Development Environment

Plugin Reloading

If you make changes to a plugin, you will generally need to shut down the server, update the plugin, and start the server again.

To enable TeamCity development mode, pass the "`teamcity.development.mode=true`" internal property. Using the option you will:

- Enforce application server to quicker recompile changed `.jsp` classes
- Disable JS and CSS resources merging/caching

The following hints can help you eliminate the restart in the certain cases:

- if you do not change code affecting plugin initialization and change only body of the methods, you can attach to the server process with a debugger and use Java hotswap to reload the changed classes from your IDE without web server restart. Note that the standard hotswap does not allow you to change method signatures.
- if you make a change in some resource (jsp, js, images) you can copy the resources to `webapps/ROOT/plugins/<plugin-name>` directory to allow Tomcat to reload them.
- change in build agent part of plugin will initiate build agents upgrade.

If you replace a deployed plugin .zip file with changed class files while TeamCity server is running, this can lead to NoClassDefFound errors.
To avoid this, set "`teamcity.development.shadowCopyClasses=true`" internal property. This will result in:

- creating "`.teamcity_shadow`" directory for each plugin .jar file;
- avoid .jar files update on plugin archive change.

See also:

Extending TeamCity: Developing TeamCity Plugins | Plugins Packaging