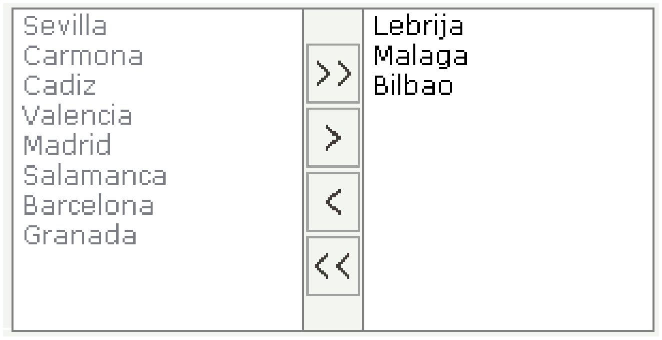
MULTISELECT

The MULTISELECT control allows comfortable input of multiple selections of items from a defined number of items.

The following topics are covered below:

- Example
- Problems with MULTISELECT
- Properties

Example



The available items are rendered on the left and are brought to the right by choosing the corresponding button. There are buttons to bring all items from the left to the right, and back.

The XML layout looks as follows:

```
<rowarea name="Control Demo">
    <itr>
        <label name="Multiselect Control" width="150">
        </label>
        <multiselect valueprop="towns" flush="server" width="300" helpid="MultiSelectHelp">
        </multiselect>
    </itr>
</rowarea>
```

The adapter code is:

```
import com.softwareag.cis.server.Adapter;
import com.softwareag.cis.server.util.MULTISELECTInfo;
import com.softwareag.cis.server.util.OPTIONInfo;

// This class is a generated one.


public class MethodLinkAdapter
    extends Adapter
{
    // ------------------------------------------------------------------------
    // property access
    // ------------------------------------------------------------------------
    // property >towns<
    MULTISELECTInfo m_towns = new MULTISELECTInfo();
    public MULTISELECTInfo getTowns() { return m_towns; }

    // ------------------------------------------------------------------------
    // public access
    // ------------------------------------------------------------------------

    public void onOutput()
    {
        StringBuffer sb = new StringBuffer();
        OPTIONInfo[] items = m_towns.getItems();
        boolean first = true;
        for (int i=0; i<items.length; i++)
        {
            if (items[i].getSelected() == true)
        {
        if (first)
            first = false;
        else
            sb.append(", ");
        sb.append(items[i].getText());
        }
        }
        outputMessage(MT_SUCCESS,sb.toString());
    }

    // ------------------------------------------------------------------------
    // standard adapter methods
    // ------------------------------------------------------------------------

    /** initialisation - called when creating this instance*/
    public void init()
    {
        m_towns.addItem(new OPTIONInfo("Sevilla", "Sevilla", false));
        m_towns.addItem(new OPTIONInfo("Carmona", "Carmona", false));
        m_towns.addItem(new OPTIONInfo("Lebrija", "Lebrija", true));
        m_towns.addItem(new OPTIONInfo("Cadiz", "Cadiz", false));
        m_towns.addItem(new OPTIONInfo("Valencia", "Valencia", false));
        m_towns.addItem(new OPTIONInfo("Madrid", "Madrid", false));
        m_towns.addItem(new OPTIONInfo("Salamanca", "Salamanca", false));
        m_towns.addItem(new OPTIONInfo("Malaga", "Malaga", true));
        m_towns.addItem(new OPTIONInfo("Barcelona", "Barcelona", false));
        m_towns.addItem(new OPTIONInfo("Bilbao", "Bilbao", true));
        m_towns.addItem(new OPTIONInfo("Granada", "Granada", false));
    }
}
```

On the server side, the control is associated with an instance of class `MULTISELECTInfo`. The items are passed as `OPTIONInfo` objects. Depending on the boolean value inside the constructor, items are either on the left side (unselected) or on the right side (selected). The `onOutput()` method shows how to derive information - which item was selected by the user.

Problems with MULTISELECT

With previous releases, the MULTISELECT control internally used the HTML control SELECT which has certain problems (see the description of the COMBOFIX control). The control is now rendered in a different way; the problems do not exist anymore.

Properties

Basic			
valueprop	Name of the adapter property representing this control on server side. The property must be of type MULTISELECTInfo. Please view corresponding documentation inside the Java API Documentation. The MULTISELECT control does not offer a STATUSPROP property in the way other controls (FIELD, ...) allow you to manipulate there input status at runtime. Instead you can set the status as method of the MULTISELECTInfo object that you create inside your adapter.	Obligatory	
width	Width of the control. There are three possibilities to define the width: (A) You do not define a width at all. In this case the width of the control will either be a default width or - in case of container controls - it will follow the width that is occupied by its content. (B) Pixel sizing: just input a number value (e.g. "100"). (C) Percentage sizing: input a percantage value (e.g. "50%"). Pay attention: percentage sizing will only bring up correct results if the parent element of the control properly defines a width this control can reference. If you specify this control to have a width of 50% then the parent element (e.g. an ITR-row) may itself define a width of "100%". If the parent element does not specify a width then the rendering result may not represent what you expect.	Obligatory	100 120 140 160 180 200 50% 100%

height	Height of the control. There are three possibilities to define the height: (A) You do not define a height at all. As consequence the control will be rendered with its default height. If the control is a container control (containing) other controls then the height of the control will follow the height of its content. (B) Pixel sizing: just input a number value (e.g. "20"). (C) Percentage sizing: input a percantage value (e.g. "50%"). Pay attention: percentage sizing will only bring up correct results if the parent element of the control properly defines a height this control can reference. If you specify this control to have a height of 50% then the parent element (e.g. an ITR-row) may itself define a height of "100%". If the parent element does not specify a width then the rendering result may not represent what you expect.	Obligatory	100 150 200 250 300 250 400 50% 100%
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.	Optional	
Appearance			
displayonly	If set to true, the FIELD will not be accessible for input. It is just used as an output field.	Optional	true false
withupdown	If set to true, corresponding up and down arrows appear on the right hand side. These arrows allow for changing the order of the selected items.	Optional	true false
align	Horizontal alignment of control in its column. Each control is "packaged" into a column. The column itself is part of a row (e.g. ITR or TR). Sometimes the size of the column is bigger than the size of the control itself. In this case the "align" property specifies the position of the control inside the column. In most cases you do not require the align control to be explicitly defined because the size of the column around the controls exactly is sized in the same way as the contained control. If you want to directly control the alignment of text: in most text based controls there is an explicit property "textalign" in which you align the control's contained text.	Optional	left center right

valign	Vertical alignment of control in its column. Each control is "packaged" into a column. The column itself is part of a row (e.g. ITR or TR). Sometimtes the size of the column is bigger than the size of the control. In this case the "align" property specify the position of the control inside the column.	Optional	top middle bottom
colspan	Column spanning of control. If you use TR table rows then you may sometimes want to control the number of columns your control occupies. By default it is "1" - but you may want to define the control to span over more than one columns. The property only makes sense in table rows that are snychronized within one container (i.e. TR, STR table rows). It does not make sense in ITR rows, because these rows are explicitly not synched.	Optional	1 2 3 4 5 50 int-value
rowspan	Row spanning of control. If you use TR table rows then you may sometimes want to control the number of rows your control occupies. By default it is "1" - but you may want to define the control two span over more than one columns. The property only makes sense in table rows that are snychronized within one container (i.e. TR, STR table rows). It does not make sense in ITR rows, because these rows are explicitly not synched.	Optional	1 2 3 4 5 50 int-value
msstyle	CSS style definition that is directly passed into this control. With the style you can individually influence the rendering of the control. You can specify any style sheet expressions. Examples are: border: 1px solid #FF0000 background-color: #808080 You can combine expressions by appending and separating them with a semicolon. Sometimes it is useful to have a look into the generated HTML code in order to know where direct style definitions are applied. Press right mouse-button in your browser and select the "View source" or "View frame’s source" function.	Optional	

Binding			
valueprop	(already explained above)		
flush	Flushing behaviour of the input control. By default an input into the control is registered within the browser client - and communicated to the server adapter object when a user e.g. presses a button. By using the FLUSH property you can change this behaviour. Setting FLUSH to "server" means that directly after changing the input a synchronization with the server adapter is triggered. As consequence you directly can react inside your adapter logic onto the change of the corresponding value. - Please be aware of that during the synchronization always all changed properties - also the ones that were changed before - are transferred to the adapter object, not only the one that triggered the synchonization. Setting FLUSH to "screen" means that the changed value is populated inside the page. You use this option if you have redundant usage of the same property inside one page and if you want to pass one changed value to all its representaion directly after changing the value.	Optional	screen server
flushmethod	When the data synchronization of the control is set to FLUSH="server" then you can specify an explicit method to be called when the user updates the content of the control. By doing so you can distinguish on the server side from which control the flush of data was triggered.	Optional	
Online Help			
helpid	Help id that is passed to the online help management in case the user presses F1 on the control.	Optional	
Miscellaneous			
testtoolid	Use this attribute to assign a fixed control identifier that can be later on used within your test tool in order to do the object identification	Optional	