Gecko and Glade mini-browser

Mini browser in less than 100 lines

Mini browser in less than 100 lines

In this example we'll use Glade# to construct the GUI of the application, and then use Gecko# to add the actual browser component.

> **Notice**
>
> To make Gecko# application run and detect Mozilla's libraries, you have to make sure that MOZILLA_FIVE_HOME directory is in the LD_LIBARY_PATH

First step - Create GUI in Glade

The design of the GUI is entirely up to you, all you need is a few buttons for basic actions, like Back,Forward,Refresh and etc, and a placeholder for Mozilla's Webcontrol.
For each button prepare a Clicked signals, and write down their names. We'll write handlers for them later on.

You can find my Glade generated at the end of the article.

Second step - Write the code

Basic idea of constructing Glade application is as follows:

1. create a class
2. on it's constructor:
    a. call Application.Init()
    b. loads the GUI xml, and connects it's signals to handlers in the code
    c. add any additional controls if needed
    d. call Application.Run()

So, here's the code:

browser.boo

```
# Simple Mozilla base Browser - booRowser
# by Eli Yukelzon <reflog@gmail.com>
# Based on examples from MonoDoc
# Note: before running, make sure that
MOZILLA_FIVE_HOME directory
# is in the LD_LIBARY_PATH
```

```
import System
# when importing , specify a 'from' to help
boo find the references
import Gtk from "gtk-sharp"
import Glade from "glade-sharp"
import Gecko from "gecko-sharp"
import Gnome from "gnome-sharp"

public class GladeApp:
    [Widget] frame1 as Frame
    [Widget] entry1 as Gtk.Entry
    [Widget] window1 as Gtk.Window
    [Widget] progressbar1 as Gtk.ProgressBar
    [Widget] statusbar1 as Gtk.Statusbar

    web as WebControl

    public def constructor (args):
        Application.Init()
        gxml = Glade.XML ("gui.glade",
"window1", null)
        try:
            # load the xml file, and setup a
controls with event handlers
            gxml.Autoconnect (self)
            # create Mozilla control
            web = WebControl()
            web.Show()
            frame1.Add(web)
            # setup event handlers
            entry1.Activated += load_url
```

```
            web.TitleChange +=
web_title_change
            web.LinkMsg += on_linkmessage
            web.Progress += on_progress
            # setup default site
            entry1.Text = "www.google.com"
            entry1.Show()
            entry1.Activate()
            statusbar1.Push(1,"Welcome to
booBrowser!")
            # start Gtk message loop
            Application.Run()
        except e as
Glade.HandlerNotFoundException:
            print "Exception caught while
constructing from Glade XML:\n" + e
        except e as Exception:
            print "Exception caught:\n" + e

    public def on_progress ( source as
object , args as ProgressArgs):
        # this will work only if target page
size is known
        progressbar1.Adjustment.Upper =
args.Maxprogress
        progressbar1.Adjustment.Value =
args.Curprogress

    public def web_title_change( source as
object, args as EventArgs):
        window1.Title = "booRowser: [
```

```
"+web.Title+" ]"

    public def OnWindowDeleteEvent (source
as object, args as DeleteEventArgs):
        Application.Quit ()
        args.RetVal = true

    public def load_url(source as object ,
args as EventArgs):
        if (entry1.Text!=null):
            web.LoadUrl(entry1.Text)

    public def on_linkmessage (source as
object, args as EventArgs):
        statusbar1.Pop (1)
        statusbar1.Push (1, web.LinkMessage)

    public def on_StopButton_clicked(source
as object ,  args as EventArgs):
        web.StopLoad()

    public def on_GoButton_clicked(source as
object ,  args as EventArgs):
        load_url(source, args)

    public def on_FwdButton_clicked(source
as object ,  args as EventArgs):
        web.GoForward()

    public def on_BackButton_clicked(source
as object ,  args as EventArgs):
```

```
        web.GoBack()

    public def
on_ReloadButton_clicked(source as object ,
args as EventArgs ):
        web.Reload(3)
        statusbar1.Pop(1)
        statusbar1.Push(1,"Refreshing Page")
```

```
# construct a glade class
GladeApp ([])
```

And here's the Glade-generated GUI file:

gui.glade

```
<?xml version="1.0" standalone="no"?> <!--*-
mode: xml -*-->
<!DOCTYPE glade-interface SYSTEM
"http://glade.gnome.org/glade-2.0.dtd">

<glade-interface>
<requires lib="gnome"/>

<widget class="GtkWindow" id="window1">
  <property name="visible">True</property>
  <property name="title"
translatable="yes">BooRowser</property>
  <property
name="type">GTK_WINDOW_TOPLEVEL</property>
  <property
name="window_position">GTK_WIN_POS_CENTER</p
roperty>
  <property name="modal">False</property>
  <property
name="default_width">230</property>
  <property
name="default_height">350</property>
  <property name="resizable">True</property>
```

```
  <property
name="destroy_with_parent">False</property>
  <property name="decorated">True</property>
  <property
name="skip_taskbar_hint">False</property>
  <property
name="skip_pager_hint">False</property>
  <property
name="type_hint">GDK_WINDOW_TYPE_HINT_NORMAL
</property>
  <property
name="gravity">GDK_GRAVITY_NORTH_WEST</prope
rty>
  <signal name="delete_event"
handler="OnWindowDeleteEvent"/>

  <child>
    <widget class="GtkVBox" id="vbox1">
      <property
name="visible">True</property>
      <property
name="homogeneous">False</property>
      <property name="spacing">0</property>

      <child>
        <widget class="GtkHBox" id="hbox1">
          <property
name="visible">True</property>
          <property
name="homogeneous">False</property>
          <property
```

```
name="spacing">0</property>

          <child>
            <widget class="GtkButton"
id="BackButton">
              <property
name="visible">True</property>
              <property
name="can_focus">True</property>
              <property
name="label">gtk-go-back</property>
              <property
name="use_stock">True</property>
              <property
name="relief">GTK_RELIEF_NORMAL</property>
              <property
name="focus_on_click">True</property>
              <signal name="clicked"
handler="on_BackButton_clicked"/>
            </widget>
            <packing>
              <property
name="padding">0</property>
              <property
name="expand">False</property>
              <property
name="fill">False</property>
            </packing>
          </child>

          <child>
```

```
            <widget class="GtkButton"
id="FwdButton">
              <property
name="visible">True</property>
              <property
name="can_focus">True</property>
              <property
name="label">gtk-go-forward</property>
              <property
name="use_stock">True</property>
              <property
name="relief">GTK_RELIEF_NORMAL</property>
              <property
name="focus_on_click">True</property>
              <signal name="clicked"
handler="on_FwdButton_clicked" />
            </widget>
            <packing>
              <property
name="padding">0</property>
              <property
name="expand">False</property>
              <property
name="fill">False</property>
            </packing>
          </child>

          <child>
            <widget class="GtkButton"
id="StopButton">
              <property
```

```
name="visible">True</property>
              <property
name="can_focus">True</property>
              <property
name="label">gtk-stop</property>
              <property
name="use_stock">True</property>
              <property
name="relief">GTK_RELIEF_NORMAL</property>
              <property
name="focus_on_click">True</property>
              <signal name="clicked"
handler="on_StopButton_clicked" />
            </widget>
            <packing>
              <property
name="padding">0</property>
              <property
name="expand">False</property>
              <property
name="fill">False</property>
            </packing>
          </child>

          <child>
            <widget class="GtkButton"
id="ReloadButton">
              <property
name="visible">True</property>
              <property
name="can_focus">True</property>
```

```
              <property
name="label">gtk-refresh</property>
              <property
name="use_stock">True</property>
              <property
name="relief">GTK_RELIEF_NORMAL</property>
              <property
name="focus_on_click">True</property>
              <signal name="clicked"
handler="on_ReloadButton_clicked" />
            </widget>
            <packing>
              <property
name="padding">0</property>
              <property
name="expand">False</property>
              <property
name="fill">False</property>
            </packing>
          </child>

          <child>
            <widget class="GtkEntry"
id="entry1">
              <property
name="visible">True</property>
              <property
name="can_focus">True</property>
              <property
name="editable">True</property>
              <property
```

```
name="visibility">True</property>
              <property
name="max_length">0</property>
              <property name="text"
translatable="yes"></property>
              <property
name="has_frame">True</property>
              <property
name="invisible_char">*</property>
              <property
name="activates_default">False</property>
            </widget>
            <packing>
              <property
name="padding">0</property>
              <property
name="expand">True</property>
              <property
name="fill">True</property>
            </packing>
          </child>

          <child>
            <widget class="GtkButton"
id="GoButton">
              <property
name="visible">True</property>
              <property
name="can_focus">True</property>
              <property
name="label">gtk-ok</property>
```

```
              <property
name="use_stock">True</property>
              <property
name="relief">GTK_RELIEF_NORMAL</property>
              <property
name="focus_on_click">True</property>
              <signal name="clicked"
handler="on_GoButton_clicked" />
            </widget>
            <packing>
              <property
name="padding">0</property>
              <property
name="expand">False</property>
              <property
name="fill">False</property>
            </packing>
          </child>
        </widget>
        <packing>
          <property
name="padding">0</property>
          <property
name="expand">False</property>
          <property
name="fill">False</property>
        </packing>
      </child>

      <child>
        <widget class="GtkFrame"
```

```
id="frame1">
          <property
name="visible">True</property>
          <property
name="label_xalign">0</property>
          <property
name="label_yalign">0.5</property>
          <property
name="shadow_type">GTK_SHADOW_ETCHED_IN</pro
perty>

          <child>
            <placeholder/>
          </child>
        </widget>
        <packing>
          <property
name="padding">0</property>
          <property
name="expand">True</property>
          <property
name="fill">True</property>
        </packing>
      </child>

      <child>
        <widget class="GtkHBox" id="hbox2">
          <property
name="visible">True</property>
          <property
name="homogeneous">False</property>
```

```
          <property
name="spacing">0</property>

          <child>
            <widget class="GtkProgressBar"
id="progressbar1">
              <property
name="visible">True</property>
              <property
name="orientation">GTK_PROGRESS_LEFT_TO_RIGH
T</property>
              <property
name="fraction">0</property>
              <property
name="pulse_step">0.10000000149</property>
            </widget>
            <packing>
              <property
name="padding">0</property>
              <property
name="expand">False</property>
              <property
name="fill">False</property>
            </packing>
          </child>

          <child>
            <widget class="GtkStatusbar"
id="statusbar1">
              <property
name="visible">True</property>
```

```
                <property
name="has_resize_grip">False</property>
              </widget>
              <packing>
                <property
name="padding">0</property>
                <property
name="expand">True</property>
                <property
name="fill">True</property>
              </packing>
            </child>
          </widget>
          <packing>
            <property
name="padding">0</property>
            <property
name="expand">False</property>
            <property
name="fill">True</property>
          </packing>
        </child>
      </widget>
    </child>
```

```
</widget>

</glade-interface>
```

Step three: Running

Since, we've specified the list of assemblies that need to be referenced in the import section, all the is left now is to run the application:

```
mono booi.exe browser.boo
```

Or even compile it into executable, and then run it:

```
mono booc.exe browser.boo
mono browser.exe
```

That's it! Have fun boo-ing!

Eli Yukelzon