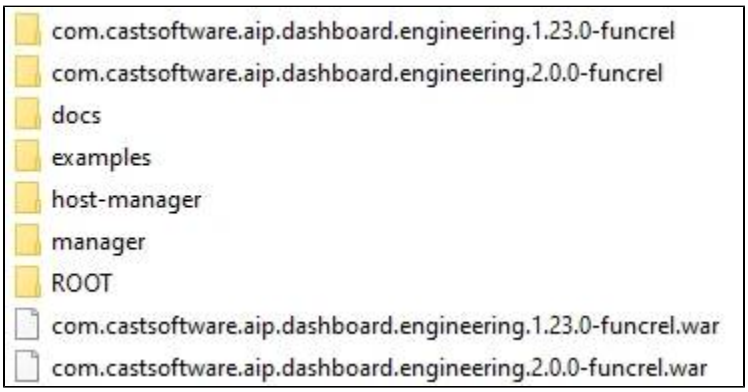
Instructions for transitioning between 1.x and 2.x WAR files with Apache Tomcat

Step 1 - Deploy the new .WAR file

Obtain the new **.WAR** file and unzip it to the following location:

```
CATALINA_HOME\webapps
```

This should give you the following file hierarchy, for example when upgrading the CAST Engineering Dashboard from 1.23 to 2.0.0:



Step 2 - Reconfigure your Dashboard

In a "side-by-side" deployment, you will need to transfer the details of your environment (stored in the configuration files such as **web.xml/context.xml /other .properties/.xml/.json files**) from the old Dashboard to the new. Please be aware that the **location, structure and content of configuration files HAS CHANGED in the 2.x WAR files**, so you should **not** perform a merge or overwrite configuration files (unless specifically requested to do so in the instructions below) - instead you must manually update the new configuration files with the details from the old configuration files.

The following files may contain configuration that you will need to copy to the equivalent file in the newly deployed Dashboard:

1.x WAR file	2.x WAR file	Notes

\META-INF**context.xml**	Does not exist. Use new sections in **\WEB-INF\classes\application.properties**.	Configuration has changed. Refer to the following documentation (**Step 2**) for more information: • **Standalone Engineering Dashboard deployment using WAR file** • **Standalone Health Dashboard deployment using WAR file**

\WEB-INF**authorizations.xml**	**\WEB-INF\classes\authorizations.xml**	See **Data authorization.**
		• **2.0.x release** - No change in configuration method.
		• **2.1.0 release** - **authorizations.xml** has been replaced with a graphical user interface to manage data authorizations for users/groups. Configuration is stored in a schema called "cast_dashboards" on a chosen CAST Storage Service/PostgreSQL instance. There are two choices for how to proceed:
		  • You can provide your authorizations.xml file and on first start-up of the web application the authorizations will be automatically transferred into the management interface (i.e. the schema called "cast_dashboards"). The authorizations.xml file will then be ignored on any subsequent web application startup and all changes from then on must be performed using the graphical interface.
		  • You can assign the data authorizations from scratch using the graphical interface.

\WEB-INF**domains.properties**	**\WEB-INF\classes\domains.properties**	Configuration has changed. Refer to the following documentation (**Step 2**) for more information: • **Standalone Engineering Dashboard deployment using WAR file** • **Standalone Health Dashboard deployment using WAR file**
\WEB-INF**license.key**	Only for the Engineering Dashboard. This file will not exist in the new Dashboard and can therefore be copied over as is to: **\WEB-INF\classes\license.key**	No change in configuration method. See **Dashboard Service license key configuration**.
\WEB-INF**license.xml**	**\WEB-INF\classes\license.xml**	No change in configuration method. See **Dashboard Service license key configuration** and **Data authorization**.
\WEB-INF**log4j2.xml**	Does not exist. Use **\WEB-INF\classes\log4j2-spring.xml**	No change in configuration method. See **Configuring the Log and Audit Trail**.
\WEB-INF**report.properties**	Does not exist. Use equivalent section in **\WEB-INF\classes\application.properties**.	No change in configuration method. See **CAST Report Generator - CAST Report Generator for Dashboards**.

\WEB-INF**roles.xml**	**\WEB-INF\classes\roles.xml**	See **User roles**.
		• **2.0.x release** - No change in configuration method.
		• **2.1.0 release** - **roles.xml** has been replaced with a graphical user interface to manage role assignments for users/groups. Configuration is stored in a schema called "cast_dashboards" on a chosen CAST Storage Service/PostgreSQL instance. There are two choices for how to proceed:
		  • You can provide your roles.xml file and on first start-up of the web application the role assignments will be automatically transferred into the management interface (i.e. the schema called "cast_dashboards"). The roles.xml file will then be ignored on any subsequent web application startup and all changes from then on must be performed using the graphical interface.
		  • You can assign the roles from scratch using the graphical interface.

\WEB-INF**security.properties**	Does not exist. Use equivalent section in **\WEB-INF\classes\application.properties**.	Configuration has changed. Refer to the following documentation: **User authentication**.
\WEB-INF**users.properties**	**\WEB-INF\classes\users.properties**	No change in configuration method. See **User authentication**.
\engineering\resources**ed.json**	**WEB-INF\classes\config\ed\ed.json**	If you have made configuration changes to .JSON files to create custom tiles or layout and you want to retain those changes in the new version of the Dashboard, you can re-use old .JSON files with the new Dashboard, however, please note that doing so will mean that you will not benefit from any improvements or features added to the current release of the Dashboard.
\portal\resources**app.json**	**WEB-INF\classes\config\hd\app.json**	

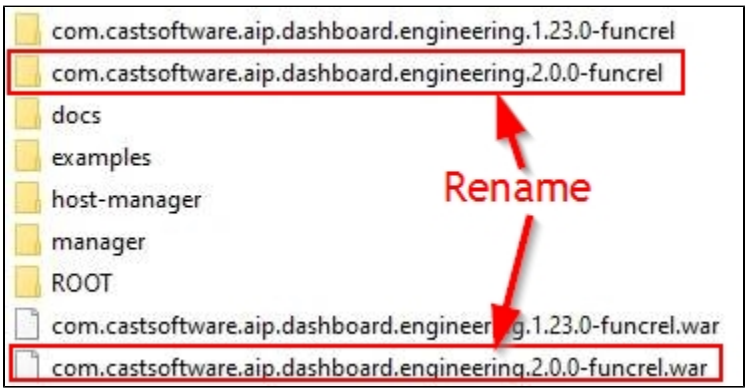
\portal\resources**cmp.json**	**WEB-INF\classes\config\hd\cmp.json**	
\portal\resources\internal**app-navigation.json**	Does not exist (withdrawn in **1.23)**.	-

Step 3 - Undeploy the existing WAR file (optional)

Undeploy the existing **war** file from Apache Tomcat - you can use Apache Tomcat manager to do this, or simply delete the relevant folder and file in `CATALINA_HOME\webapps`. This step is optional and you may wish to leave this in place until the new dashboard is fully tested and working.

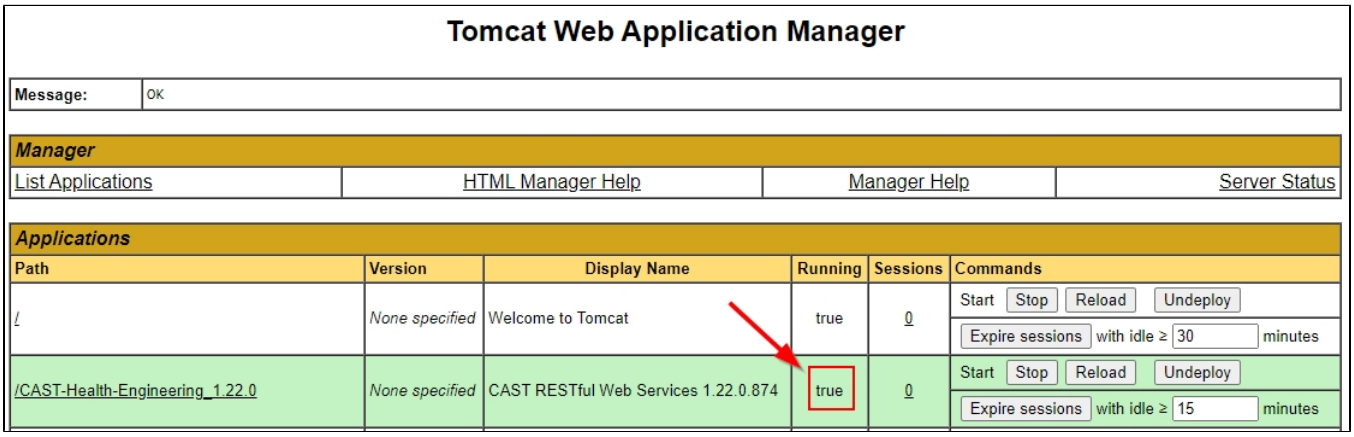
Step 4 - Rename the new WAR file and corresponding folder

If you use a specific naming convention to access the dashboards, you may need to rename the .WAR file and its corresponding folder (the two must match). The name of these two items forms the URL to access the dashboards:



Step 5 - Restart Apache Tomcat

Restart Apache Tomcat and ensure that the **webapp** has started correctly using **Apache Tomcat manager**:



Step 6 - Check access to CAST Dashboards

Browse to your Dashboard URL, login and check that you can access the data you require.

Step 7 - First login and become admin - 2.1 only

By default, the CAST Dashboard requires that at least one user is granted the **ADMIN role** following the first login after the **User authentication** configuration. This ensures that one user can access all data and configuration settings. This ensures that one user can access all data and configuration settings. This step MUST be performed from the machine on which the dashboard is running, using the URL **http://localhost:<port>** - if you attempt it from another machine on the network, you will not be able to "become admin". See **First login and become admin**. This step is not required when using **Dashboards 2.0** and can be skipped.

Instructions for transitioning between 1.x WAR file and 2.x ZIP file

From release **2.x** onwards, CAST is delivering a **ZIP file for each CAST Dashboard**, alongside the traditional **WAR file** that has always been delivered. The **ZIP file** is a method of deploying the CAST Dashboards based on **Spring Boot** and does not require a web application server (the application server is embedded in the ZIP itself). The aim of the ZIP file releases is to simplify and speed up the deployment of the CAST Dashboards. The deployment and configuration of the 2.x Spring Boot based dashboards differs slightly to the steps required for a traditional WAR file, mostly due to the fact that the files have moved and some have been removed.

To transition from 1.x WAR to 2.x ZIP is relatively simple although there is no managed "upgrade" process. You can repurpose the existing server hosting Apache Tomcat if necessary.

Step 1 - Deploy the new .ZIP file

Obtain the new **.ZIP** file and unzip it to any location on the host server.

Step 2 - Configure your new Dashboard

You will now need to transfer the details of your environment (stored in the configuration files such as **web.xml/context.xml/other .properties/.xml/.json files**) from the old Dashboard to the new. Please be aware that the **location, structure and content of configuration files HAS CHANGED in the ZIP files**, so you should **not** perform a merge or overwrite configuration files (unless specifically requested to do so in the instructions below) - instead you must manually update the new configuration files with the details from the old configuration files.

Use the following table to find the location of the equivalent configuration files in the new 2.x ZIP:

1.x WAR file	2.x ZIP file	Notes
\META-INF**context.xml**	Does not exist. Use new sections in **<unpacked_zip>\configurations\application.properties**.	Configuration has changed. Refer to the following documentation (**Step 2**) for more information: • **Standalone Engineering Dashboard deployment using ZIP file** • **Standalone Health Dashboard deployment using ZIP file**

\WEB-INF**authorizations.xml**	**<unpacked_zip>\configurations\authorizations.xml**	See **Data authorization.**
		• **2.0.x release** - No change in configuration method.
		• **2.1.0 release** - **authorizations.xml** has been replaced with a graphical user interface to manage data authorizations for users/groups. Configuration is stored in a schema called "cast_dashboards" on a chosen CAST Storage Service/PostgreSQL instance. There are two choices for how to proceed:
		  ◦ You can provide your authorizations.xml file and on first start-up of the web application the authorizations will be automatically transferred into the management interface (i.e. the schema called "cast_dashboards"). The authorizations.xml file will then be ignored on any subsequent web application startup and all changes from then on must be performed using the graphical interface.
		  ◦ You can assign the data authorizations from scratch using the graphical interface.

\WEB-INF**domains.properties**	**<unpacked_zip>\configurations\domains.properties**	Configuration has changed. Refer to the following documentation (**Step 2**) for more information: • **Standalone Engineering Dashboard deployment using ZIP file** • **Standalone Health Dashboard deployment using ZIP file**
\WEB-INF**license.key**	Only for the Engineering Dashboard. This file will not exist in the new Dashboard and can therefore be copied over as is to: **<unpacked_zip>\configurations\license.key**	No change in configuration method. See **Dashboard Service license key configuration**.
\WEB-INF**license.xml**	**<unpacked_zip>\configurations\license.xml**	No change in configuration method. See **Dashboard Service license key configuration** and **Data authorization**.

\WEB-INF**log4j2.xml**	Does not exist. Use **<unpacked_zip>\configurations\log4j2-spring.xml**	No change in configuration method. See **Configuring the Log and Audit Trail**.
\WEB-INF**report.properties**	Does not exist. Use equivalent section in **<unpacked_zip>\configurations\application.properties**.	No change in configuration method. See **CAST Report Generator - CAST Report Generator for Dashboards**.

\WEB-INF**roles.xml**	**<unpacked_zip>\configurations\roles.xml**	See **User roles**.
		• **2.0.x release** - No change in configuration method.
		• **2.1.0 release** - **roles.xml** has been replaced with a graphical user interface to manage role assignments for users/groups. Configuration is stored in a schema called "cast_dashboards" on a chosen CAST Storage Service/PostgreSQL instance. There are two choices for how to proceed:
		  • You can provide your roles.xml file and on first start-up of the web application the role assignments will be automatically transferred into the management interface (i.e. the schema called "cast_dashboards"). The roles.xml file will then be ignored on any subsequent web application startup and all changes from then on must be performed using the graphical interface.
		  • You can assign the roles from scratch using the graphical interface.

\WEB-INF**security.properties**	Does not exist. Use equivalent section in **<unpacked_zip>\configurations\application.properties**.	Configuration has changed. Refer to the following documentation: **User authentication**.
\WEB-INF**users.properties**	**<unpacked_zip>\configurations\users.properties**	No change in configuration method. See **User authentication**.
\engineering\resources**ed.json**	**<unpacked_zip>\configurations\config\ed\ed.json**	If you have made configuration changes to .JSON files to create custom tiles or layout and you want to retain those changes in the new version of the Dashboard, you can re-use old .JSON files with the new Dashboard, however, please note that doing so will mean that you will not benefit from any improvements or features added to the current release of the Dashboard.
\portal\resources**app.json**	**<unpacked_zip>\configurations\config\hd\app.json**	
\portal\resources**cmp.json**	**<unpacked_zip>\configurations\config\hd\cmp.json**	
\portal\resources\internal**app-navigation.json**	Does not exist (withdrawn in **1.23)**.	-

Step 3 - Start the new Dashboard and check access

To start the dashboard, run the following file:

```
<unpacked_zip>\startup.bat
<unpacked_zip>\startup.sh
```

By default the dashboard is configured to run on **port 8080**. Use the following URL - where **<server_name>** is equal to the host name of the current server to access the dashboard. If you are testing on the server itself, you can use `http://localhost:8080:`

```
http://<server_name>:8080
```

- If you have a service on the host server that is already using port 8080, please refer to **Modify the user access port for 2.x JAR or ZIP deployments** for more information.
- See also **Start and stop 2.x JAR or ZIP deployments**.

Step 4 - First login and become admin - 2.1 only

By default, the CAST Dashboard requires that at least one user is granted the **ADMIN role** following the first login after the **User authentication** configuration. This ensures that one user can access all data and configuration settings. This step MUST be performed from the machine on which the dashboard is running, using the URL **http://localhost:<port>** - if you attempt it from another machine on the network, you will not be able to "become admin". See **First login and become admin**. This step is not required when using **Dashboards 2.0** and can be skipped.

Step 5 - Undeploy the existing WAR file (optional)

Undeploy the existing **war** file from Apache Tomcat - you can use Apache Tomcat manager to do this, or simply delete the relevant folder and file in `CATALINA_HOME\webapps`. This step is optional and you may wish to leave this in place until the new dashboard is fully tested and working.

Instructions for transitioning between 2.x ZIP files

To transition between 2.x ZIP files (for example 2.0.0 to 2.1.0) is relatively simple although there is no managed "upgrade" process.

Step 1 - Deploy the new .ZIP file

Obtain the new **.ZIP** file and unzip it to any location on the host server.

Step 2 - Configure your new Dashboard

You will now need to transfer the details of your environment (stored in the configuration files such as **web.xml/context.xml/other .properties/.xml/.json files**) from the old Dashboard to the new. Please be aware that you should **not** perform a merge or overwrite configuration files (unless specifically requested to do so in the instructions below) - instead you must manually update the new configuration files with the details from the old configuration files.

Use the following table to find the location of the configuration files:

2.x ZIP file	Notes
<unpacked_zip>\configurations\application.properties.	Refer to the following documentation for more information: • **Standalone Engineering Dashboard deployment using ZIP file** • **Standalone Health Dashboard deployment using ZIP file** See also **CAST Report Generator - CAST Report Generator for Dashboards.**
<unpacked_zip>\configurations\authorizations.xml	See **Data authorization - 2.x and above**. **From 2.0.0 to 2.1.0 release** **authorizations.xml** has been replaced with a graphical user interface to manage data authorizations for users/groups. Configuration is stored in a schema called "cast_dashboards" on a chosen CAST Storage Service/PostgreSQL instance. There are two choices for how to proceed: • You can provide your authorizations.xml file and on first start-up of the web application the authorizations will be automatically transferred into the management interface (i.e. the schema called "cast_dashboards"). The authorizations.xml file will then be ignored on any subsequent web application startup and all changes from then on must be performed using the graphical interface. • You can assign the data authorizations from scratch using the graphical interface. **From 2.1.0 release** All authorization data is stored in the schema "cast_dashboards" and the authorizations.xml file is no longer used. CAST recommends re-using the "cast_dashboards" schema so that all existing authorizations are re-used. This is configured in the **<unpacked_zip>\configurations\application.properties** file.
<unpacked_zip>\configurations\domains.properties	Refer to the following documentation for more information: • **Standalone Engineering Dashboard deployment using ZIP file** • **Standalone Health Dashboard deployment using ZIP file**
Only for the Engineering Dashboard. This file will not exist in the new Dashboard and can therefore be copied over as is to: **<unpacked_zip>\configurations\license.key**	See **Dashboard Service license key configuration**.

<unpacked_zip>\configurations\license.xml	See **Dashboard Service license key configuration** and **Data authorization**.
<unpacked_zip>\configurations\log4j2-spring.xml	See **Configuring the Log and Audit Trail**.

<unpacked_zip>\configurations\roles.xml	See **User roles - 2.x and above**.
	
	From 2.0.0 to 2.1.0 release
	
	• **roles.xml** has been replaced with a graphical user interface to manage role assignments for users /groups. Configuration is stored in a schema called "cast_dashboards" on a chosen CAST Storage Service/PostgreSQL instance. There are two choices for how to proceed:
	  • You can provide your roles.xml file and on first start-up of the web application the role assignments will be automatically transferred into the management interface (i.e. the schema called "cast_dashboards"). The roles.xml file will then be ignored on any subsequent web application startup and all changes from then on must be performed using the graphical interface.
	  • You can assign the roles from scratch using the graphical interface.
	
	From 2.1.0 release
	
	All roles data is stored in the schema "cast_dashboards" and the roles.xml file is no longer used. CAST recommends re-using the "cast_dashboards" schema so that all existing roles are re-used. This is configured in the **<unpacked_zip>\configurations\application.properties** file.

<unpacked_zip>\configurations\users.properties	See **User authentication**.
<unpacked_zip>\configurations\config\ed\led.json	If you have made configuration changes to .JSON files to create custom tiles or layout and you want to retain those changes in the new version of the Dashboard, you can re-use old .JSON files with the new Dashboard, however, please note that doing so will mean that you will not benefit from any improvements or features added to the current release of the Dashboard.
<unpacked_zip>\configurations\config\hd\app.json	
<unpacked_zip>\configurations\config\hd\cmp.json	

Step 3 - Start the new Dashboard and check access

To start the dashboard, run the following file:

```
<unpacked_zip>\startup.bat
<unpacked_zip>\startup.sh
```

By default the dashboard is configured to run on **port 8080**. Use the following URL - where **<server_name>** is equal to the host name of the current server to access the dashboard. If you are testing on the server itself, you can use `http://localhost:8080`:

```
http://<server_name>:8080
```

- If you have a service on the host server that is already using port 8080, please refer to **Modify the user access port for 2.x JAR or ZIP deployments** for more information.
- See also **Start and stop 2.x JAR or ZIP deployments**.

Step 4 - First login and become admin

By default, the CAST Dashboard requires that at least one user is granted the **ADMIN role** following the first login after the **User authentication** configuration. This ensures that one user can access all data and configuration settings. This step MUST be performed from the machine on which the dashboard is running, using the URL **http://localhost:<port>** - if you attempt it from another machine on the network, you will not be able to "become admin". See **First login and become admin**.

Step 5 - Remove existing dashboard (optional)

Remove the existing Dashboard deployment if necessary. This step is optional and you may wish to leave this in place until the new dashboard is fully tested and working.

Instructions for transitioning to 2.5 JAR files

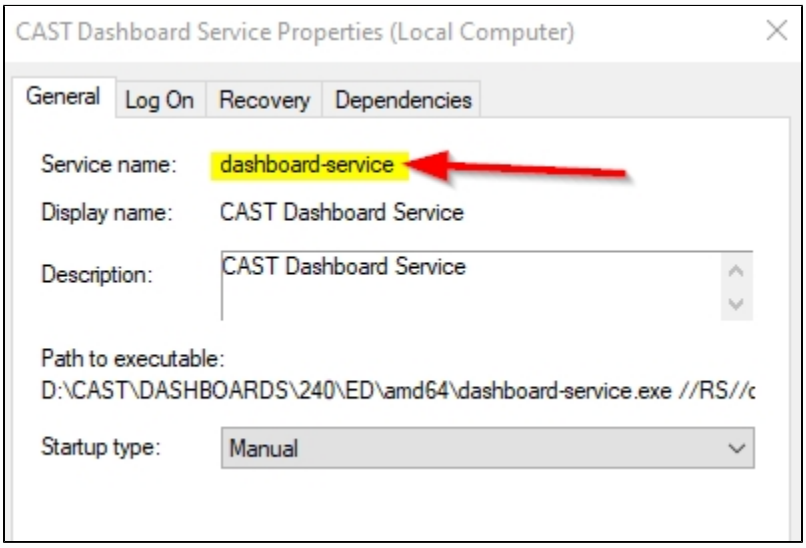
To transition to 2.5 JAR files files there is no managed upgrade process. Instead install the new dashboard side by side with the existing dashboard.

Step 1 - Deploy the new JAR file

Note that when deploying in **Microsoft Windows**, the installer will ask you whether you want to install a **Windows Service** to manage the deployment. If you have already installed a Windows Service for use with the ZIP files shipped in releases 2.0 - 2.4, then CAST highly recommends uninstalling this service before you proceed. You can do so using the following file:

```
<unpacked_zip>\dashboard-service-uninstall.bat
```

or using the following command in command prompt (opened with elevated Administrator permissions), where <service_name> can be found in the properties of the Windows Service:




```
sc.exe delete <service_name>
```

Obtain the new **.JAR** file and deploy it on the target server as described in:

- **Standalone Engineering Dashboard deployment using JAR file**
- **Standalone Health Dashboard deployment using JAR file**
- **Standalone RestAPI deployment using JAR file**

Step 2 - Configure your new Dashboard

You may now need to transfer the details of your environment (stored in the configuration files such as **web.xml/context.xml/other .properties/.xml/.json files**) from the old Dashboard to the new. Please be aware that you should **not** perform a merge or overwrite configuration files (unless specifically requested to do so in the instructions below) - instead you must manually update the new configuration files with the details from the old configuration files.

Use the following table to help. If you change any files in the new deployment, please ensure that you restart the Dashboard so that any changes are taken into account.

Existing file	Notes

authorizations.xml	See **Data authorization - 2.x and above**. **From 2.0.0 to 2.1.0 release** **authorizations.xml** has been replaced with a graphical user interface to manage data authorizations for users/groups. Configuration is stored in a schema called "cast_dashboards" on a chosen CAST Storage Service/PostgreSQL instance. There are two choices for how to proceed: • You can provide your authorizations.xml file and on first start-up of the web application the authorizations will be automatically transferred into the management interface (i.e. the schema called "cast_dashboards"). The authorizations.xml file will then be ignored on any subsequent web application startup and all changes from then on must be performed using the graphical interface. • You can assign the data authorizations from scratch using the graphical interface. **From 2.1.0 release** All authorization data is stored in the schema "cast_dashboards" and the authorizations.xml file is no longer used. CAST recommends re-using the "cast_dashboards" schema so that all existing authorizations are re-used. This is configured in the **application.properties** file.
license.xml	See **Dashboard Service license key configuration** and **Data authorization**.
log4j2-spring.xml	See **Configuring the Log and Audit Trail**.

roles.xml	See **User roles - 2.x and above**.
	
	From 2.0.0 to 2.1.0 release
	
	• **roles.xml** has been replaced with a graphical user interface to manage role assignments for users/groups. Configuration is stored in a schema called "cast_dashboards" on a chosen CAST Storage Service/PostgreSQL instance. There are two choices for how to proceed:
	  • You can provide your roles.xml file and on first start-up of the web application the role assignments will be automatically transferred into the management interface (i.e. the schema called "cast_dashboards"). The roles.xml file will then be ignored on any subsequent web application startup and all changes from then on must be performed using the graphical interface.
	  • You can assign the roles from scratch using the graphical interface.
	
	From 2.1.0 release
	
	All roles data is stored in the schema "cast_dashboards" and the roles.xml file is no longer used. CAST recommends re-using the "cast_dashboards" schema so that all existing roles are re-used. This is configured in the **<unpacked_zip>\configurations\application.properties** file.

***.json**	If you have made configuration changes to .JSON files to create custom tiles or layout and you want to retain those changes in the new version of the Dashboard, you can re-use old .JSON files with the new Dashboard, however, please note that doing so will mean that you will not benefit from any improvements or features added to the current release of the Dashboard.

Step 3 - Undeploy the existing WAR file (optional)

Undeploy the existing **war** file from Apache Tomcat if you are using it - you can use Apache Tomcat manager to do this, or simply delete the relevant folder and file in `CATALINA_HOME\webapps`. This step is optional and you may wish to leave this in place until the new dashboard is fully tested and working.