dx-sync-module

dx-sync-module synchronizes changes in files with Magnolia repositories.
Also it provides a set of tasks and strategies for updating Magnolia resources.

It allows to edit templates scripts, CSS, JavaScript and other files directly from IDE or editor of your choice. Changes are instantly propagated to Magnolia repositories of a running Magnolia instance.
So you can preview your changes on the website as soon as you save your files.
Creating, updating, deleting and renaming files operations are supported.

Content

Supported resource types

The following resource types are supported at the moment:

- JavaScript files (js and processedJs resources)
- CSS files (css and processedCss resources)
- Freemarker template scripts (stored in templates workspace)
- Files stored in DMS
- Zip archives (extracted to DMS document trees)
- Module files (from mgnl-files resource folder, including Freemarker and JSP templates)
- Groovy scripts
- Groovy classes

Initial setup

Download the latest module version.
Note: it is going to be available as a Maven artefact soon.

Download

Go to Downloads page and download the latest version of dx-sync-module.

Install

First, you need to install jnotify http://jnotify.sourceforge.net/ library to your servlet container, including both java and native library parts.
This library is used to get notifications from the file system about file updates.
You can download it from here:

- jnotify-0.93.jar
- jnotify.dll
- jnotify_64bit.dll
- libjnotify.so
- libjnotify_64bit.so

If you use Tomcat you can:

- Put jnotify-0.93.jar to Tomcat's lib directory
- For Windows, 32-bit JRE: put jnotify.dll to Tomcat's bin directory
- For Windows, 64-bit JRE: put jnotify_64bit.dll to Tomcat's bin directory and rename the file to jnotify.dll
- For Linux, 32-bit JRE: put jnotify.so to Tomcat's bin directory
- For Linux, 64-bit JRE: put jnotify_64bit.so to Tomcat's bin directory and rename the file to libjnotify.so
- For Mac OS X, refer to jnotify documentation how to obtain native library and put it to the place Tomcat can find it.

Steps are very similar for other servlet containers.

Then you need to add dx-sync-module to your project.
If you're going to use File-based configuration, you can just put dx-sync-module.jar to Magnolia'a WEB-INF/lib directory.
If you need advanced features, including DxModuleVersionHandler, you need to add dx-sync-module to your project.
Put dx-sync-module to your local Maven repository:

```
mvn install:install-file -DgroupId=com.devexperts.web -DartifactId=dx-sync-module -Dversion=1.1.2 -Dpackaging=jar
-Dfile=dx-sync-module-1.1.2.jar -DgeneratePom=true
mvn install:install-file -DgroupId=com.devexperts.web -DartifactId=dx-sync-module -Dversion=1.1.2 -Dpackaging=jar
-Dclassifier=sources -Dfile=dx-sync-module-1.1.2-sources.jar -DgeneratePom=false
```

Add the following dependency to your Maven project:

```
<dependency>
    <groupId>com.devexperts.web</groupId>
    <artifactId>dx-sync-module</artifactId>
    <version>1.1.2</version>
</dependency>
```

Configuration

Version Handler based configuration

You can specify resources which are to be synchronized in version handler for your module.
In this case

1. Specify DxModuleVersionHandler as a base class of your module version handler
2. Pass folder where your module is placed as a parameter to the DxModuleVersionHanlder constructor.
3. Specify resources which need synchronization in your module version handler constuctor.

For example:

```
super("web-devspace");
keepUpToDateResources("/mgnl-files", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.MGNL_FILE);
keepUpToDateResources("/abc/img", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.DMS_FILE);
keepUpToDateResources("/abc/themes/img", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.DMS_FILE);
keepUpToDateResources("/dev/docs", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.DMS_FILE);
keepUpToDateResources("/abc/js", Scope.FOLDER_ONLY, ResourceType.RESOURCE_PROCESSED_JS);
keepUpToDateResources("/abc/js", Scope.SUBFOLDERS_ONLY, ResourceType.RESOURCE_JS);
keepUpToDateResources("/abc/themes/abc/css", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.RESOURCE_PROCESSED_CSS);
keepUpToDateResources("/abc/emails", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.TEMPLATE);
keepUpToDateResources("/abc/paragraphs", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.TEMPLATE);
keepUpToDateResources("/abc/templates", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.TEMPLATE);
keepUpToDateResources("/search-module/templates", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.TEMPLATE);
keepUpToDateResources("/abc/groovy", Scope.FOLDER_AND_SUBFOLDERS, ResourceType.GROOVY_SCRIPT);
```

DxModuleVersionHandler may work in 2 modes.

- Development mode.
  This mode is in use when -DdevPlatform=true JVM argument is specified. In this case all changes in source files are instantly propagated to the running Magnolia CMS.
- Production mode.
  This mode is used by default. All resources are synchronized on module start up using usual Magnolia tasks.
  Note: you don't need jnotify library in Production mode.

Note: dependent on how you run your project you may still to specify path to the root of your source files via dx.sync.sources system property. See the next section for details.

File-based configuration

You can specify which files are to be synchronized in a configuration file.
Synchronization is performed on a directory basis. Each directory that is needed synchronization should be listed in the configuration file.

Example configuration file:

```
# Line format is:
#    source path, scope, recource type, destination
path                                                                \
# Possible scopes are:
# - FOLDER_AND_SUBFOLDERS
# - FOLDER_ONLY
# - SUBFOLDERS_ONLY
# Possible resource types are:
# - MGNL_FILE
# - GROOVY_CLASS
# - GROOVY_SCRIPT
# - DMS_FILE
# - DMS_ZIP
# - RESOURCE_JS
# - RESOURCE_PROCESSED_JS
# - RESOURCE_CSS
# - RESOURCE_PROCESSED_CSS
# - TEMPLATE

# test javadoc link
web-devspace\docs, FOLDER_ONLY, DMS_ZIP, /dev/apidocs
# test downloads
web-devspace\resources\downloads, FOLDER_AND_SUBFOLDERS, DMS_FILE, /dev/downloads
```

2 parameters should be specified for this module.

- dx.sync.sources - path to the root of source files. Directory paths in the configuration file are relative to this path.
  By default, the directory which is parent to Tomcat location is used (for example, if Tomcat is located under C:\work\project\apache-tomcat-6.0.32, the directory C:\work\project will be used).
- dx.sync.config - location of the configuration file.
  By default, sync.csv in directory which is parent to Tomcat location is used (for example, if Tomcat is located under C:\work\project\apache-tomcat-6.0.32, the file C:\work\project\sync.csv is used).

Specify them by adding -Ddx.sync.sources=<path to source files root> -Ddx.sync.config=<path to configuration file> to the JVM startup command line.

Synchronizing module resources

DxModuleVersionHandler has an additional functionality of synchronizing resources on start-up.
You can use it instead of writing BootstrapSingleModuleResource tasks in your version handler and updating your module version.

All XML files you need to be bootstraped can be placed under /dx-resources folder in module resources.
Files under resource directory /dx-resources/keepUpToDate (including subdirectories) will be bootstraped on module start up, if they have been changed.
Files under resource directory /dx-resources/installOnce (including subdirectories) will be bootstraped on module start up, if a corresponding node doesn't exist in repository.
For files under resource directory /dx-resources/remove (including subdirectories) a check if a corresponding node exists in repository will be performed on module start up. If it is, it will be removed.

Note that DxModuleVersionHandler usually works significantly faster than Boostrap*** tasks, and you don't need to write code and update module versions.

License

The module is released under GPL license. Contact us if you need this module under different license.

Questions

We really appreciate your feedback, feature requests, bug reports or any kind of comments.
Contact us at avasiliev@devexperts.com