

15 Seconds : Incorporating SQL Server Reporting Services into your Web Site

Don Franke
03/31 /2005

This article discusses a pretty simple combination of C#, Javascript, an IFRAME, and SQL Server Reporting Services to create a one-click web page for viewing all the reports for your web application.

This solution was the result of the need for a consolidated view for all reports for a particular web site. All the user has to do is select the report and click the button 'View Report'; no jumping back and forth between a report-selection page and report-viewing page (such as the default Reporting Services page is set up.)

Note: This article does not discuss how to create reports using Reporting Services. It is assumed the reports have already been created, and that you have a .NET web site ready to add reports to.

User Walk Thru

When a user first comes into the reports page, they will see something like Figure 1.

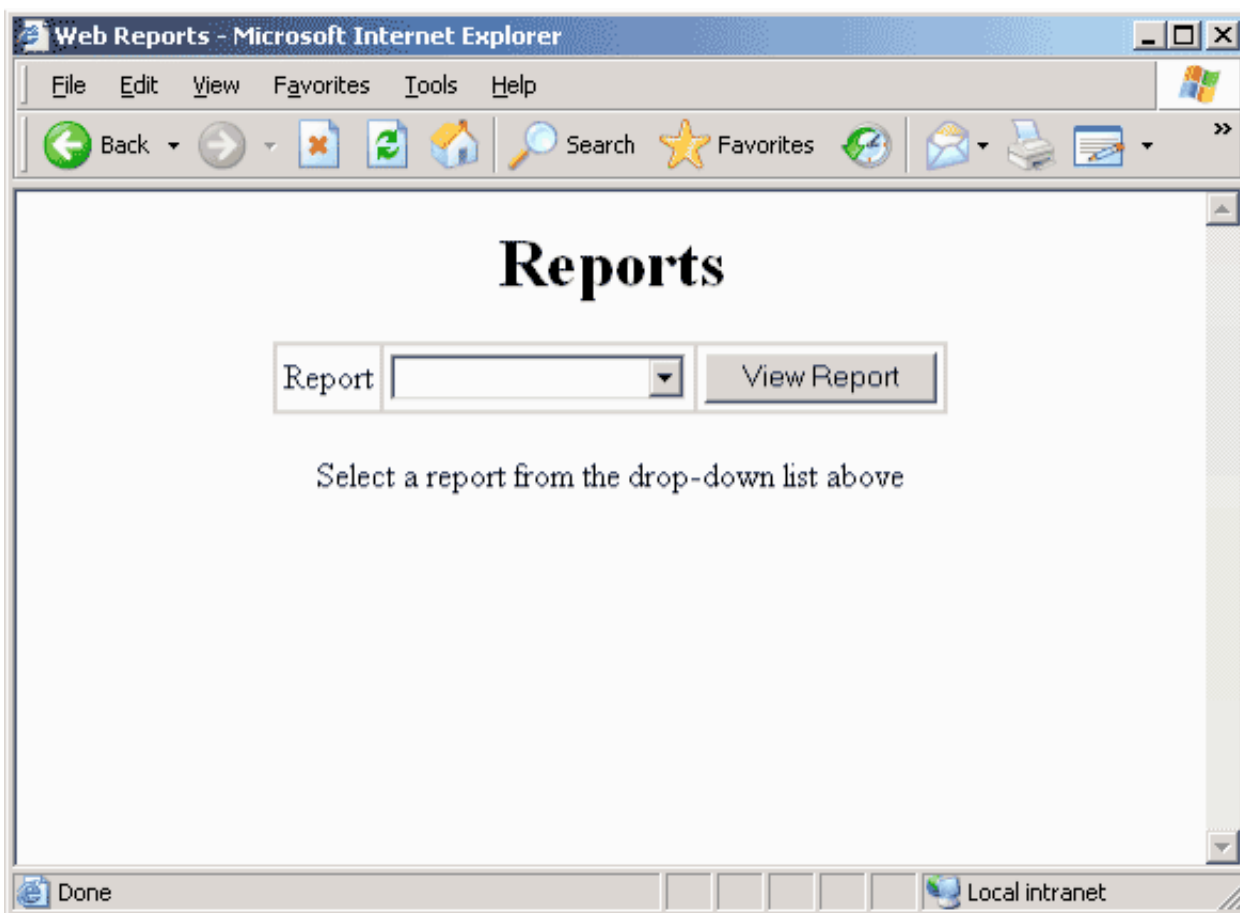


Figure 1: Reports.aspx

When they initially come to the page, no report has been selected, therefore:

- The report list box shows blank

- The message "Select a report from the list box above" is shown

If the user then selects "List of Customers" from the report list box a screen similar to Figure 2 is generated:

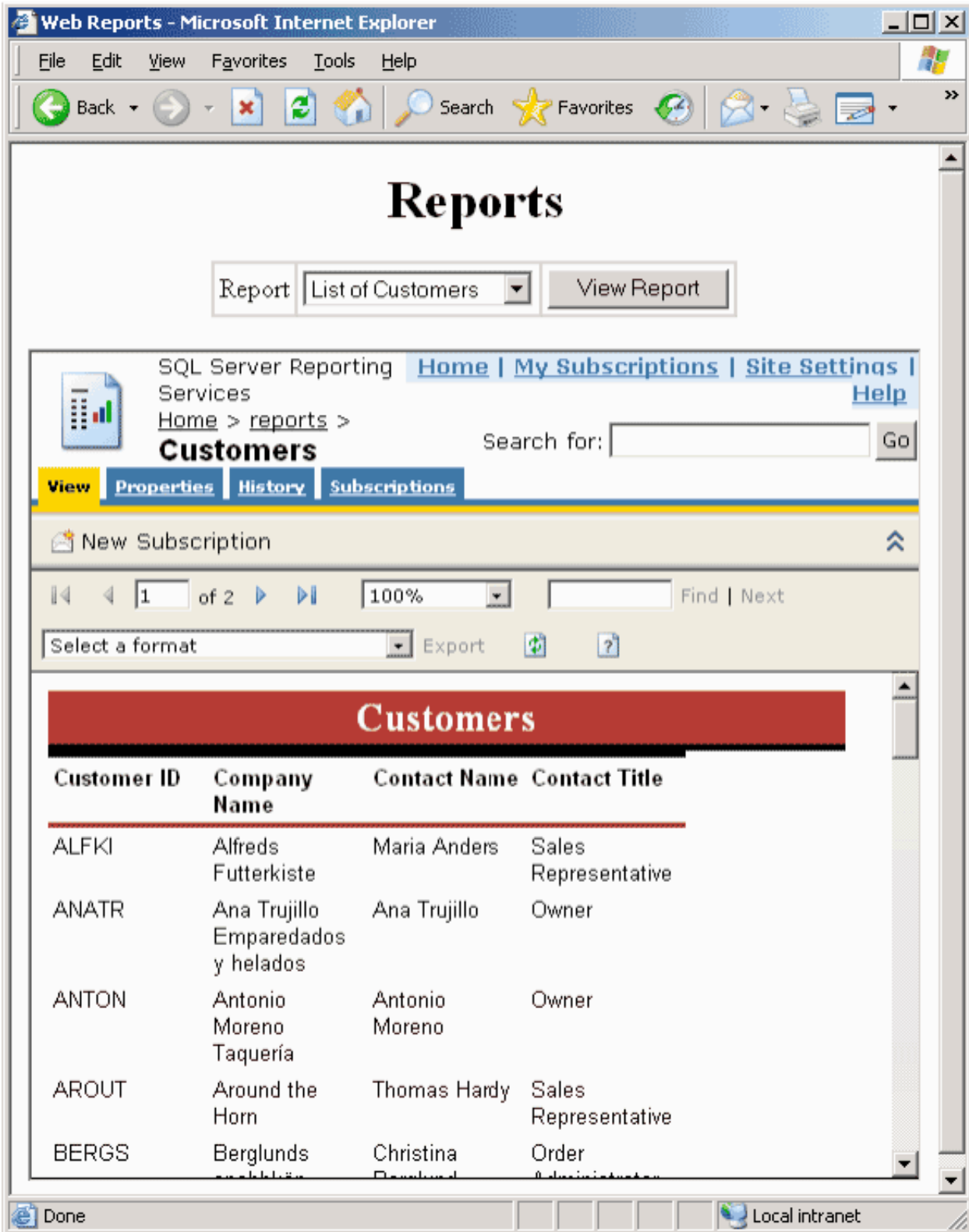


Figure 2: Customer report (data from Northwind.mdb)

The report selection list box stays at the top. The bottom part of the page (which contains an IFRAME) has its source changed to reflect the report that was selected. Even though the web page is served up from one web server, and the Reporting Services that delivers the report resides on a different server, it is all transparent to the user.

The Code

I used C# .NET, but of course any scripting language could be used. The two main dynamic functions the page must be able to do are:

- Receive a parameter (e.g. Request.QueryString)
- Set the IFRAME source based on the value of the received parameter

```
<script language="c#" runat="server">
void View_Report(Object sender, EventArgs e) {
    Response.Redirect("reports.aspx?rpt=" + rpt.SelectedItem.Text);
}

void Page_Load(Object sender, EventArgs e) {
    string sReport="";
    sReport = Request.QueryString["rpt"];

    int i= 0;
    foreach(ListItem Lst in rpt.Items) {
        if(Lst.Text.Equals(sReport)) {
            rpt.SelectedIndex = i;
        }
        i++;
    }
}
</script>
```

Listing 1

Listing 1 shows that when the page is loaded, the list box of reports is iterated through, looking for a match between a list box item and what is received by Request.QueryString. If there's a match, the list box will reflect the user's report selection.

```
<asp:listbox ID="rpt" Rows="1" SelectionMode="Single" runat="server">
    <asp:ListItem />
    <asp:ListItem Value="List of Customers"/>
    <asp:ListItem Value="Current Employees"/>
</asp:listbox>
```

Listing 2

Listing 2 is the code for the list box. Nothing special here.

```
<%
string sReport = "";
switch(Request.QueryString["rpt"]) {
    case "List of Customers":
        sReport="Customers";
        break;
    case "Current Employees":
        sReport="Employees";
        break;
    default:
        Response.Write("<P class='note'>Select a report from the drop-down list
above</P>");
}
```

```
        break;
    }

    string sReportPath = "http://myreportserver/Reports/Pages/Report.aspx?
ItemPath=%2freports%2f";

    if(!sReport.Equals("")) {
        %>
        <iframe src="<%=sReportPath%><%=sReport%>" height="500" width="100%" scrolling="no">
        </iframe>
        <%
    }
%>
```

Listing 3

Listing 3 is the meat of the page, where the source of the report IFRAME is dynamically set to reflect the user's report choice. The switch statement finds a match for the parameter the page receives (which is stored in the string variable rpt.) If the user selects "List of Customers", the value for the string variable sReport is set to "Customers." If no selection is made (such as when this web page is first hit) the switch statement falls through to default, and the IFRAME isn't displayed at all.

After sReport is set, the report path (sReportPath) is built. The variable sReportPath is initialized to the server path. Then, in the "if" statement that follows, if a report has been selected, the IFRAME source is set to to the report path (sReportPath) + Report Name (sReport.)

Conclusion

And that's about it, it's that simple. IFRAME is a great way for allowing parts of a web page to be set dynamically without needing a full page refresh. Just remember that whenever you want to add a report to the page:

- Add an item to the List Box
- Add a matching case statement to the switch statement

About the Author

Don Franke is a contractor currently residing in San Antonio, TX. He can be reached at donfranke@satx.rr.com.

[Back to article](#)