

This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. The article is believed to be still valid for the original topic scope.



## Contents

- [1 About this widget](#)
- [2 WidSets Scripting Language code: hello\\_world.he](#)
- [3 Widget.xml](#)
- [4 See also](#)

## About this widget

This is a basic Hello World demo widget.

## WidSets Scripting Language code: hello\_world.he

```
class hello_world
{
    //It is recommended to store command ids to static constants
    const int CMD_BACK = 1;

    //MenuItems are displayed over phone's softkeys
    //Usually to go back, ok, open options menu etc
    MenuItem BACK = new MenuItem(CMD_BACK, "Back");

    //WidSets framework calls createElement() per
    //each script-element it finds from views being
    //created by createView()

    Component createElement(String viewName,
                            String elementName,
                            Style style,
                            Object context)
    {
        //return simple label with style defined in widget.xml,
        //in this case "text"
        if (elementName.equals("hello")) {
            //to get the label aligned in the middle of the
            //view you need to contain it inside a Flow which
            //you then return here
            return new Label(style, "Hello World");
        }
        return null;
    }

    void startWidget ()
    {
        //instantiate minimized view in startup
        setMinimizedView(createView("viewMini", getStyle("bg")));
    }
}
```

## Hello\_World

```
}

Shell openWidget ()
{
    //instantiate maximized view when user opens this widget
    Flow view = createView("viewMaxi", getStyle("bg"));
    return new Shell(view);
}

MenuItem getSoftKey(Shell shell, Component focused, int key)
{
    //return the key to be displayed at position=SOFTKEY_BACK
    //this is usually the right softkey, for other key
    //positions return null

    if (key == SOFTKEY_BACK) {
        return BACK;
    }
    return null;
}

void actionPerformed(Shell shell, Component source, int action)
{
    //when CMD_BACK event comes in, pop the current shell (this widget)
    if (action == CMD_BACK) {
        popShell(shell);
    }
}
}
```

## Widget.xml

```
<?xml version="1.0" encoding="utf-8"?>

<widget spec_version="2.0">

    <info>
        <name>hello_world</name>
        <version>0.1</version>
        <author>render</author>
        <clientversion>0.98</clientversion>
        <shortdescription>Very simple widget</shortdescription>
        <longdescription>Simplest possible widget saying hello to the world.</longdescription>
        <tags>test example hello world</tags>
    </info>

    <parameters>
        <parameter type="string" name="widgetname">Hello World</parameter>
    </parameters>

    <resources>
        <code src="hello_world.he"/>
        <stylesheet>
            bg {
                color-1: white;
                background: solid black;
            }
        </stylesheet>
    </resources>
</widget>
```

## Hello\_World

```
        align: hcenter vcenter;
        border: 1 1 1 1;
        border-type: rectangle white;
    }

    text {
        color-1: white;
        padding: 2 2 2 2;
    }
</stylesheet>
</resources>

<layout minimizedheight="65sp">
    <view id="viewMini" class="bg">
        <label class="text">${widgetname}</label>
    </view>

    <view id="viewMaxi" class="bg">
        <script id="hello" class="text"/>
    </view>

    <webview>
        <weblabel class="top: 0px; left: 10px;" style="color: black;">${widgetname}</weblabel>
    </webview>
</layout>

</widget>
```

## See also

- [WidSets SDK](#)
- [WidSets Client](#)
- [WidSets Scripting Language](#)
- [Widget examples](#)
  - ◆ [WidClock](#)
  - ◆ [Memory Game](#)
  - ◆ [Filter test](#)
  - ◆ **Hello World**
  - ◆ [UITest](#)