

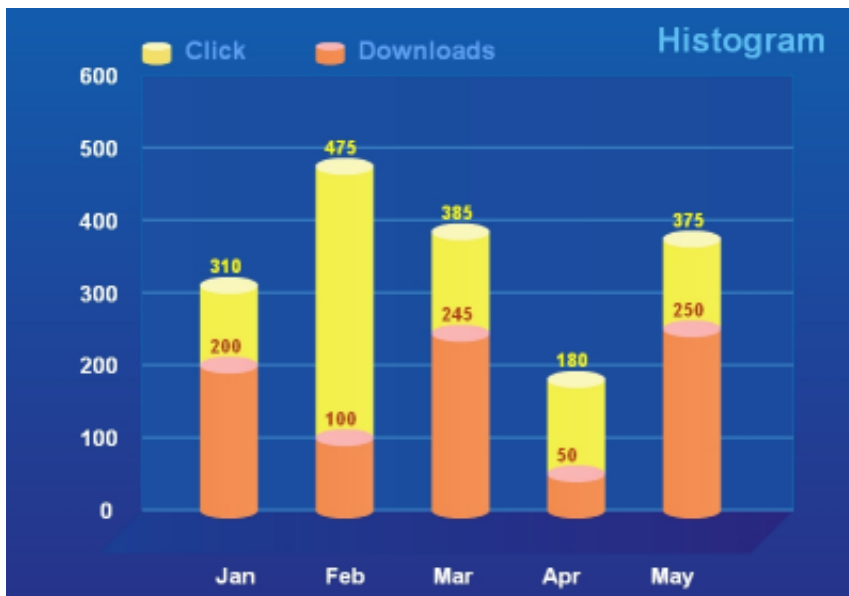
MACROMEDIAFLASH

The MACROMEDIAFLASH control defines an area in which a Macromedia Flash movie is run. It provides for a client-side API that is to be used within the movie's action script. With that, the application (page adapter) can react on clicks that are captured by the movie's action script.

The following topics are covered below:

- Example
 - Creating the Action Script
 - Properties
-

Example



The XML layout definition is:

```
<pagebody>
  <itr takefullwidth="true" height="100%">
    <macromediaflash valueprop="flashInfo" width="100%" height="100%">
      </macromediaflash>
    </itr>
  </pagebody>
```

The Java code of the adapter is:

```
import com.softwareag.cis.server.*;
import com.softwareag.cis.server.util.MACROMEDIAFLASHInfo;
import com.softwareag.cis.util.*;

// This class is a generated one.

public class MacroMediaFlashAdapter
  extends Adapter
```

```

{
    public class MyMACROMEDIAFLASHInfo extends MACROMEDIAFLASHInfo
    {
        MyMACROMEDIAFLASHInfo(Adapter owner)
        {
            super(owner);
        }

        public void reactOnFlushEvent(String flushInfo)
        {
            outputMessage(MT_SUCCESS, flushInfo);
        }
    }

    // property >flashInfo<
    MyMACROMEDIAFLASHInfo m_flashInfo = new MyMACROMEDIAFLASHInfo(this);
    public MyMACROMEDIAFLASHInfo getFlashInfo() { return m_flashInfo; }

    /** */
    public void init()
    {
        // content byte array contains the flash movie
        byte[] content = ...;
        String key = "FlashMovie_" + UniqueIdMgmt.createPseudoGUID() + ".swf";
        SessionBuffer sb = findCISSessionContext().getSessionBuffer();
        String url = sb.addDocument(key, content, "application/x-shockwave-flash");
        m_flashInfo.showDocument(url);
    }
}

```

Creating the Action Script

The MACROMEDIAFLASH control provides for a client-side API to be invoked within the movie's action script. When creating the flash movie on the fly (for example, using Flagstone API), use the method `generateActionScript` of the `MACROMEDIAFLASHInfo` class as shown below:

```
String actionScript = m_flashInfo.generateActionScript("This string is flushed to server on click");
```

When creating the movie in a static way (for example, using Adobe's Flash Editor), the action script to be included looks like this:

```
FSGetUrl("javascript:(setProperty('flashInfo.flushInfo', 'This string is flushed to server on click');
        invokeMethodInModel('flashInfo.reactOnFlushEvent');");
```

where `flashInfo` is the property name of the page adapter returning the `MyMACROMEDIAFLASHInfo` object.

Properties

Basic			
valueprop	Server side property representation of the control.	Optional	
width	<p>Width of the control.</p> <p>There are three possibilities to define the width:</p> <p>(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.</p> <p>(B) Pixel sizing: just input a number value (e.g. "100").</p> <p>(C) Percentage sizing: input a percentage 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.</p>	Optional	<p>100</p> <p>120</p> <p>140</p> <p>160</p> <p>180</p> <p>200</p> <p>50%</p> <p>100%</p>
height	<p>Height of the control.</p> <p>There are three possibilities to define the height:</p> <p>(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.</p> <p>(B) Pixel sizing: just input a number value (e.g. "20").</p> <p>(C) Percentage sizing: input a percentage 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.</p>	Optional	
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.	Optional	