

Data View Web Parts! Heaven or Hell?

در این مطلب يك مثال عملي استفاده از DVWP (Data View Web Part) آورده مي‌شود. ضمن این مثال نقاط قوت و ضعف آن و راه‌کارهایی برای استفاده‌ی مناسب‌تر بررسی می‌شود.

صورت مسئله

برای ثبت و بررسی اطلاعات کاربرگ‌ها سیستمی بر اساس شیرپوینت پیاده‌سازی شده است. کاربرگ‌ها برگه‌های ساده‌ای هستند که اشخاص کارکرد روزانه‌ی خود را بر اساس اینکه برای چه پروژه‌ای کار کرده‌اند، ثبت می‌کنند. به این ترتیب روی هر برگه‌ی کاربرگ يك کاربر برای يك تاریخ معین ساعت و پروژه‌ای را که روی آن کار کرده‌است را وارد می‌کند.

ایده‌ی اولیه این بود که پس از ساخته‌شدن این لیست ساده در شیرپوینت می‌شود با امکانات ساخت نماها (View) که در شیرپوینت ارائه می‌شود، انواع گزارش‌های مورد نظر را پوشش داد. اما در عمل بعد از اینکه سایت ساخته شد و به هنگام طراحی نماها مشکلاتی پیش آمد که عملاً امکان استفاده از نماها را زیر سوال برد. و نتیجه‌گیری شد که امکانات نماهای استاندارد شیرپوینت برای نیازمندی‌های این سایت کفایت نمی‌کند. زیر:

- امکان ساختن فیلترهای پیچیده آنچنانکه از نیازمندی‌ها این سیستم است در نماهای استاندارد وجود ندارد. برای مثال یکی از کلیدی‌ترین فیلترهای مورد نیاز فیلتر تاریخ است. اینکه به شود بر حسب نیاز يك کاربر اطلاعات کاربرگ‌ها را از يك تاریخ تا يك تاریخ دیگر فیلتر کرد. امکانات شیرپوینت برای فیلترهای سفارش کاربر، (آنچه با استفاده از کلید فیلتر در بالای صفحه انتخاب می‌شود) فقط به شکلی است که کاربر می‌تواند اطلاعات را برای يك تاریخ مشخص فیلتر کند.
- امکانات شیرپوینت برای گروه‌بندی (Group By) کافی نیست چرا که اولاً این گروه‌بندی توسط کاربر انتخاب نمی‌شود و بایستی در مرحله طراحی نما انجام شود و ثانیاً در صورتی که حجم اطلاعات زیاد باشد معمولاً گروه‌بندی و به خصوص جمع‌ها درست نمایش داده نمی‌شوند. برای مثال در صورتیکه تعداد کاربرگ‌های يك پروژه بیشتر از 100 تا باشد و شما نما را طوری تنظیم کرده باشید که حداکثر 100 رکورد نمایش داده شود، وقتی شیرپوینت جمع کارکرد در این گروه را نمایش می‌دهد فقط 100 رکورد را پردازش می‌کند، که طبیعتاً بسیار مشکل ساز خواهد بود. از آنجاییکه طبیعت سیستم کاربرگ به شکلی است که گزارشات آن معمولاً به شکل گروه‌بندی شده می‌تواند مفید باشد طبیعی است این مشکلات به شکلی جدی توانایی‌های سیستم را محدود خواهند کرد.
- با بروز این اشکالات استفاده از DVWP و مزایای آن مورد بررسی قرار گرفت. مزایای عمده عبارتند از:
- این وب‌پارت، همانطور که بعداً به آن اشاره شد می‌تواند امکانات بسیار مناسب‌تری برای فیلتر کردن و همچنین گروه‌بندی اطلاعات را در اختیار بگذارد.
- این وب‌پارت امکانات بسیار مناسب‌تری برای نمایش و Rendering اطلاعات دارد.
- این وب‌پارت روش‌های بسیار مناسب‌تری برای سفارشی کردن و کنترل در زمان اجرا (Run Time) دارد.
- از این وب‌پارت می‌شود در سایت‌های دیگر استفاده کرد. بطور مشخص می‌شود اطلاعات لیستی از يك سایت را در سایت دیگری نمایش داد.

بر اساس این مزایا استفاده از این وب‌پارت مورد بررسی و پیاده‌سازی قرار گرفت. در حین کار البته مزایای دیگر و همچنین معایب آن مشخص شد که در محل خود به آنها خواهیم پرداخت.

DVWP چگونه کار می‌کند؟ و مزایای این روش کار

- تکنیک مورد استفاده در DVWP بسیار جالب است:
- "DVWP به Data Service هایی که خروجی XML تولید می‌کنند وصل می‌شود و خروجی این سرویس‌ها را توسط XSL که توسط کاربر معرفی می‌شود به صورت HTML در می‌آورد."
- این تکنیک باعث می‌شود که:
1. از آنجاییکه Data Service ها می‌توانند Query ها پیچیده‌ای را پردازش کنند، می‌شود در DVWP از فیلترهای پیچیده‌تری (نسبت به نماهای شیروینت) استفاده کرد.
 2. استفاده از روش استاندارد XSL امکان تغییر در شکل ظاهری و تولید گزارش‌های سفارشی و خاص را به نحو موثری فراهم می‌سازد.
 3. از آنجاییکه هم معرفی Query و هم خود XSL از نوع فایل‌های XML می‌باشد، می‌شود به راحتی و بدون نیاز به ابزارهای پیچیده و خاص (نظیر Report Writer ها و یا رابط کاربر معرفی نما در شیروینت) آنها را به سادگی تغییر داد. همچنین انتقال يك DVWP از يك سایت به سایت دیگر و یا از يك صفحه به يك صفحه‌ی دیگر براحتی امکان پذیر خواهد بود.
- علاوه بر موارد فوق ظرایف دیگری هم در DVWP هست که کاربردهای آنرا به شدت افزایش می‌دهد:
- عملیات تبدیل (XSL Transformation) در DVWP بر روی سرور صورت می‌گیرد. این مسئله به شکل مطلوبی نگرانی‌های انتقال حجم بزرگی از اطلاعات را از سرور به کلاینت‌ها بر طرف می‌سازد.
 - DVWP روش جالبی برای برقراری پارامترهای مورد نیاز با کنترل‌های سمت کلاینت معرفی می‌کند. این روش برای مثال این امکان را فراهم می‌سازد که مثلاً يك پارامتر که در Query استفاده می‌شود را به يك کنترل روی صفحه‌ی HTML وصل کرد. به این ترتیب می‌شود فیلترهای پویا ایجاد کرد. این را در مثال عملی خواهیم دید.
- استفاده از OM برای دسترسی به وب‌پارته‌ها هم ممکن است امکانات دیگری را فراهم سازد که نگارنده در این زمینه تجربه‌ی ندارد.

استفاده از DVWP در FrontPage

يك نکته‌ی بسیار مهم در DVWP امکانات حیرت‌آور FrontPage در ارائه‌ی يك رابط کاربر مناسب برای ایجاد و ویرایش DVWP هست. این امکان باعث می‌شود که کاربری که اطلاعاتی از XSL و یا DSP ندارد بتواند تا حد زیادی از امکانات DVWP ها بهره‌مند شود. در اینجا در واقع FrontPage يك جور الگوی عمومی و آماده برای XSL دارد که رابط کاربر می‌تواند به شکل مناسبی بر اساس انتخاب‌های کاربر آنرا تغییر دهد.

عدول از امکانات FrontPage

اگرچه الگوی آماده‌ی FrontPage برای عموم کاربران می‌تواند مناسب باشد اما کاربر حرفه‌ای و آشنا به XSL نهایتاً نمی‌تواند از دستکاری‌های حرفه‌ای خود برای حصول به نتیجه‌ی بهتر خودداری کند. این دستکاری‌ها نهایتاً منجر به این خواهد شد که الگوی FrontPage آنقدر تغییر کند که دیگر توسط FrontPage قابل شناسایی نباشد. در این وضعیت رابط FrontPage برای ویرایش DVWP از کار می‌افتد و کار بر حرفه‌ای تنها با

دستکاری مستقیم در کد می‌تواند کار خود را ادامه دهد. هر که طاووس خواهد جور هندوستان کشد!

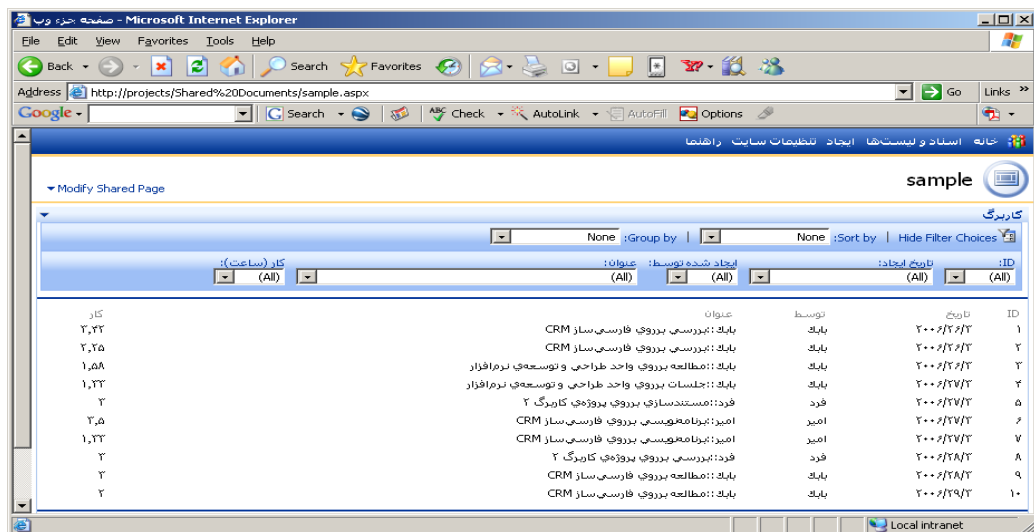
مثال عملی

در این مثال امکانات DVWP برای گزارش‌گیری از روی لیست کاربرگ مورد توجه قرار گرفته است. کارمندان سازمان کارکرد روزانه خود بر روی پروژه‌های مختلف را در لیست کاربرگ ثبت می‌کنند. برای سهولت در این مثال فرض می‌شود لیست کاربرگ يك لیست ساده‌ی شیرپوینت است که ستون‌های زیر را دارد:

1. تاریخ : تاریخ کاربرگ
2. مرکز هزینه: مرکز هزینه‌ای که کارکرد روی آن شارژ می‌شود.
3. سرویس: نوع کارکرد و یا سرویسی که شخص ارائه کرده است.
4. توسط: نام شخص
5. دپارتمان: واحد سازمانی شخص
6. کار : ساعت کار
7. عنوان : عنوان کار انجام شده

برای اینکار به سرعت کار شروع می‌شود. ابتدا يك Web Part Paper در شیرپوینت ایجاد می‌کنیم. سپس آنرا به FrontPage برای ویرایش منتقل می‌کنیم. در FrontPage با يك DVWP ایجاد می‌شود. این DVWP به لیست کاربرگ‌ها متصل شده فیلدهای مناسب را برای نمایش انتخاب می‌کنیم. سپس با استفاده از گزینه‌های DVWP نمایش فیلتر و Toolbar را فعال می‌کنیم. به این ترتیب به سرعت به چیزی شبیه آنچه در شکل زیر نمایش داده شده است می‌رسیم:

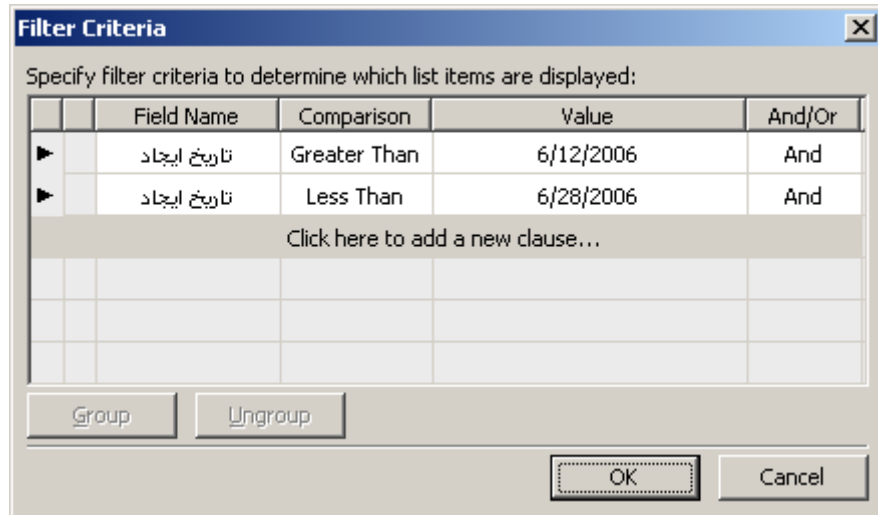
روی این صفحه کاربر می‌تواند براحتی از طریق کنترل‌ها مربوط به فیلتر و سورت و



گروه‌بندی به اطلاعات مورد نظر خود دسترسی پیدا کند.

فیلتر از تاریخ تا تاریخ

امکانات فیلتری که به صورت پی‌فرض در DVWP ارائه می‌شود، بعضاً کارآمد نیست. برای مثال در این مورد بخصوص کاربر علاقه‌مند است کاربرگ‌ها در يك محدوده‌ی تاریخی بررسی کند و بنابراین فیلتری به شکل از تاریخ تا تاریخ لازم است. این فیلتر به هیچ‌وجه توسط امکانات آماده‌ی DVWP قابل ارائه نیست. البته مفهوم این فقط آنست که اینکار از طریق انتخاب‌ها ئ رابط کاربر FrontPage امکان پذیر نیست ولی، با دقت در Query متوجه می‌شویم این Query می‌تواند فیلترهای پیچیده‌ای داشته باشد. برای مثال نگاه کنید به این پنجره در تنظیم فیلتر :



که در آن فیلتر مناسبی برای تاریخ بین دو مقدار مشخص تعریف شده است. مشکل آنست که این مقادیر معین بایستی توسط کاربر معرفی شوند. و چالش اصلی در واقع برقراری ارتباط بین این مقادیر و کنترل‌های مناسب بر روی صفحه‌ی نمایش است. در میان انتخاب‌هایی که برای مقدار Value در این پنجره هست، یکی هم این [INPUT PARAMETER] است. اگر این مقدار انتخاب شود با مراجعه به کد مربوط به DVWP این قسمت برای فیلتر دیده می‌شود:

```
<dsp:Where xmlns:dsp="http://schemas.microsoft.com/sharepoint/dsp">
<dsp:And>
<dsp:Gt>
<dsp:FieldRef Name="Created"/>
<dsp:Value Type="xsd:string"><udc:ClientParameterValue
xmlns:udc="http://schemas.microsoft.com/data/udc" Name="filterParam"/></dsp:Value>
</dsp:Gt>
<dsp:Lt>
<dsp:FieldRef Name="Created"/>
<dsp:Value Type="xsd:string"><udc:ClientParameterValue xmlns:udc="http://schemas.microsoft.com/data/udc"
Name="filterParam"/></dsp:Value>
</dsp:Lt>
</dsp:And>
</dsp:Where>
```

از ظاهر این کد اینطور استنباط می‌شود که DVWP می‌تواند پارامترهای مربوط به Query را به یک نوع پارامتر که در این مثال با نام filterParam به آن اشاره شده است متصل کند. در واقع چند سطر پایین‌تر در کد همین صفحه می‌بینیم که:

```
<ParameterBindings xmlns="http://schemas.microsoft.com/office/fpddw">
<ParameterBinding Name="dvt_adhocmode" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_adhocfiltermode" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_fieldsort" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_sortfield" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_groupfield" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_groupdisplay" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_sortdir" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_groupdir" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_grouptype" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_sorttype" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_groupsorttype" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_filterfield" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_filterval" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_filtertype" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_firstrow" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_nextpagedata" Location="Postback;Connection"/>
<ParameterBinding Name="dvt_apos" Location="Postback;Connection"/>
<ParameterBinding Name="filterParam" Location="Postback;Connection"/>
<ParameterBinding Name="UserID" Location="CAMLVariable" DefaultValue="CurrentUserName"/>
<ParameterBinding Name="Today" Location="CAMLVariable" DefaultValue="CurrentDate"/>
```

```
</ParameterBindings>]]></dvw:ParamBindings>
```

مایکروسافت لطف کرده است و توضیحی در مورد این بخش از DVWP ارائه کرده است:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odc_fp2003_ta/html/odc_fpParameterBinding.asp

به نظر می‌رسد اشکالی نداشته‌باشد که ما هم پارامترهای خودمان را به این بخش اضافه کنیم:

```
<ParameterBinding Name="filterDate1" Location="PostBack;Connection" Default="2007-01-01">
<ParameterBinding Name="filterDate2" Location="PostBack;Connection" Default="2006-01-01"/>
```

البته ظاهراً برای این کار یک بخش دیگری هم به شکل زیر هست که بایستی تصحیح شود چند خط بالا تر می‌بینیم:

```
<udc:ClientParameterBindings>
  <udc:ClientParameterBinding Name="dvt_sortfield" Location="PostBack;Connection" Item="dvt_sortfield"/>
  <udc:ClientParameterBinding Name="dvt_sortdir" Location="PostBack;Connection" Item="dvt_sortdir"
  DefaultValue="Asc"/>
  <udc:ClientParameterBinding Name="dvt_groupfield" Location="PostBack;Connection" Item="dvt_groupfield"/>
  <udc:ClientParameterBinding Name="dvt_groupdir" Location="PostBack;Connection" Item="dvt_groupdir"
  DefaultValue="Asc"/>
  <udc:ClientParameterBinding Name="filterParam" Location="PostBack;Connection" DefaultValue="" />
</udc:ClientParameterBindings>
```

یعنی بایستی پارامترهای خودمان را اینجا هم اضافه کنیم:

```
<udc:ClientParameterBinding Name="filterDate1" Location="PostBack;Connection" DefaultValue="" />
<udc:ClientParameterBinding Name="filterDate2" Location="PostBack;Connection" DefaultValue="" />
```

و بعد برگردیم به قسمت Query و آنرا اینطوری درست کنیم:

```
<dsp:Where xmlns:dsp="http://schemas.microsoft.com/sharepoint/dsp">
<dsp:And>
<dsp:Gt>
<dsp:FieldRef Name="Created"/>
<dsp:Value Type="xsd:string"><udc:ClientParameterValue xmlns:udc="http://schemas.microsoft.com/data/udc"
Name="filterDate1"/></dsp:Value>
</dsp:Gt>
<dsp:Lt>
<dsp:FieldRef Name="Created"/>
<dsp:Value Type="xsd:string"><udc:ClientParameterValue xmlns:udc="http://schemas.microsoft.com/data/udc"
Name="filterDate2"/></dsp:Value>
</dsp:Lt>
</dsp:And>
</dsp:Where>
```

کد تولید شده را آزمایش می‌کنیم و می‌بینیم با توجه به تاریخ‌هایی که به عنوان Default وارد می‌کنیم و در کمال تعجب همه چیز کار می‌کند. خب زیاد خوشحال نشویم با این همه مشقت ما هنوز مشکل را حل نکرده‌ایم! مشکل این بود که کاربر بتواند این مقادیر را بسادگی معرفی کند. در همان مقاله‌ی کذایی در MSDN یک روش ساده برای اینکار پیشنهاد شده است، استفاده از QueryString و مشخص کردن فیلتر در URL برای اینکار می‌شود، پارامترها را اینطور تعریف کرد:

```
<ParameterBinding Name="filterDate1" Location="PostBack;Connection;QueryString(Date1)" Default="2007-01-01"/>
```

که به این ترتیب می‌شود با مراجعه به این صفحه به این شکل:

```
http://mypage.aspx?Date1="2006-01-01"
```

فیلتر را از طریق URL دریافت کرد. به این ترتیب اگر کنترلی روی صفحه گذاشته می‌شود، با انتخاب تاریخ در این کنترل صفحه بایستی به URL جدید منتقل شود. این روش میتواند پاسخگویی نیاز ما باشد. ولی وقتی به دقت روش داخلی DVWP را به هنگام تنظیم فیلتر از طریق رابط معمولی آن بررسی می‌کنیم متوجه می‌شویم که در اینجا روش دیگری مورد استفاده قرار گرفته است.

این روش در واقع استفاده از نوعی تابع PostBack است. که مبتنی بر روش کار متداول Net. برای PostBack به نظر می‌رسد. نگارنده آشنایی زیادی با این روش کار

ندارد. اما با بررسی کدهای Client Side تولید شده در صفحه متوجه می‌شویم که با PostBack کردن پارامترها به شکلی شبیه:

```
...Parametername={ParameterValue}...
```

می‌شود پارامترها را برای سرور فرستاد. به این ترتیب و با کمی کنجکاوی در می‌یابیم که سرویسی در DDWRT (بعداً این اسم را توضیح می‌دهم) برای تولید کدهای مورد نیاز وجود دارد. در واقع در بخشی از کد صفحه عباراتی شبیه اینرا پیدا می‌کنیم:

```
<xsl:value-of select="ddwrt:GenFireServerEvent('dvt_firstrow={1}')"/>
```

که در واقع کدی شبیه این تولید می‌کند:

```
__doPostBack(... dvt_firstRow={value})
```

با شبیه سازی این کد برای کنترل‌های مربوط به تاریخ در مثال خودمان می‌توانیم به نتیجه‌ی مطلوب برسیم:

```
<xsl:variable
name="clvar_filter_date_1">'+document.getElementById("_filterDate_1").TZdateValue+'</xsl:variable>
<xsl:variable
name="clvar_filter_date_2">'+document.getElementById("_filterDate_2").TZdateValue+'</xsl:variable>
<xsl:variable name="onDateChagedText">
javascript:
//alert("Babak");
//var val1= document.getElementById("_filterDate_1").TZdateValue;
//alert(val1);
<xsl:value-of select="ddwrt:GenFireServerEvent(concat(
                                'NotUTF8;filter_Date_1={',
                                $clvar_filter_date_1,
                                '}',
                                'filter_Date_2={',
                                $clvar_filter_date_2,
                                '}',
                                )
                                )"/>
</xsl:variable>
```

کمی پیچیده به نظر می‌رسد. ولی خب دنیای Client Side است و... به علت عدم آشنایی نگارنده با مزایا و معایب استفاده از این دو روش (PostBak و URL) در اینجا نمی‌توانک توصیه‌ای برای استفاده داشته‌باشم. فقط يك نکته است که مرا بیشتر به استفاده از URL ترغیب می‌کند. و آن اینست که کاربر نهایی می‌تواند با تنظیم HyperLink‌هایی از صفحات دیگر، مقادیر پارامترها را به هنگام باز شدن صفحه کنترل کند. برای مثال می‌تواند لینکی ایجاد کند به "کاربرگ‌های این ماه من"

```
http://Report.aspx?filterDate1="2006-01-01";filterDate2="2006-01-30"&User=Me
```

يك نکته مهم درباره‌ي فیلتر پیش فرض

فیلتر کردن خروجی اساساً می‌تواند در دو قسمت انجام شود:

1. در دستور Query

2. به هنگام XSL TRANSFORM یعنی با تنظیم SELECT در XSL.

فیلتری که DVWP به عنوان پیش‌فرض از آن استفاده می‌کند از نوع دوم است. یعنی وقتی کاربر فیلتر را انتخاب می‌کند، در واقع Query فرستاده شده برای Database تغییر نمی‌کند بلکه آن دسته از رکوردهایی که به هنگام ترانسفورمیشن انتخاب می‌شوند فیلتر خواهد شد. در حالیکه فیلتری که در مثال بالا به آن اشاره شد عملاً در Query برای Database اعمال می‌شود. دقت در این موضوع و انتخاب روش مناسب در لیست‌های بزرگ مهم خواهد شد. برای مثال دقت کنید که DVWP برای پر کردن لیست‌های کنترل‌های انتخاب فیلتر (Drop Down Lists) مجبور است تمامی رکوردهای گرفته شده از Database را بررسی کند. اینکار ممکن است بسیار وقت‌گیر باشد. در مثال ما با توجه به آنکه لیست "مراکز هزینه" از طریق آسان‌تری قابل دسترسی است، نگارنده برای بهبود سرعت مجبور شد این روش پر کردن لیست‌ها را کنار بگذارد و از يك Web Service استفاده کند.

گروه‌بندی

گروه‌بندی (Group By) یکی از امکانات مهم DVWP است. در حالیکه نماهای استاندارد شیرپوینت گروه‌بندی را فقط به هنگام طراحی نما امکان‌پذیر ارائه می‌کند در DVWP گروه‌بندی در حال اجرا و توسط کاربر معرفی می‌شود. به این ترتیب لزوم طراحی نماهای مختلف در مواقعی که گروه‌بندی‌های مختلفی ممکن است مورد نیاز باشد عملاً از بین می‌رود.

گروه‌بندی در DVWP تقریباً به شکل پیچیده‌ای در کدهای XSL صورت می‌گیرد:

```
<xsl:variable name="NewGroup_0"><xsl:choose xmlns:xsl="http://www.w3.org/1999/XSL/Transform"><xsl:when test="$dvt_groupfield_1"><xsl:value-of select="ddwrt:NameChanged(string(*[name()=$dvt_groupfield_1] | @*[name()=$dvt_groupfield_1] | current()[name(.) = $dvt_groupfield_1]), 0)"/></xsl:when><xsl:otherwise></xsl:otherwise></xsl:choose></xsl:variable>
<xsl:variable name="NewGroup_1"><xsl:choose xmlns:xsl="http://www.w3.org/1999/XSL/Transform"><xsl:when test="$dvt_groupfield_2"><xsl:value-of select="ddwrt:NameChanged(string(*[name()=$dvt_groupfield_2] | @*[name()=$dvt_groupfield_2] | current()[name(.) = $dvt_groupfield_2]), 1)"/></xsl:when><xsl:otherwise></xsl:otherwise></xsl:choose></xsl:variable>
<xsl:choose>
  <xsl:when test="0"/>
  <xsl:when test="($dvt_groupfield_1) and (not($NewGroup_0=") and position() >= $FirstRow and position() <= $LastRow or ($FirstRow = position()))">
    <xsl:if test="not ((position()=1) or (position()=$FirstRow)) and ($dvt_groupfield_2)">
      <xsl:call-template name="dvt_1.groupfooter1">
        <xsl:with-param name="fieldtitle">مرکز هزینه</xsl:with-param>
      </xsl:call-template>
    </xsl:if>
  </xsl:when>
</xsl:choose>
```

همانطور که مشخص است از کدهای نسبتاً پیچیده‌ای برای پشتیبانی از گروه‌بندی در الگوی آماده DVWP استفاده می‌شود.

DVWP اصولاً به دو شکل مختلف گروه‌بندی را ارائه می‌کند. در یک روش گروه‌بندی بر اساس انتخاب کاربر و از طریق کنترل انتخابی بر روی نوار ابزار انجام می‌شود. در شکل دیگر گروه‌بندی به هنگام تنظیم Query انتخاب می‌شود. این دو روش به صورت همزمان هم قابل استفاده هستند ولی در این صورت گروه‌بندی تنظیم شده در Query به صورت شکل اولیه نمایش ارائه می‌شود و با انتخاب گروه‌بندی توسط کاربر عملاً جایگزین می‌شود.

به خاطر پیچیدگی کدهای XSL در گروه‌بندی عملاً هر گونه عدول از روش‌های پیش‌فرض کار را خیلی پیچیده می‌کند. در مثال ما مثلاً اضافه کردن یک گروه‌بندی اضافه به نوار ابزار استاندارد عملاً خیلی وقت‌گیر شد.

نتیجه

استفاده از DVWP می‌تواند کاربردهای زیادی در شیرپوینت داشته باشد. به خصوص این ابزار می‌تواند در Integration و برقراری ارتباط‌های گزارش‌گیری از بانک‌های اطلاعاتی بسیار سودمند باشد. تکنولوژی مورد استفاده در DVWP آنرا به ابزاری قدرتمند و در عین حال با انعطاف‌پذیری بالایی تبدیل می‌کند. این مطلب تنها مقدمه‌ای بر این ابزار است و هنوز نکات زیاد دیگری در مورد آن قابل بررسی است.