

Office Web Components in pagine ASP.NET

da <http://escher07.altervista.org>

Generalità

L'obiettivo è realizzare una pagina ASP.NET che utilizzando il componente Web Pivot permetta di ottenere un risultato come il seguente:

Analisi Documenti Contabili							
cod_anno_doc							
2007							
		Mesi					Totale complessivo
		feb	nov	gen	feb	mar	
cod_tipo_doc	Valore	Valore	Valore	Valore	Valore	Valore	
AFF				7571,4		159632,62	
IMP	30400	10128,1	234175,9		1000	286024,09	
Totale complessivo	30400	10128,1	241747,4		787,28	445656,71	

Buttons: Connect, Fill

Nel quale in una form compare un componente Tabella Pivot caricato con i dati di una tabella proveniente da un DB che permette di eseguire analisi OLAP (es. Drill Down) sui dati in questione.

Nella realizzazione finale i dati saranno prelevati da DB Oracle mediante WS. Qui presenteremo essenzialmente una soluzione esemplificativa in cui di dati sono presi direttamente da codice da un DB access, dando solo alcuni cenni relativamente al passaggio alla soluzione "da ambiente di produzione".

Implementazione

Il "vero" obiettivo per cui sono arrivato a scrivere queste considerazioni era realizzare in pratica il seguente schema di principio:

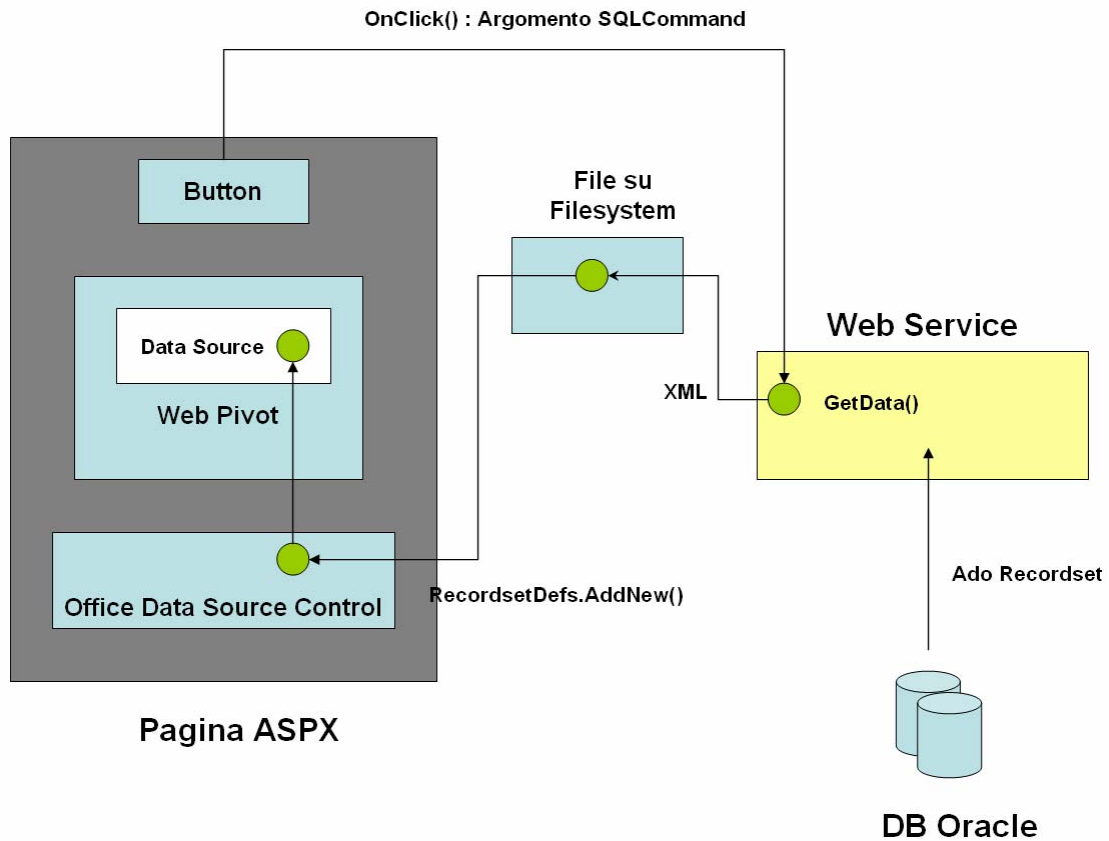


Figura 1

Molti dei concetti relativi a questo sono stati spiegati nella realizzazione in ambiente client/server, per cui in questo documento si ipotizzerà uno schema semplificato quale il seguente:

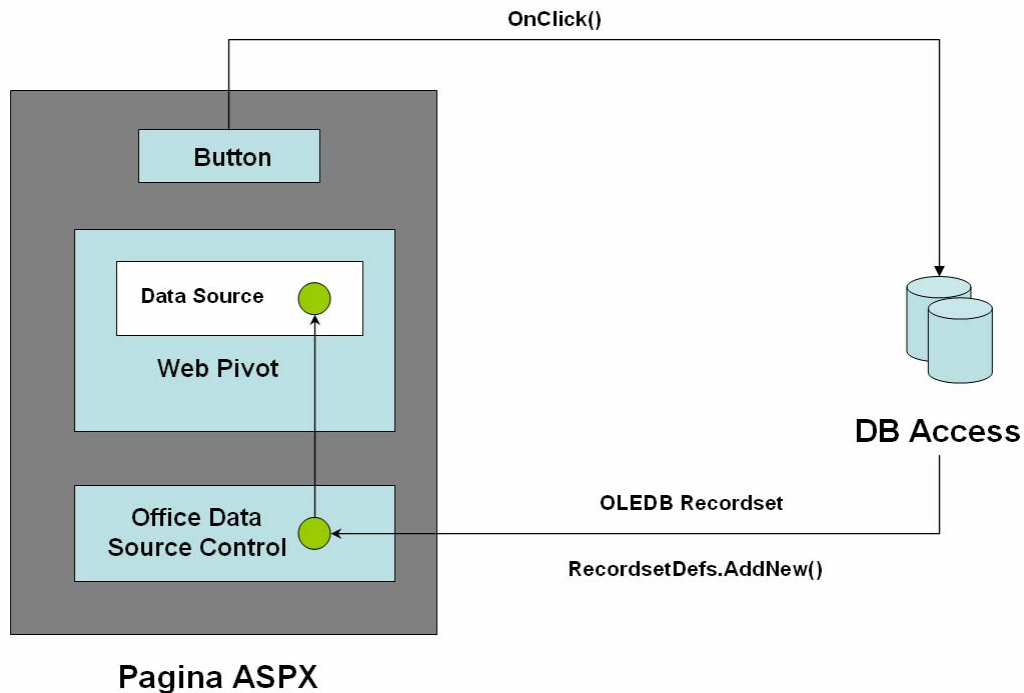


Figura 2

In cui l'accesso ai dati viene eseguito direttamente tramite provider di tipo OLEDB ad un DB Access.

Si tratterà di inserire due controlli di tipo Office Web Pivot e Microsoft Office Data Source in una pagina Web e poi scrivere del codice opportuno con cui venga prima popolato il DSC con i dati estratti dal DB e poi venga associata tale data source all'omonima proprietà della Tabella Pivot.

Codice e Aspetti Pratici (sullo schema di Figura 2)

In teoria il disegno precedente è molto "pulito" e semplice, in realtà la realizzazione pratica dà svariati problemi che ho risolto in sintesi per tentativi. Di sicuro la soluzione che qui illustro non è il top ma, rispetto ai tanti esempi che ho trovato in rete, è anche l'unica che si è rivelata funzionante.

Il primo problema è il riferimento ai nostri componenti Web Pivot e Data Source Control (oggetti COM) in una pagina .NET. La soluzione più "bella" mi sembrava quella di fare l'hosting in una pagina Web del controllo "Pivot" sviluppato in ambiente Windows Form (di cui si parla nel documento "[OWC e C#](#)") e gestirlo attraverso i suoi metodi e proprietà esposti. Probabilmente lo era ma non mi funzionava...

Altra soluzione possibile per fare riferimento ad un oggetto COM in ASP.Net è quello di usare il tag OBJECT ed il corrispondente CLSID. Ho optato per questa trovando in rete il giusto CLSID per la versione degli OWC che avevo in precedenza scaricato.

Caratteristica interessante degli OWC è che per scelta progettuale (ovvero essenzialmente motivi di sicurezza e possibilità di eseguire analisi anche se fisicamente disconnessi dal DB), sono oggetti che "girano" lato client. Il che vuol dire che la gestione di questi, ovvero

l'assegnazione dei valori alle varie proprietà (es. connection string etc..) non può avvenire nel code behind nella pagina ma necessariamente in script client side, scritti in Javascript o VBScript.

Mi era sembrata cosa logica utilizzare lo stesso linguaggio per entrambe le routine che di dovevano realizzare (caricamento DSC ed associazione DSC-Tabella Pivot) ma ho verificato che l'unica soluzione funzionante era quella con lo script per il DSC in VB e l'altro in JS!

Premesso tutto questo, diamo un'occhiata al codice. La pagina ASPX si presenta così:

```
<%@ Page Language="C#" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Esempio OWC</title>
  <object classid="clsid:0002E553-0000-0000-C000-000000000046"
id="DSC"></object>
</head>
<body>
  <form id="form1" runat="server">
    <div></div>
    <table style="width: 472px">
      <tr>
        <td>
        </td>
        <td style="width: 386px">
        </td>
        <td>
        </td>
      </tr>
      <tr>
        <td style="height: 343px">
        </td>
        <td style="width: 386px; height: 343px">
          <OBJECT classid="clsid:0002E552-0000-0000-C000-000000000046"
id="Pivot" name="Pivot"></object>
        </td>
        <td style="height: 343px">
        </td>
      </tr>
      <tr>
        <td>
        </td>
        <td style="width: 386px; text-align: center">
          <input id="Button2" type="button" value="Connect"
language="vbscript" onclick="Connect()" />
          <input id="Button1" type="button" value="Fill"
language="javascript" onclick="Fill()" style="width: 68px" /></td>
        <td>
        </td>
      </tr>
    </table>
    &nbsp;
  </form>
</body>
```

```

<script language="VBScript">

Function Connect()

    Dim sSQL, sConn
    sSQL = "SELECT * FROM QRY_DOCUMENTI"
    sConn = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
            "Data Source=c:\temp\db1.mdb"

    DSC.ConnectionString = sConn
    If DSC.RecordsetDefs.Count < 1 Then
        DSC.RecordsetDefs.AddNew sSQL, DSC.constants.dscCommandText,
"dscData"
    End If
End Function

</script>

<script language="javascript" type="text/javascript">

function Fill()
{
    var pivot;
    var text;
    pivot = document.getElementById("Pivot");
    dsc = document.getElementById("DSC");
    pivot.DataSource = dsc;
    pivot.DataMember = "dscData";
    pivot.DisplayFieldList = true;
}

</script>

</html>

```

Con la funzione Connect() si aggiunge al DSC il recordset prelevato dal DB con l'accesso OLEDB. Con la funzione Fill() si associa il DSC alla proprietà Data Source della Tabella Pivot, facendo in particolar modo all'unico recordset "dscData".

Codice e Aspetti Pratici sullo schema di Figura 1

Quando andremo al realizzare lo schema di Figura 1 credo che sia opportuno dividere il caricamento in tre fasi che saranno:

- Extract Data
- Connect
- Fill

Durante la prima i dati saranno estratti dal DB con il WS e posti in un file XML. Questa fase può essere molto lunga, specie se si interroga una tabella con molti dati. Per questo è opportuno anche valutare l'eventuale impatto delle regole di timeout del proxy su questa, se la navigazione avviene tramite proxy.

L'invocazione di un metodo di un web service può essere fatta in vari modi. Ad esempio con questo codice lato server (C#) :

```

private void ExtractData()
{
    localhost.WSced objWebMethod = new localhost.WSced();
    string strResult = "";
    string strDataFile = "";
    string strSQLSelect = "SELECT * FROM QRY_DOCUMENTI";
    strDataFile = "c:\\inetpub\\wwwroot\\mydata.xml";
    strResult = objWebMethod.GetData(strSQLSelect);
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.LoadXml(strResult);
    xmlDoc.Save(strDataFile);
}

```

...fatta l'ipotesi che sulla nostra macchina esista un WS che espone un metodo pubblico GetData() che riceve in input il comando SQL da passare e restituisce in formato XML i dati derivanti alla nostra interrogazione al DB (Oracle nel nostro caso).

Riguardo allo script di connessione si presenterà così:

```

<script language="VBScript">

    Function Connect()

        Dim sXML
        Dim sFile
        Dim sConn
        sFile = "c:\\inetpub\\wwwroot\\mydata.xml"
        sConn = "Provider=MSPersist.1"
        DSC.ConnectionString = sConn
        If DSC.RecordsetDefs.Count < 1 Then
            DSC.RecordsetDefs.AddNew sFile, DSC.constants.dscCommandFile,
"dscData"
        End If
    End Function

</script>

```

Come si vede la modalità del caricamento del DSC è passata da dscCommandText (ovvero da stringa SQL) a dscCommandFile (ovvero da file di dati, cioè il mydata.xml che abbiamo generato).

Lo script di riempimento sarà invece uguale al precedente.

Links

<http://aspnet.4guysfromrolla.com/articles/052604-1.aspx>

Hosting a Windows Control in a Web Form

<http://www.dbazine.com/sql/sql-articles/charran3>

Implementing Office Web Component Pivot Tables with ASP.NET

<http://support.microsoft.com/kb/315695/it>

Come utilizzare l'Office Web Components con i dati XML generati da un servizio Web XML creato utilizzando un servizio ASP.NET

http://forums.codecharge.com/posts.php?post_id=84527
XML, AJAX, Office Web Components (OWC) tips and tricks

[http://msdn2.microsoft.com/en-us/library/aa155723\(offic.10\).aspx#odc_chap4owc_intro#odc_chap4owc_intro](http://msdn2.microsoft.com/en-us/library/aa155723(offic.10).aspx#odc_chap4owc_intro#odc_chap4owc_intro)
Pagine disponibili in rete di un buon libro su OWC e ASP