# xcommand

## Table of contents

# 1. XCommand

## 1.1. Welcome to xcommand

### 1.1.1. Introduction

XCommand is the implemention of the extended command pattern. This article describes the concept behind XCommand. XCommand's website is available at xcommand.sourceforge.net.

# 2. Collage

## 2.1. Collage: 5 Minutes Tutorial

This tutorial will get you going with collage in 5 minutes. It will describe how to get the required jar files and shows a first usage example.

### 2.1.1. Prerequisites

The idea of this tutorial is to get most possible new users up an running smoothly in a very short time. Therefor I need to make certain assumptions which result in the following requirements to run this tutorial. If you do not like some of them check out the alternative hints behind them.

- Java 1.4.x or newer
- Apache Ant (1.6.x or newer)
- Available Internet while running ant scripts

### 2.1.2. Preparation

- Create an empty directory called `X001` in any location you like.
- Then create a new file named `build.xml` with the following content inside it:

```
<?xml version="1.0"?>
<project name="collage_5mintutorial" default="init" basedir=".">
  <property name="rep" value="http://www.sven-ehrke.de/tmp"/>
  <get src="${rep}/build.xml" dest="lib-build.xml" verbose="true"
usetimestamp="true"/>
  <import file="lib-build.xml"/>
</project>
```

This small ant file is able to bootstrap your first collage example.

- open a command window in your new `X001` directory and invoke: `ant compile run`

You should see an output similar to this one:

```
Buildfile: build.xml
      [get] Getting: http://www.sven-ehrke.de/tmp/build.xml
      [get] .
      [get] last modified = Sun Feb 18 10:52:07 CET 2007

init:
    [mkdir] Created dir: Z:\Daily\2007-02-18\lib
    [mkdir] Created dir: Z:\Daily\2007-02-18\output\classes
      [get] Getting: http://www.sven-ehrke.de/tmp/commons-lang-2.2.jar
      [get] .......................
      [get] last modified = Sun Feb 18 10:15:35 CET 2007
      [get] Getting: http://www.sven-ehrke.de/tmp/commons-logging.jar
      [get] ..
      [get] last modified = Sun Feb 18 10:15:24 CET 2007
      [get] Getting: http://www.sven-ehrke.de/tmp/javacc.jar
      [get] .....
      [get] last modified = Sun Feb 18 10:15:54 CET 2007
      [get] Getting: http://www.sven-ehrke.de/tmp/javassist.jar
      [get] .......
      [get] last modified = Sun Feb 18 10:16:21 CET 2007
      [get] Getting: http://www.sven-ehrke.de/tmp/junit.jar
      [get] ...
      [get] last modified = Sun Feb 18 10:16:20 CET 2007
      [get] Getting: http://www.sven-ehrke.de/tmp/xcommand-core.jar
      [get] ..
      [get] last modified = Sun Feb 18 10:33:38 CET 2007
      [get] Getting: http://www.sven-ehrke.de/tmp/xcommand-misc.jar
      [get] .
      [get] last modified = Sun Feb 18 10:33:43 CET 2007
      [get] Getting: http://www.sven-ehrke.de/tmp/collage.jar
      [get] ...
      [get] last modified = Sun Feb 18 10:33:43 CET 2007
    [mkdir] Created dir: Z:\Daily\2007-02-18\src
    [mkdir] Created dir: Z:\Daily\2007-02-18\src\org\collage\example
      [get] Getting: http://www.sven-ehrke.de/tmp/X001.java
      [get] .
      [get] last modified = Sun Feb 18 10:37:50 CET 2007

gensrc:

compile:
    [javac] Compiling 1 source file to
Z:\Daily\2007-02-18\output\classes

init:
      [get] Getting: http://www.sven-ehrke.de/tmp/commons-lang-2.2.jar
      [get] local file date : Sun Feb 18 10:15:36 CET 2007
      [get] Not modified - so not downloaded
      [get] Getting: http://www.sven-ehrke.de/tmp/commons-logging.jar
      [get] local file date : Sun Feb 18 10:15:24 CET 2007
      [get] Not modified - so not downloaded
      [get] Getting: http://www.sven-ehrke.de/tmp/javacc.jar
      [get] local file date : Sun Feb 18 10:15:54 CET 2007
      [get] Not modified - so not downloaded
      [get] Getting: http://www.sven-ehrke.de/tmp/javassist.jar
      [get] local file date : Sun Feb 18 10:16:22 CET 2007
      [get] Not modified - so not downloaded
      [get] Getting: http://www.sven-ehrke.de/tmp/junit.jar
      [get] local file date : Sun Feb 18 10:16:20 CET 2007
      [get] Not modified - so not downloaded
      [get] Getting: http://www.sven-ehrke.de/tmp/xcommand-core.jar
      [get] local file date : Sun Feb 18 10:33:38 CET 2007
      [get] Not modified - so not downloaded
      [get] Getting: http://www.sven-ehrke.de/tmp/xcommand-misc.jar
      [get] local file date : Sun Feb 18 10:33:44 CET 2007
      [get] Not modified - so not downloaded
      [get] Getting: http://www.sven-ehrke.de/tmp/collage.jar
      [get] local file date : Sun Feb 18 10:33:44 CET 2007
      [get] Not modified - so not downloaded
      [get] Getting: http://www.sven-ehrke.de/tmp/X001.java
      [get] local file date : Sun Feb 18 10:37:50 CET 2007
      [get] Not modified - so not downloaded
Overriding previous definition of reference to id.classpath

run:
     [java] Hi Uli. How are you?

BUILD SUCCESSFUL
Total time: 19 seconds
```

Your small ant file downloaded anther ant file with targets for getting the required jars, compilation and to run the example. This ant file is imported into your ant file. After having downloaded the required jar files the `compile` and the `run` target get invoked.

### 2.1.3. Hello ${firstname}

The interesting part is of course the Java code:

```
package org.collage;

import org.collage.template.Template;
import org.collage.template.TemplateSource;
import org.collage.dom.evaluator.text.TextTemplate;

import java.util.Map;
import java.util.HashMap;

public class X001
{
  public static void main(String[] args)
  {
    Map dataCtx = new HashMap();
    dataCtx.put("firstname", "Uli");
    Template t = new TextTemplate(new TemplateSource("Hi ${firstname}.
How are you?"));
    String s = t.getStringResult(dataCtx);
    System.out.println(s);
  }
}
```

First a new `HashMap dataCtx` is created and populated with data (here the firstname of my wife). Next a new `TemplateSource` is created from the String `Hi ${firstname}. How are you?` from which a new `TextTemplate` is created. With `TextTemplate.getStringResult(dataCtx)` a String with all placeholders replaced with the values in `dataCtx` is returned. This leads to the expected output: `Hi Uli. How are you?`

## 2.2. Collage usage examples

hallo du da

### 2.2.1. Hello ${firstname}

```
package org.collage;

import org.collage.template.Template;
import org.collage.template.TemplateSource;
import org.collage.dom.evaluator.text.TextTemplate;

import java.util.Map;
import java.util.HashMap;

public class X001
{
  public static void main(String[] args)
  {
    Map dataCtx = new HashMap();
    dataCtx.put("firstname", "Uli");
    Template t = new TextTemplate(new TemplateSource("Hi ${firstname}.
How are you?"));
    String s = t.getStringResult(dataCtx);
    System.out.println(s);
  }
}
```

# 3. Best Practices

## 3.1. Best Practices

### 3.1.1. Best Practices

Best Practices ...

# 4. All