

# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany

Author: Frithjof Eckhardt, updated: 25. Okt. 2013

**Note:** The package comes with the namespace /UKW/, which is the namespace for Wuerzburg University Hospital, Germany. You can't create objects in this namespace in your SAP system. Therefore map the namespace prefix of each object from /UKW/ to any name starting with Z or Y, or your own customer namespace.

## Demo program 1

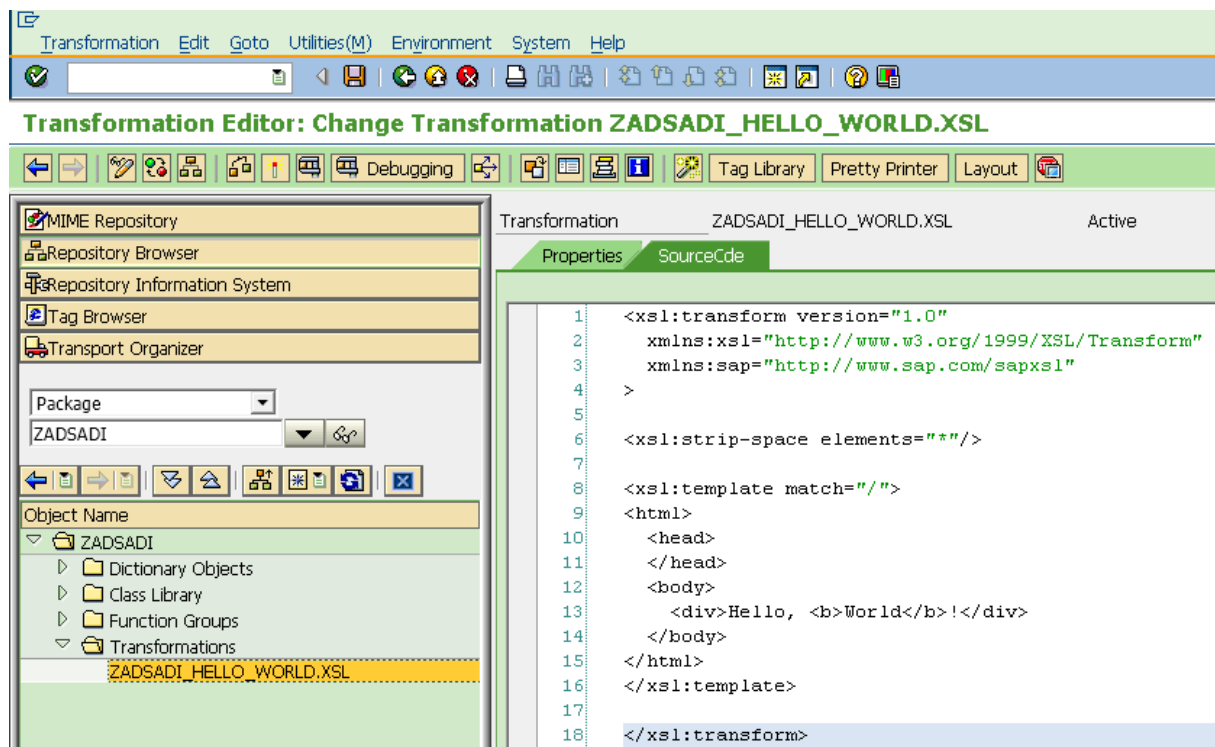
### Create XSL stylesheet

```
<xsl:transform version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:sap="http://www.sap.com/sapxsl"
>

<xsl:strip-space elements="*" />

<xsl:template match="/">
<html>
  <head>
  </head>
  <body>
    <div>Hello, <b>World</b>!</div>
  </body>
</html>
</xsl:template>

</xsl:transform>
```



# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany

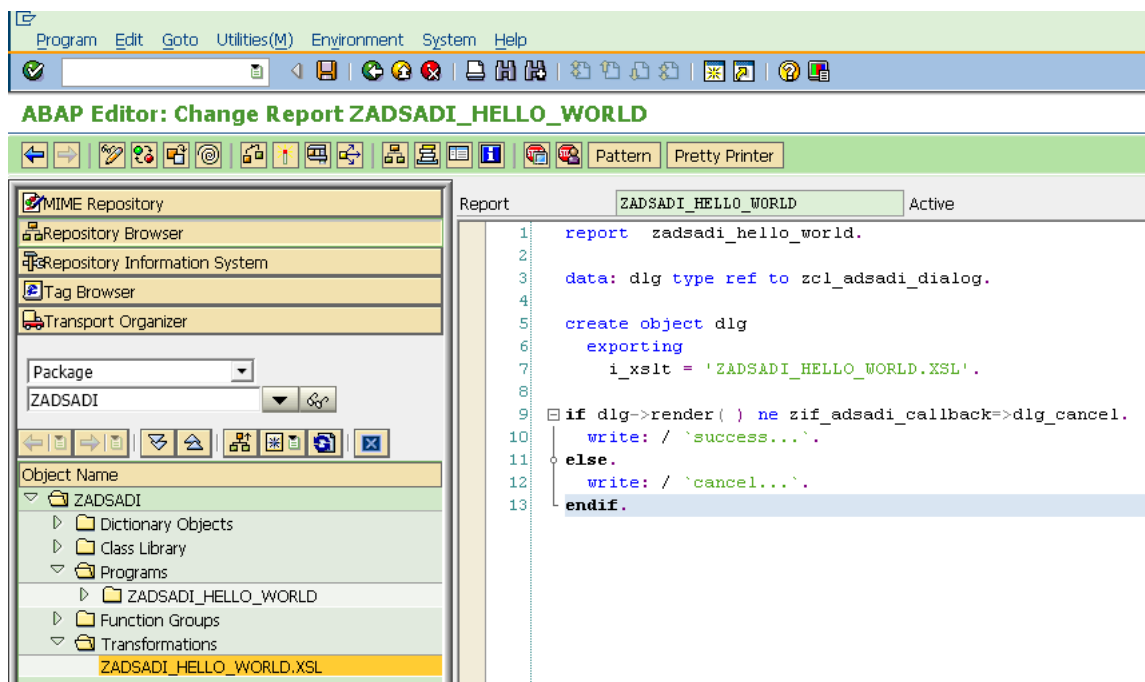
## Create program

```
report zadsadi_hello_world.

data: dlg type ref to /ukw/adsadi_dialog.

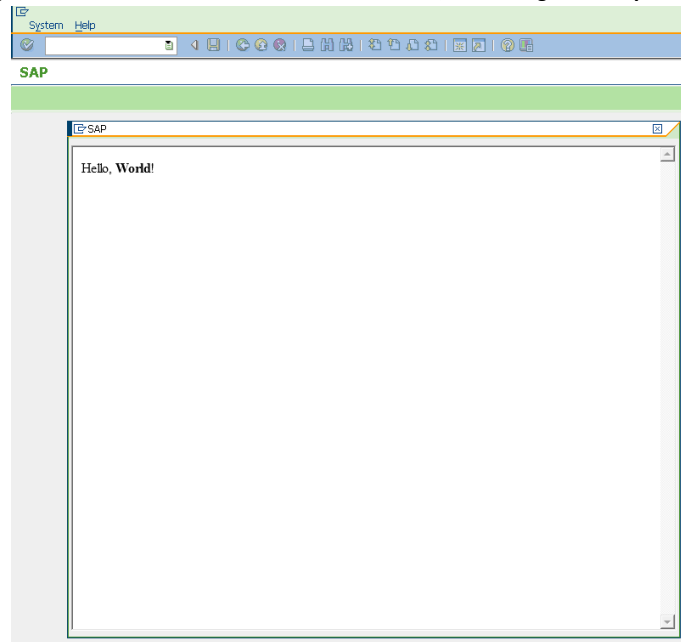
create object dlg
  exporting
    i_xslt = 'ZADSADI_HELLO_WORLD.XSL'.

if dlg->render( ) ne /ukw/adsadi_dialog=>dlg_cancel.
  write: / `success...`.
else.
  write: / `cancel...`.
endif.
```



# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany



## Demo program 2

---

```
report zadsadi_hello_world2.

data: dlg type ref to /ukw/adsadi_dialog.

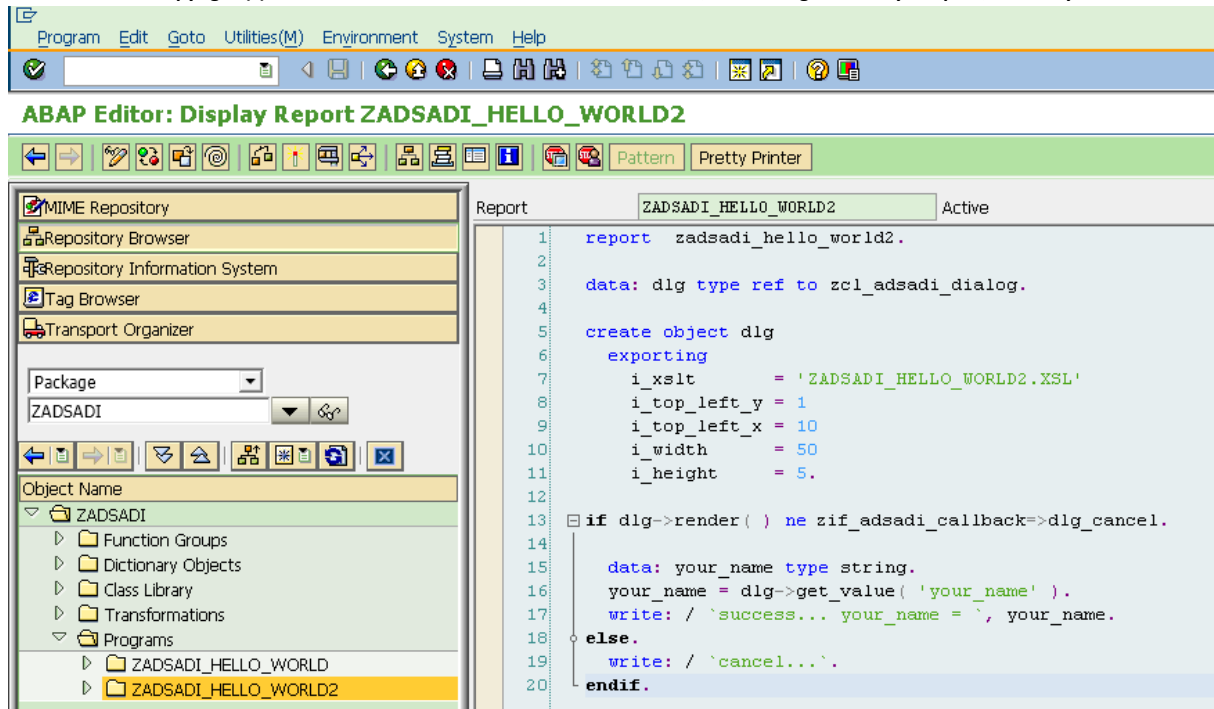
create object dlg
  exporting
    i_xslt      = 'ZADSADI_HELLO_WORLD2.XSL'
    i_top_left_y = 1
    i_top_left_x = 10
    i_width     = 50
    i_height    = 5.

if dlg->render( ) ne /ukw/adsadi_dialog=>dlg_cancel.

  data: your_name type string.
  your_name = dlg->get_value( 'your_name' ).
  write: / `success... your_name = `, your_name.
else.
  write: / `cancel...`.
endif.
```

# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany



```
<xsl:transform version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:sap="http://www.sap.com/sapxsl"
>

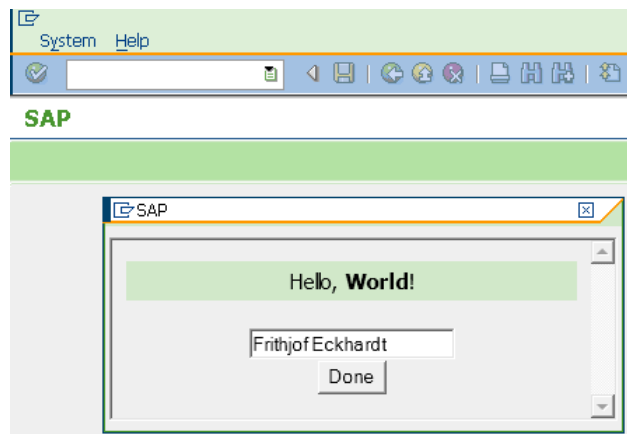
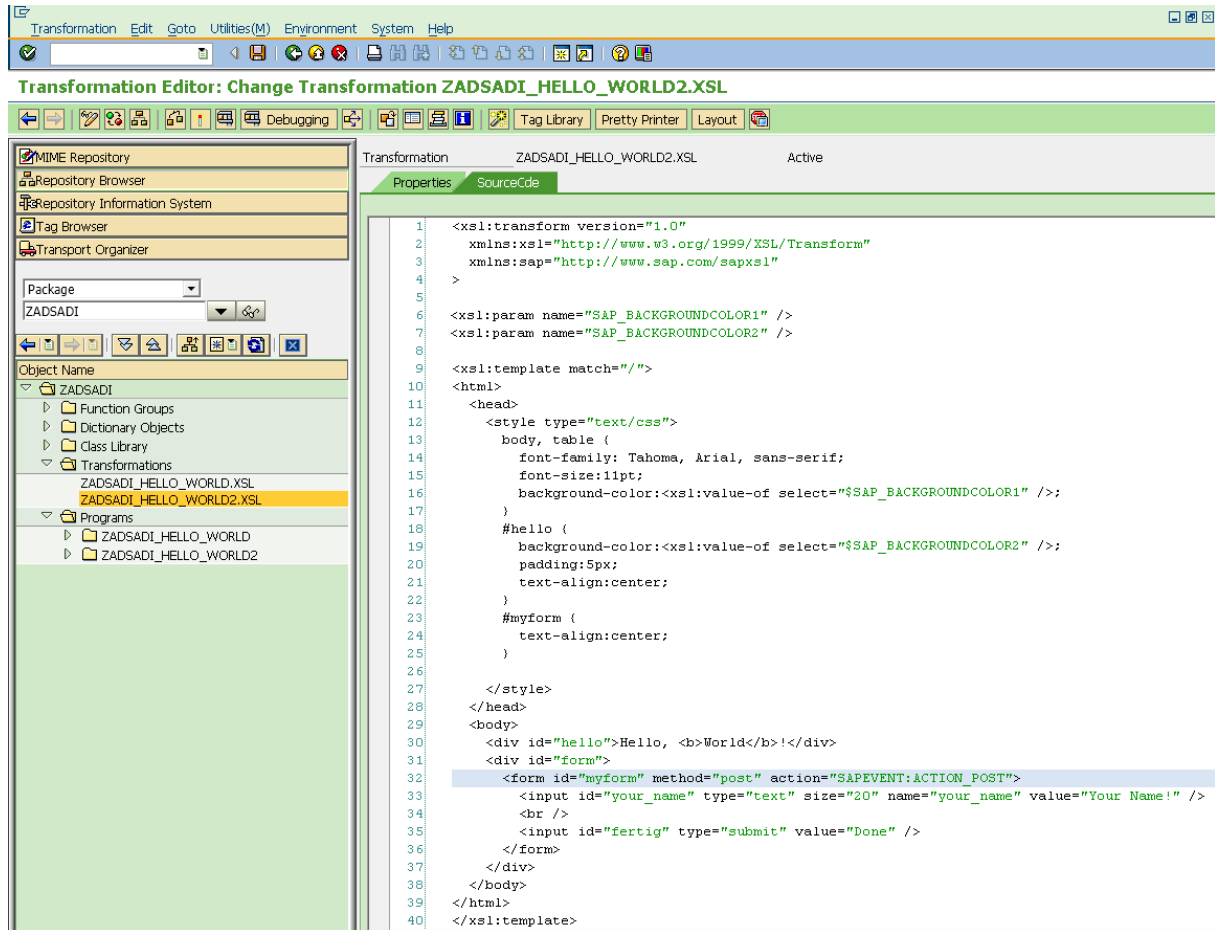
<xsl:param name="SAP_BACKGROUND_COLOR1" />
<xsl:param name="SAP_BACKGROUND_COLOR2" />

<xsl:template match="/">
<html>
  <head>
    <style type="text/css">
      body, table {
        font-family: Tahoma, Arial, sans-serif;
        font-size: 11pt;
        background-color: <xsl:value-of select="$SAP_BACKGROUND_COLOR1" />;
      }
      #hello {
        background-color: <xsl:value-of select="$SAP_BACKGROUND_COLOR2" />;
        padding: 5px;
        text-align: center;
      }
      #myform {
        text-align: center;
      }
    </style>
  </head>
  <body>
    <div id="hello">Hello, <b>World</b>!</div>
    <div id="form">
      <form id="myform" method="post" action="SAPEVENT:ACTION_POST">
```

# AdSaDi - Advanced SAP Dialog

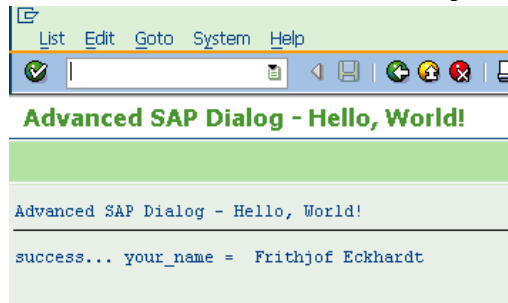
Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany

```
<input id="your_name" type="text" size="20" name="your_name" value=
"Your Name!" />
<br />
<input id="fertig" type="submit" value="Done" />
</form>
</div>
</body>
</html>
</xsl:template>
</xsl:transform>
```



# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany



## Demo program 3

---

AdSaDi works on top of the CL\_GUI\_HTML\_VIEWER (<http://help.sap.com/>, <http://goo.gl/NpvFU>) and therefore has to deal with the limitations of the SAPHTMLP protocol, i.e. entries in form fields must not have more than 512 characters. If longer text is entered into an HTML input field or textarea, an ABAP short dump will show up.

Further more, if the length of text is between 250 and 512 characters all the content beyond the 250th character will be lost, due to internal data types of CL\_GUI\_HTML\_VIEWER.

So to meet this challenge, we have to chop up such long texts into smaler pieces, each not longer than 250 characters. This needs to be done on the HTML client side before the HTML form is submitted to SAP. In this example we use JQuery ([www.jquery.com](http://www.jquery.com)) to do this tedious job.

```
<script type="text/javascript">
    jQuery.noConflict();
    jQuery(document).ready(function($) {
        $('#myform').submit( function(e) {
            var content = $('#editor').val();

            // Replace newline with html tag to prevent ABAP from adding dirty characters
            content = content.replace(/\n\r?/g, '<br />');

            // SAPHTMLP protocol processes only values that are no longer than 512 characters and
            // ABAP stores these values later in fields of size 250. In order to prevent
            // runtime errors and loss of content, slice the content in 250-1 long
            // chunks, add an end marking to prevent automatic right-sided whitespace
            // optimization of ABAPs string processing
            // (right-hand whitespace is not always preserved).
            var len = 250-1, prev = 0;
            output = [];
            var cnt = 0;
            var cnt_str = "";

            while(content.charAt(prev)) {
                cnt += 1;
                if (cnt <= 10) {
                    cnt_str = "00" + cnt;
                } else if (cnt <= 100) {
                    cnt_str = "0" + cnt;
                } else {
                    cnt_str = cnt;
                }
            }
        });
    });
```

# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany

```
output.push({
  name: $('#editor').attr("name") + "_chunk_" + cnt_str,
  value: content.substr(prev, len) + "|"
});
prev += len;
};

// Add chunks as hidden fields to the form
$.each(output, function(i,param){
  $('<input />')
    .attr('type', 'hidden')
    .attr('name', param.name)
    .attr('value', param.value)
    .appendTo('#myform');
});

// Remove original field to prevent the oversized
// value to be transferred to SAP on form submit.
$('#editor').remove();

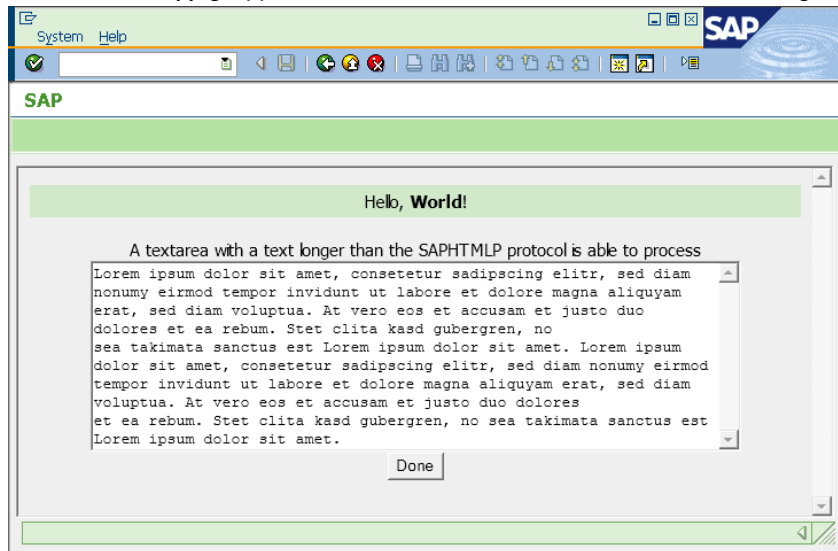
return true;
});
});
</script>

...

<body>
  <div id="hello">Hello, <b>World</b>!</div>
  <div id="form">
    <form id="myform" method="post" action="SAPEVENT:ACTION_POST">
      <label for="editor">
        A textarea with a text longer than the SAPHTMLP protocol is able to process</label>
      <br />
      <textarea id="editor" name="long_text" cols="68" rows="10" wrap="PHYSICAL">
        >Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempo
r invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam
et justo duo dolores et ea rebum. Stet clita kasd gubergren, no
sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sa
dipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat
, sed diam voluptua. At vero eos et accusam et justo duo dolores
et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet
.</textarea>
      <br />
      <input id="fertig" type="submit" value="Done" />
    </form>
  </div>
</body>
```

# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany



```
data: dlg type ref to /ukw/adsadi_dialog.

create object dlg
exporting
  i_xslt = 'ZADSADI_HELLO_WORLD3.XSL'
  i_size = 'FULL'.

if dlg->render( ) ne /ukw/adsadi_dialog=>dlg_cancel.

  data: long_text type string.
  data: line_len type i value 72.
  data: len type i.

  long_text = dlg->get_value( 'long_text' ).
  len = strlen( long_text ).

  write: / `long_text:` no-gap, len.

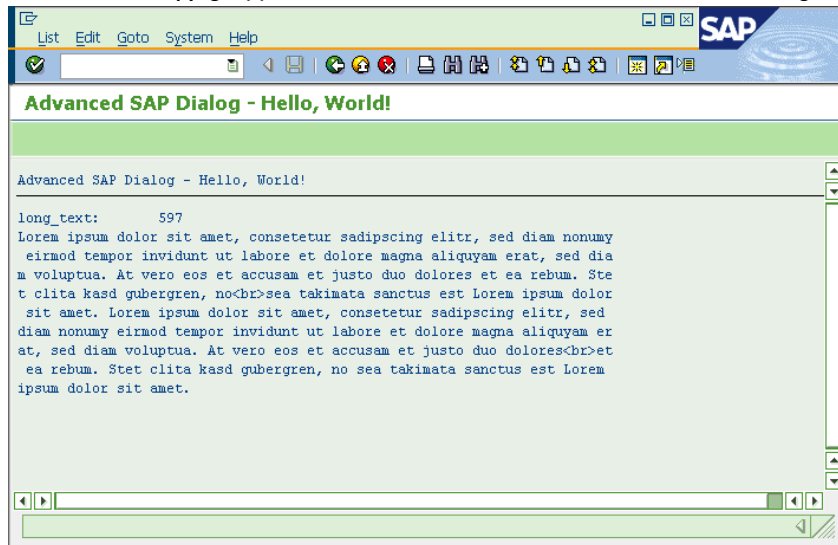
  while len > line_len.
    write: / long_text(line_len).
    long_text = long_text+line_len(*).
    len = strlen( long_text ).
  endwhile.
  write: / long_text(len).

endif.
```



# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany



## Demo XML Program 1

In this example we use static XML to be transformed into a dialog.

```
data: dlg type ref to /ukw/adsadi_dialog.
```

```
data: xml type string.
```

```
concatenate
```

```
  '<?xml version="1.0" encoding="utf-8" ?>'
  '<my-dialog title="Risk factor analysis">'
  '  <field type="select" text="Emergency admission" name="emergency" value="">'
  '    <option text="Outpatient" value="outpatient" />'
  '    <option text="Inpatient" value="inpatient" selected="yes" />'
  '    <option text="Day-Unit" value="day-unit" />'
  '  </field>'
  '  <field type="radio" text="Level of care III" name="care_level3" value="" />'
  '  <field type="radio" text="Stay in hospital last 12 month" name="stay_12month" value="" />'
  '  <field type="input" text="Comment" name="comment" value="" />'
  '</my-dialog>'
into xml.
```

```
create object dlg
```

```
  exporting
```

```
    i_xslt = 'ZADSADI_XML_DEMO_1.XSL'
    i_xml  = xml
    i_top_left_y = 1
    i_top_left_x = 1
    i_width      = 50
    i_height     = 10.
```

```
if dlg->render( ) ne /ukw/adsadi_dialog=>dlg_cancel.
```

```
  data: param type string.
```

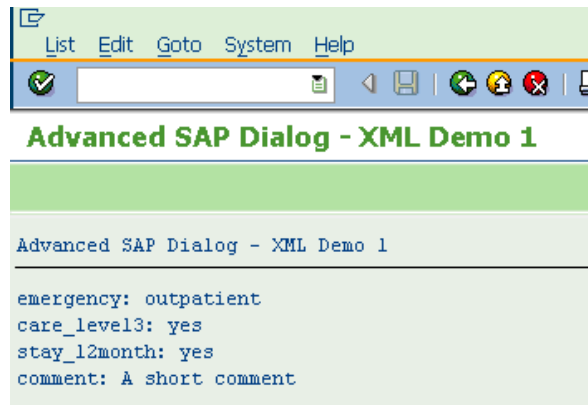
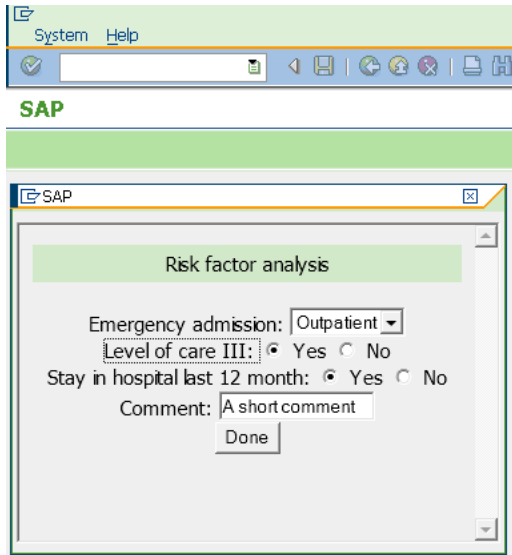
```
  param = dlg->get_value( 'emergency' ).
  write: / `emergency: ` no-gap, param.
```

```
  param = dlg->get_value( 'care_level3' ).
  write: / `care_level3: ` no-gap, param.
```

# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany

```
param = dlg->get_value( 'stay_12month' ).  
write: / `stay_12month: ` no-gap, param.  
  
param = dlg->get_value( 'comment' ).  
write: / `comment: ` no-gap, param.  
  
endif.
```



In the dialog we use again JQuery, now for validating the input: any set of radio buttons needs a valid selection, yes or no. We are able to use the full power of JQuery, so we animate the error message with the pulsate effect.

```
<script type="text/javascript">  
  jQuery.noConflict();  
  jQuery(document).ready(function($) {  
    $('#myform').submit( function(e) {  
      $('#error').attr('innerHTML', '');  
  
      var isOkay = true;  
      var fieldIsOkay = false;  
  
      <xsl:for-each select="/my-dialog/field[@type='radio']">  
        fieldIsOkay = false;  
        $("input:radio[name='<xsl:value-of select="@name" />']").each(function() {  
          if(this.checked) {fieldIsOkay = true;}  
        });  
        if(fieldIsOkay==false){  
          isOkay = false;  
          $('<div />')  
            .attr('innerHTML', "<xsl:value-of select="@text" /> missing!")  
            .appendTo('#error');  
        }  
      </xsl:for-each>  
  
      if(isOkay==false) {  
        $('#error').effect("pulsate", { mode:'show', times:3 }, 2000)  
      }  
  
      return isOkay;  
    });  
  });  
</script>
```

# AdSaDi - Advanced SAP Dialog

Copyright (c) 2011-2013 Servicecenter for Medical Informatics, Wuerzburg University Hospital, Germany

System Help

SAP

Risk factor analysis

Level of care III missing!  
Stay in hospital last 12 month missing!

Emergency admission: Inpatient

Level of care III: ☐ Yes ☐ No

Stay in hospital last 12 month: ☐ Yes ☐ No

Comment:

Done