



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

The screenshot displays the TimeLoc application interface. At the top, it shows the date '08/02/2007' and a prompt to click in the 'Worked (mins)' column to edit recorded time. Below this is a table with the following data:

Job	Chargeable	Hourly Rate (\$)	Worked (mins)	Notes
Business Planning	<input type="checkbox"/>	0	20	
Generic Non Billable	<input type="checkbox"/>	0	10	
Monday's Job	<input checked="" type="checkbox"/>	120	22	
Overnight	<input checked="" type="checkbox"/>	100	50	
Testing TimeLoc	<input checked="" type="checkbox"/>	90	8	
Thursday's Job	<input checked="" type="checkbox"/>	150	78	

Below the table are buttons for 'Save Changes' and 'Undo Changes'. To the right, a digital clock shows '00:33' and a 'TimeLoc (Rec)' control panel with a red record button and a 'rad3' logo. Below the table, two pie charts are shown. The left chart is for 'Thu Aug 2 2007' and the right chart is for 'August 2007'. Each chart has a corresponding table below it:

Thu Aug 2 2007

	Duration	Amount (\$)
Chargeable	2:38	334.33
Non-chargeable	0:30	0

August 2007

	Duration	Amount (\$)
Chargeable	3:02	373.33
Non-chargeable	0:30	0

Buttons for 'Save CSV for Day', 'Save CSV for Month', and 'Close' are also visible.

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

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();
}
```



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

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.





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

info@rad3.com
www.rad3.com

