



An overview of Koalog Configurator™

Copyright 2002-2006 Koalog™

Solaris, Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both. Windows, Windows 2000 and Windows XP are registered trademarks of Microsoft Corporation. Linux is a registered trademark of Linus Torvalds.

Introduction

Koalog Configurator™ is a configurator written in the Java™ programming language. It includes:

- a model maintenance application;
- an XML importer;
- a customizable Swing front-end;
- a customizable web front-end.

With Koalog Configurator™, end users can rapidly and easily configure products or services while satisfying all relevant business rules.

Features

We detail here the main features of Koalog Configurator™.

Configure in any order

Other configurators impose a predefined sequence of selections on the user, thus restricting the manner and order in which choices can be expressed. This can make the configuration task a slow and tedious process. In particular, when no solution corresponding to the user choices is found at the end of it, the user is invited to restart from the beginning.

With Koalog Configurator™, there is no such constraint. The user can express preferences and wishes in any order, and modify them at any time.

Selections violation

Moreover, Koalog Configurator™ comes with an interactive prioritization module (or explanation module) that allows the user to violate previously made selections.

If the user makes a selection which conflicts with earlier selections, the configurator will prevent the selection being made. Instead, the configurator will show the user which previous selections are incompatible with the most recent failed

selection, and then it will prompt the user to prioritise among them. The configurator will discard the low-priority selections, and will continue as if those selections had not been made.

Pricing rules

In addition to configuration rules, Koalog Configurator™ support pricing rules. With pricing rules, you can define individual configuration item prices depending on user selections. Hence, bundles or version-dependent prices can be implemented.

During the configuration, the user is informed about the impact of his selections on the final price range. Moreover, the user can directly modify the price range: the configuration engine reduces the set of possible configurations accordingly.

Technology

This section provides a brief high-level overview of the technology powering Koalog Configurator™ (figure 1 details the architecture of Koalog Configurator™).

Constraint Programming

Koalog Configurator™ is powered by the technology of constraint programming: it relies on Koalog Constraint Solver™, a Java™ library for constraint solving. It is hence possible to consider all potential configurations at once, and Koalog Configurator™ can compute a single solution, all solutions, or an optimal solution (eg optimizing on price) to a given configuration problem.

Two layers

Koalog Configurator™ consists of two software layers:

- a configuration state layer;

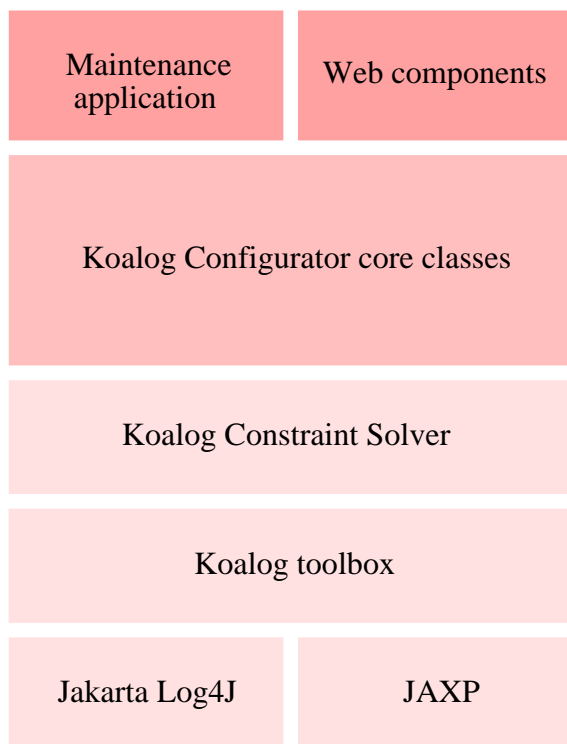


Figure 1: Koalog Configurator™ architecture

- a configuration engine layer, which is stateless.

The configuration state layer is responsible for recording the user choices during the configuration session. The configuration engine layer is responsible for satisfying the constraints of the model, in accordance with the user's choices.

This software architecture makes it possible to efficiently remote and share the engine among different configuration sessions (for example by using sockets or in an EJB framework), and thus enables scalability, reliability, and high performance.

Configuration Models

We explain here how Koalog Configurator™ can be fed with configuration data.

Model maintenance

The maintenance application included with Koalog Configurator™ has a graphical user interface that allows configuration modelers to easily create and maintain configuration models.

Model import

Koalog Configurator™ includes an XML importer, capable of rapidly importing huge configuration models, generated either by the maintenance application or by external legacy systems.

User Front Ends

Koalog Configurator™ comes with two highly customizable front ends:

- a Swing-based GUI;
- a web-based GUI.

Both are built with a set of components that can be reused to create a front-end that perfectly match your needs.

The web components are W3C compliant and integrate to Java™ Server Pages or to Java™ Servlets.

Quality

Koalog Configurator™ has been produced to the highest quality standards:

- it is fully documented;
- it is internationalizable;
- it is fully tested (its test suite uses JUnit¹ and Koalog™ Code Coverage);
- it uses a well-known and highly customizable logging system (Log4J²).

¹JUnit is an open source project initiated by Kent Beck (the author of "Extreme Programming").

²Log4J is an open-source project of the Apache Software Foundation.

Supported platforms

Koalog Configurator™ is a Java™ application and thus smoothly integrates to any recent Java™ environment. See table 1 for detailed information.

Operating System	Java™ Virtual Machine			
	Sun J2SE™ SDK		IBM® JDK	
	1.4.2	1.5.0	1.4.2	1.5.0
SuSE Linux 9.3	yes	yes	yes	yes
Fedora Core 3	yes	yes	yes	yes
Windows® XP operating system	yes	yes	n/a	n/a
Solaris™ Sparc 2.8	yes	yes	n/a	n/a

Table 1: Koalog Configurator™ supported platforms

Commercial information

To evaluate or purchase the latest release of Koalog Configurator™ (v1.1), please contact sales@koalog.com.