

# XSTAMPP

## Setup Guide

Lukas Balzer

### Contents

<b>How to build XSTAMPP</b>	<b>1</b>
<b>Working on XSTAMPP</b>	<b>2</b>
Language skills . . . . .	2
Setting up the environment . . . . .	2
Running XSTAMPP from Eclipse . . . . .	2
Contribute . . . . .	3
Create a new plugin: . . . . .	4
Create a new Version . . . . .	4
<b>Known Issues</b>	<b>5</b>

### How to build XSTAMPP

1. The build requires Maven ( $\geq 3.3$ ) and Java ( $> 1.7$ ) either from *https://maven.apache.org/* or from the IDE
2. go to the xstampp.parent directory in the root path of the xstampp project (where this file is located)
3. open a command in the xstampp.parent dir and execute
  - (a) 'mvn clean verify' to build xstampp with xstpa and cast already included
  - (b) 'mvn clean install' as 3.1 but also installs xstampp on *[user]/.m2* for usage as local dependency of other builds
4. the build artifacts are located in the astpa.repository/target

# Working on XSTAMPP

## Language skills

1. XSTAMPP is written in **Java 8** (depending on the Plugin)
2. The documentation (help contents in xstampp.[plugin]/html) is provided in **html 4.0** and styled with **CSS 3**
3. The hazx schema is given in **XMLSchema 1**

## Setting up the environment

- Eclipse for RCP and RAP Developers (Plug-in Development)<sup>1</sup> (> *Lunar*)
- At least JavaSE 1.8
- To install gef ( *help*→*install new software*→<http://download.eclipse.org/tools/gef/updates/releases/>)
- To install nebula grid from eclipse.org<sup>2</sup>
- To install maven<sup>3</sup>
- import/clone xstampp projects using the included git
  1. open the *Import* Dialog selecting *File*→*Import*
  2. in the Import menu click *Git*→*Projects from Git* and follow the steps of the import wizard
- To resolve upcoming error messages refer to Known Issues Section [chap:issues]

## Running XSTAMPP from Eclipse

1. Go to *xstampp.repository*→*xstampp.product*
2. In the product editor click on *Testing*→ *Launch an Eclipse Application*
3. The run fails on the first try, which is normal because we haven't included the required plugins yet
4. In the last step Eclipse has created a *Run configuration* for us which we are going to use now
  - (a) right click on the *xstampp* project and select *Run As*→ *Run Configurations*..

---

<sup>1</sup><http://eclipse.org/downloads>

<sup>2</sup><http://download.eclipse.org/technology/nebula/snapshot/>

<sup>3</sup><https://maven.apache.org/download.cgi>

- (b) in the opening dialog search for the Plug-ins Tab (see figure [fig:runConfig])(you may need to adjust the size of the window)
- (c) you can now include/exclude the xstampp plug-ins included in your runtime
- (d) finally find/press the button *Add Required Plug-ins* and Apply/Run the run configuration

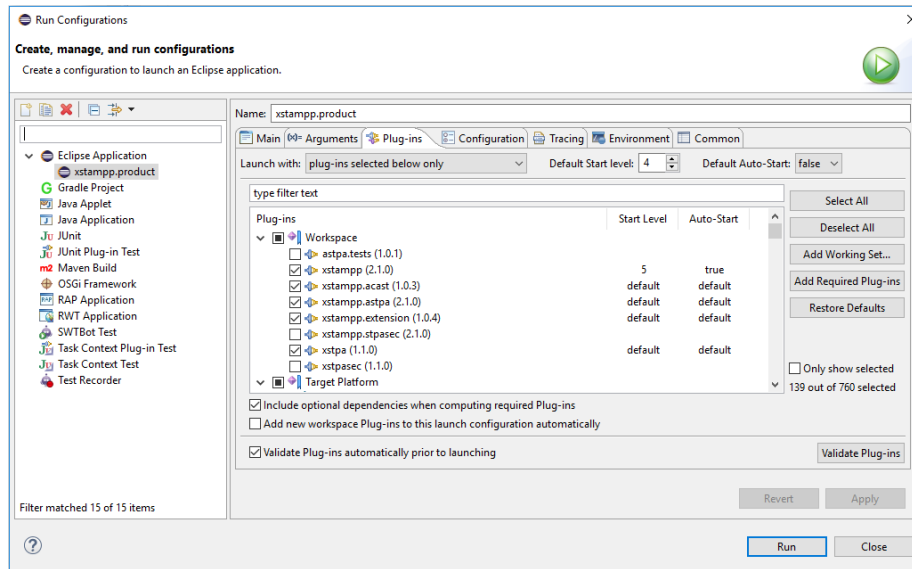


Figure 1: Before eclipse can successfully run xstampp the required plug-ins must be included in the runtime

## Contribute

- Setting up Eclipse Preferences (open *Eclipse*→*Window*→*Preferences*):
  1. Go to *XML*→*XML Files*→*Editor*
    - (a) set the *Line width* to **120**
    - (b) check the radio box *Indent using spaces*
    - (c) set *Indentation size* to **4**
  2. Go to *Java*→*Code Style*→*Formatter*
    - (a) Press *Import...*
    - (b) Import the *java\_formatter.xml* in `< repo >/xstampp/misc/java_formatter.xml`

### Create a new plugin:

- Contributing plugins should be named as *xstamp.< yourPlugin >*
- Create a new plugin by clicking *New→Others..→Plug-in Development→Plug-in Project*
- Add dependencies *xstamp* and *xstamp.extension*
- Add the extension *xstamp.extension.steppedProcess* to your plugin
- Create a class implementing *IDataModel*
- Create *stepEditors* which must extend *StandartEditorPart* and implement *IViewBase*
- *Xstamp* loads the files which are selected in the load Dialog or already located in the workspace by directly calling a load command registered as command in the *steppedProcess* extensionPoint herefore it needs:
  - a load job which extends *AbstractLoadJob*
  - a load Handler extending *AbstractHandler* which is registered as default handler for the load command
  - let your handler.execute() return a new instance of your load job
- *XSTAMPP* uses Eclipse Tycho as build tool, to include a plugin into its build process it need to be configured as Maven plugin<sup>4</sup>

### Create a new Version

- All changes must be recorded in the *CHANGELOG.md*
- If *misc/docu/README.tex* has been changed than:
  - Download LaTeX(MikTex<sup>5</sup> for Windows or MacTex<sup>6</sup> for Mac)
  - This should contain an html(for eclipse help), md(for GitHub) and a pdf version of the Readme this can be achived by using Pandoc<sup>7</sup>

```
* cd misc/docu
* pandoc -s README.tex -o README.pdf -toc
* pandoc -s README.tex -o README.html
* pandoc -s README.tex -o README.md
* cp README.html ../../README/html/
```

---

<sup>4</sup><http://www.vogella.com/tutorials/EclipseTycho/article.html>

<sup>5</sup><https://miktex.org/>

<sup>6</sup><http://tug.org/mactex/>

<sup>7</sup><https://pandoc.org>

- \* `cp README.pdf ../../`
- Update the *xstamp/html/CHANGELOG.html* (using Pandoc):
  - `cd ../../`
  - `pandoc -s CHANGELOG.md -o CHANGELOG.html`
  - `cp CHANGELOG.html xstamp/html/`
- *createFiles.cmd* is a Windows batch script that executes all of the above commands to create the release files

## Known Issues

### An API baseline has not been set for this Workspace

1. Go to the Eclipse Problems View (*Window → ShowView → Problems*)
2. Right click the 'API baseline' error
3. In the context menu select *QuickFix*
4. A Preference Window filtered for the API Baselines opens up
5. in that Dialog find the field *MissingAPIBaseline* and set it to *Ignore* (see figure [fig:APIerror])

### Plugin Execution not covered by lifecycle configuration

1. Go to *Window → Preferences → Maven → Error/Warnings*
2. find the line 'Plugin Execution not covered..'
3. Set the Value to ignore, by choosing selecting 'ignore' in the combo box
4. Click on Apply/Ok to rebuild the projects

### When cloning into/ importing xstamp and its sub projects to eclipse the project dependencies must be located sometimes

1. In the Project Explorer right click on the project 'Build Path->Configure Build Path'
2. In the 'Java Build Path' Page click on 'Source', by doing that java relocates the source folders in the projects andsets the dependencies
3. hit Apply/Ok to store the settings

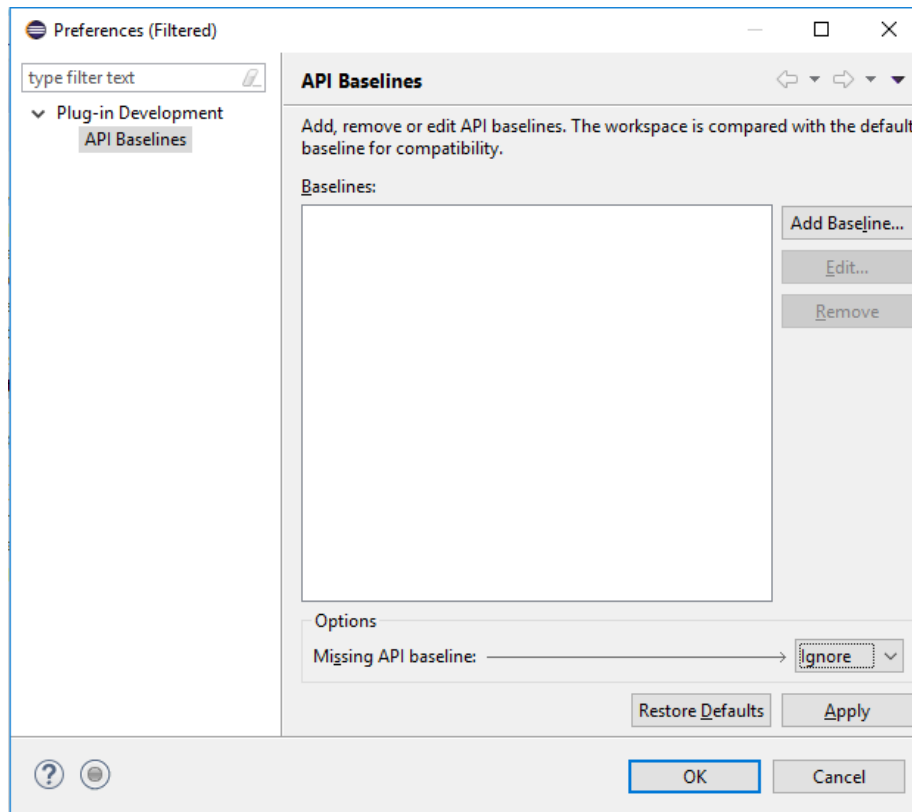


Figure 2: The API baseline can be ignored