Guideline

Project Acronym:	**PEPPOL**
Grant Agreement number:	**224974**
Project Title:	**Pan-European Public Procurement Online**

Transport Infrastructure
ICT
Services-Components

PEPPOL-Silicone
Setting up the development infrastructure

Version: 1.5.1
Status: In Use

Editors:
Philip Helger, PEPPOL.AT

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	**Public**	**X**
C	**Confidential, only for members of the consortium and the Commission Services**	

Revision History

Version	Date	Editor	Org	Description
0.1	2011-01-12	PH	BRZ	Initial version
0.2	2011-01-14	PH	BRZ	Added input from IBX (Ana) and BRZ (Jakob)
0.3	2011-01-19	PH	BRZ	Added workaround for m2eclipse problem – thanx to Joan
0.4	2011-02-09	PH	BRZ	Updated several items based on recent finding
1.0	2011-02-25	PH	BRZ	Added chapter on what to do if a project does not compiler after import Added specific note to use Metro 2.0.1 instead of 2.1
1.1	2011-04-20	PH	BRZ	Added recommendation on the usage of Metro 2.1 Specified details on the usage of 64bit software Fixed project paths
1.2	2012-02-19	PH	BRZ	Updated to the latest version of the components Added notes on Eclipse 3.7 and Subversion 1.7 Requiring Maven 3.x Updated to paths for Google Code repository Updated to new layout
1.3	2012-03-21	PH	BRZ	Updated to new template Minor adjustments for Java 7
1.4	2012-03-30	PH	BRZ	Added a new chapter on Portecle Fixed some typos Added a statement on Metro 2.2
1.5	2012-04-02	PH	BRZ	Extended compatibility list for IDEs Fixed some typos Added section on configuration files Fixed some typos in section on logging
1.5.1	2012-04-11	PH	BRZ	Fixed some typos Added some additional explanations and clarifications

Contributors

Organisations

PEPPOL.AT/BRZ (Bundesrechenzentrum)[1], Austria, http://www.brz.gv.at/
IBX/Cap Gemini
alfa1lab
UPRC (University of Piraeus)

Persons

Philip Helger, BRZ
Ana Chiritescu (IBX/Cap Gemini)
Jakob Frohnwieser (BRZ)
Joan Farfan (alfa1lab)
Ioannis Tsampoulatidis (UPRC)
Alexandru Pislaru (Cap Gemini)

[1] English: Austrian Federal Computing Centre

Table of Contents

1 Introduction

1.1 Objective and Scope

This document is the introduction document on how to develop for the PEPPOL Silicone implementation. This document does not contain information that is specific to BusDox or PEPPOL components (SML, SMP and AP).
All elements listed here where tested under Windows XP, Vista and 7. Theoretically everything is platform independent and can be built on any platform that supports the required software components. It works with 32 and 64 Bit systems, Java 6 and Java 7 as well as an arbitrary set of application servers.
It was reported that the project setup works for Eclipse, Netbeans and IntelliJ Idea.

1.2 Audience

The audience for this document is organizations in need for a short introduction to the PEPPOL Silicone development process. These may include the following PEPPOL stakeholders:

- PEPPOL Community Governance
- Contracting Authorities
- Economic Operators
- ICT Providers
- Service Providers

More specific it is the following roles:

- ICT Architects
- ICT Developers
- ICT Governing participants

2 Software components

Prerequisites to developing and running the PEPPOL Silicone components:

- It is assumed that you have checked out the Google Code SVN components.
- Internet access

The following software components are recommended for development

- Latest Sun JDK 1.6.x/1.7.x with Metro 2.1.1 (it works also with 2.0.1 but 2.1.1 is recommended; version 2.2 is not yet tested)
- Eclipse 3.6/3.7 JEE (Netbeans and IntelliJ Idea will also work)
    - M2eclipse plugin
    - Subclipse plugin
- Apache Maven 3.x

Optional software components but strongly recommended:

- Subversion client (either command line or TortoiseSVN or both)
- Subclipse (Eclipse Subversion plugin)

For running some of the applications:

- Apache Tomcat 6.x/7.x or Jetty 7.x
- MySQL 5.x (or another relational database)

2.1 Prerequisites

Check out the PEPPOL Silicone implementation from the Google Code Subversion (SVN) directory. Use **https**://peppol-silicone.googlecode.com/svn/trunk/ as the SVN base directory if you are a developer or use **http**://peppol-silicone.googlecode.com/svn/trunk/ for read-only access. Either use the command line client (svn co **https**://peppol-silicone.googlecode.com/svn/trunk/) or use TortoiseSVN (Checkout… and enter **https**://peppol-silicone.googlecode.com/svn/trunk/ into the field "URL of repository").

It is strongly recommended to avoid blanks in the path where you check out, as this may result in severe and hard to track problems when building! See chapter 2.5 for details on Subversion.

2.2 Sun JDK

The latest Sun JDK (1.6.0_31 at the time of writing) can be downloaded from the following website http://www.oracle.com/technetwork/java/javase/downloads/index.html
Because of updates of components that are part of the runtime library (rt.jar), a few parts of the Metro framework must be present in the JDK directory:

1. Download version 2.1.1 of the Metro framework from http://metro.java.net/2.1.1/
   Newer Metro versions (like 2.2) have not yet been tested.
2. Unzip the Metro download and copy the file "webservices-api.jar" into your JDK endorsed directory (e.g. C:\Program Files\Java\jdk1.6.0_31\jre\lib\endorsed)

Ensure that the "JAVA_HOME" environment variable points to this Java installation – see chapter 5.1 for details.
You may also choose to use Java 7 for development and deployment, but it hasn't been tested extensively.

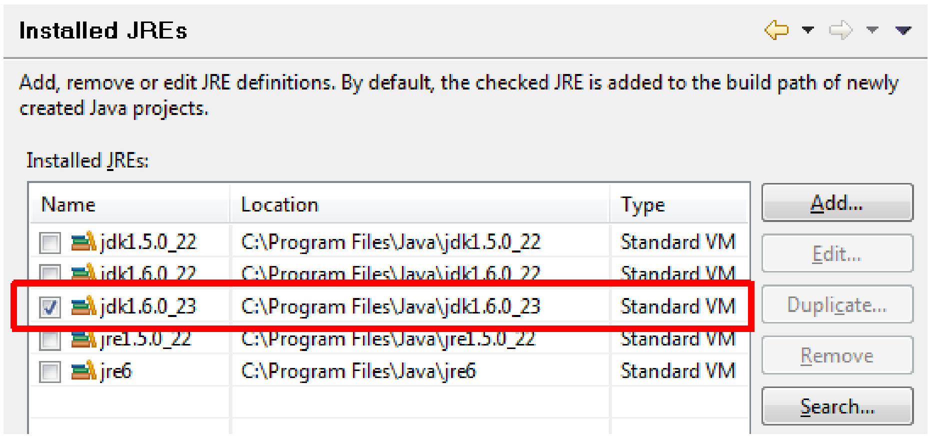
2.3 Eclipse 3.6/3.7

Because of potential incompatibilities with other Eclipse plugins it is strongly recommended to setup a new Eclipse instance (separate installation directory) and separate workspace for PEPPOL development.
Download the latest Eclipse 3.6 JEE edition from http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/heliossr2 or Eclipse 3.7 JEE edition from http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/indigosr1. Choose your platform on the right side.
After downloading and unzipping, run eclipse.exe for the first time and select a new workspace directory.
Then install the following Eclipse plugins (via Help | Install new Software...):

- M2eclipse (Update site http://download.eclipse.org/technology/m2e/releases)
  Note: some new Eclipse installations already come with a bundled m2eclipse
- Subclipse (Update site http://subclipse.tigris.org/update_1.6.x for Subversion 1.6.x or http://subclipse.tigris.org/update_1.8.x for Subversion 1.7.x)
  When using 64 bit Java you may be presented with an obscure error message. This can be resolved by locally installing a native 64 Bit SVN command client like SlikSVN: http://www.sliksvn.com/en/download

Afterwards restart Eclipse. Modify the Eclipse preferences (Window | Preferences) as follows:

- Java
  - Code style
    - Formatter: Import the formatter from the SVN file "java/00config/eclipse-formatter-peppol.xml"
  - Installed JREs: ensure that the previously installed JDK is contained and selected in this list. If it is already contained, press "Edit..." and select "Restore Default" to ensure that the endorsed libraries (Metro) are correctly contained. If it is not contained, use "Search..." and locate the Java directory; afterwards select the latest JDK 1.6/1.7 instance.



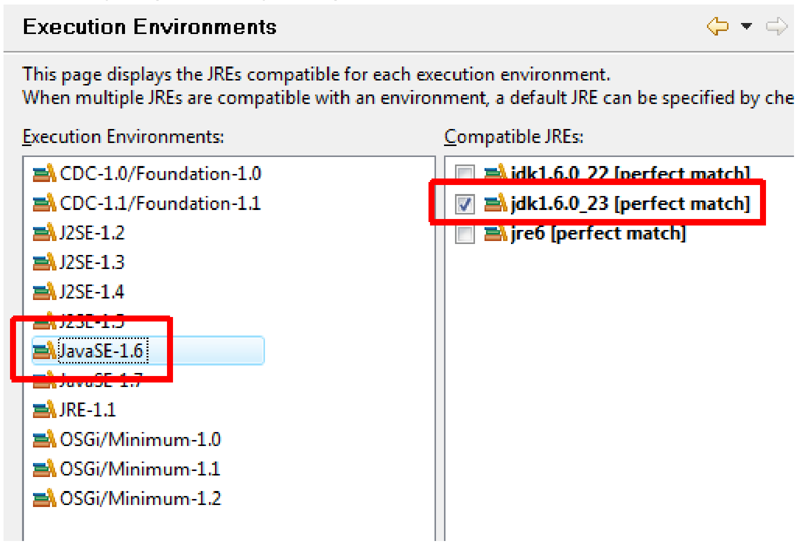
Installed JREs

Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

Name	Location	Type
☐ jdk1.5.0_22	C:\Program Files\Java\jdk1.5.0_22	Standard VM
☐ jdk1.6.0_22	C:\Program Files\Java\jdk1.6.0_22	Standard VM
☑ jdk1.6.0_23	C:\Program Files\Java\jdk1.6.0_23	Standard VM
☐ jre1.5.0_22	C:\Program Files\Java\jre1.5.0_22	Standard VM
☐ jre6	C:\Program Files\Java\jre6	Standard VM

- Execution Environments: Ensure that the execution environment “JavaSE-1.6” is mapped to the JDK you previously setup.



Execution Environments

This page displays the JREs compatible for each execution environment.
When multiple JREs are compatible with an environment, a default JRE can be specified by che

Execution Environments:
- CDC-1.0/Foundation-1.0
- CDC-1.1/Foundation-1.1
- J2SE-1.2
- J2SE-1.3
- J2SE-1.4
- J2SE-1.5
- JavaSE-1.6
- JavaSE-1.7
- JRE-1.1
- OSGi/Minimum-1.0
- OSGi/Minimum-1.1
- OSGi/Minimum-1.2

Compatible JREs:
- ☐ jdk1.6.0_22 [perfect match]
- ☑ jdk1.6.0_23 [perfect match]
- ☐ jre6 [perfect match]

Now import the following projects – that are required by most projects - from the checked out SVN directory:

- java/commons-busdox
- java/commons-peppol

The import can be done inside the “Package Explorer” using “Import…” | “Existing projects into workspace” and selecting the project base directory.

2.3.1 Running Eclipse with a JDK

By default Eclipse uses the public JRE for executing itself. Maven sometimes expects Eclipse to run with a JDK. Therefore you need to modify the *eclipse.ini* file that resides in your Eclipse installation directory. For Windows you may simple add the following lines to *eclipse.ini* (of course the path to the JDK must be adopted):

```
-vm
C:/Programme/Java/jdk1.6.0_31/bin/javaw.exe
```

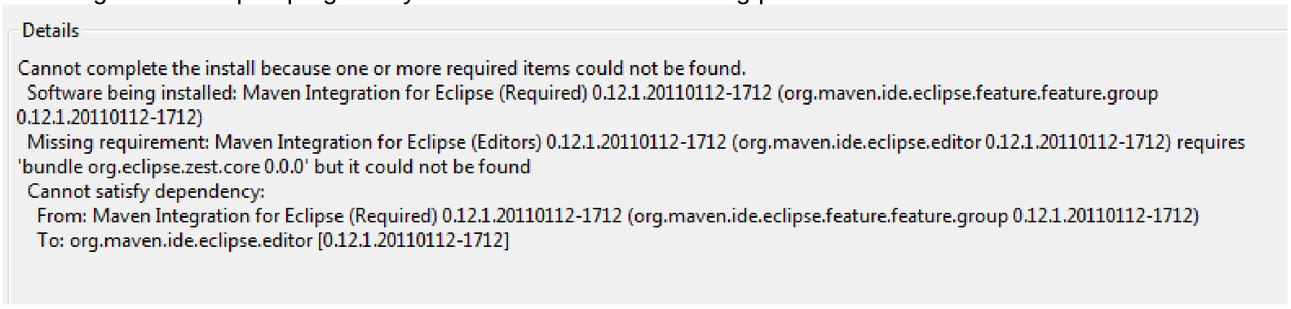
Since the format of the eclipse.ini file is very specific, it is strongly recommended to read the following web pages before performing the modifications to ensure the Eclipse will start up correctly:

- http://wiki.eclipse.org/FAQ_How_do_I_run_Eclipse%3F#eclipse.ini
- http://wiki.eclipse.org/Eclipse.ini

This modification requires a restart of Eclipse in case you did it while running Eclipse.

2.3.2 Issues with the m2eclipse plugin

Installing the m2eclipse plugin may fail because of the following problem:



```
Details

Cannot complete the install because one or more required items could not be found.
  Software being installed: Maven Integration for Eclipse (Required) 0.12.1.20110112-1712 (org.maven.ide.eclipse.feature.feature.group 
0.12.1.20110112-1712)
  Missing requirement: Maven Integration for Eclipse (Editors) 0.12.1.20110112-1712 (org.maven.ide.eclipse.editor 0.12.1.20110112-1712) requires 
'bundle org.eclipse.zest.core 0.0.0' but it could not be found
  Cannot satisfy dependency:
    From: Maven Integration for Eclipse (Required) 0.12.1.20110112-1712 (org.maven.ide.eclipse.feature.feature.group 0.12.1.20110112-1712)
    To: org.maven.ide.eclipse.editor [0.12.1.20110112-1712]
```

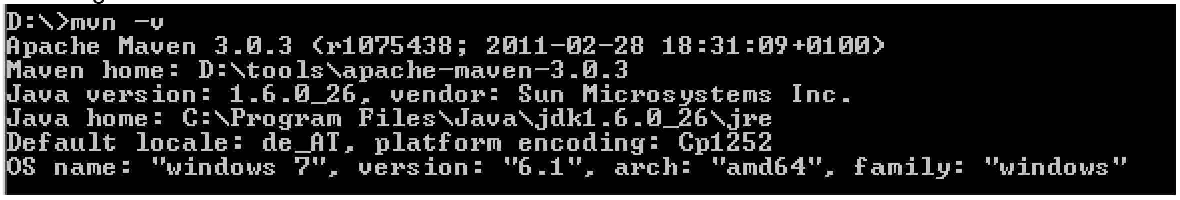
In that case you need to add the following update site:
http://download.eclipse.org/tools/gef/updates/releases/
From that update site you need to install the “zest visualization toolkit” library before you can install m2eclipse. Performing an Eclipse restart after installing “zest visualization toolkit” is necessary.

2.3.3 Projects not compiling after initial import

If you have the issue that projects are not compiling after import (especially “commons-busdox” is a candidate), try calling “Run as” “Maven generate-sources” to create all generated source files. This seems to be necessary due to a bug in m2eclipse that won’t execute these actions automatically upon project import.

2.4 Apache Maven 3.x

Download the latest Apache Maven 3.x from http://maven.apache.org and unzip the file to a common directory. Set the environment variable “M2_HOME” to the unzipped directory (the directory that includes the “bin” directory). Extend the environment variable “PATH” to contain “%M2_HOME%/bin”. Use the correct separator to append this item to the existing PATH (“;” on Windows and “:” on Unix).
After setting up Maven open a command line and type “mvn –v”. The output should look similar to the following screen:



```
D:\>mvn -v
Apache Maven 3.0.3 (r1075438; 2011-02-28 18:31:09+0100)
Maven home: D:\tools\apache-maven-3.0.3
Java version: 1.6.0_26, vendor: Sun Microsystems Inc.
Java home: C:\Program Files\Java\jdk1.6.0_26\jre
Default locale: de_AT, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
```

Maven provides support for repository managers (mirrors), proxy servers etc. Please see the documentation on the Maven website for details.

Note: only use “mvn install” when building your projects, as “mvn deploy” tries to deploy to a phloc.com server, what can currently only be done by Philip!

2.4.1 Maven repository managers

If you are using a Maven repository manager (like Sonatype Nexus, Apache Archiva or JFrog Artifactory) you may need to add the following repositories to the proxied ones (Please contact your Maven administrator for the details):

- http://repo.phloc.com/maven2/ (for releases and snapshots; Maven 2 layout)

In case you cannot modify your repository manager you can work around this issue by temporarily commenting out the element “<mirrorOf>” in your local Maven 2 settings.xml (*userHome*/.m2/settings.xml).

2.5 Subversion

TortoiseSVN is a Windows explorer integration that lets you perform SVN command via the GUI. Download it at http://tortoisesvn.net/downloads.html and select either 32 bit or 64 bit version as you need it.
A Windows command line SVN client can be acquired from SlikSVN: http://www.sliksvn.com/en/download. Also choose correctly between 32 bit and 64 bit version.

As Subversion 1.7.x is a major update, it must be ensured that all Subversion components (command line client, TortoiseSVN and the Eclipse plugin) use the same version. So either all versions are on 1.6.x or on 1.7.x - mixing will result in weird errors!

2.6 Portecle

As PEPPOL makes heavy use of key stores and trust stores, using the standard JDK commandline application "keytool" may be unnecessarily complex. Instead the open source application Portecle can be used. It's standalone tool which allows you to manage your Java key stores and trust stores in a graphical way. It can be downloaded from http://portecle.sourceforge.net/ or directly started via Java WebStart. It works on all operating systems with a JRE installed.

3 Special cases

3.1 Proxy server

If you're behind a proxy server you may face issues when using Subversion or Maven.

3.1.1 Maven

For Maven update both your personal settings.xml (see http://maven.apache.org/settings.html for details) and - if this does not solve all issues (e.g. for the "commons-busdox" project using the wsimport plugin) - pass the required parameters on the command line like this:

```
mvn install -Dhttp.proxyHost=10.0.0.1 -Dhttp.proxyPort=8080
```

See http://download.oracle.com/javase/6/docs/technotes/guides/net/proxies.html for a list of all proxy related command line parameters.

3.1.2 Subversion

Please see http://subversion.apache.org/faq.html#proxy for details on how to configure Subversion to use a proxy server.

4 Default components

4.1 Logging

Source level logging is done via SLF4J (http://www.slf4j.org/) only. SLF4J is a logging façade comparable to Apache commons-logging but has less known issues and offers binding to Log4J and other logging frameworks. Don't use commons-logging or java.util.logging in PEPPOL Silicone code. The preferred way to create a logger is as follows:

```
private static final Logger s_aLogger = LoggerFactory.getLogger (MyClass.class);
```

"MyClass" has to be replaced with the actual class name. Logger is org.slf4j.Logger and LoggerFactory corresponds to org.slf4j.LoggerFactory. The field name "log" or "s_aLogger" should be used.
All messages with lower priority than "info" have to be prefixed with a query for performance reasons[2]:

```
If (s_aLogger.isDebugEnabled ())
  S_aLogger.debug ("Doing this and that")
```

For log messages with a log level >= "info" there is no necessity to prefix the statements, as "info" is the default log level that is emitted.

[2] If you e.g. have log.debug(„My value is „ + sValue") the preceding „isDebugEnabled" check may spare you the string concatenation operations.

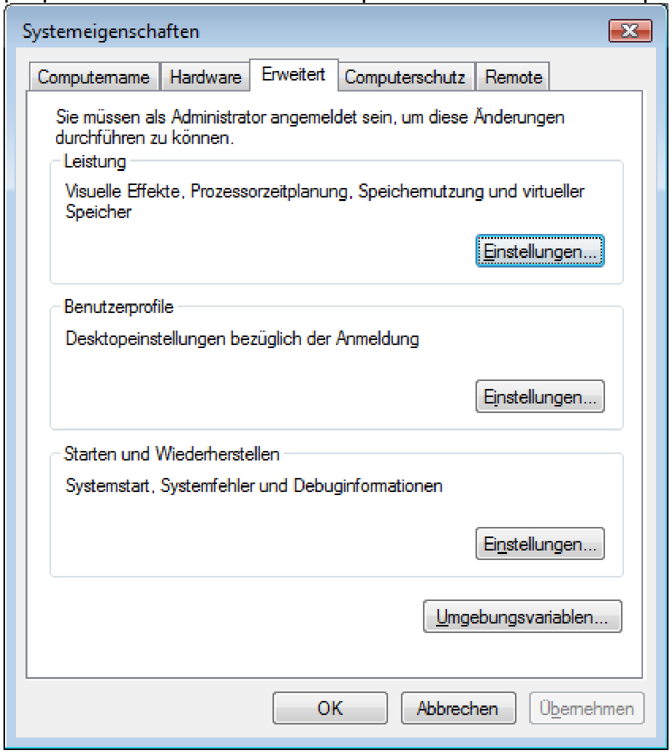
4.2 Configuration files

As the PEPPOL Silicone implementation makes heavy usage of configuration files, it ships with a single class that encapsulates this handling: at.peppol.commons.utils.ConfigFile
As configuration files usually need to be changed, before PEPPOL Silicone components can be ran, the code usually contains special handling for configuration files prefixed with “private-“. If you e.g. want to modify the START AP server configuration file called “configServer.properties” and you have a local file called “private-configServer.properties”, the one with the “private-“ prefix takes precedence. Also in all directories where properties files reside, files with the prefix “private-“ are ignored and won’t be committed to the SVN to minimize an accidental commit of a password to the public SVN.

5 Appendix

5.1 Setting environment variables in Windows

Setting environment variables in Windows is a hard to find if you want set values system wide.
On Windows XP press <Windows key>+<Pause>. On Windows Vista you can execute the application “SystemPropertiesAdvanced.exe” or press <Windows key>+<Pause> followed by selecting the “advanced properties” text in the left task pane. You should end up with a window like this:


Systemeigenschaften

Computername | Hardware | Erweitert | Computerschutz | Remote

Sie müssen als Administrator angemeldet sein, um diese Änderungen durchführen zu können.

Leistung
Visuelle Effekte, Prozessorzeitplanung, Speichernutzung und virtueller Speicher
Einstellungen...

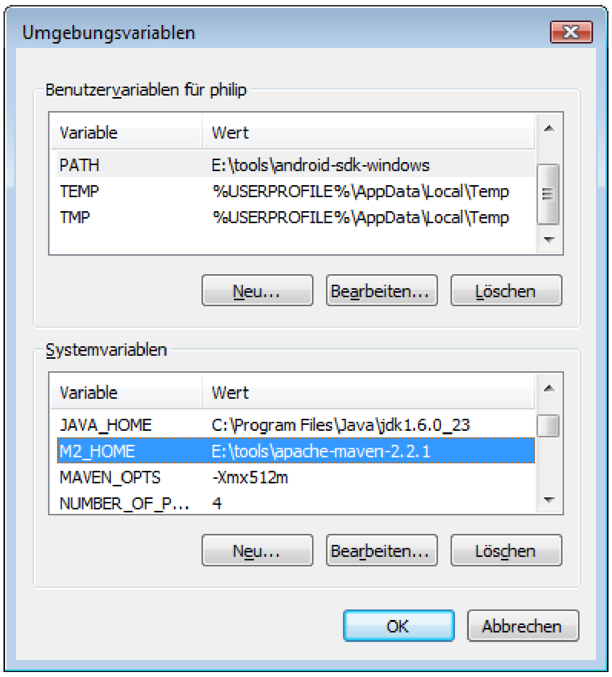
Benutzerprofile
Desktopeinstellungen bezüglich der Anmeldung
Einstellungen...

Starten und Wiederherstellen
Systemstart, Systemfehler und Debuginformationen
Einstellungen...

Umgebungsvariablen...

OK | Abbrechen | Übernehmen

Go to the “Advanced” tab (already selected in the screenshot above) and select the “Environment variables…” button (in the above screenshot it is the button directly above “OK” and “Cancel”).


Umgebungsvariablen

Benutzervariablen für philip

Variable	Wert
PATH	E:\tools\android-sdk-windows
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Systemvariablen

Variable	Wert
JAVA_HOME	C:\Program Files\Java\jdk1.6.0_23
M2_HOME	E:\tools\apache-maven-2.2.1
MAVEN_OPTS	-Xmx512m
NUMBER_OF_P...	4

The top list shows environment variables that are user specific. This means only your account has access to them. I prefer to set the variables system wide but there are also good reasons to set it in the personal section. After you changed the values here, you need to close and re-open existing command line windows, as changes only take effect after a restart of the command line application.