rad3

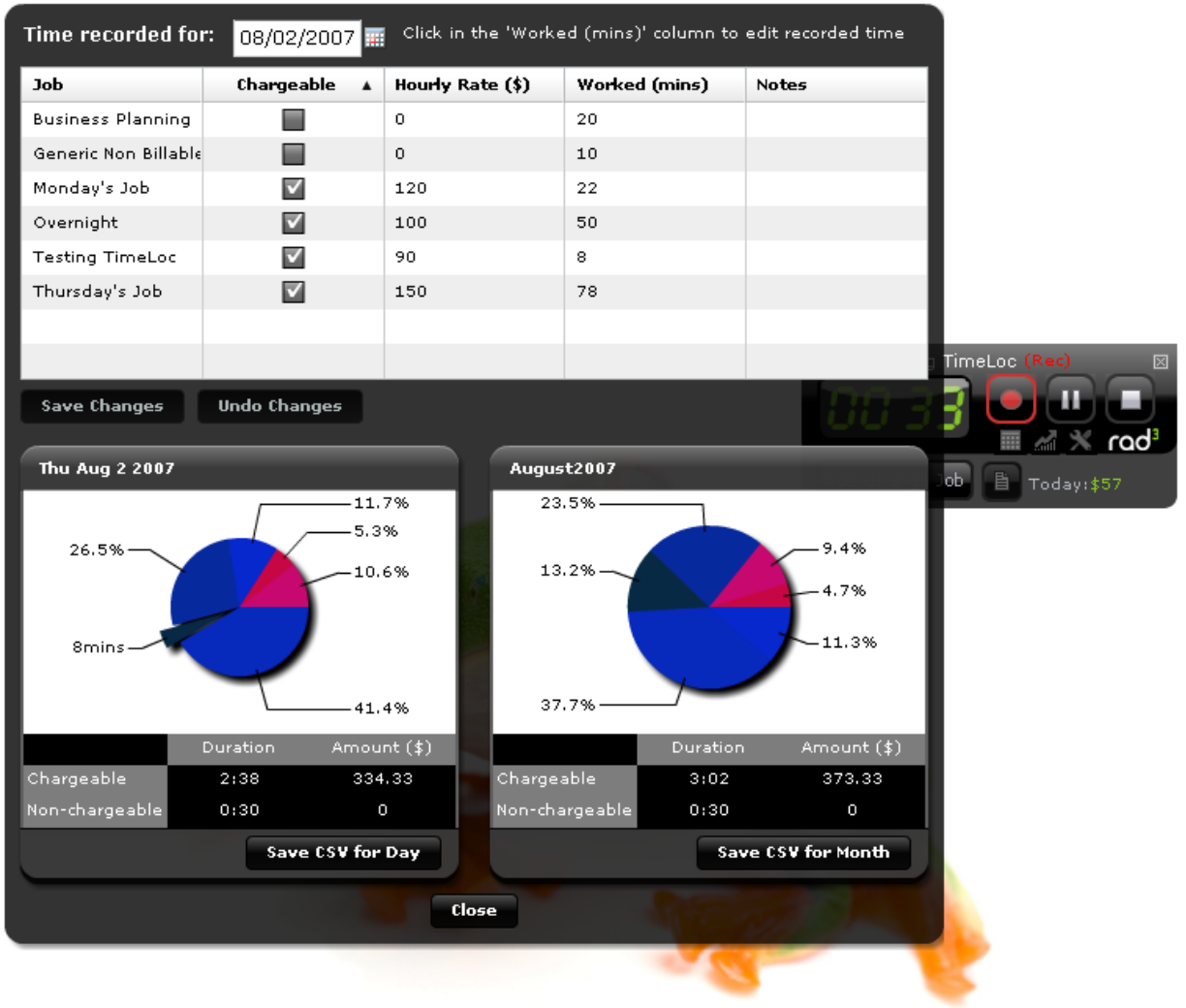
rich application development

TimeLoc – Custom Chrome

John Clarkson, Greg Hart

TimeLoc – a business utility from Rad3

- Desktop utility for recording time spent on jobs (projects)
  - Create chargeable and non-chargeable jobs with individual billing rates
  - Pause time recording for phone calls and breaks
  - Specify a minimum billing increment
  - View daily and monthly records and save to file
  - Add notes to jobs
- Created using Adobe Flex (Builder 3) as an AIR application
- TimeLoc uses a range of Flex and AIR features including custom chrome, charting, sound effects and file system access
- Try the beta version available at www.rad3.com



Application Properties and Style

TimeLoc.css

```
Application {
	background-color:"";
	background-image:"";
	padding: 0px;
	margin-top: 0;
	margin-right: 0;
	margin-bottom: 0;
	margin-left: 0;
}
```

TimeLoc-app.xml

```
<rootContent systemChrome="none"
transparent="true" visible="true">[SWF reference
is generated]</rootContent>
```

TimeLoc.mxml

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
width="240" height="105" xmlns:ns1="assets.views.*" creationComplete="init()"
backgroundImage="assets/images/tlocbkgrnd.png" themeColor="haloSilver">

...

</mx:Application>
```

Window Properties and Styles

Note: *Adding Flex components directly to the display list of an application-created NativeWindow is not supported in the Beta release of AIR.*

When additional windows are displayed they are also created with no system chrome and with window transparency set to true…

```
private function displayInANewWindow(pDisplayedObject:DisplayObject):void
{
    // Create a new window
    aNewWindow = new Window();

    // Set the window to be transparent and without OS or Flex header or frame
    aNewWindow.setStyle( "showFlexChrome", false );
    aNewWindow.transparent = true;
    aNewWindow.systemChrome = NativeWindowSystemChrome.NONE;
    aNewWindow.type = NativeWindowType.LIGHTWEIGHT;

    // Add events so that the window can be moved and closed
    aNewWindow.addEventListener(Event.ADDED_TO_STAGE, popupPosition);
    aNewWindow.addEventListener(Event.CLOSING, subWindowClosed);
    aNewWindow.addEventListener(MouseEvent.MOUSE_DOWN, popupWindowMove);
    aNewWindow.addEventListener(MouseEvent.MOUSE_WHEEL, popupOpacity);

    // Add the contents (pDisplayedObject) to the window and open it
    aNewWindow.addChild(pDisplayedObject);
    aNewWindow.open();
    aNewWindow.orderToFront();
}
```

Some other development notes…

The clock display – consists of a background image, an embedded font and an overlay image to provide the glassy highlight effect.

```
@font-face {
    src: url("../fonts/LCD2U___.TTF") ;
    fontFamily: DigitalBold;
    unicodeRange:
        U+0030-U+0039, /* Numbers [0..9] */
        U+002E-U+002E, /* Period [.] */
        U+0020-U+0020, /* Space [ ] */
        U+003A-U+003A; /* Colon [:] */
}
```

The flashing colon – as an experiment we applied a repeating effect to make the colon flash, but this used at least 4 times more average CPU compared to making the Label component visible/not visible.

rad3

rich application development

Head Office
19 Newton Road
PO Box 68261
Newton
Auckland
New Zealand

info@rad3.com
www.rad3.com