Eclipse Integration
Lessons from the Trenches

Tan T. Phan

EclipseCon 2004
February 4, 2004
Discussion Topics

• Background Information
• SlickEdit Studio Integration Overview
• Fitting Duplicated Functionalities
• Rebinding Keys
• Logging Errors
• Integrating Help
• Running on the UI Thread
• Maintaining Proper Z-Order
• Product Branding
• Summary
Background Information

• Advanced code editing solutions since 1988
• Visual SlickEdit
  – Comprehensive and flexible code editor
• SlickEdit Plug-In for WebSphere Studio & Eclipse
  – Advanced code editor plug-in
  – Validated as Ready For IBM WebSphere Studio Software
• SlickEdit Studio
  – Fully supported multilanguage IDE built on Eclipse and SlickEdit technology
SlickEdit Studio Integration Overview

- Merging of SlickEdit technologies with the Eclipse Framework
  - Design goal: to provide a “cohesive user experience”
- Based on Visual SlickEdit v8.0.1, Eclipse 2.1.0, CDT 1.1.0
- Uses JDK 1.4.1 and Cygwin 1.5
- Implemented as three plug-ins: branding, documentation, editor and tools
- Written in Java/SWT, C for JNI, C/C++ and Slick-C for the editor core

![Diagram of SlickEdit Studio integration with Eclipse base]
Fitting Duplicated Functionalities (1 of 3)

Problem

- The merging of SlickEdit editing technologies and the Eclipse Framework exposed duplicated features. The overlapping needs to be eliminated or reduced to minimize user confusion.
  - Design goal: to provide a cohesive user experience
- Examples include search and replace, bookmarks, project support, preferences, key bindings

Solution

- Modified the Eclipse source
  - Removed, renamed, and/or enhanced actions
  - Enhanced menu listeners
- Added ActionSets in plugin.xml
  - Modified the Search menu and contributed to the button toolbar
Solution (continued)

- Implemented `EditorActionBarContributor` class (contributorClass attribute) of `org.eclipse.ui.editors` extension point in `plugin.xml`
  - Contributed actions to the Edit menu
- Acted on menu activation using `MenuListener`
  - Contributed actions to the Navigate, Project, and Help menus
  - Added and managed additional menus: Source, Display, Build, Macro, and Tools
- Redefined and reprogrammed SlickEdit operations
  - Masked out SlickEdit’s bookmarks
  - Retrofitted SlickEdit’s project support and build system
Fitting Duplicated Functionalities (3 of 3)

Remaining Issues

- Two ways to change user preferences
- Slight overlap for searching
- More complicated key rebinding
Rebinding Keys (1 of 2)

Problem
• The SlickEdit editor offers nine keyboard emulations (CUA, Emacs, vi, Brief, Visual C++, etc.). When the SlickEdit editor is active, Eclipse must pass through the needed key combinations.

Solution
• Not possible before Eclipse 2.1.0
• With Eclipse 2.1.x, the user can manually rebind the needed key combinations using Preferences > Workbench > Keys preference page
• Programmatically rebind the Eclipse key combinations to SlickEdit action on each emulation change using the internal interfaces for CoreRegistry, LocalRegistry, PreferenceRegistry, and SequenceBinding
Rebinding Keys (2 of 2)

Remaining Issues

- Accelerator text not being displayed in menus
- Automatic pass through of Eclipse key combinations for new SlickEdit Editor key bindings
- Avoid the use of internal interfaces
- Providing a common key bindings dialog
Logging Errors

Problem
• The SlickEdit editor technology does not have an error logging or tracing mechanism

Solution
• Leverage on the Eclipse built-in MultiStatus Error Dialog and Error Log
• Control the verbosity of output traces using the \texttt{-debug} command line argument and the \texttt{.options} file
Integrating Help (1 of 2)

Problem

• SlickEdit legacy help system (WinHelp on Windows and proprietary home-brew on Linux)
• Need an index (topical search)

Solution

• First Integration (SlickEdit Plug-In)
  – Converted documents to PDF and used Eclipse’s ability to display embedded PDFs in its Help viewer
• Second Integration (SlickEdit Studio)
  – Rewrote/Converted documents to HTML
  – Integrated into the Eclipse help system using `org.eclipse.help.toc` extension point in a separate documentation plug-in
Integrating Help (2 of 2)

Solution (continued)

• Modified org.eclipse.help.webapp plug-in
  – Added an Index view to the left pane of the Eclipse Help viewer
  – Filled the Index view with a pre-built index from the documentation plug-in

Remaining Issues

• Index is only for SlickEdit editor technologies
• Context-sensitive help disabled in SlickEdit’s native dialog boxes as well as for Slick-C macro function help
Running on the UI Thread

Problem
• SlickEdit editor and many of its tools, such as DIFFzilla and Beautifier, are UI-based and must run on the Eclipse UI (Display) thread. If an entry point is not made from the UI thread, unexpected crashes occur.

Solution
• Manually crafted all SlickEdit entry points to run on the UI thread
  – JNI entries
  – Progress monitors
Maintaining Proper Z-Order

Problem
- Message boxes and modal dialog boxes displayed by the native SlickEdit editor and tools must stay above those of Eclipse. A modal dialog box or message box can “fall behind” an Eclipse modal dialog box, become inaccessible, and block all keyboard and mouse inputs.

Solution
- Query the top-most Eclipse dialog box and specify its window handle (HWND on Windows and Window on Linux/X Window) as the owner (transientFor property on Linux/X Window) of the native SlickEdit dialog box or message box.
Product Branding

Problem
• Rebrand Eclipse platform without modifying its source

Solution
• Created a branding plug-in
  – Customized about.ini (for product images and welcome page), about.mappings (for build ID), about.properties (for product name and copyrights)
  – Customized plugin_customization.ini (for help system defaults)
• Changed feature.default.id in install.ini to bootstrap the branding plug-in
Summary

- The flexible Eclipse framework is key to the successful integration of SlickEdit Studio
- Eclipse’s open source model enables tighter technology integration which further enhances the cohesive user experience
- Advances in both SlickEdit technologies and Eclipse 3.0 will help resolve remaining issues.

For More Information

- Stop by the SlickEdit Booth
- Visit www.slickedit.com
- Call 1.919.934.EDIT or +1 919.473.0070
References

• eclipse.org
• Eclipse online help
• SlickEdit Studio v2 User Guide
• *Ready for WebSphere Studio Software Integration Requirements* and its companion *Pre-Test Instructions.* Version 2.
EclipseCon 2004.

slickedit

Develop an edge.