Development Support: Getting Started

This is a guide for setting up your development environment.

Terms and Locations

The DNSR

On Devnet you will have access to the Devnet Software Repository (DNSR). This is a basic drupal site we use for indexing approved software. If you'd like to install a program on your Devnet machine, start by searching the DNSR. If the program you want isn't available, you can request it.

You can access the DNSR at http://devcas024/dnsr/.

Nexus

http://nexus.devnet.local/

Jira

https://jira/jira/

Jenkins

https://build/hudson/view/Khaos/

Crucible

https://devlvs033/

Setting up Your Machine

Installing Java

You will need a JDK for Java development. At the time of this writing, the latest downloads can be found here:

- Java 7 JDK - http://devcas024/dnsr/node/676
- Java 6 JDK - http://devcas024/dnsr/node/677

You'll likely want to install the 64-bit versions of both. Most of our software is still Java 6, but we will soon migrate to 7.

After install, ensure that you have a `JAVA_HOME` system environment variable pointing to the correct location. (e.g. `C:\Program Files\Java\jdk1.7.0_09`). You may also want to add the java `bin` folder to your system `PATH` so you can run the java tools from the command line.

Installing and Configuring Maven

Maven is a Java build and dependency management system. Most of our projects use Maven to handle configuration.

Installing

1. Create a directory at `C:\maven`.
2. Extract `\\devslo020\data\devsupport\installers\maven\apache-maven-3.0.3-bin.zip` into the new folder.
3. `C:\maven\apache-maven-3.0.3` should now contain `bin`, `conf`, `lib`, etc.

Configuring

1. Add `C:\maven\apache-maven-3.0.3\bin` to your system `PATH`.
2. Open command prompt and run the following to make a `.m2` folder in your user directory: `mkdir %USERPROFILE%\.m2`
3. Copy the `settings.xml` file from `\\devslo020\data\devsupport` to the new `.m2` directory.

Proxy Fix

1. Copy the directory `\\devslo020\data\devsupport\PROXY_FIX_JAVA7` to a temporary location on your machine.
2. Open it and run `installcert.bat`.
3. Replace the cacerts file found in `<jdk home>/jre/lib/security` with the newly generated file. If you have more than 1 java install, you should replace the file for each.

TODO java7 vm args

Eclipse Setup

See DevelopmentSupport/EclipseSetup

Installing Tomcat

See DevelopmentSupport/InstallingTomcat

Installing Flashbuilder

1. Copy the directory `\\devslo020\data\devsupport\installers \Adobe Flash Builder 4.5` to a temporary location on your machine.
2. Run `Set-up.exe` and go through all the installation steps.

3. Run a text editor as Administrator and open `FlashBuilder.ini` from your Flash Builder install directory. (Likely `C:\Program Files (x86)\Adobe \Adobe Flash Builder 4.5`) It's best to use something like Notepad++ since Notepad can't handle unix line endings and the entire file will show up as a single line.
4. Replace the following lines to give Flash Builder more memory:

Old	New
-Xmx515m	-Xmx1024m
-XX:MaxPermSize=256m	-XX:MaxPermSize=512m
-XX:PermSize=64m	-XX:PermSize=128m

Configuration

1. Run Flash Builder as Administrator.
2. Go to Help -> Install New Software...
3. Click the "Available Software Sites" link.
4. Select all the sites in the list and click Remove. You can't get to them since there's no internet access but Flash Builder will keep trying.
5. Click Add and enter the following URL: `http://sloeclipseupdates /flashbuilder` This is our internal eclipse update site for Flash Builder on devnet. You can leave Name blank since it will auto-fill later.
6. Click OK to return to the Install dialog.
7. In the "Work with" box, select the new update site.
8. Ensure that "Hide items that are already installed" and "Group items by category" is checked at the bottom.
9. Select the following plugins to install:

Name	Explanation
m2e - Maven Integration for Eclipse	This plugin adds support for reading maven project configuration from pom.xml files.
flexmojos connector for m2e	This plugin works with m2e to automatically configure your flex projects (build path, source directories, etc.) using information from the pom.
cdm-version connector for m2e	Most CDM projects have a version file that's generated on each build to provide easy programmatic access to version information. This plugin enables that in Flash Builder.
Subversive SVN Connectors	Enables connection to SVN
Subversive SVN Team Provider	
SVNKit **1.7.5v1** Implementation	

10. Proceed through the installation steps, accepting the license agreements and security warnings.

11. When asked to restart, decline and *exit* Flash Builder. (If you click restart it will restart as Administrator which we don't want.)
12. Start Flash Builder again using your regular account.
13. Go to Window -> Preferences.
14. On the Maven tab uncheck "Download repository index updates on startup"
15. On the Maven/Installations tab, click Add... and select the directory where you installed Maven 3.0.3 earlier.
16. Set this new installation as the default.

Using the Flash Builder plug-in in Eclipse

"THIS IS STILL BEING REVISED - IT MAY BE WRONG!!!"

First, download and install the latest Flash Player (version 11.0.2 at this time). It is available in the repository. Also make sure you install the plug-in for whatever browser you're using.

Second, install the plugin: Go into the utilities folder inside the Flash Builder install directory and run "Adobe Flash Builder 4.X Plug-in Utility.exe". Follow the prompts and finish installing.

Now, launch Eclipse and follow these steps. If Eclipse does not start, please see note below.

1. Open Eclipse preferences and go to General -> Workspace -> Linked Resources
2. Add a new Path Variable with the name `TMM_WEBAPP_DIR` pointing to the `target/tmm` folder in your `tmm-webapp` project.
3. Right click in Project/Package Explorer and select "Import...". In the dialog choose "Existing Maven Projects". Click Next and then browse to the location of the `tmm/tmm-client` folder under the main `tmm` project. Finish the wizard. This project will be automatically imported.
4. Go to Windows->Properties and select File Associations. Make sure both ActionScript and MXML files are checked. Click OK.
5. Right click the tmm-client project and select Project Preferences. Select "Flex Compiler" and choose "Use default SDK" under Flex SDK Version (*Note: If you are using Flash Builder 4.7, you must use the SDK from Flash Builder 4.5. You can copy this SDK from FB4.5 to FB4.7 in the 'sdks' folder or download 4.5.1 SDK from the repository. Then go to Installed Flex SDKs and check Flex 4.5 as the default). Under Adobe Flash Player options, select 'Use minimum version required by the Flex SDK. A dialog will popup saying you will no longer be able to open this project under an older version of Flash Builder. Man up and click 'OK." Select Close (you may be required to enter your local admin password).*
6. *Right click the `tmm-client` project, select Debug As -> debug configuration. Select the Flex Web Application. Under Project browse to the `tmm-client` project (it should be the only one available). Un-check 'Use Default" and enter '* *https://localhost:8443/tmm/index.html." At the top, name your debug configuration, something like "debug client." You are now set up to debug the app. Click Debug and let it run once to make it the default debug settings.*

Note: The first time I installed the plug in, the registration dialog came up for the flex plug-in, but after selecting "Continue Trial" nothing happened. If this happens, try going into the

Eclipse install directory, in the `dropins` folder, and **moving** the Flash Builder link file out to the desktop. Run Eclipse normally. Quit Eclipse, copy the link back in, and then re-launch Eclipse. This worked for me.

Development Setup

See DevelopmentSetup

DevelopmentSupport/GettingStarted (last edited 2013-06-12 19:56:01 by jnetzley)