

Setting up your sandbox

- Create your development environment
- Setup your DW server connection
- Checkout and install current codebase
- Preparing the environment
 - Building project
 - Usage
- Aliases
- Resources

Create your development environment

Note that these are very high level instructions which assume a certain level of Eclipse knowledge and are therefore more intended as an aide memoire.



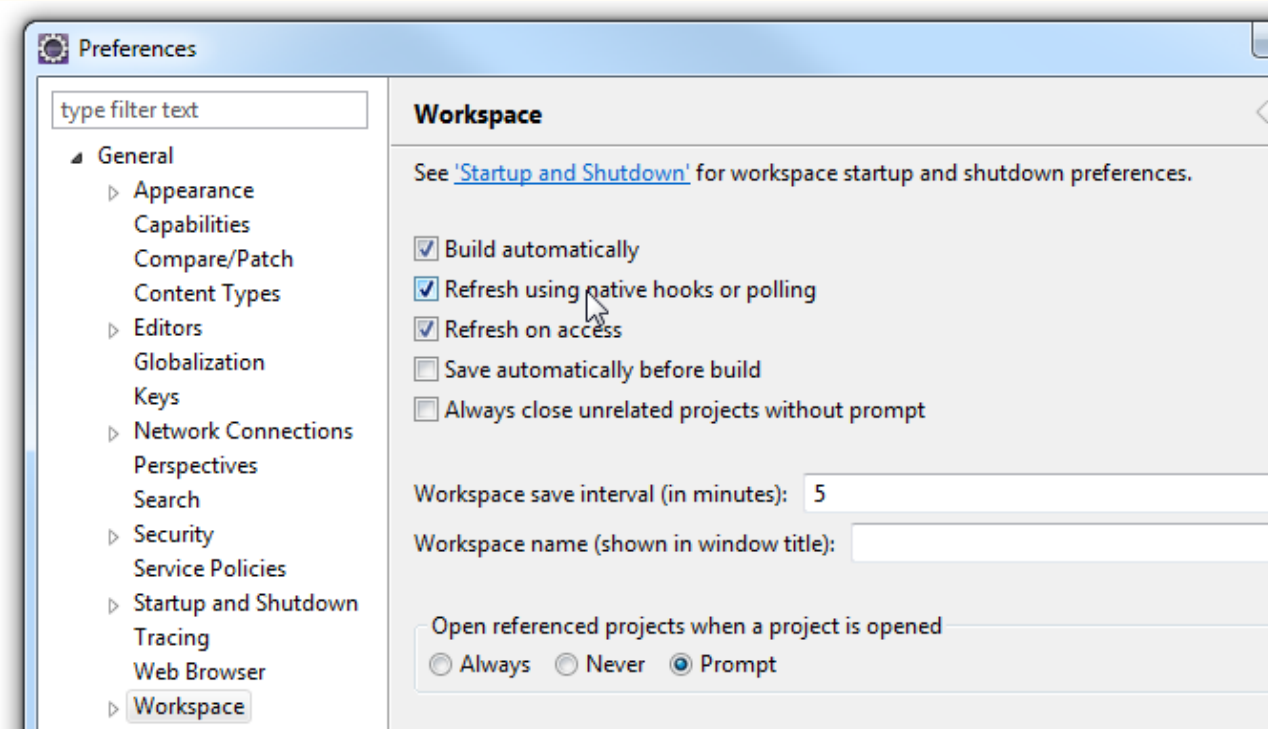
Create Project Directory

Create the **PROJECT_DIRECTORY** directory `c:/projects/Aurora`

1. Download latest stable Eclipse version
 - a. e.g. Juno Sr2 (there is 32 and 64 bit versions available) by following the link below:
[Eclipse IDE for Java EE Developers - http://www.eclipse.org/downloads/packages/release/juno/sr2](http://www.eclipse.org/downloads/packages/release/juno/sr2)
2. Install Eclipse Studio under the **PROJECT_DIRECTORY**/eclipse
3. Choose workspace directory: **PROJECT_DIRECTORY**/workspaces/workspace_latest
4. Install UXstudio
 - a. Go to Help/Install New Software
 - b. In the Available Software dialog box click Add
 - c. Enter a name for the site, i.e. demandware, and paste the following into the URL http://updates.demandware.com/uxstudio_ea/4.5, click OK.
 - d. Select the Demandware checkbox and install the software.
5. Install Subclipse
 - a. Go to Help/Install New Software
 - b. In the Available Software dialog box click Add
 - c. Enter a name for the site, i.e. subclipse, and paste the following into the URL:
 - i. for SVN client < 1.8 - http://subclipse.tigris.org/update_1.8.x
 - ii. for SVN client 1.8 - http://subclipse.tigris.org/update_1.10.x
 - d. Select the Subclipse and SVNKit checkboxes and install the software



It's good idea to enable the auto refresh for files in eclipse: Window -> Preferences -> General -> Workspace -> Refresh using native hooks or polling



Setup your DW server connection

1. Launch eclipse, switch into the Demandware development perspective.
2. Select a sandbox that has not been used yet. Available sandboxes / credentials can be found here: [Demandware Sandbox Enviroments](#). Check that you can login to the business manager. Update the wiki page to indicate the sandbox that you have been assigned.



Users

Make sure you **do not** use the admin user, but create your own user - please refer to this page [Demandware Sandbox Enviroments#Users](#) for more information .

3. In eclipse select File/New/Demandware Server connection. Enter a name for the project i.e. **devXX-web-XX.demandware.net**, set the hostname, username and password of the sandbox that you have just allocated to yourself
4. Choose "next"
5. On the 'Code Staging Directory' window select 'Target Version Directory' = 'version1'



Alternatively you can create a new code version by going to BM and under Administration - Site Development - Code Deployment, create a new version and activate it. Then use that when creating the Demandware Server connection.

Checkout and install current codebase



Create Repository Directory

Create the **REPOSITORY_DIRECTORY** directory **PROJECT_DIRECTORY/repository**

1. Use a the svn client of your choice to checkout the current codebase from <https://svn.javellingroup.com/svn/arp/trunk> into the **REPOSITORY_DIRECTORY**. You may need to pass in some credentials, if you do not have permission to access this repository then please request access from your team lead.
2. Launch eclipse, switch into the Demandware development perspective
3. Import each of the cartridges which have been checked out into **REPOSITORY_DIRECTORY/cartridges** (this should be into the workspace that contains your Demandware server connection). File/Import/General/Existing Projects Into Workspace, you can select the

- cartridges folder itself and all of the cartridges within it will be imported into your workspace
4. Make sure that when prompted to link the projects to the Demandware Server you click Yes
 5. Clean the project i.e. Project/Clean
 6. Import each of the projects which have been checked out into **REPOSITORY_DIRECTORY/sites**, do not link those to the server connection if prompted.



Site template structure details can be found here: [Site Configurations and Site Templates](#)

Checkout metadata spreadsheet and keepass dabase



Create Repository Directory

Create the **SHARED_DIRECTORY** directory **PROJECT_DIRECTORY/shared**

1. Use a the svn client of your choice to checkout the current codebase from <https://svn.javelingroup.com/svn/arp/shared> into the **SHARED_DIRECTORY**. You may need to pass in some credentials, if you do not have permission to access this repository then please request access from your team lead.
2. Launch eclipse, switch into the Demandware development perspective
3. Import the project from **SHARED_DIRECTORY**

Preparing the environment

1. Install node.js (choose *.msi and install from here: [node-v0.10.36-x64.msi](#)) . Do not get latest version it requires python.
2. Install node package manager (should be included in latest node.js installations, but can be downloaded from here: <https://www.npmjs.com/package/npm>)
3. Install Grunt command line interface from command line:

```
npm install -g grunt-cli
```

Building project

1. Go to the **REPOSITORY_DIRECTORY/cartridges** directory and start command prompt there
2. Run the installation command which is going to download all the required dependencies

```
npm install
```

Usage

1. Go to the **REPOSITORY_DIRECTORY/cartridges** directory and start command prompt there
2. The following commands are available

commands of note	usage/description
grunt sass	Compiles all the scss files into one styles.css
grunt browserify	Compiles all *.js files under app_storefront_richUI/js into app_storefront_richUI/static/defaults/js/app.js <ul style="list-style-type: none"> • https://github.com/substack/browserify-handbook • http://browserify.org/
grunt watch	Watches for changes in scss and js files and executes the commands above automatically

grunt jshint	<p>Validates the javascript in the projects - checks for unused variables, issues, etc</p> <pre>Running "jshint:target" <jshint> task app_storefront_richUI\cartridge\js\pages\cart.js line 5 col 5 'product' is defined but never used. app_storefront_richUI\cartridge\js\product-tile.js line 4 col 5 'product' is defined but never used. !! 2 warnings Warning: Task "jshint:target" failed. Use --force to continue. Aborted due to warnings.</pre>
grunt autoprefixer	
grunt jscs	<p>Javascript code style checker</p> <pre>Running "jscs:src" <jscs> task Missing space before opening round brace at app_storefront_richUI/cartridge/js/product-tile.js : 94 : var \$tiles = \$(<.tiles-container .product-tile>); 95 : if (<\$tiles.length === 0>) <return; > 96 : imagesLoaded(<.tiles-container>).on('done', function() { 97 : \$tiles.syncHeight() 98 : .each(function (idx) { > 1 code style errors found! Warning: Task "jscs:src" failed. Use --force to continue. Aborted due to warnings.</pre>

Aliases

1. Edit host file and add the sites. Example:

```
# aurora - dev02, km realm
185.50.160.165 oasis-stores.com
185.50.160.165 www.oasis-stores.com

185.50.160.165 www.karenmillen.com
185.50.160.165 karenmillen.com
185.50.160.165 m.karenmillen.com
185.50.160.165 fashion.karenmillen.com
185.50.160.165 reviews.karenmillen.com
185.50.160.165 www.karenmillenblack.com
185.50.160.165 m.karenmillenblack.com
185.50.160.165 www.warehouse.co.uk
185.50.160.165 warehouse.co.uk
185.50.160.165 www.warehouse-london.com
185.50.160.165 warehouse-london.com
```

Resources

1. <http://gruntjs.com/installing-grunt>
2. <http://gruntjs.com/getting-started>
3. <https://bitbucket.org/demandware/sitegenesis-community>
4. <https://xchange.demandware.com/docs/DOC-25778>
5. Site Configurations and Site Templates