Integrating Software Development Kits into the Eclipse Platform

Steve Forsyth
Senior Software Engineer
Hewlett-Packard Company
Software Global Business Unit
Agenda

- What are we talking about?
- Features of SDK Integrator Plug-In
- Technical Underpinnings of SDK Integrator Plug-In
- Demo
- Possible Extensions
Use Case Model

- Develop XYZ SDK
- Develop Eclipse Plug-in(s) for XYZ SDK
- Learn XYZ SDK
- Develop MyApp utilizing XYZ SDK

XYZ SDK Provider

XYZ SDK User
Deployment Model

- SDK Integrator Plug-in
- XYZ SDK Provider's Eclipse
- XYZ SDK Plug-in(s)
- XYZ SDK User's Eclipse
Goals and Objectives

- Codify our experiences integrating Java SDKs into the Eclipse Platform.
  - HP OpenCall Media Platform SDK.
  - HP Internet Usage Manager SDK.
- Enhance SDK User’s ability to explore an SDK.
- Make it trivial for SDK Provider to create a basic, but useful, Eclipse plug-in for their SDK.
- Use existing Eclipse paradigms where appropriate.
SDK Integrator Features

- Generation of a basic PDE Plug-in Project, plus:
  - Generation of Java project wizards, including wizards to expose the SDK’s samples.
  - Other details like integration of the SDK’s license and a generated Ant build script.

- A separate wizard that automates the creation of an on-line help plug-in for the SDK.
Technical Details

- Extended org.eclipse.pde.ui.projectGenerators, providing a new Plug-In Project code generation wizard

- Extended org.eclipse.pde.ui.newExtension which enables SDK Provider to add samples to the generated plugin using the plugin.xml editor.

- Invented a new extension point that is declared and extended by the SDK Provider’s plug-in. Extensions describe the code samples to be included in a particular Java project wizard.
Technical Details (cont.)

- Populate the SDK Provider’s plug-in with the source code that processes the new extension point.

- Populate the SDK Provider’s online help plug-in with standard Eclipse help infrastructure and content.
Demo

- Integrate Log4J into Eclipse 2.1, that is, play the role of a developer on the Log4J project tasked with creating an integration to Eclipse.
NewJavaProjectWizardBase

- Calls subclass's `processPages()` during Wizard `performFinish`

ExtensibleJavaProjectWizard

- Manages the Library elements.

ExtensibleSampleProjectWizard

- Manages extensions of ExtensibleJavaProjectWizard
- Manages the Sample and CodeSkeleton elements.
Going beyond the basics

- External Tools for building samples (e.g. MX4J needs an RMI compilation step for some examples)

- Support for SDK’s with a deployment context other than just a JVM
  - Deployment capabilities.
  - Specialized launch configurations

- SDK Starter Code

- Support for other languages
More info on HP Eclipse Work

- Visit http://devresources.hp.com