The "complete" KickStart manual

Users, Administrators and Developers Guide to KickStart

Rick-Rainer Ludwig

version: 1.0.0
date: 29th July 2010

This is the "official" manual for KickStart.

Contents

List of Figures

List of Tables

Part I

Overview

Chapter 1

About KickStart

KickStart is a Java application launcher framework for easy installation and usage. The most annoying thing in Java is the deployment procedure where a lot of different Jar files need to be packed and within the manifests need to be coupled together by there CLASSPATH configuration. A simple add, remove or change of different classes or a refactoring within this Jars is always time consuming.

KickStart tries to avoid this by starting itself on a pure JRE installation and collection afterwards all additionally needed libraries and collecting them for an enhanced CLASSPATH. A coupling of Jars is not needed anymore and Jars can be replaced, splitted and merged as wanted, as soon as there are all dependencies in range of KickStart during startup.

Part II

Installation and Configuration

Chapter 2

Installation of KickStart

The Installation can be done anywhere wanted. On Unix like System an installation directory like `/opt/JavaApps` is suitable. Within this directory a recommendation is to create the following subdirectories:

- **bin**: Within this directory all executable scripts are located which actually start KickStart and choose the Class in which the needed main() method can be found for startup.
- **config**: This directory should be treated as class library directory and is used for configuration files which are placed into the CLASSPATH and loaded as resource.
- **extlib**: To `extlib` all libraries from external vendors should go in.
- **lib**: Should be the directory where all classes go in which are from own developmens and which are changed during development cycles.
- **log**: Within this directory all log files which are generated by the tools should be located. KickStart itself does not generate any log files.

All directory naming is meant to be inside the installation directory of the KickStart framework.

Chapter 3

System Configuration

3.1 PATH variable

On Unix one adds KickStart to a global `PATH` variable by changing `/etc/profile`, `/etc/profile`, `/etc/bash.bashrc` or `/etc/bash.bashrc.local`. The location depends on the System used. For user related changes the files `/.profile` or `/.bashrc` should be checked.

For Windows in the System Control Panel the path can be adjusted.

3.2 Cron

To add the KickStart `bin` directory into the Cron PATH variable modify `/etc/crontab` and add a line like `PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin`.

Part III

Usage

Part IV

Appendices

Nomenclature

ASCII American Standard Code for Information Interchange - Code table for a number to character mapping

Design Patterns Design Patterns (in german: Entwurfsmuster) are generalized approaches to repeating programming issues. For the most issues there is a fitting approach to solve the issue with a generalized design. Design patterns are well known with generalized names and software which implements and documents these patterns are very easy mainanable. Other programmers have a good chance to understand the code and the design of the software in short time. (see: [1, 2])

RefactoringRefactoring Refactoring is the process of permanent code improvement even after implementing functionality. This process is essential to keep the software code in a clean state for later use. Without Refactoring the code will get out of shape and the implementation process for new functionallity will become more expensive. Therefore, Refactoring is a way to keep code maintanable. (see: [3, 4])

Bibliography

[1] Eric Freeman & Elisabeth Freeman. *Head First Design Patterns*. O'Reilly, 1st edition, Oct 2004.

[2] Erich Gamma & Richard Helm & Ralph Johnson & John Vlissides. *Design Patterns – Elements of Reusable Object-Oriented Software*. Addison-Wesley, 36th edition, July 2008.

[3] Andrew Hunt & David Thomas. *The Pragmatic Programmer*. Addison-Wesley, 22nd edition, January 2008.

[4] Martin Fowler. *Refactoring – Improving the Design of Existing Code*. Addison-Wesley, 22nd edition, August 2008.

Index