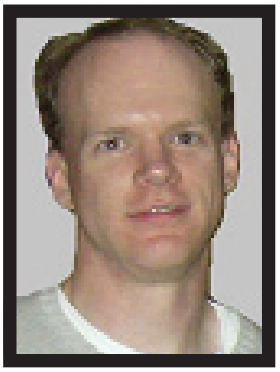
Integration with a Timeline Widget

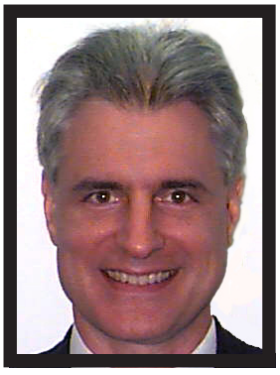
An Oracle "How To" Paper 2007

Presented By:

Jeremy Blankenship



William Indest



Time is what prevents everything from happening at once.

John Archibald Wheeler
Physicist

PURPOSE

The value propositions driving this work are:

1. Show how fast and easy it is to integrate with an AJAX widget.
2. Demonstrate our philosophical support for open source software.
3. Provide enhanced functionality to customers that are asking for the ability to review a case in a graphical way.
4. Give the Sales Consultant community demonstrable "eye-candy" for "customer interactions to remember".
5. Increase Oracle's coolness factor.
6. Provide a topic for engaging conversations during cocktail parties.

What the fuss is about...

WHAT IS IT?

It is easier to understand the Timeline widgetry if you interact with an existing widget: go to http://simile.mit.edu/timeline/ To interact with the widget, place your cursor in the timeline, hold down your mouse button and drag.

The Timeline widget is an Open Source AJAX-based widget created by MIT. The documentation for the widget can be found at http://simile.mit.edu/timeline/docs. The widget reads in an XML file to determine where the individual points should be placed on the timeline and what information should be shown for each point. Based upon the input data, the widget creates a scrolling timeline with time markers added appropriately.

For our purposes, we associate the Timeline Widget with the HLS Case Object and plot Activities for the Case in the Timeline view. The idea is to provide a more comprehensive and strategic view of how a case is worked from an Activity perspective. The Timeline can be implemented with other objects but will require eScript recoding and changes in the target Siebel objects.

Time is an illusion.
Lunchtime doubly so.

Douglas Adams
Writer

In Siebel Tools

HOW TO BUILD IT: CONFIGURATION

1. Import the objects from the Timeline.sif file. The objects will be imported into the Timeline Project.
2. Check the location of the Siebel Installation where the Timeline will be used. If the installation directory is something other than D:\80DQSSIA, you need to change the Timeline VBC BS Business Service code. In the Timeline VBC BS Business Service, go to the Declaration section and change the value of the sFileLocation variable to match the location of the Siebel installation.
3. Lock the PUB Case project to add the Timeline Business Component to the Business Object. Add the Timeline BC Business component to the HLS Case Business Object using the HLS Case/Timeline BC link to join the BC.
4. Add the Timeline View to the HLS Cases Screen. Compile the Timeline and PUB Case projects.

In the App

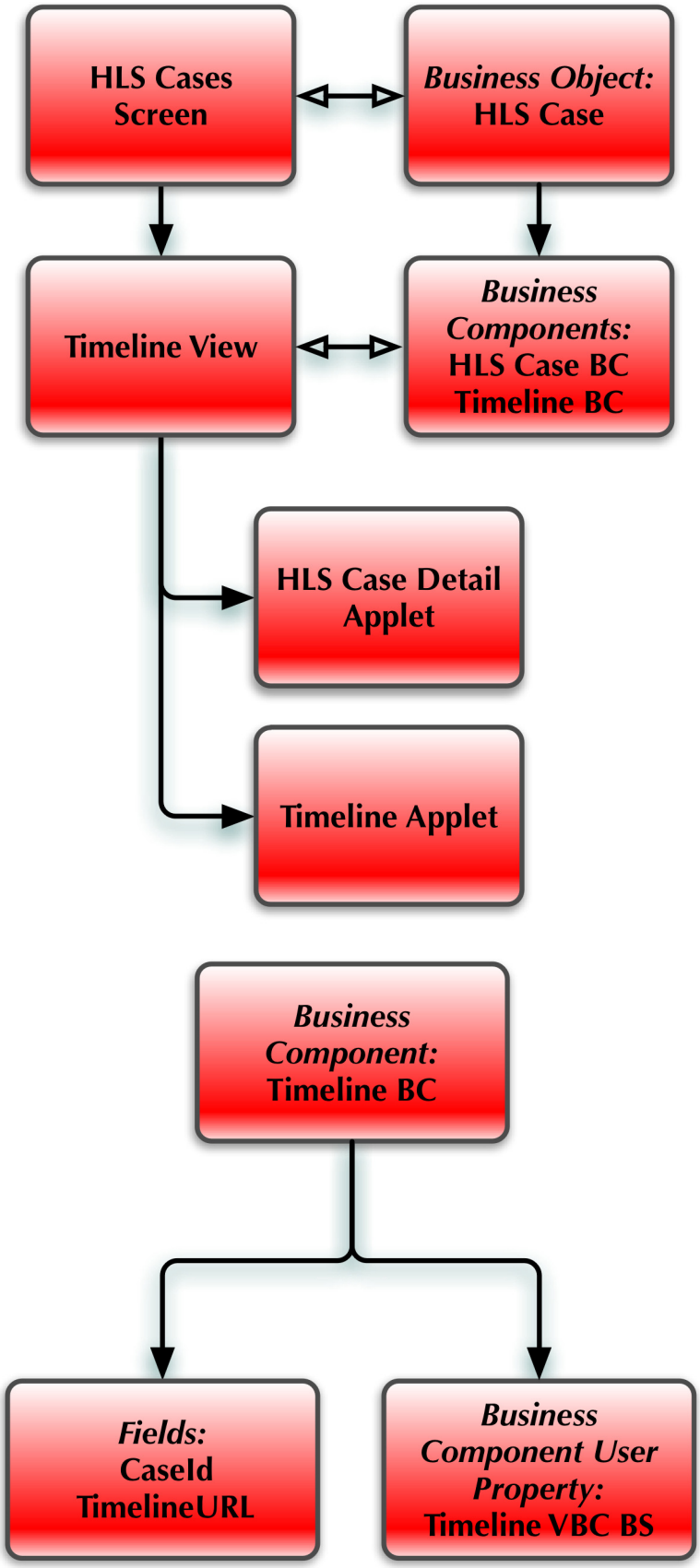
HOW TO BUILD IT: ADMINISTRATION

1. Login to the Public Sector application as SADMIN.
2. Add the Timeline View to the View Administration screen
3. Add the appropriate Responsibilities to the View.
4. Logout and login again.
5. Go to the Timeline view; if nothing shows up make sure there are activities associated with the Case.

DESIGN

The majority of the logic for displaying the Timeline widget is contained in a Virtual Business Component called Timeline BC. A VBC makes it easy to associate the Case Object and allows for file manipulation prior to rendering any data. The Timeline VBC BS Business Service provides the Timeline VBC its source as well as performing several other functions in order to make the Timeline work. When the Business Service is called, an HTML and an XML file are created in the web server directory. Unique HTML and XML files are created each time to avoid any issues with caching. The HTML file is then rendered in an IFRAME component on the Timeline Applet in the Timeline View. Below are the objects that are relevant to this configuration.

Objects Laid Out



I've been on a calendar, but never on time.

Marilyn Monroe
Actor

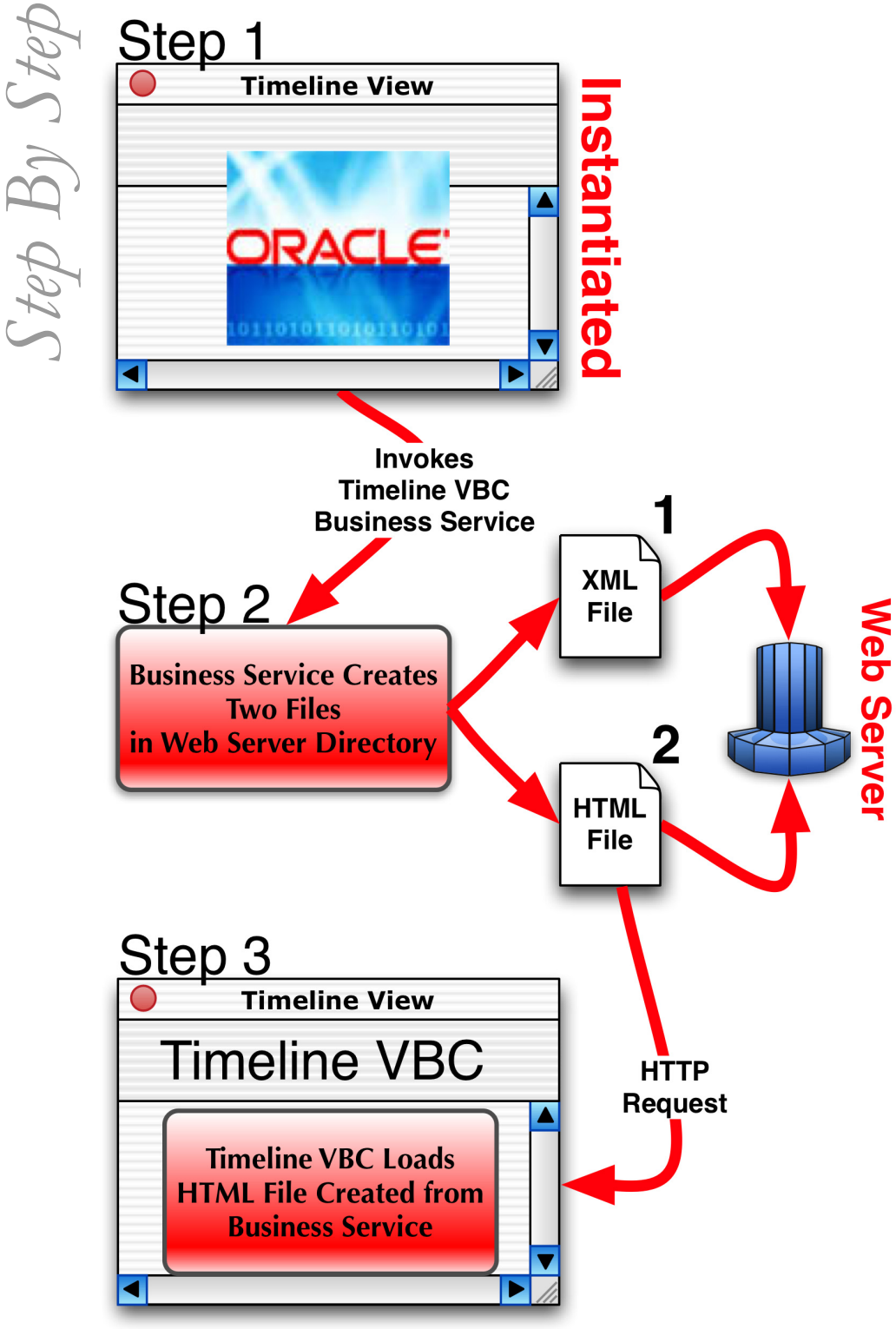
HOW IT WORKS

Step 1. When the view is instantiated, the Business Service on the Timeline BC is invoked.

Step 2. The Business Service creates an XML file and an HTML file in the Web Server directory.

Step 3. The VBC requests the HTML file from the web server.

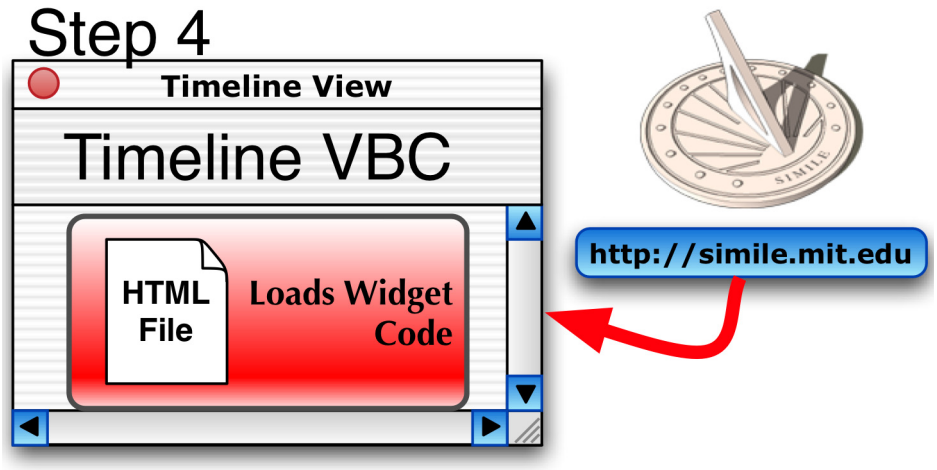
Step By Step

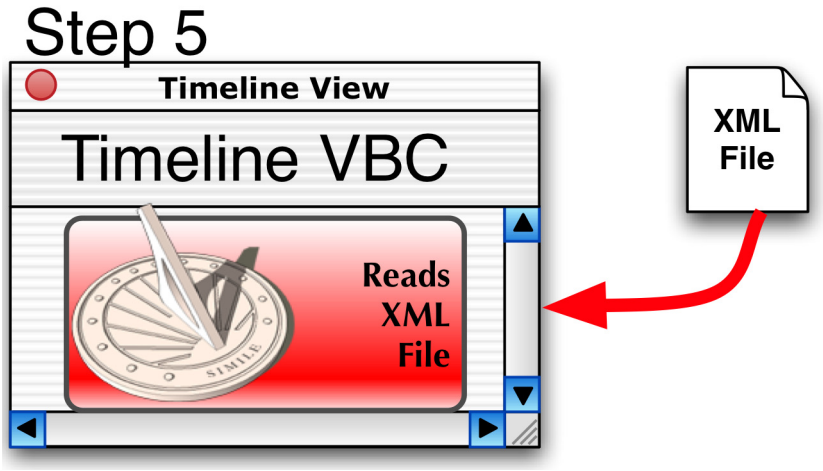


ORACLE

Step 4. The HTML file invokes the Timeline widget from http://simile.mit.edu/timeline/.

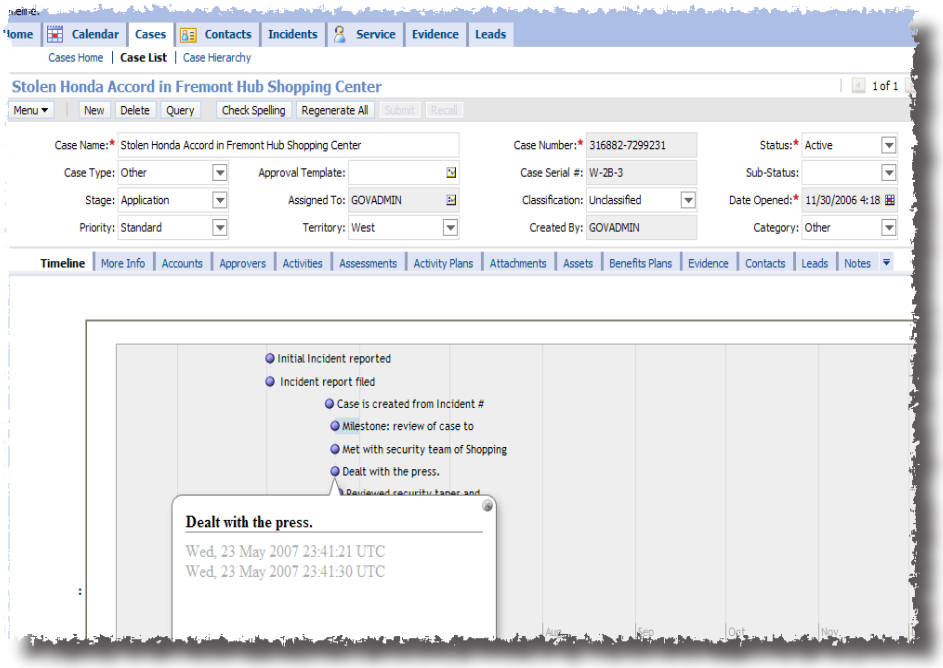
Step 5. The widget reads in the information from the XML file created by the Business service.





Step 6. The information is displayed in the Timeline View.

Voila!



CONFIGURATION NOTES

Here are some ideas about potential tweaks:

1. The Timeline was originally developed against a widescreen monitor. If the widget does not fit appropriately in a standard resolution monitor, the widget can be resized by setting IFRAME sizing parameters. To change the sizing, go to the QueryFunction function located in the Timeline VBC BS and find the line that starts with "PSOutputRow. SetProperty…". The end of this line will contain Height and Width parameters. Increase or decrease the values of these parameters to adjust the way the widget displays in the applet.
2. In order for the HTML pages to display properly, the HTML files must be placed in the Web Server directory. The code contained in the SIF will place the files in the local Web Server directory. If this functionality is used in a server environment, the location of the files will need to change to the Web Server directory on the server. The value of the sFileLocation variable in the Declation section of the Timeline VBC BS controls the directory where the HTML files will be created.
3. The structure of the HTML files is produced in the CreateHTML function of the Timeline VBC BS. There are specific settings for colors and so forth that can be set by changing parameters for the HTML file.

THE SCRIPT ON THE BUSINESS COMPONENT

Finally we leave you with the script snippets that do all of the heavy lifting. This code is included when you import the sif; we show it here in order for you to understand what is going on.

Business Service code

//Declaration Code

```
//The variable, sFileLocation, specifies the location of the Public ENU
//folder.  For an Object Manager demo this will need to be under the
//SweApp directory
var sFileLocation = "D:\\80DQSSIA\\client\\PUBLIC\\enu\\";

var EarliestDate;
```

//PreInvokeMethod for the Business Service (standard fare)

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
   if (MethodName == "Init")
   {
     InitFunction(Inputs,Outputs);
     return(CancelOperation);
    }
   if (MethodName == "Query")
   {
       QueryFunction(Inputs,Outputs);
       return(CancelOperation);
    }
   return (ContinueOperation);
}
```

//Init Function

```
function InitFunction(Inputs,Outputs)
{
   Outputs.SetProperty("CaseId","");
   Outputs.SetProperty("TimelineURL","");
}
```

//Query Function

```
function QueryFunction(Inputs,Outputs)
{
   var PSOutputRow = TheApplication().NewPropertySet();
```

```
    var psInputs = TheApplication().NewPropertySet();

    psInputs = Inputs.GetChild(0);
    var sFieldName = psInputs.GetFirstProperty();
    var sCaseId = psInputs.GetProperty(psInputs.GetFirstProperty());
    var curDate = new Date();
    var iDateSeconds = curDate.getTime();

    EarliestDate = "";

    CreateXML(sCaseId, iDateSeconds);

    CreateHTML(sCaseId, iDateSeconds);

    PSOutputRow.SetProperty("CaseId",sCaseId);

    PSOutputRow.SetProperty("TimelineURL","<IFRAME SRC="'
+ sCaseId + iDateSeconds + ".htm' height='500' width='1200'></
IFRAME>");

    Outputs.AddChild(PSOutputRow);
}
```

//CreateXML Function

```
function CreateXML(CaseId, DateSeconds)
{
    var BOAction = TheApplication().GetBusObject("Action");
    var BCAction = BOAction.GetBusComp("Action");
    var dtEarliestDate;

    var file = Clib.fopen(sFileLocation + CaseId + DateSeconds + ".xml",
"w");
    Clib.fprintf(file,"<data>\n");

    with (BCAction)
    {
        ClearToQuery();
        SetViewMode(AllView);
        ActivateField("Case Id");
        ActivateField("Planned");
        ActivateField("Planned Completion");
        ActivateField("Description");

        SetSearchSpec("Case Id", CaseId);
        ExecuteQuery();
```

```
        var isRecord = FirstRecord();

        if (isRecord)
        {
           while (isRecord)
           {
              var sId = GetFieldValue("Id");
              var sPlanned = GetFormattedFieldValue("Planned");
              var dtPlanned = new Date(sPlanned);
              if (GetFieldValue("Planned Completion") != "")
                 var sPlannedCompletion = GetFormattedFieldValue("Planne
d Completion");
              else
                 var sPlannedCompletion = sPlanned;

              var dtPlannedCompletion = new Date(sPlannedCompletion);
              if (EarliestDate != "")
              {
                 dtEarliestDate = new Date(EarliestDate);
              }
              else
              {
                 EarliestDate = sPlanned;
                 dtEarliestDate = new Date(EarliestDate);
              }

              if (dtPlanned.getTime() < dtEarliestDate.getTime())
              {
                 EarliestDate = sPlanned;
              }

              var XMLEntry = "<event start=\"" +
GetFieldValue("Planned") + "\" end=\"" + sPlannedCompletion + "\"
title=\"" + GetFieldValue("Description") + "\"/>\n";

              Clib.fprintf(file,"%s",XMLEntry);
              isRecord = NextRecord();
           }

        }
     }
     Clib.fprintf(file,"</data>\n");
     Clib.fclose(file);
     BCAction = null;
     BOAction = null;
}
```

//CreateHTML Function

```
function CreateHTML(CaseId, DateSeconds)
{
   var file = Clib.fopen(sFileLocation + CaseId +
DateSeconds + ".htm", "w");

   Clib.fprintf(file,"<html>\n");
   Clib.fprintf(file,"  <head>\n");

   Clib.fprintf(file,"    <style>\n");
   Clib.fprintf(file,"       .timeline-default {\n");
   Clib.fprintf(file,"              font-family: Trebuchet MS,
Helvetica, Arial, sans serif;\n");
   Clib.fprintf(file,"              font-size: 8pt;\n");
   Clib.fprintf(file,"              border: 1px solid #aaa;\n");
   Clib.fprintf(file,"              margin: 2em;\n");
   Clib.fprintf(file,"       }\n");
   Clib.fprintf(file,"   </style>\n");
   Clib.fprintf(file,"  <script src=\"http://simile.mit.edu/
timeline/api/timeline-api.js\" type=\"text/javascript\"></
script>\n");
   Clib.fprintf(file,"      <script type=\"text/javascript\">\
n");
   Clib.fprintf(file,"function onLoad() {\n");
   Clib.fprintf(file,"  var eventSource = new Timeline.
DefaultEventSource();\n");
   Clib.fprintf(file,"  var theme = Timeline.ClassicTheme.
create();\n");
   Clib.fprintf(file,"  theme.event.bubble.width = 320;\n");
   Clib.fprintf(file,"  theme.event.bubble.height = 220;\n");
   Clib.fprintf(file,"  theme.ether.backgroundColors[1] =
theme.ether.backgroundColors[0];\n");
   Clib.fprintf(file,"  var d = Timeline.DateTime.
parseGregorianDateTime(\"1953\")\n");
   Clib.fprintf(file,"  var bandInfos = [\n");
   Clib.fprintf(file," \n");
   Clib.fprintf(file,"    Timeline.createBandInfo({\n");
   Clib.fprintf(file,"       eventSource:    eventSource,\n");
   Clib.fprintf(file,"       date:           \"%s\",\
n",EarliestDate);
   var tempwidth = "       width:          \"70%\", \n";
   Clib.fprintf(file,"%s",tempwidth);
   Clib.fprintf(file,"       intervalUnit:   Timeline.DateTime.
```

```
MONTH, \n");
    Clib.fprintf(file,"        intervalPixels: 100,\n");
    Clib.fprintf(file,"        theme:          theme\n");
    Clib.fprintf(file,"    }),\n");
    Clib.fprintf(file,"    Timeline.createBandInfo({\n");
    Clib.fprintf(file,"        eventSource:    eventSource,\n");
    Clib.fprintf(file,"        date:           \"%s\",\n",EarliestDate);
    tempwidth = "       width:          \"30%\", \n";
    Clib.fprintf(file,"%s",tempwidth);
    Clib.fprintf(file,"        intervalUnit:   Timeline.DateTime.YEAR, \n");
    Clib.fprintf(file,"        intervalPixels: 200,\n");
    Clib.fprintf(file,"        showEventText:  false,\n");
    Clib.fprintf(file,"        theme:          theme\n");
    Clib.fprintf(file,"    })\n");
    Clib.fprintf(file,"  ];\n");

    Clib.fprintf(file,"  bandInfos[0].syncWith = 1;\n");
    Clib.fprintf(file,"  bandInfos[0].highlight = false;\n");

    Clib.fprintf(file,"  bandInfos[1].eventPainter.setLayout(bandInfos[0].
eventPainter.getLayout());\n");

    Clib.fprintf(file,"  var tl = Timeline.create(document.getElementById(\
"my-timeline\"), bandInfos);\n");
    Clib.fprintf(file,"  Timeline.loadXML(\"" + CaseId + DateSeconds +
".xml\", function(xml, url) { eventSource.loadXML(xml, url); });\n");
    Clib.fprintf(file,"}\n");

    Clib.fprintf(file,"var resizeTimerID = null;\n");
    Clib.fprintf(file,"function onResize() {\n");
    Clib.fprintf(file,"    if (resizeTimerID == null) {\n");
    Clib.fprintf(file,"        resizeTimerID = window.setTimeout(function()
{\n");
    Clib.fprintf(file,"            resizeTimerID = null;\n");
    Clib.fprintf(file,"            tl.layout();\n");
    Clib.fprintf(file,"        }, 500);\n");
    Clib.fprintf(file,"    }\n");
    Clib.fprintf(file,"}\n");

    Clib.fprintf(file,"</script>\n");
    Clib.fprintf(file,"</head>\n");


    Clib.fprintf(file,"<body onload=\"onLoad();\" onresize=\
"onResize();\">\n");
    Clib.fprintf(file,"<div id=\"my-timeline\" class=\"timeline-default\"
 style=\"height: 400px;border: 1px solid #aaa\">\n");
```

```
    Clib.fprintf(file,"    </div>\n");

    Clib.fprintf(file,"</body>\n");
    Clib.fprintf(file,"</html>\n");
    Clib.fclose(file);

}
```

parting is such sweet sorrow.

Juliet
Doomed Lover

End

Integration with a Timeline Widget
Authors: Jeremy Blankenship and William Indest
Managers: James Perry and Todd Kyle

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com