

FL Installation Guide

prepared by A. Rozhenko

Contents

- [1 Introduction](#)
 - [1.1 Scope](#)
 - [1.2 The Software Needed for Development](#)
 - [1.3 The Software Needed in Runtime](#)
- [2 Installation Procedure](#)
- [3 Essential Changes in FL 4](#)
 - [3.1 Changes in FL Petal Model](#)
 - [3.2 Changes in FL](#)
- [4 Migration from FL 3](#)
 - [4.1 Conversion of Rose Model](#)
 - [4.2 Conversion of Sources](#)
- [5 Revision History](#)

1 Introduction

1.1 Scope

The document describes the installation procedure for the Foundation Layer 4.0. It also explains the basic distinctions between FL 3 and FL 4, and describes the migration to a new FL version.

1.2 The Software Needed for Development

- Java Development Kit (JDK) 1.1.x or 1.2.x.
- The Foundation Layer 4.0.x released for JDK 1.1 or 1.2.
- The com.sun.java.util.collections package from Sun Microsystems, Inc. for JDK 1.1.x (it is included into the FL distribution).
- Rose Enterprise Edition 98i modeling tool of version 6.1 or later from [Rational Rose](#).

1.3 The Software Needed in Runtime

- Java Runtime Environment (JRE).
- fl.jar library.
- collections.jar library (for JDK 1.1 only).
- Swing 1.1.x for JDK 1.1 or the jawax.swing package for JDK 1.2.

2 Installation Procedure

The Foundation Layer supports both JDK 1.1 and JDK 1.2 platforms. So, you have to decide at first what FL distribution you need, and download corresponding zip-archive. For example, the version FL 4.0.5 released for JDK 1.2 is archived in the fl4_0_5-jdk1_2.zip file.

To install the Foundation Layer, you have to:

1. Install Java Development Kit. This step may be omitted if JDK is already installed anywhere in the local network;
2. Set the environment variable JAVA_HOME to the JDK home directory;
3. Create a home directory for the Foundation Layer;
4. Unpack the FL distribution;
5. Set the environment variable FL_HOME to the FL home directory.

The environment variables JAVA_HOME and FL_HOME are used for code generation via the flgen.cmd file stored in the FL home directory, and for conversion to new FL via the cvt.cmd file stored in the convert subdirectory of the FL home directory. So, if you don't use these commands you need not set these variables.

The FL home directory contains the following files and subdirectories:

flenv.cmd

sets the environment variables PATH and CLASSPATH to use FL libraries with Java. It uses JAVA_HOME and FL_HOME environment variables.

flgen.cmd

runs FL Generator to generate Java classes by an FL petal model.

convert

the subdirectory containing all the information needed for conversion FL 3 application sources to FL 4.

doc

the subdirectory containing an additional documentation for FL.

gen

the subdirectory containing the basic FL model, attributes, libraries and scripts needed for conversion an old FL petal model to FL 4 form and for generation Java classes from a new model.

libs

the subdirectory containing jar-files needed for compilation of FL based Java classes and for runtime.

3 Essential Changes in FL 4

3.1 Changes in FL Petal Model

The basic distinction of the new FL petal model is the use of stereotypes:

- All **persistent** classes should have now the <<entity>> stereotype;
- You must use the <<value>> stereotype for **transient** classes;
- All exceptional classes should be mark with the <<exception>> stereotype;
- Primitive types are marked by <<primitive>> stereotype;
- Standard java classes which are not the subject of generation have no stereotype.

The next significant change of the new model is its strict syntax. **All types used in operations's signature must be specified.** This requires a certain discipline in modeling and may lead to difficulties in the conversion of an old model to the new syntax. The benefit of the strict syntax is more exact description of the model which allows to generate more correct Java code.

3.2 Changes in FL

The FL 4 release contains incompatibilities with previous versions. The following changes are introduced:

- The FL now uses standard collection classes:
 - o com.sun.java.util.collections for JDK 1.1 version,
 - o java.util for JDK 1.2 version;
- The table model interface is now taken from javax.swing package. JDK 1.1 users should use a swing 1.1.* distribution;
- Transaction interfaces are moved to com.novosoft.fl.transactions;
- The FL generator written in Java is introduced;
- The petal model format is changed;
- Optimistic locking for transactions is introduced;
- Week associations are introduced;
- The logging service is added for FL testing and debugging.

4 Migration from FL 3

The migration from FL 3 to FL 4 consists of two steps: conversion of FL Rose petal model and conversion of Java sources of FL based application to be compatible with the new FL.

4.1 Conversion of Rose Model

To convert an old model to the new format, you should:

1. Launch Rose;
2. Load the old model;
3. Remove classes java.lang.String, java.lang.Object, and java.util.Date from the model;
4. Select Tools @ Options menu;
5. Select "Diagram tab";
6. Select "label" in the "Stereotype display" option;
7. Check "Show labels on relations and associations" on in the "Stereotype display";
8. Press "OK" button in the "Option" dialog;
9. Execute script flconvert.ebs from the directory gen\script relative to the FL home directory;
10. If no errors occur, save the converted model.

All classes not defined in the model are stored into the package Unresolved Classes. To fix them, you should manually move them into the packages which they belong to.

4.2 Conversion of Sources

The distribution contains the Conversion Tool which automatically converts to FL 4 application sources written for FL 3. The distinction between FL 3 and FL 4 consists in some difference of package names used. The table below shows name replacements produced by the Conversion Tool.

FL 3	FL 4
com.novosoft.java.util	com.sun.java.util.collections (for JDK 1.1)
com.novosoft.java.util	java.util (for JDK 1.2)
com.novosoft.transaction	com.novosoft.fl.transaction
ru.novosoft.jdbc.fltn	com.novosoft.fl.jdbc.fltn

To produce the conversion you have to do the following:

1. Open the base project source directory. For example, let it look like this:
2. com
3. ru
4. business
5. Create "convert" subdirectory in it, and
6. Go to the "convert" subdirectory;
7. Start "cvt.cmd <root-package>". For example, to convert the "business" package you should type "cvt.cmd business". To convert all the packages at a time, type "cvt.cmd com ru business".

Converted files will be stored in the "convert" directory.

Note that to start cvt.cmd you have to set the JAVA_HOME and FL_HOME environment variable to the home directories of JDK and FL respectively.

If you use IDE (VisualAge, JBuilder, etc) you should export files, then convert them, and then import them back to the project.

5 Revision History

March 29, 2000

Initial version.