# Installing and Building the DataScript Tools

Harald Wellmann <HWellmann@harmanbecker.com>

Revision History
Revision 0.1 28 September 2006

## Table of Contents

1. Introduction ........ 1
2. Installing and Running ........ 1
3. Building from Source ........ 1
3.1. Installing Eclipse ........ 1
3.2. Installing the Subclipse plugin ........ 2
3.3. Creating a local dstools project from the Subversion repository ........ 2
3.4. Setting up the project properties ........ 3
3.5. Building the dstools project ........ 3

# 1. Introduction

This is a short guide for installing and running the datascript tools and for getting the source and building the tools from source.

# 2. Installing and Running

T.b.d.

# 3. Building from Source

The only supported build environment for the dstools project is Eclipse 3.1 or 3.2. If you know what you are doing, you will also be able to build dstools in other environments, but Eclipse is the only one that the authors will document and support.

## 3.1. Installing Eclipse

Subclipse is a Subversion client plugin for Eclipse. Using this plugin, you can directly access Subversion repositories from Eclipse.

- Install JDK 1.5.0 from `http://java.sun.com`.
- Install Eclipse 3.2 from `http://www.eclipse.org`.
- Start Eclipse and set the proxy options in *Window* | *Preferences* | *Install/Update*.

- Goto *Help* | *Software Updates* | *Find and Install* to install the Eclipse Modelling Framework (EMF).
- Select *Callisto Discovery Site*. EMF is in the category *Models and Model Development*.

## 3.2. Installing the Subclipse plugin

Subclipse is a Subversion client plugin for Eclipse. Using this plugin, you can directly access Subversion repositories from Eclipse.

- Goto *Help* | *Software Updates* | *Find and Install*. Select *Search for new features to install* and click *Next*.
- Click on *New Remote Site...* Enter name `Subclipse` and URL `http://subclipse.tigris.org/update_1.0.x`.
- Select the Subclipse Site and click *Finish*. The Search Results should display a feature named *Subclipse*.
- Select the Subclipse feature and click Next. Accept the license terms and click Next. Click Finish.
- There will be a warning *You are about to install an unsigned feature*. Simply click *Install*.
- You will be prompted to restart the workbench. Click *Yes*.
- Select *Window* | *Open Perspective* | *Other...* | *SVN Repository Exploring*.
- Select *Window* | *Show View* | *SVN Repository*.
- If you are forced to use a proxy for HTTP and HTTPS, you have to edit a configuration file so that Subversion will use your proxy. Using any text editor, open the file servers in the folder `%APPDATA%\Subversion`. If this folder does not exist, make sure you did not miss any of the preceding steps. The folder gets created when you first open the SVN Repository view.

  Go to the `[global]` section at the end of the file, uncomment and edit the lines `http-proxy-host` and `http-proxy-port` to reflect the proxy settings at your site.

## 3.3. Creating a local dstools project from the Subversion repository

- In Eclipse, go to the SVN Repository view in the SVN Repository Exploring perspective.
- Select *New* | *Repository Location* from the context menu.
- Fill in the URL `https://svn.sourceforge.net/svnroot/dstools`. Click

*Finish*.

- When prompted for accepting a digital certificate, click *Accept Permanently*.
- Expand the repository tree and select the subnode `trunk/dstools`.
- Select *Checkout...* from the context menu of this node.
- Enter a project name. If you are expecting to work on multiple versions in parallel (e.g. trunk and development), make sure to select a meaningful name, e.g. `dstools-trunk`. Click *Finish*.

## 3.4. Setting up the project properties

- Switch to the Java perspective and select your new project `dstools-trunk`.
- Select the Ant build file `build.xml` from the root directory and open *Run As | 2 Ant Build...* from the context menu. This will open the *External tools* dialog.
- Select the *Refresh* tab and activate the checkbox *Refresh resources upon completion*.
- Select the *Build* tab and deactivate *Build before launch*.
- Select the *Classpath* tab and click *Add External JARs...* Add `%ECLIPSE_HOME%\plugins\org.junit_3.8.1\junit.jar`.
- Select the *JRE* tab and activate the radiobutton *Run in the same JRE as the workspace*.
- Click *Apply* and *Close*.
- Select *Window | Show View | Ant*.
- Goto the Ant view and select *Add Buildfiles...* from the context menu.
- Select `dstools-trunk/build.xml` and click OK.

## 3.5. Building the dstools project

Go to the Ant view, open the dstools node and double-click on the all target. This will run an Ant build that writes diagnostic messages to a console tab of the Eclipse workbench.

The build produces class files in `build/classes` and a JAR file `datascript.jar` in `build/jar`.