

# Joget v6 to DX Upgrade Notes

- [Upgrading From An Existing Joget v5 or v6 Folders](#)
- [Installing Joget In A New Folder](#)
- [Important Note About Joget DX New Process Record ID Using UUID](#)
- [Configuring Nginx To Proxy WebSocket](#)
- [Configuring Apache To Proxy WebSocket](#)
- [Perform Thorough App Testing](#)
- [Important](#)

## Upgrading From An Existing Joget v5 or v6 Folders

The following are the changes you need to run Joget DX if you are upgrading from Joget v6 folders:

- Joget DX uses Glowroot for Java APM, so you need to add a new argument in the startup script, example below:

```
set JAVA_OPTS=-Xmx768M -Dwflow.home=./wflow/ -javaagent:./wflow/aspectjweaver-1.8.5.jar -javaagent:./wflow/glowroot/glowroot.jar
```

- Install Joget DX in a temporary folder and copy the whole content of folder "wflow\glowroot\\*" to your new Joget "wflow" folder.
- Do also note the higher default maximum memory allocation pool for the JVM in "-Xmx768M". Joget DX requires more heap space and if your server has the additional RAM, do allocate more "-Xmx" memory for better performance.
- Default Joget DX installation is running on "jre11.0.2". However Joget DX can still run on Java 8.

The upgrade steps are as follows if you are upgrading to Joget DX in v5 or v6 folders:

1. Backup your Joget v6 files and v6 database.
2. Make the above changes in startup script and Glowroot folders.
3. Delete ".\apache-tomcat-8.5.41\work\\*" folder.
4. Delete ".\apache-tomcat-8.5.41\webapps\jw" folder and jw.war file.
5. Copy Joget DX "jw.war" file into ".\apache-tomcat-8.5.41\webapps\".
6. Edit ".\wflow\app\_datasource-default.properties" to point to your existing v6 database.
7. Start Joget Apache Tomcat and monitor the joget.log.

**Tip** To save time in the initial DX testing, you can delay the copying of the ".\wflow\app\_formuploads" folder (may have too many files) and ".\wflow\app\_plugins" folder (to first test Joget DX with zero custom plugin) until after everything is running smoothly.

**Tip** For Joget installations before v7.0.12, you are recommended to increase the cacheMaxSize to "100000" in ".\apache-tomcat-8.x.xx\conf\context.xml" to avoid the following warning: "WARNING [localhost-startStop-1] org.apache.catalina.webresources.Cache.getResource Unable to add the resource at [/WEB-INF/classes/templates/userview/login.ftl] to the cache for web application [/jw] because there was insufficient free space available after evicting expired cache entries - consider increasing the maximum size of the cache".

## Installing Joget In A New Folder

A second quicker solution where Joget DX is a new install folder:

1. Backup your Joget v6 database or clone it for Joget DX use (Joget DX will automatically create the new tables and fields it needs).
2. Install Joget DX in a new folder using the Windows or Linux Installer.
3. Copy over all files from v6 ".\wflow\\*" to the same folder in Joget DX.
4. Ensure Joget DX can access the v6 database (check the "app\_datasource-default.properties" settings).
5. Edit the ".\apache-tomcat-8.5.41\conf\server.xml" if you wish to run Joget DX on a different port.
6. Edit to set a higher "-Xmx" memory setting in startup script "Joget-start.bat or .sh".
7. Start Joget Apache Tomcat and monitor the joget.log.

Lastly, you can download and install new plugins specially for Joget DX from <https://archives.joget.org/addons/> to try out.

## Important Note About Joget DX New Process Record ID Using UUID

One of the top requested features for Joget DX is the enhancement of process form data security, by allowing the **record ID** to use **UUID** instead of reusing the **process ID**. Using UUID for process record ID makes the URL difficult to guess, thus protecting unauthorized access to the process records in a Form.

With the old record ID of "[seq#]\_appid\_process1" which is the same format as the process ID, it was easy for users to access other process records by just incrementing the process seq number in the browser URL and viewing records not belonging to them.

You can revert Joget DX to use sequential process id numbering via the new option in ["General Settings > Run Process Primary Key"](#).

Here is a process id related checklist if you are migrating apps from v5 or v6 to Joget DX:

- In Beanshell, have two separate variables to store the @processid@ and @recordid@ at the top of your code:
  - Recordid contains the UUID primary key (id). This is used when you have to query or update the process database table using SQL syntax.
  - Processid is used when you are using Joget methods to manipulate the process data.
  - See an example app using Recordid in BeanShell code at "Multiple Approval App For JogetDX":<https://dev.joget.org/community/display/DX7/Demonstration+Multiple+Approval+App+For+Joget+DX>.
- Similarly, if you use process Database Update Tool where SQL is included, ensure you replace any "where id = '#assignment.processid#' " with "where id = '#process.recordid#' ".
- If your forms (that are used in processes) has subform or Ajax subform with references to the parent form id stored in a hidden field, ensure you initialize the hidden field with the "#process.recordid#" hash and not the "#process.processid#".

The apps that has zero BeanShell or SQL in JDBC calls should have no issues running on Joget DX. Nevertheless, please thoroughly test all older apps if you are migrating them to Joget DX.

Related Documentation:

- [General Settings > System Administration Settings > Run Process Primary Key option.](#)
- [Hash Variable > Workflow Process Hash Variable](#)

## Configuring Nginx To Proxy WebSocket

WebSocket is used by Joget DX for App Console **Logs** menu.

Refer to the following links to configure your Nginx to proxy WebSocket:

- <http://nginx.org/en/docs/http/websocket.html>
- <https://www.serverlab.ca/tutorials/linux/web-servers-linux/how-to-configure-nginx-for-websockets/>
- <https://stackoverflow.com/questions/12102110/nginx-to-reverse-proxy-websockets-and-enable-ssl-wss>

Here is a simple sample configuration:

```
server {
    listen 80;
    listen [::]:80;

    server_name myjoget.com;

    location /jw/web/applog/ {
        proxy_pass http://localhost:8080/jw/web/applog/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
    location / {
        proxy_set_header Host $http_host;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8080/;
    }
}
```

## Configuring Apache To Proxy WebSocket

Apache config would be as follows:

```
<Proxy balancer://wscluster>
BalancerMember ws://tomcat1:8181 route=node01
BalancerMember ws://tomcat2:8282 route=node02
Order deny,allow
Allow from all
</Proxy>

ProxyPass /jw/web/applog balancer://wscluster/jw/web/applog stickysession=JSESSIONID|jsessionId
ProxyPassReverse /jw/web/applog balancer://wscluster/jw/web/applog
```

## Perform Thorough App Testing

We highly recommend that you spend some time to perform a thorough test on the Joget apps that you have created based on Joget v5 or v6. An app test can quickly determine if your app is fully compatible with Joget DX. Be sure to check the server log too as it would generate error logs on possible missing/incompatible plugins or other issues.

## Important

- Useful read and reference: [Optimizing Joget Platform](#)
- Apache Tomcat user needs full Read Write access to the whole ".\apache-tomcat.x.x.x\" and ".\wflow\" directories on Linux/Ubuntu OS.
- Starting from v6.0.15, JAVA 8 is required due to fixes for vulnerability issues of a dependent library.