A plug-in for creating and deploying SmartFrog applications

Introduction

SmartFrog™ (Smart Framework for Object Groups) is a framework for configuring and automatically activating distributed applications. The framework is both a descriptive language for describing component collections, components being the Java class files making up your applications, and a runtime environment, which activates and manages these components. A SmartFrog deployment descriptor can be used to deploy and configure Java and native applications on systems ranging from your local system to a remote cluster.

The SmartFrog Eclipse plug-in offers you a common integrated development environment (IDE) in which to develop, test, deploy, and debug SmartFrog applications. It provides specialized editors enabling you to integrate the deployment of your SmartFrog applications into your Eclipse-based development processes.

More about the SmartFrog framework

SmartFrog is a general-purpose framework for deploying and managing software components based on a specification of their configuration. SmartFrog can be applied to multiple problem domains, though recently the focus has been to use SmartFrog to instantiate complex services and applications on utility infrastructures.

The approach adopted by SmartFrog is to think of services as composed of distributed components that must collaborate to deliver the complete service. The configuration details of each component are kept as separate as possible from the functionality of the component. This means that components can be as general-purpose as possible and can adopt different configuration-driven behavior when deployed.

To express the configuration details of individual components and groups of components, mechanisms are provided to facilitate highly flexible manipulation and composition of configuration data. It is this configuration data that specify the behavior required from the group of components that comprises the service. The configuration data are used at run-time to orchestrate the activities of component groups to deliver correctly configured, running services. This is referred to as creating *configuration-driven systems*.

The SmartFrog framework consists of three major elements:

1. configuration language

2. runtime deployment engine

3. set of components to be deployed

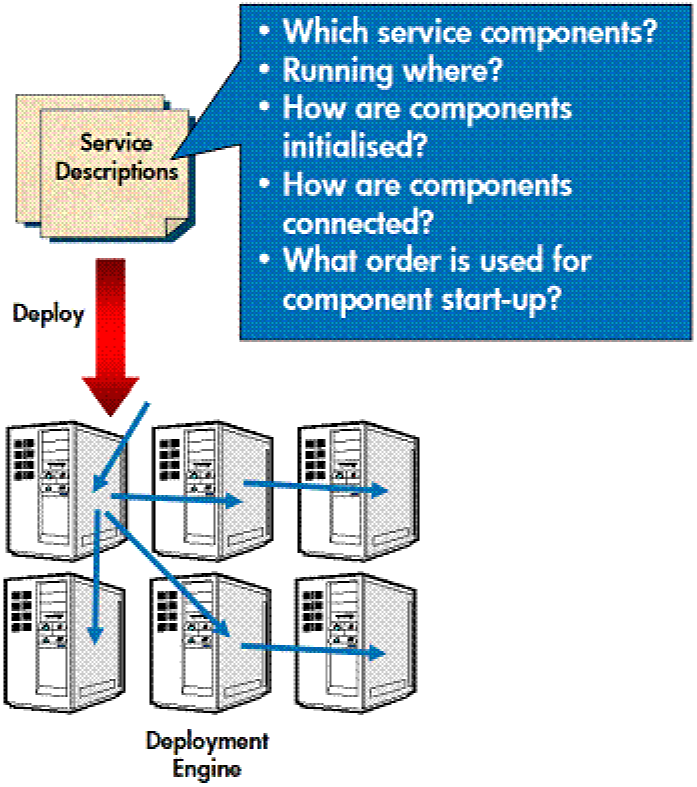

Service Descriptions
• Which service components?
• Running where?
• How are components initialised?
• How are components connected?
• What order is used for component start-up?
Deploy
Deployment Engine

Figure 1: SmartFrog Framework

More about the SmartFrog plug-in

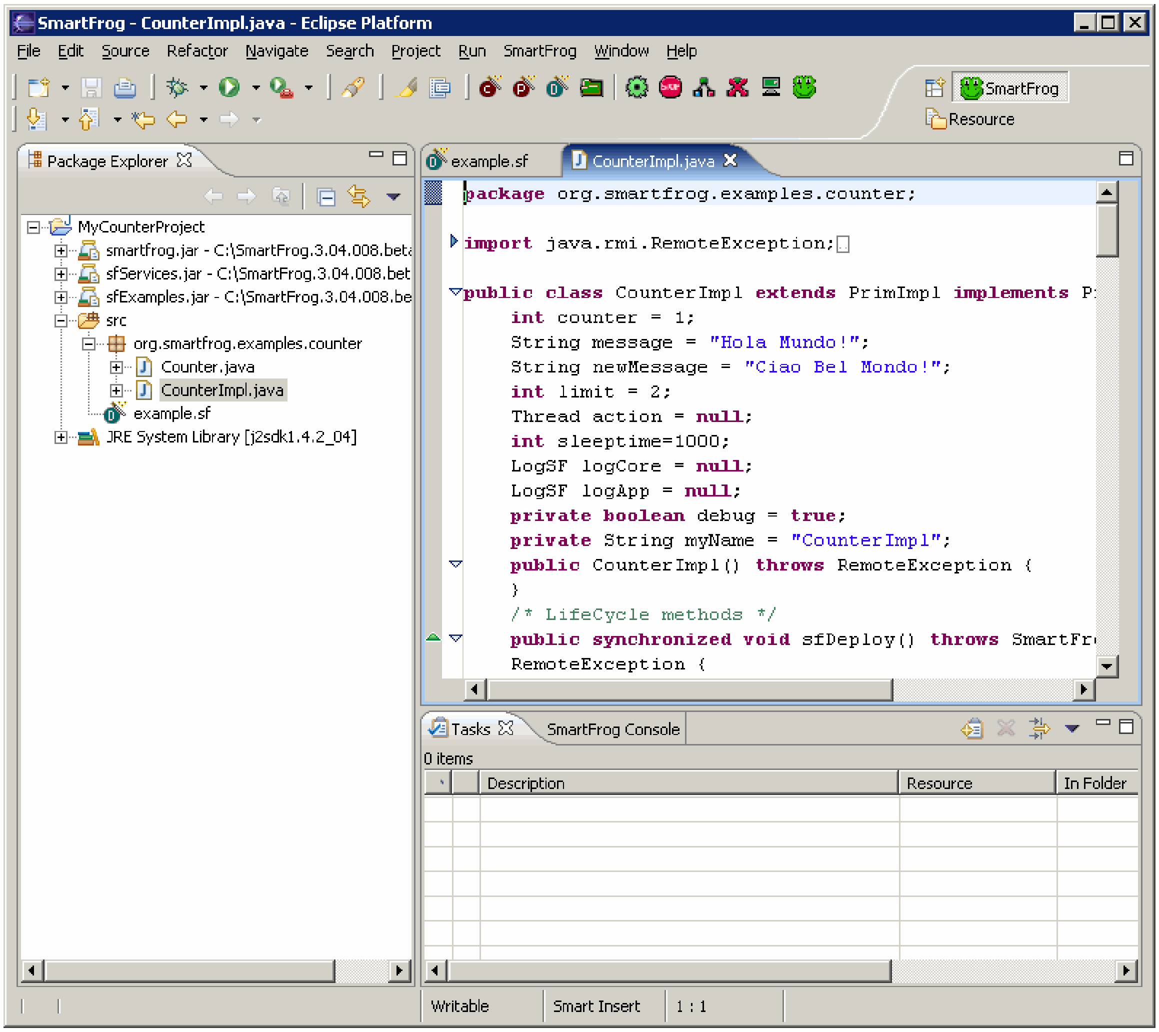
The SmartFrog Eclipse plug-in automatically provides you with the many benefits inherent to the Eclipse platform. These include project management, source version control, sophisticated Java tooling, an intuitive GUI providing perspectives and views customized for the task at hand, and integrated online help with search capabilities.

In addition, we have provided specialized wizards and editors that allow you to create SmartFrog projects for specific applications and then populate them with the necessary description and component files. The project concept provides an excellent means of organizing your SmartFrog application files.

The specialized SmartFrog description editor provides such features as syntax color highlighting, creation of expandable areas in your source code, and content assistance for SmartFrog notation keywords.

The component wizard allows you to create both types of SmartFrog Java component files, namely Prim and Compound. You can select the SmartFrog lifecycle method stubs you want automatically generated for your component file at the time of creation. In addition, you can automatically add more SmartFrog method stubs as necessary after the file is generated.

Once you have implemented your application, you can compile, test, and deploy it within this common IDE. The RMI compiler has been integrated into the plug-in so that all the necessary stub and skeleton classes needed for remote deployment are generated when you build your SmartFrog project.
Many of the SmartFrog GUI icons have been integrated into the SmartFrog Eclipse plug-in toolbar. You can start and stop the SmartFrog daemon, start and stop your SmartFrog application, launch the SmartFrog Management Console, and launch the SmartFrog GUI all from the SmartFrog Eclipse plug-in toolbar. When your application is running, you can see messages logged to the SmartFrog Console view in the SmartFrog plug-in perspective.



```
package org.smartfrog.examples.counter;

import java.rmi.RemoteException;

public class CounterImpl extends PrimImpl implements P
    int counter = 1;
    String message = "Hola Mundo!";
    String newMessage = "Ciao Bel Mondo!";
    int limit = 2;
    Thread action = null;
    int sleeptime=1000;
    LogSF logCore = null;
    LogSF logApp = null;
    private boolean debug = true;
    private String myName = "CounterImpl";
    public CounterImpl() throws RemoteException {
    }
    /* LifeCycle methods */
    public synchronized void sfDeploy() throws SmartFr
    RemoteException {
```

Figure 2: SmartFrog Perspective

The SmartFrog Eclipse plug-in also has integrated online help providing conceptual information about the SmartFrog framework and procedural information for using the plug-in.

Next steps

The SmartFrog Eclipse plug-in simplifies the process of implementing and deploying SmartFrog applications. It allows you to create SmartFrog application projects for your description and Java component files in a common IDE offering a customized perspective, specialized wizards and editors, and sophisticated Java tooling.

The SmartFrog Eclipse plug-in is now available. You can download the SmartFrog core framework and the Eclipse plug-in from http://sourceforge.net/projects/smartfrog. To learn more about SmartFrog and the Eclipse IDE, you can explore the links listed below.

For more information

SmartFrog Website:
http://www.smartfrog.org/

SmartFrog Reference Manual:
http://www.smartfrog.org/manuals/current/sfReference.pdf

SmartFrog User Manual:
http://www.smartfrog.org/manuals/current/sfUserManual.pdf

SmartFrog GUI Quick Reference Guide:
http://www.smartfrog.org/manuals/current/SFGuiManual.pdf

SmartFrog Work Flow Manual:
http://www.smartfrog.org/manuals/current/sfWorkflow.pdf

SmartFrog Dynamic Web Server Example:
http://www.smartfrog.org/manuals/current/sfDynamicWebServerExample.pdf

Eclipse Website:
http://eclipse.org

About the authors

This article was written by Diana Friedland, Julio Guijarro, and Elwin Ho.

Diana is a Learning Products Engineer, supporting technologies and tools for the Adaptive Enterprise. She develops and delivers online help for many of these tools and assists R & D with usability and human factors engineering.

Julio is a researcher in the Application Systems Department at HP Labs Bristol. His research interests include configuration, management and security of distributed systems.

Elwin is a Software engineer currently focusing on development environments for implementing voice-enabled Web applications and on manageability tools. He has worked in the areas of Application server, Web server, and transaction server development. He is a committer for various open source projects.