

Albireo: Swing/SWT Integration Building a Wish List

Bruno Haible, ILOG
Gordon Hirsch, SAS

Goals

- Embedding Swing GUIs in SWT

Non-Goals

- Embedding pure AWT GUIs in SWT
- Embedding native applications in SWT
 - ◆ too application dependent
- Embedding SWT and Swing
 - ◆ low user demand
 - ◆ single process approach with `SWT_AWT.new_Shell` maybe a dead-end

Our Role in the Eclipse Community

- Provide functionality based on SWT_AWT
- Speak up for SWT_AWT users
 - ◆ interface to the SWT team

Status Overview



State

- Usable
- Basic API stable
- **Alpha Testers: ready to launch!**
- Documentation to be written

Ease of Use

- Problem: 7 reported SWT_AWT “bugs” are actually methodology problems
- Albireo: do the right thing under the hood

```
SwingControl control =  
    new SwingControl(parent, SWT.NONE) {  
        protected JComponent createSwingComponent() {  
            return new JMagicPanel();  
        }  
        public Control getLayoutComponent() {  
            return parent;  
        }  
    };
```

Synchronized Look&Feel

- Problems:
 - ◆ SWT look&feel based on native platform widgets, Swing look&feel sometimes Metal
 - ◆ Different default fonts
- Albireo
 - ◆ activates native platform look&feel in Swing if possible
 - ◆ sets default fonts and colours in Swing
- Remaining:
 - ◆ JFace look&feel != SWT look&feel

Flicker

- Problem:
 - ◆ Flickering when control is resized
- Albireo
 - ◆ solves it for Windows
- Remaining:
 - ◆ find a solution also for gtk and MacOS

Dialogs

- Problems:
 - ◆ SWT GUI can still be acted upon while AWT modal dialogs are shown
 - ◆ AWT dialogs are sometimes shown behind the SWT window
- Albireo
 - ◆ provides workarounds
- Remaining:
 - ◆ Flickering when clicking on main window

Focus Handling

- Problems:
 - ◆ It takes 2 clicks to assign the focus
 - ◆ Focus lost after reactivating window
 - ◆ Focus lost when clicking on view's tab
 - ◆ Traverse out from AWT to SWT (or possibly even to another embedded AWT window) does not work
 - ◆ Selection not deactivated when another selection is made
- Albireo
 - ◆ provides workarounds for all these
- Remaining:
 - ◆ Focus wrong at initial display

Keystroke Contention

- Problems:
 - ◆ Some keystrokes are sent to both the AWT and SWT system. For example, for the “Exit” key, AWT may exit the application without running the SWT exit handlers.
 - ◆ Upon Shift-F10, AWT may react and SWT may “see” an F10 and react too, differently.
- Albireo
 - ◆ provides API for overriding this behaviour

Size/Layout Management

- Problems:
 - ◆ Preferred size on the Swing side is ignored on the SWT side
 - ◆ Size assigned on the SWT side does not resize the Swing component
- Albireo
 - ◆ connects the two size/layout protocols
 - ◆ sends events when embedded Swing component's size changed
- Remaining:
 - ◆ SWT_AWT does not support SWT.BORDER style correctly

SWT Popup Menus

- Problems:
 - ◆ Swing popup menus stay open when an SWT menu is opened
 - ◆ SWT popup menus may not appear on Gtk
- Albireo
 - ◆ fixes it
- Remaining:
 - ◆ SWT popup menus don't work on some JComponents

Cursors

- Remaining problems:
 - ◆ No API for converting between SWT and AWT cursors
 - ◆ BusyIndicator disappears on SWT_AWT controls

Tooltips

- Remaining problems:
 - ◆ AWT Tooltip not shown if window does not have focus
 - ◆ Tooltip may be shown on wrong screen (multi-screen)

Actions

- Remaining problems:
 - ◆ Swing actions cannot be specified in the plugin.xml file

Multithreading

- Problems:
 - ◆ Exceptions during `asyncExec` are not visible by default
- Albireo
 - ◆ shows exceptions on the console
- Remaining:
 - ◆ Thread-hopping is complicated to program. One needs either a single event thread for both SWT and AWT, or a task/queue/state based API.

Other Topics

- ?

More about Albireo

- Project home page:
<http://www.eclipse.org/albireo>
- Project wiki:
http://wiki.eclipse.org/Albireo_Project