

# Upgrading ART

Take the following steps to upgrade an existing ART installation to a more current version.

- Download the latest ART version
- Extract the zip package to a temporary location
- Review the **changelog.txt** file to see what has changed
- Backup the ART directory contained on the application server
- Backup the ART database
- Connect to the ART database and run any database upgrade scripts included in the download package to bring the ART database up to date
- Deploy the new **art.war** file to the application server. There are a number of ways to do this. You can do the following if using Apache Tomcat.
  - Stop Tomcat
  - Delete the art directory and art.war file in the **TOMCAT\_HOME\webapps** directory
  - Delete ART's Tomcat cache by deleting the art directory in the **TOMCAT\_HOME\work\Catalina\localhost** directory
  - Copy the new art.war file to the webapps directory
  - Start Tomcat
- Copy the contents of the backup **art\WEB-INF\work** directory to the new version's work directory
- Copy the contents of the backup **art\js-templates** directory to the new version's js-templates directory
- If you had modified the **art\WEB-INF\art-custom-settings.json** file, copy the file from the backup directory to the new version directory
- If you were using the demo HSQLDB database, after running the upgrade scripts on the ArtRepositoryDB database in the backup location, copy the updated files to the **art\WEB-INF\hsqldb** directory of the new version, overwriting the existing files. You can look at the [Tips](#) documentation to see how you can upgrade an HSQLDB database.
- If you had put any **JDBC drivers** in the **art\WEB-INF\lib** directory, copy these drivers from the backup lib directory to the new version's lib directory.
- Restart Tomcat
- Clear your browser's cache e.g. for Firefox, use the History | Clear Recent History menu

The upgrade is complete. All users, reports, scheduled jobs etc should continue to work as before.

## Notes

Some upgrades may require special attention or additional effort due to feature changes and incompatibilities with previous versions.

## 3.0

ART 3.0 is a major upgrade and may require extensive testing to verify that it continues to fulfill the needs of an existing installation.

Item	Comments
<b>URL changes</b>	URLs that were being used in bookmarks or other links will need updating. Review the new application to see the appropriate URL to use or look at the admin manual to see new URLs.
<b>ART Database configuration</b>	The details of the ART Database will need to be entered afresh
<b>Settings</b>	Application settings will need to be entered afresh
<b>Internet Explorer</b>	If using Internet Explorer, you will need to use version 8 at a minimum. Some features may not work with Internet Explorer 8 so it is recommended, if using Internet Explorer, to use the latest version.
<b>Java</b>	Java 7 and above is required
<b>Application server</b>	An application server that supports servlet 3.0 will be required e.g. Tomcat 7 and above
<b>Renamed concepts</b>	Some concepts have been renamed e.g. "Queries" are now "Reports", "Graphs" are now "Charts", "View Modes" are now "Report Formats", "Report on Column" report types are now "Group" reports.
<b>jXLS templates</b>	jXLS templates need to be recreated. Review the sample templates located in the <b>ART_HOME\WEB-INF\work\templates</b> directory, or the jXLS website, <a href="http://jxls.sourceforge.net/">http://jxls.sourceforge.net/</a> , to see the new template structure.
<b>Chained parameters</b>	Reports that use chained parameters will need to be modified. Review the chained parameter example in the demo database or the admin manual to see the new method of specifying chained parameters.
<b>Directory location changes</b>	The templates directory has been moved to <b>ART_HOME\WEB-INF\work\templates</b> . The export directory has been moved to <b>ART_HOME\WEB-INF\work\export</b> .
<b>XMLA Repors</b>	For Mondrian XMLA and Microsoft XMLA reports, the url, username and password fields have been removed from the report definition page and instead a datasource needs to be created from the Datasource Configuration page and then used by selecting the Datasource field in the report definition.
<b>Query tags</b>	The special hardcoded tags or placeholders than can be used within query sql, <b>:USERNAME</b> , <b>:DATE</b> and <b>:TIME</b> have been changed to include a colon at the end i.e. <b>:USERNAME:</b> , <b>:DATE:</b> and <b>:TIME:</b>
<b>Dashboard XML</b>	If you have xml special characters within tag content, ensure to replace them with the relevant xml character entities e.g. if the <URL> tag contains a url like <code>http://localhost:8080/art/app/runReport.do?reportId=1&amp;p-param1=test</code> , you'll need to replace the & character with <code>&amp;amp;</code> , to have the url as <code>http://localhost:8080/art/app/runReport.do?reportId=1&amp;amp;p-param1=test</code> . For a list of the 5 special xml characters and their respective replacement strings or character entities, see <a href="http://support.esri.com/technical-article/000005870">http://support.esri.com/technical-article/000005870</a>
<b>LOOKUP rule values</b>	If you had any rule values of type <b>LOOKUP</b> , these will no longer be in effect as there is no longer an option for lookup rule values. Instead, define a user group, assign rule values to this user group and put the required users into this user group.
<b>Non-labelled multi parameters</b>	For reports that use multi parameters where the parameter name is not included in the sql source e.g. <code>... IN( #param# )</code> , the reports will need to be updated to include the parameter placeholder in the sql source.
<b>Non-labelled rules</b>	For reports that use rules, if the <code>#rules#</code> label doesn't exist in the sql source, such reports will need to be updated to include this label, so that rules can be applied.
<b>Dynamic datasources</b>	Dynamic datasources must now use the Datasource ID as the value to pass to the parameter. If using an LOV report to provide the dynamic datasource list, the first column must be the datasource ID. If using a static lov this may look something like '3
<b>Long parameters in Jasper-Reports</b>	Parameters that were defined with a data type of <code>java.lang.Long</code> in JasperReports need to be modified to <code>java.lang.Integer</code>