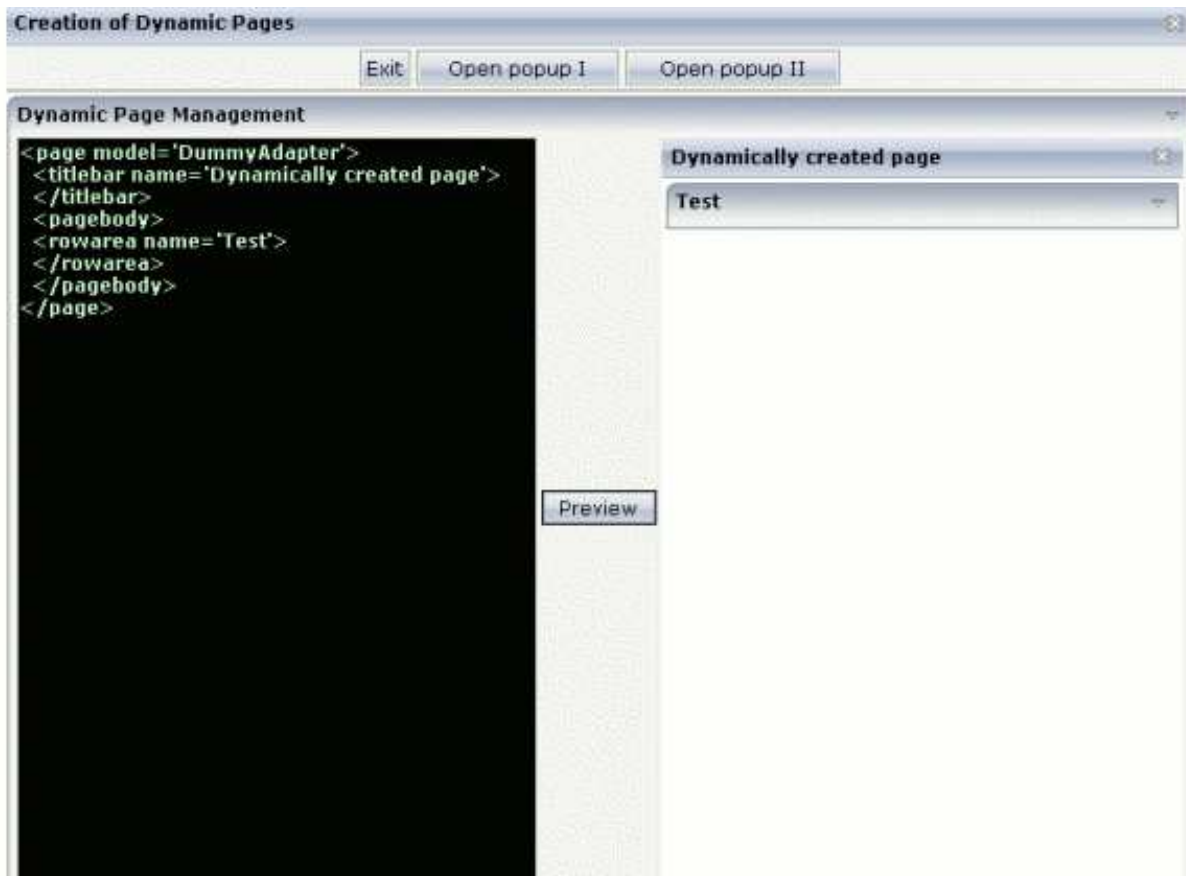


Programming Dynamic Pages

This section gives an example on how to program the dynamic page management.



On the left of the example page, you can enter any XML layout definition. When choosing the **Preview** button, the layout definition is translated into an HTML page. The page is shown in the right area.

The main controls that are used are:

- The TEXT control - for entering the text.
- The SUBCISPAGE control - for displaying the preview.

The XML layout definition looks as follows:

```
<rowarea name="Rowarea" height="100%">
  <itr takefullwidth="true" height="100%">
    <text valueprop="layoutXML" width="50%" height="100%"
      textareastyle="background-color:#000000;color:#D0FFD0;font-weight:bold">
    </text>
    <hdist>
    </hdist>
    <button name="Preview" method="onPreview">
    </button>
    <hdist>
    </hdist>
  </itr>
</rowarea>
```

```

        <SUBCISPAGE valueprop="subpageURL" width="50%" height="100%" borderwidth="0">
        </SUBCISPAGE>
    </itr>
</rowarea>

```

The adapter code looks as follows:

```

// This class is a generated one.

import com.softwareag.cis.server.Adapter;
import com.softwareag.cis.server.IDynamicPageMgmt;

public class dynamicPagesAdapter
    extends Adapter
{
    String m_layoutXML;

    public String getLayoutXML() { return m_layoutXML; }
    public void setLayoutXML(String value) { m_layoutXML = value; }

    String m_subpageURL = "/HTMLBasedGUI/empty.html";
    public String getSubpageURL() { return m_subpageURL; }
    int m_counter = 0;
    // -----
    // public adapter methods
    // -----

    public void onPreview()
    {
        try
        {
            m_counter++;
            //      pass page to dynamic page management
            IDynamicPageMgmt dpm = m_sessionContext.getDynamicPageMgmt();
            dpm.addDynamicPage(m_layoutXML, "DYNAMIC_EXAMPLE" + m_counter);
            //      switch to dynamic page
            m_subpageURL = dpm.createURLString("DYNAMIC_EXAMPLE" + m_counter);
        }
        catch (Throwable t)
        {
            outputMessage(MT_ERROR, t.toString());
        }
    }
}

```

When you choose the **Preview** button, the method `onPreview()` is called. This method takes the text that is entered inside the TEXT control and passes this text to the dynamic page management. The dynamic page management is internally organized on session level - you access it inside an adapter via the member `m_sessionContext`.

Directly inside the method `addDynamicPage(...)`, the XML that is passed gets translated. If the XML contains errors, the corresponding error messages are thrown.

With the message `createURLString(...)`, a URL is built that you can use for referencing the dynamic page. Maybe you want to have a look of what is created as the URL: it is the call of the servlet `StartCISPage` with a parameter indicating the current session and the name of the dynamic page. You should not build this URL on your own but always go through the API.

The URL that is created is used inside the SUBCISPAGE control. If it changes, the control displays the new URL in its content area. This is also the reason why an explicit counter is used: by using the counter, each previewed page gets a URL of its own - the SUBCISPAGE control does not reload its content area if the URL stays stable.