

Utilizing DOCStudio dynamic templates in non-plugin applications

While DOCStudio was designed to accommodate perl applications using its plug-in architecture, other applications, including those hosted by other sites, may incorporate dynamically generated DOCStudio templates via javascript. The benefit of using DOCStudio templates over hard-coded templates include:

1. No need to recode applications when templates change
2. User experience remains consistent
3. Navigation remains constant
4. Breadcrumbs generated for user
5. Pageviews logged in DOCStudio stats
6. No need to code CSS
7. Faster development time

I. Create a page in DOCStudio to “host” the application template

1. Create a new page in the appropriate section.
2. In the Sort Order menu, select “Do not show links to this page” (this page will be sending javascript, not html, so you do not want users to access it as a normal page).
3. Title the page something that will be appropriate for the entire application (the title of the page will appear in the breadcrumbs throughout the application).
4. Abstract and Keywords may be left blank, though it may be helpful to include a note in the abstract that this page serves up a template for an application and should not be altered... just in case an admin decides to “clean up” the site.
5. Uncheck the box to include in searches and sitemap (remember, you do not want anyone to access this page).
6. Filename may be anything, but it may be helpful to include the word “template”. Remember what you type – you will need the filename later.
7. Ignore other fields and scroll to Page Body
8. The Page Body field will not be used, so just enter a space or dummy text. (The application will be the page body.)
9. In the Template menu, select “nondocsjs_template.htm”
10. Select a CSS if necessary.
11. Save the page.

If the application is being hosted by another company, an internal DOCStudio administrator will have to create the “host” pages.

After you click Save and DOCStudio displays the page for you, you will see why you don’t want the page to be accessed as a normal site page! DOCStudio sends this page as javascript.

Should you need to edit this page later, you won't be able to access it via the Admin buttons that normally appear at the bottom of every page. Instead, go to the main Admin page, click "Manage Content", then "Edit a Page". Find the page in the sitemap and click to edit.

In the DOCStudio stats Page Requests report, all pageviews for the application will be counted under this page. If you want to track multiple pages in the application, create a DOCStudio "host" page for each one you want to track.

II. Create/Edit the application html

Each page in the application will need to incorporate some javascript code to call the template. Note that the DOCStudio template takes care of the following elements and they should **not** be duplicated by the application:

1. CSS
2. <meta> tags
3. <body> tag
4. Sitewide navigation
5. Breadcrumbs
6. Targeted elements
7. Copyright
8. Page update date

The application should adhere to the CSS selected for the DOCStudio "host" page. Programmers should be provided with the CSS for their reference; if they need additional styles, a new CSS should be created and selected for the "host" page.

III. Insert javascript into top of all application pages

The javascript at the top of the page will write the template up to where the main content of the application starts. The top of every application page must start with

```
<html>
<head>
<title> Page Title Here </title>

    <script language="javascript"
src="http://www.yourdomain.com/host-filename.js"></script>

    <h1>Start Application Content</h1>
    <p>Etc...
```

NOTES:

- Replace "host-filename" with the filename of the page you created in DOCStudio to host the template.

- No <meta> tags
- No </head> tag
- No <body> tag
- Any additional javascripts needed by the application may be inserted after the <head> tag.
- The application's content is everything it would include between (but **not including**) <body> and </body>.

IV. Insert javascript into bottom of all application pages

The javascript at the bottom of the page finishes the template. The bottom of every page must end with

```
<script language="javascript">  
    document.write(template_bottom);  
</script>
```

```
</body>  
</html>
```

NOTE:

- </body> and </html> tags **after** javascript.

V. Sample application page

DOCStudio javascript code noted in blue. *Some of the ASP code has been omitted for space.*

```
<%
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "survey","",""

' Build Query String
sqlselct = "SELECT id, institution, department, country, MailCity, MailState
FROM [DeptReport] "
sqlorder = "ORDER BY country, institution, MailState, MailCity"
If Request.Form("sqlwhere") = "" Then
    sqlwhere = createWhereClause()
Else
    sqlwhere = Replace(Request.Form("sqlwhere"), "~", "'")
End If
strSQL = sqlselct & sqlwhere & sqlorder
' Create Recordset
Set oRs = Server.CreateObject("ADODB.Recordset")
oRs.CursorType = 3
oRs.CursorLocation = 3
oRs.ActiveConnection = conn
oRs.Open strSQL
%>

<html>
<head>
<title>Survey of Staffing</title>

    <script language="javascript" src="http://www.yourdomain.com/host-
filename.js"></script>

<!-- start application content -->

<h1>Search Results</h1>
<p>
Your search returned <b><%=intRecordCount%></b> records.

<%If intRecordCount > 0 Then%>
You are now viewing records <b><%=intStart%> through <%=intFinish%></b>.
<br>&nbsp;
</p>
<TABLE BORDER=0 CELSPACING=0 cellpadding="2" WIDTH=100%>
<THEAD>
<TR bgcolor="#003300">
```

```

<TD WIDTH="30%" class=s2><b><font color="#ffffff">Institution</font></b></td>
<TD WIDTH="20%" class=s2><b><font color="#ffffff">Department</font></b></td>
<TD WIDTH="10%" class=s2><b><font color="#ffffff">Country</font></b></td>
<TD WIDTH="20%" class=s2><b><font color="#ffffff">City</font></b></td>
<TD WIDTH="20%" class=s2><b><font color="#ffffff">State</font></b></td>
</TR>
</THEAD>
<TBODY>
<%
For intRecord = 1 to oRs.PageSize
    If intRecord Mod 2 = 0 Then
        rowtype = "#eeeeee"
    Else
        rowType = "#ffffff"
    End If
    Response.Write "<TR VALIGN=TOP bgcolor=" & rowtype & ">" & CHR(10)
    Response.Write "<TD class=s2><A HREF='profile.asp?id=" &
oRs.Fields("id").Value & "'>" & oRs.Fields("institution").Value &
"</A><BR></TD>" & CHR(10)
    Response.Write "<TD class=s2>" & oRs.Fields("department").Value &
"<BR></TD>" & CHR(10)
    Response.Write "<TD class=s2>" & oRs.Fields("country").Value &
"<BR></TD>" & CHR(10)
    Response.Write "<TD class=s2>" & oRs.Fields("MailCity").Value &
"<BR></TD>" & CHR(10)
    Response.Write "<TD class=s2>" & oRs.Fields("MailState").Value &
"<BR></TD>" & CHR(10)
    Response.Write "</TR>" & CHR(10)
    oRs.MoveNext
    If oRs.EOF Then Exit For
Next

Response.Write "</TBODY><TFOOT></TFOOT></TABLE>&nbsp;<br>" & CHR(10)
Response.Write "<FORM METHOD=post ACTION=searchresults.asp>" & CHR(10)
Response.Write "<INPUT TYPE=hidden NAME=prevPage VALUE='" & intPage - 1 & "'>"
& CHR(10)
Response.Write "<INPUT TYPE=hidden NAME=nextPage VALUE='" & intPage + 1 & "'>"
& CHR(10)
Response.Write "<INPUT TYPE=hidden NAME=sqlwhere VALUE='" & Replace(sqlwhere,
"'", "~") & "'>" & CHR(10)
' Check to see if the current page is greater than the first page
' in the recordset. If it is, then add a "Previous" link.
If cInt(intPage) > 1 Then
    Response.Write "<INPUT TYPE=submit NAME=prev VALUE='<< Prev'> " & CHR(10)
End If
' Check to see if the current page is less than the last page

```

```
' in the recordset. If it is, then add a "Next" link.
If cInt(intPage) < cInt(intPageCount) Then
    Response.Write "<INPUT TYPE=submit NAME=next VALUE='Next >>'>" &
CHR(10)
    End If
End If
Response.Write "</FORM>"

oRs.close()
conn.close()
%>

<!-- end application content -->

<script language="javascript">
    document.write(template_bottom);
</script>

</body>
</html>
```