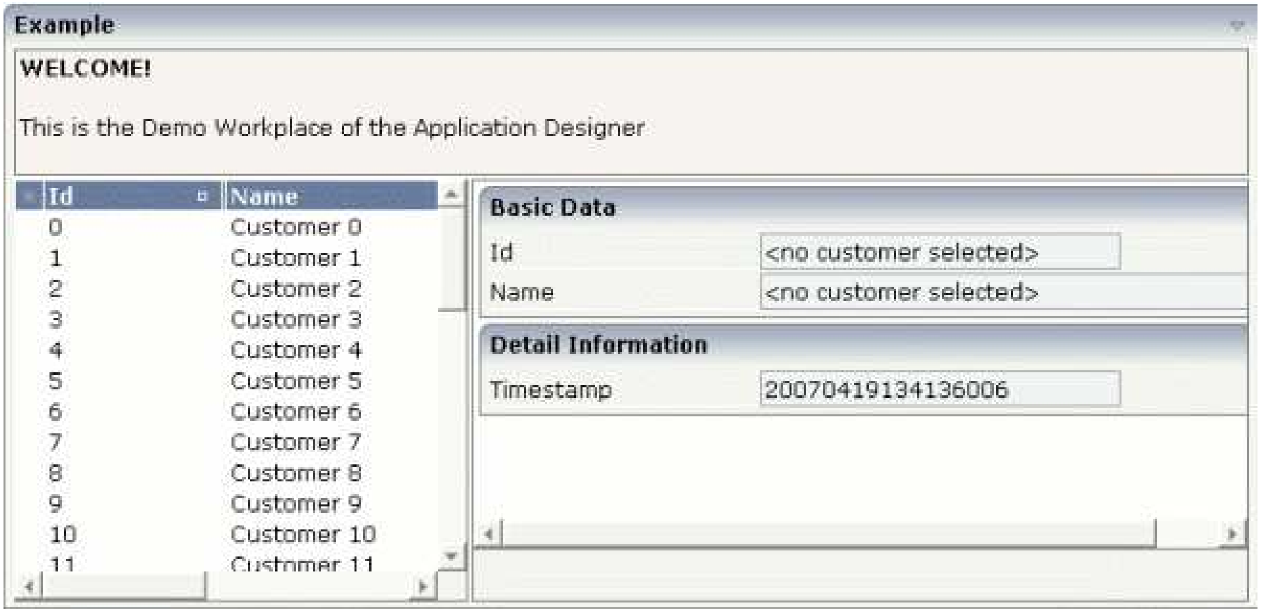
Example

The example that will be built in this section produces the following output:



Example

WELCOME!

This is the Demo Workplace of the Application Designer

Id	Name
0	Customer 0
1	Customer 1
2	Customer 2
3	Customer 3
4	Customer 4
5	Customer 5
6	Customer 6
7	Customer 7
8	Customer 8
9	Customer 9
10	Customer 10
11	Customer 11

Basic Data

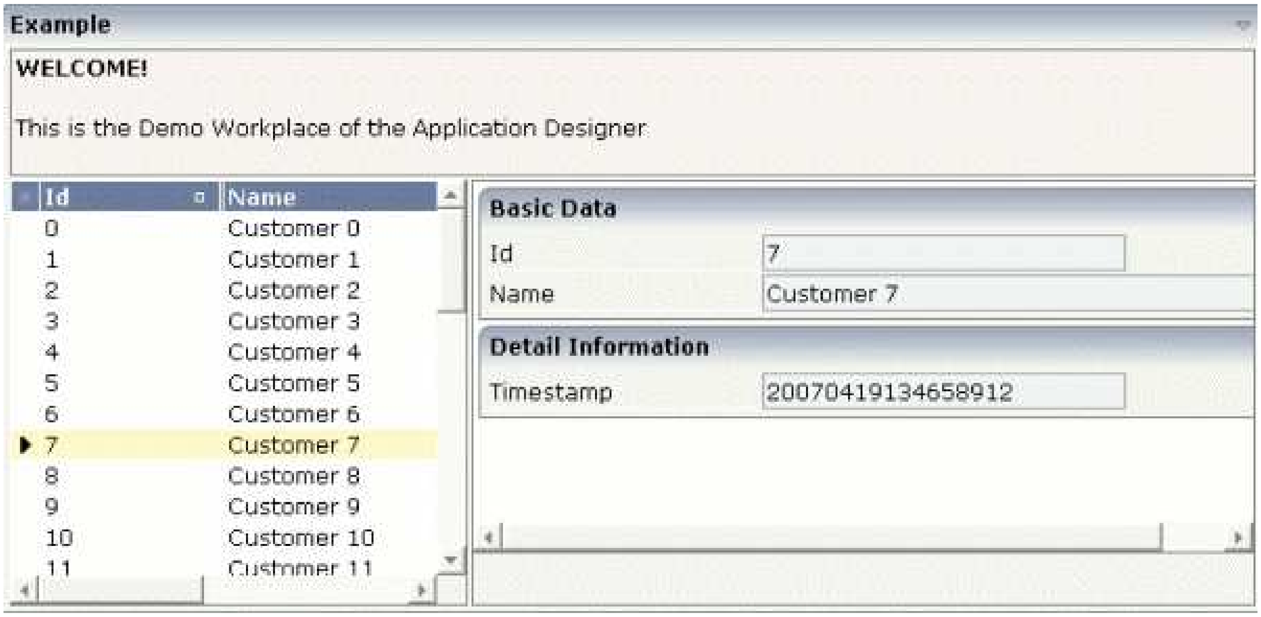
Id <no customer selected>

Name <no customer selected>

Detail Information

Timestamp 20070419134136006

When selecting a customer on the left, the customer detail screen is displayed on the right:



Example

WELCOME!

This is the Demo Workplace of the Application Designer

Id	Name
0	Customer 0
1	Customer 1
2	Customer 2
3	Customer 3
4	Customer 4
5	Customer 5
6	Customer 6
7	Customer 7
8	Customer 8
9	Customer 9
10	Customer 10
11	Customer 11

Basic Data

Id 7

Name Customer 7

Detail Information

Timestamp 20070419134658912

When the user selects another record on the left, the screen on the right is updated accordingly.

This chapter covers the following topics:

- The Multi Frame Page Around

- The Left Frame
- The Right Frame

The Multi Frame Page Around

First let us have a look at the multi frame page itself. The layout definition is as follows:

```
<mfpage separation="rows" sizing="70,*">
    <mfhtmlframe target="HEADER"
                 url="../HTMLBasedGUI/workplace/welcome.html"
                 resizable="true"
                 withborder="false"
                 scrolling="false"
                 framestyle="border: 1px #808080 solid">
    </mfhtmlframe>
    <mfframeset target="AROUND"
                separation="cols"
                sizing="200,*">
        <mfcisframe target="INNERLEFT"
                   cisurl="/cisdemos/25_mfinnerleft.html"
                   framestyle="border-right: 1px solid #808080;
                   border-bottom: 1px solid #808080">
        </mfcisframe>
        <mfcisframe target="INNERRIGHT"
                     cisurl="/HTMLBasedGUI/empty.html"
                     framestyle="border: 1px solid #808080">
        </mfcisframe>
    </mfframeset>
</mfpage
```

The page is subdivided into three frames: "HEADER", "INNERLEFT" and "INNERRIGHT". Two of them are Application Designer frames, one is an HTML frame. Every frame is pointing to a certain page.

The Left Frame

The INNERLEFT frame’s page displays a text grid and lets the user select from the list of items. The layout definition is:

```
<page model="MFInnerLeftAdapter">
    <pagebody horizdist="false" takefullheight="true">
        <itr height="100%" fixlayout="true" width="100%">
            <textgrid2 griddataprop="customers" width="100%" height="100%" selectprop="selected"
                       singleselect="true" hscroll="true" directselectmethod="onSelect"
                       directselectevent="onclick">
                <column name="Id" property="id" width="100">
                </column>
                <column name="Name" property="name" width="400">
                </column>
            </textgrid2>
        </itr>
    </pagebody>
</page>
```

The adapter implementation is done in the following way:

```
import com.softwareag.cis.server.Adapter;
import com.softwareag.cis.server.ServerLog;
import com.softwareag.cis.server.util.TEXTGRIDCollection;

public class MFInnerLeftAdapter
        extends Adapter
{
        // ------------------------------------------------------------------------
        // inner classes
        // ------------------------------------------------------------------------

        public class CustomerInfo
        {
            boolean m_selected;
            String m_id;
            String m_name;
            public String getId() { return m_id; }
            public String getName() { return m_name; }
            public boolean getSelected() { return m_selected; }
            public void setId(String string) { m_id = string; }
            public void setName(String string) { m_name = string; }
            public void setSelected(boolean b) { m_selected = b; }
        }
        // ------------------------------------------------------------------------
        // members
        // ------------------------------------------------------------------------

        TEXTGRIDCollection m_customers = new TEXTGRIDCollection();

        // ------------------------------------------------------------------------
        // property access
        // ------------------------------------------------------------------------

        public TEXTGRIDCollection getCustomers() { return m_customers; }

        // ------------------------------------------------------------------------
        // public methods
        // ------------------------------------------------------------------------

        public void init()
        {
                super.init();
                for (int i=0; i<40; i++)
                {
                        CustomerInfo info = new CustomerInfo();
                        ci.setId(""+i);
                        ci.setName("Customer " + i);
                        m_customers.add(ci);
                }
        }

        public void onSelect()
        {
                try
                {
                        CustomerInfo info = (CustomerInfo)m_customers.findLastSelectedItem();
                        // prepare adapter of right frame
                        MFInnerRightAdapter mfira =
                        (MFInnerRightAdapter)findAdapter(MFInnerRightAdapter.class);
                        mfira.prepare(ci.getId());
                        // preload adapter so that only one request is executed
                        includeAdapterInResponse("../_DevelopersGuide/mfinnerright.html",false);
                        // refersh target
                        refreshTarget("INNERRIGHT");
                }
                catch (Throwable t) { ServerLog.appendException(t); }
        }
}
```

The class contains the following:

- An inner class for the text grid items.
- An `init` method for filling the text grid.
- A `onSelect()` method that is called when the user selects a text grid line.

The "critical" lines of code are inside the `onSelect()` method. Inside the method

- the selected line is determined,
- the adapter of the right neighbor screen is prepared so that it shows the data of the selected line,
- the right page is switched to the detail page (if first call) or
- the right page is refreshed to present the correct adapter information.

The Right Frame

The right frame is loaded with */HTMLBasedGUI/empty.html* first. With the first selection in the text grid, the detail page is opened inside the right frame. Afterwards, the detail page is refreshed to update its content.