Building a Dashboard in SQL Server Reporting Services

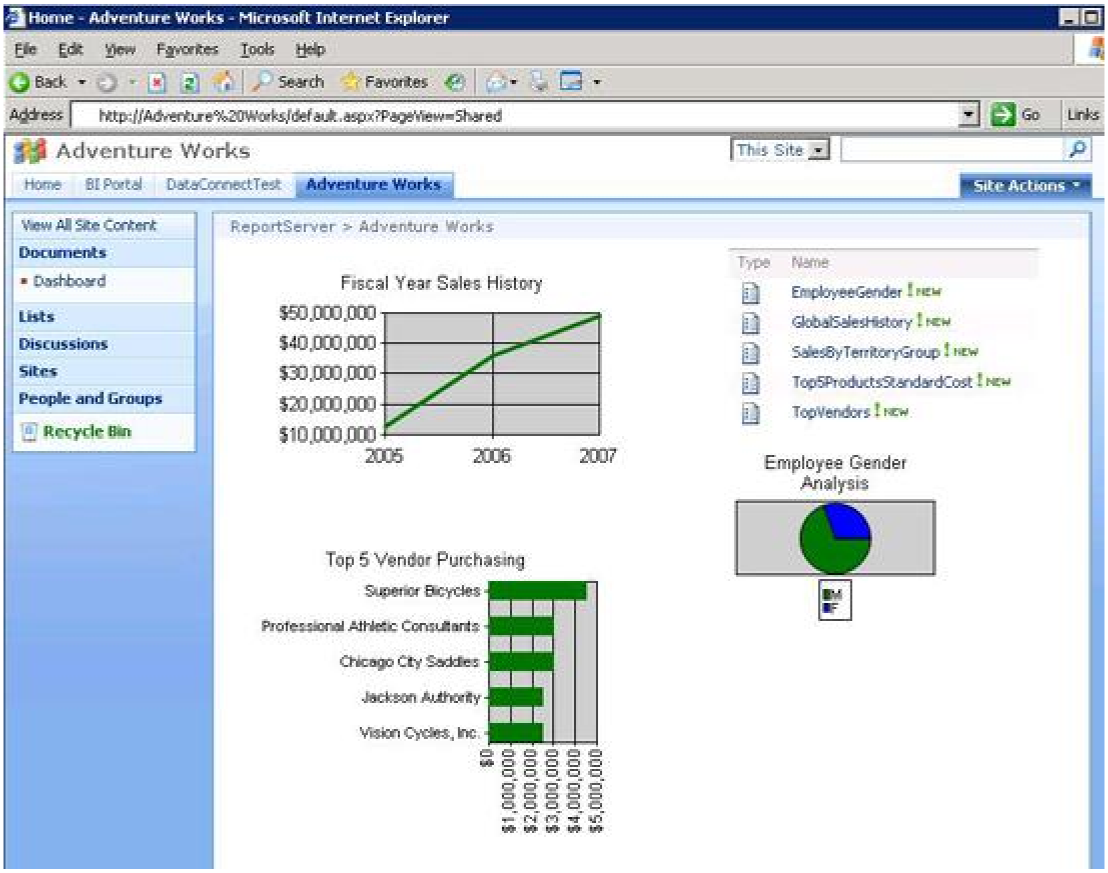
Problem

Your data warehouse is rock solid. You have multiple sets of fantastic reports. Departments utilize their own designated reporting areas, and you've increased productivity and value (ROI on the BI program) by focusing operations on critical areas by offering specialized filters and sorting as well as exception reports.

But top level executive management still doesn't have a clear picture of what direction the company is going in and can't make the strategic decisions needed to plan into the future. Your voluminous sets of reports look to them like the local telephone directory. They just don't have the time to wade through every report in all the (wonderful) departmental reporting solutions you have created.

Solution

Time for a Dashboard. (Or several).



Before we begin, let's begin with a basic definition of what a dashboard is. I like this one...

"A digital dashboard, also known as an enterprise dashboard or executive dashboard, is a business management tool used to visually ascertain the status (or "health") of a business enterprise via key business indicators. Digital dashboards use visual, at-a-glance displays of data pulled from disparate business systems to provide warnings, action notices, next steps, and summaries of business conditions."

The Hard Part: Determining Content and Audience

The hard parts of building a dashboard are not technical at all. They are determining what key indicators should be presented and figuring out who the audience should be. For that, you need to work with the top level executives (perhaps through your boss and chain of command), on planning what these key indicators should be and who should be allowed to see them.

You may find you have to create one dashboard for executives and another for the rank and file (middle management and low level staff). If that is the case, then so be it. You just need to find out the score (the business requirements) and move on.

In order to determine content, you may have to ask leading questions like:

"If your BI portal could look like the front page of the NY Times, what would you like to see?"

"What are the top 10 metrics that are important to you as an individual and the Executive Management team?"

"For a top level overview, do you prefer charts and graphics, or are you strictly a numbers person?"

"Should we build one dashboard for top level executives and another for all personnel or should everyone see the same thing?"

Trust me on this, getting these answers will take longer than actually building your dashboard.

One more word of advice before we begin. Your dashboard is not your dashboard. It's the company's dashboard. As such, you need to find a high level business owner that will partner with you on it as you develop it. You need to show them your work at several points during the construction process, and make modifications when needed. Consider it a partnership.

The Tools

There are many commercial software products that provide dashboard solutions. I am sure, for the most part, they can add a great deal of value, and add a level of sophistication that is impressive. Consider them something to grow into. Even if you want to start there, you will want to try building your own mini-version, so you will more fully understand the features and functionality to look for in a commercial product. So regardless over whether you are going to "build or buy", this exercise will be useful for everyone.

I will make the assumption that you already have Microsoft SQL Server 2005 in some production worthy flavor (Standard or Enterprise). You also need an instance of Sharepoint running of one of the following: WSS2, WSS3, Sharepoint Portal 2003, MOSS 2007. Installation of Sharepoint and/or Reporting Services either in Sharepoint Integrated mode or

Standard (pre SP2) mode will be outside the scope of this article.

Checklist:
Microsoft SQL Server 2005 (with Reporting Services)
Sharepoint (WSS2, WSS3, Sharepoint Portal 2003, or MOSS 2007)

Now that we have that out of the way, we have review some the integration features between Reporting Services and Sharepoint. From this point forward I will only be referring to Sharepoint as WSS2 or WSS3. Everything I say about WSS2 also goes for Sharepoint Portal 2003. Equally, WSS3 can be equated with MOSS 2007 in this regard.

If you are using Reporting Services in Native Mode, then you will likely want/need to use Sharepoint WSS2 as your portal/dashbaord. If you are using Sharepoint Integrated mode, by it's very nature you will be using WSS3 (or MOSS 2007).

Is it possible to cross cross these, either entirely or partially? Well yes, especially if you are good with Sharepoint Designer. But that is outside the scope of this article. For this article we will assume that if you are not using Sharepoint Integrated mode of SQL Reporting Services, then you are using WSS2.

Is this a bad thing? No.

Is it worth upgrading? (Ah, the eternal question.) It depends. If you do not have Reporting Services in Sharepoint Integrated mode, then I suggest you at least try it on a test server or virtual machine. It's worth the experience. You need to go through that experience in order the answer the "Is it worth upgrading?" question for yourself.

Native Mode

You need to install the Reporting Services web parts in order to get the Report Explorer web part. I really have no use for the Report Viewer web part, but it gets installed at the same time.

The installation is a little tricky. Follow these instructions which are direct from SQL Server BOL:

1. Copy the RSWebParts.cab to a folder on the SharePoint server. The .cab is installed with Reporting Services. By default, it is located in the C:\Program Files\Microsoft SQL Server\90\Tools\Reporting Services\SharePoint folder. You can copy it to any folder on the SharePoint server, and then delete it later after you install the Web Parts.

2. On the computer that has the installation of the SharePoint product or technology, open a Command Prompt window and navigate to the folder that has the Stsadm.exe tool. The path will vary depending on which version of Windows SharePoint Services you are running. If you are using Windows SharePoint Services 3.0, the path is C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\BIN.

3. Run Stsadm.exe on the .cab, using the following syntax:

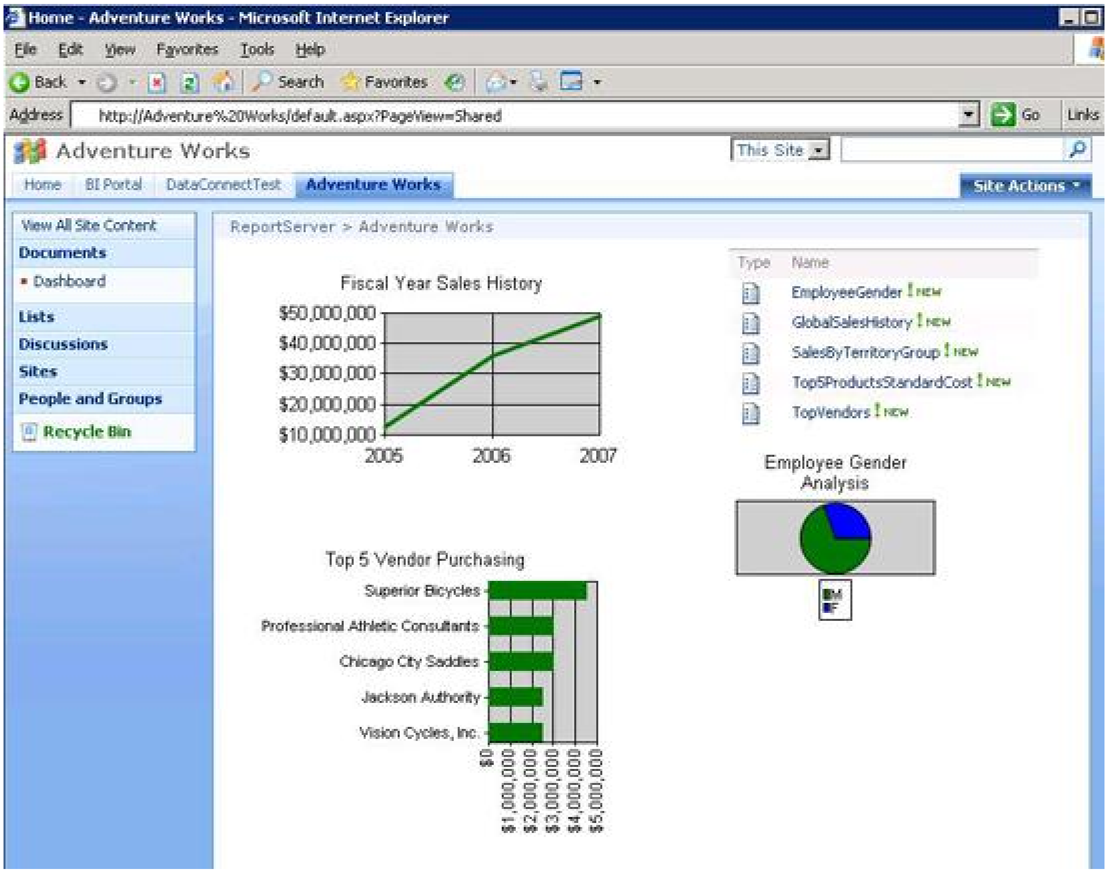
```
STSADM.EXE -o addwppack -filename "C:\ Program Files\Microsoft SQL Server\90\Tools\Reporting Services\SharePoint\RSWebParts.cab" -globalinstall
```

Sharepoint Integrated Mode

If you got through the Report Services installation in Integrated Mode, then you don't need to do anything else. I recently saw a very detailed (Word 2007) document on all the installation screens.

The Sharepoint Dashboard

Below is an example of a Dashboard I created against Microsoft's sample Adventure Works database.

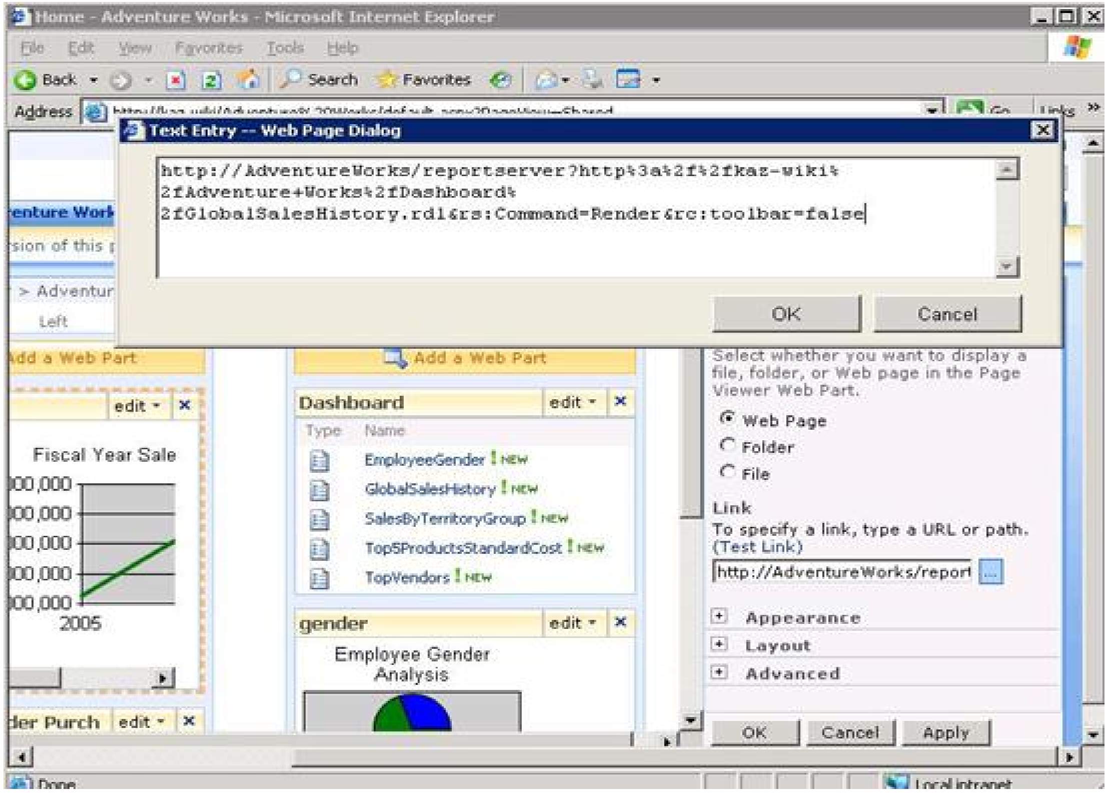


The top right report list is a scaled down view of the actual reporting library. The other three objects are actual Reporting Services charts run in real time.

They are displayed using standard Page Viewer Web Part in Sharepoint (not the special Reporting Services web part). I find this much easier and more direct than the Reporting Services web part.

What you don't see is the standard Report Server toolbar. That is because I have hidden it by appending the URL Access Parameter option after the Report Server URL path with the following syntax: &RC:toolbar=false

In order to use this, you must append it to the Report Server URL path (not the Report Manager or Sharepoint server path). The following is an example of that configuration.



Home - Adventure Works - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Text Entry -- Web Page Dialog

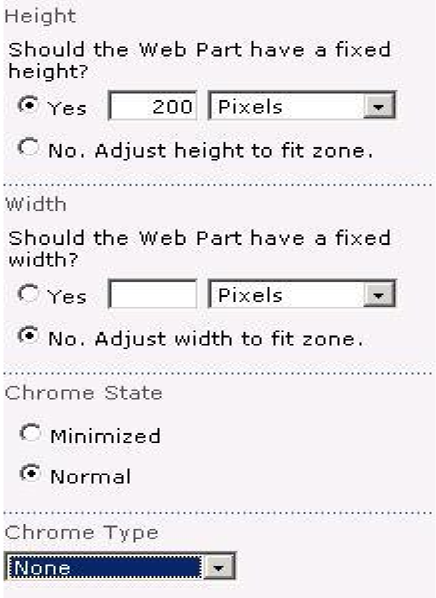
```
http://AdventureWorks/reportserver?http%3a%2f%2fkaz-wiki%
2fAdventure+Works%2fDashboard%
2fGlobalSalesHistory.rdl&rs:Command=Render&rc:toolbar=false
```

Select whether you want to display a file, folder, or Web page in the Page Viewer Web Part.

● Web Page
○ Folder
○ File

Link
To specify a link, type a URL or path. (Test Link)
http://AdventureWorks/report

The other options I used were in the Appearance section of the web part. You may need to adjust the default settings for height and width of the web page. Additionally, you will want set the "Chrome Type" to None.



Height

Should the Web Part have a fixed height?

● Yes 200 Pixels

○ No. Adjust height to fit zone.

Width

Should the Web Part have a fixed width?

○ Yes Pixels

● No. Adjust width to fit zone.

Chrome State

○ Minimized

● Normal

Chrome Type

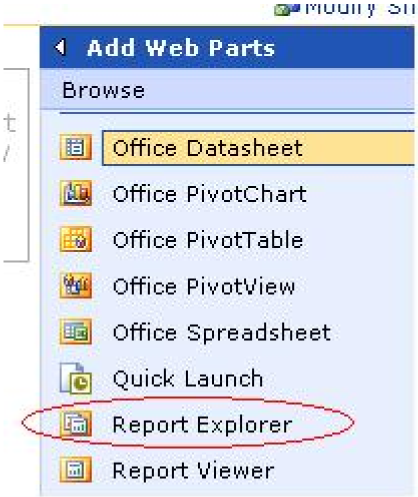
None

Native Mode

In a WSS2 Sharepoint dashboard you use the Page Viewer Web Page exactly the same way as in the Sharepoint Integrated Mode. Use the Page Viewer Web Part to display your charts and reports opened up.

In order to display the Reporting Services library, you cannot use a document library. Instead, you have to use the Reporting Services Report Explorer mentioned above (under "The Tools" section).

Drag the Report Explorer web part to one of the Web Part Zones.



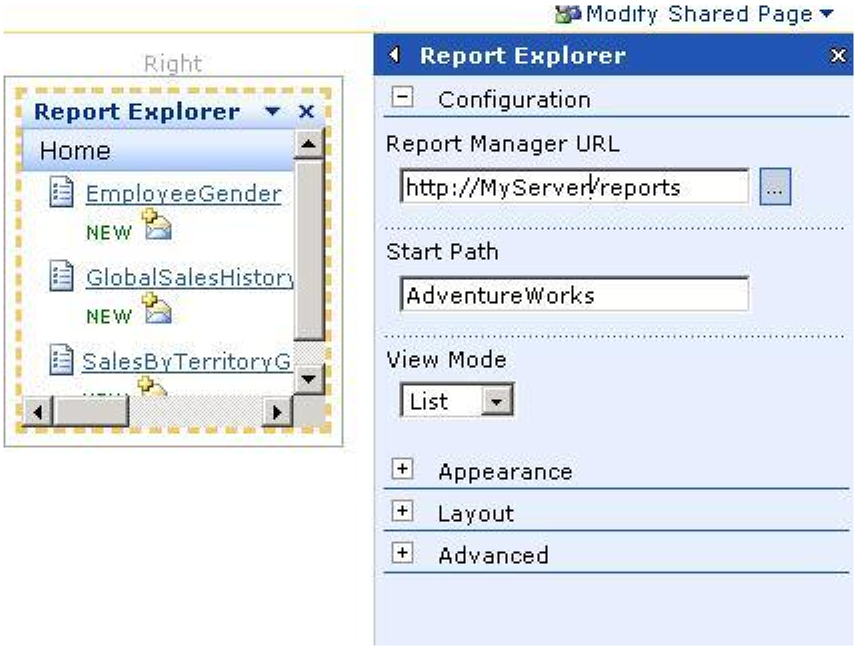
- In the configuration of the Report Explorer web part, enter the Report Manager URL (Not the Report Server URL).

  I'll digress for a minute here. In Native Mode, the URLs of the Report Server and Report Manager are as follows:

Native Mode Reporting Servers	Starting URL path
Report Manager	http://YourServerName/reports
Report Server	http://YourServername/reportserver

- Enter the **Start Path** where the reports were deployed. You find this in the Report Designer in the Configuration Manager. (Right click on the solution and select properties.)

- Change the **View Mode** the **List**.



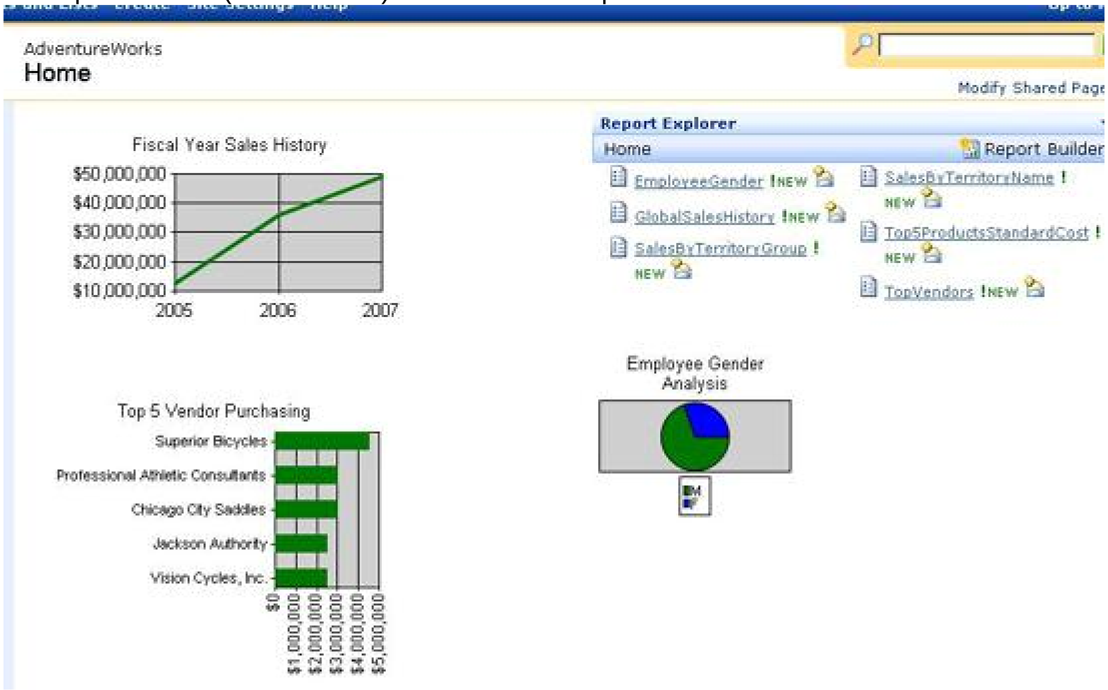
Report Manager URL: http://MyServer/reports

Start Path: AdventureWorks

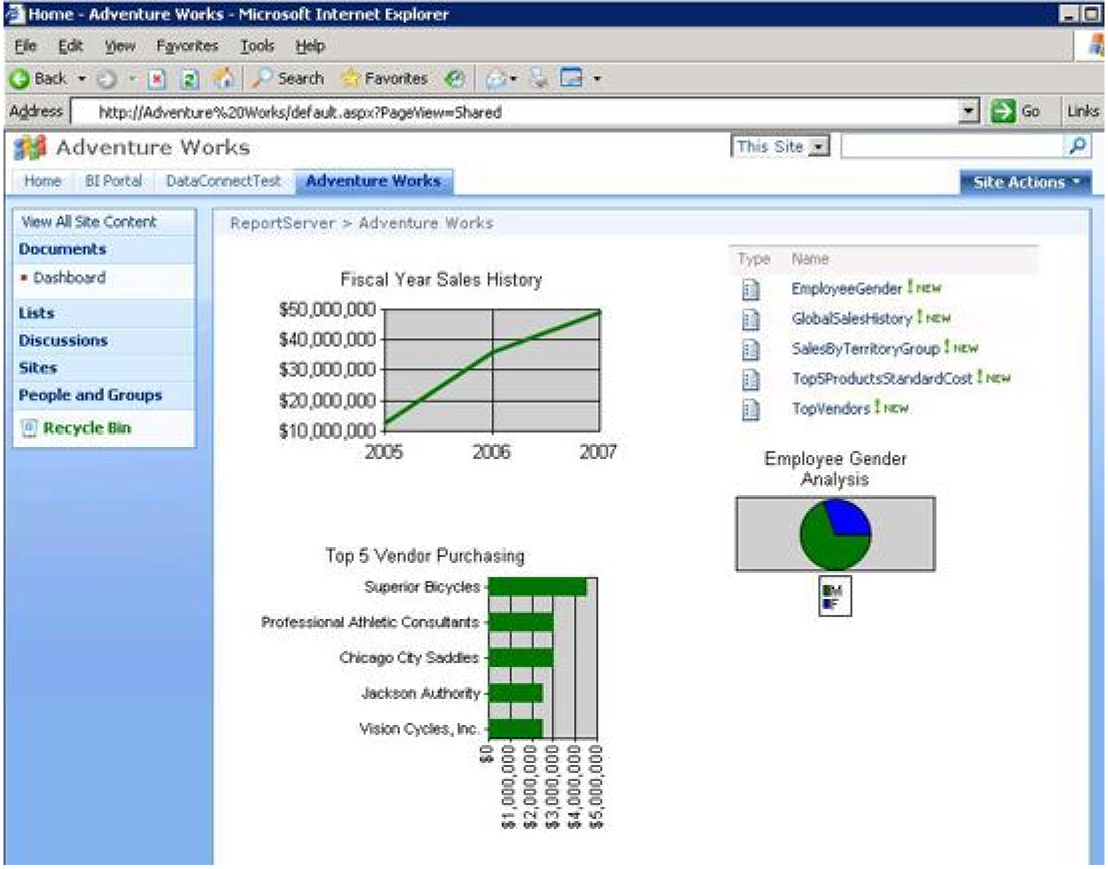
View Mode: List

- Here's the WSS2-based dashboard. I'll show the WSS3-based dashboard right below it again so you can see the similarities and differences.

Sharepoint **WSS2** (Native Mode) Dashboard example:



Sharepoint **WSS3** (Sharepoint Integrated Mode) Dashboard example:



The primary difference is the appearance and functionality of the report library web part (top right) of each page. In WSS2, the Report Explorer pops open a new browser when the user clicks on the report. That's actually an advantage over the WSS3 library. In contrast, the WSS3 library is more compact and appears in a single column, where the WSS2 always takes up 2 columns.

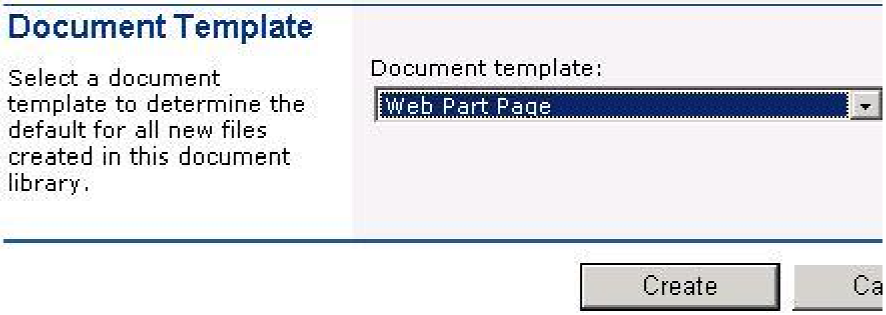
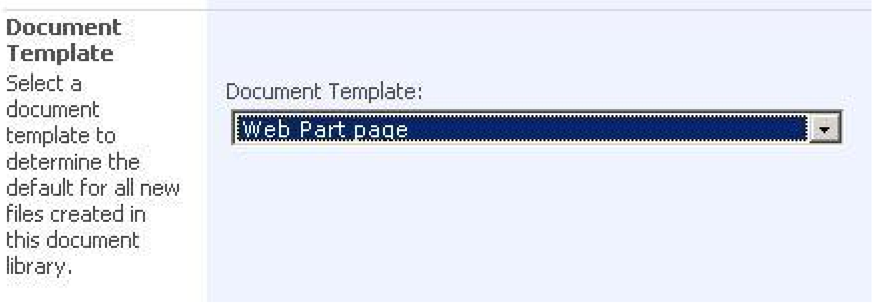
Dashboarding

At this point, we have a single high level web page with various charts and report libraries. In the reports we can add drilldowns and drill-throughs or even links to other sub-level dashboard pages that get narrower in focus. But in Sharepoint, how are we going to create web pages for other dashboards?

Well you could create another site for each sub-level dashboard. And you may want to do that, as your portal scales up. But it is not necessary to do in the beginning or even ever, if you plan carefully.

Here's where you begin:

Instead of a new site, start by creating a new document library. When you create the new document library, you are prompted for a Document template. Change the default type to Web Part Page.

When creating a document library for additional dashboards, choose theWeb Part Page document template. See screenshots below.	
WSS2	 Document Template Select a document template to determine the default for all new files created in this document library. Document template: Web Part Page
WSS3	 Document Template Select a document template to determine the default for all new files created in this document library. Document Template: Web Part page

Then start a new page in the document library. When you do, you will be prompted for a web part layout. Choose one that seems appropriate for the dashboard style you want. Then, you can create a new Web Part page for each dashboard you want to build. In WSS2, you cannot apply permissions to each page, so you will to plan your security setup and the functional hierarchy carefully. Perhaps you will need multiple Web Part Page libraries in order to allow for different security boundaries. In WSS3, you can apply security granularity right down to the document level within a single document library so the planning may be a bit different.