

Point and Click Web Pages with Design-Time Controls and SAS/IntrNet®

Vincent DelGobbo, SAS® Institute Inc., Cary, NC

John Leveille, iBiomatrics LLC™, Cary, NC

ABSTRACT

SAS® Design-Time Controls (DTCs) are a powerful and exciting addition to SAS Web Technologies. These add-in software components integrate back-end SAS® software functionality with your desktop HTML editor. You can now create Web pages that contain SAS content in a WYSIWIG, point-and-click fashion using any one of several popular HTML editors. In most cases, you do not have to write a single line of code.

INTRODUCTION

This paper provides a brief overview of the SAS/IntrNet Application Dispatcher. You do not need to know how to use this product in order to understand this paper. However, you need a working knowledge of the Application Dispatcher to use DTCs to generate Web pages.

This paper explains SAS Design-Time Controls and how you can use them to ease the task of building attractive Web pages. It also explains how to generate Web pages that use Java Server Page (JSP) and Active Server Page (ASP) technology.

SOFTWARE AND HARDWARE REQUIREMENTS

In order to use Design-time Controls, you must meet the following software and hardware requirements.

- You must license SAS/IntrNet software Version 6.09E, Version 6.12, Release 8.0 or later and have SAS installed and ready to run.
- You must license SAS/GRAPH® software if you wish to use Design-Time Controls that produce graphics.
- You must license SAS/EIS® software to use the MDDB Report control in Version 6 of the SAS System. If you are using Version 8, you must license either the SAS/EIS® software or SAS OLAP Server software. If you wish to build a SAS Multidimensional Database (MDDB) in Version 6 or 8, you must license SAS/MDDB® Server software.
- You must have a Web server in order to store and serve the Web pages you create. If you wish to use Design-Time Controls to create JSP-based or ASP-based Web pages, the Web server you choose must support these application types.
- You are not required to have the SAS System installed on your development machine in order to use SAS Design-Time Controls. However, SAS must be installed and configured to run as a SAS/IntrNet Application Server on some machine on your network. You must also have a Web server installed on some machine on your network.
- You must be running the Microsoft Windows operating system on the machine that you use to create the Web pages. Microsoft Internet Explorer Version 4.0 or later and one of the following HTML editors must be installed on this machine.
 - Microsoft FrontPage 98 or 2000
 - Softquad HoTMetaL Pro 5 or 6
 - Microsoft Visual InterDev
 - Macromedia Dreamweaver 2, 3, or 4
 - Macromedia Dreamweaver UltraDev
 - webAF™ 2.0 (part of AppDev Studio™ 2.0)
 - Macromedia Drumbeat 2000

Note that the Microsoft Windows requirement applies only to the machine used to develop the Web pages. The machine that serves the pages and the machines that browse the pages can be any operating system.

APPLICATION DISPATCHER OVERVIEW

The Application Dispatcher is a component of SAS IntrNet software. The Dispatcher allows you to execute SAS programs from a Web browser. These programs can consist of any combination of DATA step, PROC, MACRO, or SCL code, allowing you to immediately leverage your SAS skills and deploy a truly thin-client Web application.

APPLICATION DISPATCHER ARCHITECTURE

The Application Dispatcher is comprised of three components, as shown in Figure 1:

- Thin Client
- SAS Application Broker
- SAS Application Server

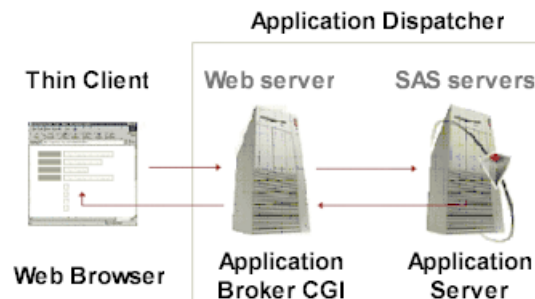


Figure 1. SAS Application Dispatcher Architecture and Execution Flow

The thin client is simply your Web browser. It is used to interact with any SAS programs that execute on the SAS Application Server. The user interface is typically an HTML page with a number of HTML form elements, and possibly some client-side script such as JavaScript. SAS software is not required on the client machine.

The SAS Application Broker is a very lightweight program that must be installed on your Web server machine. It is used to "broker" communication between the Web browser and the SAS Application Server using the Common Gateway Interface (CGI). You do not need CGI programming experience to use the Application Dispatcher.

The final component of the Application Dispatcher is the SAS Application Server. The server executes the SAS programs that you write and the results are returned to the Web browser via the Application Broker. Later you will see that Design-Time Controls alleviate this requirement to write your own SAS programs.

TYPICAL APPLICATION DISPATCHER EXECUTION FLOW

Figure 1 illustrates the typical execution flow of an Application Dispatcher request. As previously mentioned, the user interface is usually an HTML page containing HTML form elements. A user presented with this page will make appropriate choices and then submit the form for processing.

At this point, any data that are contained in the form are passed to the SAS Application Broker. The Application Broker determines which SAS Application Server is to handle the request and forwards all data to that server. The data are then converted to SAS macro variables and are also stored in an SCL list. Thus, the data contained on the HTML form are made available to your SAS program.

You should write your SAS program so that it uses the data passed from the Application Broker. After the SAS program finishes execution, you can display in your Web browser, the content that is generated by the program. HTML generated by the SAS Output Delivery System (ODS) or by the SAS HTML Formatting Tools are examples of appropriate content. A typical Application Dispatcher program is responsible for creating an entire page of content. This content is passed to the Application Broker, which in turn sends it back to the Web browser.

WHAT ARE DESIGN-TIME CONTROLS?

Design-Time Controls are add-in components that can be used with your Windows-based HTML editor. DTCs allow you to easily add SAS content to your Web pages. Using the interface of your HTML editor, you can insert a Design-Time Control into your Web page. The control assists you by presenting a dialog of choices and settings. After you make selections and close the dialog, the Design-Time Control will cause an Application Dispatcher program to execute, and the appropriate SAS-based content will be written into your Web page. Design-Time Controls act like page component wizards that help you build parts of your Web page.

After you publish your Web page anyone can browse the content that you have created. The Web browser user does not have to have Design-Time Controls installed on his/her machine in order to see the pages you have created, just as other people do not need your HTML editor to see pages you have created. Thus, the Windows restriction is imposed at design time, but not on clients that wish to view the page.

Design-Time Controls are very powerful. They can generate many forms of Web page content including HTML, JavaScript, Java applets, ActiveX controls, ASP, and JSP code. They present a user-friendly, intuitive interface that insulates you from much of the complexity that comes along with sophisticated Web content design. The SAS Design-Time Controls allow you to access the power of SAS Software, surface the results in your Web page, and still control the look and feel of your pages in a WYSIWYG editor environment.

Design-Time Controls contrast with Application Dispatcher programs in several ways.

- They generate only part of a page.
- You typically do not have to write the SAS program.
- The HTML editor handles HTML layout.
- They integrate with ASP and JSP.

As previously noted, typical Application Dispatcher programs generate a full page of content. Since the HTML editor is responsible for handling layout, the Design-Time Controls are solely responsible for generating fragments of SAS content. As will be discussed later, you can use Design-Time Controls to generate ASP- and JSP-based Web pages.

WHO USES SAS DESIGN-TIME CONTROLS?

SAS Design-Time Controls are intended to be used by someone who has knowledge of both the desired Web pages and the underlying data structures on which those pages are based. This may or may not be the same person in charge of the aesthetic design of the Web pages. It is possible for one person to lay out a series of Web pages including backgrounds, color schemes, and graphics, and a second person to use DTCs to insert the SAS content in the appropriate locations on these pages. Each Design-Time Control encapsulates its content, making it safe for anyone to edit the Web page without disturbing either the layout or the SAS content.

Of course, the page designer and SAS content builder may be the same person. In this case the SAS Design-Time Controls are still of considerable benefit. In the past, creating Web pages with a SAS server often meant having to write HTML and SAS code by hand. SAS Design-Time Controls make both tasks easier. Writing HTML is easy when you use a WYSIWYG HTML editor. Additionally, writing code is much simpler when you use the control property dialogs because they write it all for you.

When combined with the SAS/ACCESS® software products, the SAS Design-Time Controls have the added benefit of being able to reach from your Web page editor through SAS to external data sources, such as Oracle, DB2, and Excel. This means that a person in charge of creating Web-based reports for these other databases can also use SAS Design-Time Controls.

When your task is to analyze, construct reports, and surface that content to the Web, SAS Design-Time Controls make life easier for everyone involved.

HOW SAS DESIGN-TIME CONTROLS WORK WITH SAS/INTRNET SOFTWARE

You are not required to have the SAS System installed on your development machine in order to use SAS Design-Time Controls. However, SAS must be installed and configured to run as a SAS/IntrNet Application Server on some machine on your network. You must also have a Web server installed on some machine on your network. The Web server must be accessible from the development machine where you are creating the Web pages and from your users' machines where they will view the pages using Web browsers. The following diagram illustrates the required infrastructure to run Design-Time Controls with SAS/IntrNet.

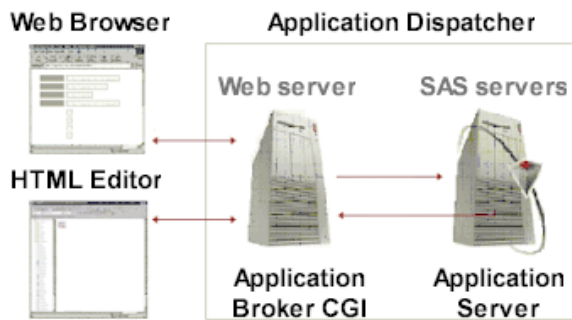


Figure 2. Design-Time Control and SAS/IntrNet Infrastructure

It is possible for the development, Web server, and SAS server machines to be three different machines, two different machines, or a single machine. It is also likely that the development machine (with the HTML editor) will function as a Web browser machine.

When you insert a SAS Design-Time Control into a Web page it reads the Windows registry and loads your default SAS server connection information. This information is set when the Design-Time Controls are installed on your machine. The control does not connect to the SAS server at this time. When you invoke the properties dialog for the control and make selections on the various tabs of the dialog, the control makes connections to the SAS server. The control queries the server for information such as a list of available data sets or a list of variables, and presents this information to you in the properties dialog.

A series of events occur when you save the Design-Time Control and when the Web page is browsed. The specific events that occur depend upon the current processing mode of the control. The processing mode can be set so that the control generates static content or dynamic content. Static content is content that does not change after construction of the page is complete. Conversely, a page that contains dynamic content, which is achieved through the use of ASP and JSP, will change if the underlying data changes.

If the control is set to generate static content, certain events occur, as shown in Table 1.

If the control's processing mode is set to static	
When the control is saved	When the Web page is browsed
- Control property settings are serialized into the Web page as an HTML comment. - Control connects to the SAS server and sends the current property settings. - SAS server runs a program stored on the server using the current control settings as input. - SAS server returns static content (usually HTML). - Control deposits content into the Web page	Static content is returned to the browser each time the page is viewed. No SAS processing is performed at this point.

Table 1. Static Processing Mode

If the control is set to generate dynamic content you have a choice of generating ASP code or JSP code. In either case, the events occur as shown in Table 2.

If the control's processing mode is set to dynamic	
When the control is saved	When the Web page is browsed
- Control property settings are serialized into the Web page as an HTML comment. - Control generates a URL string containing all the property settings. - Control wraps the URL string in ASP or JSP code and deposits the code into the Web page. No SAS processing is performed at this point.	- ASP or JSP code connects to SAS server and sends the current property settings. - SAS server runs a program stored on the server using the current control settings as input. - SAS server returns content (usually HTML) - ASP or JSP code deposits content into the Web page.

Table 2. Dynamic Processing Mode

When viewed in a Web browser, the resulting page should initially look identical for both the static and dynamic cases. However, over time the dynamic output may change if the underlying data changes. For static pages, the SAS processing is performed once, at the time when you construct the page. This is advantageous when your data is not changing because you only pay the processing cost once. It is not as beneficial if you need to see the latest data values and your data source is changing. For dynamic pages, the SAS processing is performed for each user that visits the page. This is advantageous because it shows up to the minute data values, which is very important for volatile data sources. However, this can become prohibitive if the processing takes a long time or if the page gets a lot of visitors. It is recommended that you consider the pros and cons of dynamic publishing before deploying your Web pages.

Sometimes you may find it necessary to blend the publishing modes of static and dynamic. For more information how to get the best of both techniques, see the "Advanced Topic: Scheduled Publishing" section of this paper.

AVAILABLE CONTROLS

Currently there are seven controls that you can use. Table 3 on the following page lists the controls and shows the minimum required release of the SAS/IntrNet Application Dispatcher software that is required for each control.

Control Name	Output Description	SAS Release or Version
SAS Critical Success Factor	Critical Success Factor Java Applet	6.09E, 6.12, or 8
SAS MDDDB Report	Multidimensional database report in HTML form	6.12 or 8
SAS Stored Program	Output from your SAS/IntrNet program	6.09E, 6.12, or 8
SAS Table	HTML table from a SAS data set	6.09E, 6.12, or 8
SAS Tabular Report	HTML table from PROC TABULATE	8
SAS Thin Client Graphics	SAS graphic Java applet or ActiveX control	6.09E, 6.12, or 8
SAS TreeView	Java applet to visualize hierarchical data	8.2

Table 3. Available Design-Time Controls

These controls are described as follows:

- The SAS Critical Success Factor control is a point and click interface to the SAS RangeView HTML Generator (DS2CSF Macro).
- The SAS MDDDB Report control is modeled after the Layout page of the SAS MDDDB Report Viewer.
- The SAS Stored Program control is used to run any standard SAS/IntrNet Application Dispatcher program. It provides a way for you to integrate your existing or new program output into a Web page.
- The SAS Table control is a point and click interface to the SAS HTML Data Set Formatter.
- The SAS Tabular Report control surfaces some of the functionality of the SAS TABULATE procedure.
- The SAS Thin Client Graphics control is a point and click interface to the SAS Graph Applet HTML Generator.
- The SAS TreeView control is a point and click interface to the SAS TreeView HTML Generator.

BUILDING YOUR FIRST WEB PAGE

In this section you will learn how to use a SAS Design-Time Control to add some SAS-based content to your Web page. For the purpose of illustration, the SAS Table control will be used. You can adapt the steps outlined below to work with any of the Design-Time Controls.

Once you install the SAS Design-Time Controls on your development machine and set up the SAS Application Dispatcher, you are ready to incorporate SAS content into your Web pages. You use the functionality of your HTML editor to do the entire layout of your page. In most cases, you can create an attractive Web page without having to write any HTML code.

Start your HTML editor and spend a few minutes adjusting the layout and appearance of your page. You may want to change the background color or add some text and images to make the

page more attractive. Once all the layout tasks are complete, you are ready to insert a SAS Design-Time Control into the page.

The act of inserting a Design-Time Control is different based on the HTML editor you are using. Typically, you can do this by finding the correct menu, such as **Insert**, and then selecting the proper submenu item, such as **Design-Time Control**. In Microsoft FrontPage, for example, the menu path is:

Insert ➤ Advanced ➤ Design-Time Control

Dreamweaver and Dreamweaver UltraDev users can insert Design-Time Controls through the **Insert** menu or via the SAS panel of the object palette. Other editors may have slightly different mechanisms for inserting controls. Please consult the documentation for your HTML editor if you are unsure how to insert an object.

Choose to insert a SAS Table DTC. At this point a thumbnail icon representing the Design-Time Control is placed into the page. Note that this is simply a placeholder to indicate that there is some piece of content at this point in your HTML page. You then must display the properties dialog of the Design-Time Control in order to make appropriate selections.

Some HTML editors may automatically display this dialog. Others will require you to invoke the dialog by double clicking the control, accessing a right mouse button menu, or some other technique. Access to the properties dialog in Dreamweaver and Dreamweaver UltraDev is provided through the property inspector for that object. If you are unable to access the properties dialog for the Design-Time Controls please consult the documentation for your HTML editor.

In Microsoft FrontPage, you can access the properties dialog either by double clicking the thumbnail icon or by selecting the icon and then choosing **Design-Time Control Properties...** from the popup menu. Figure 3 shows the **Dataset** tab of the SAS Table control properties dialog.

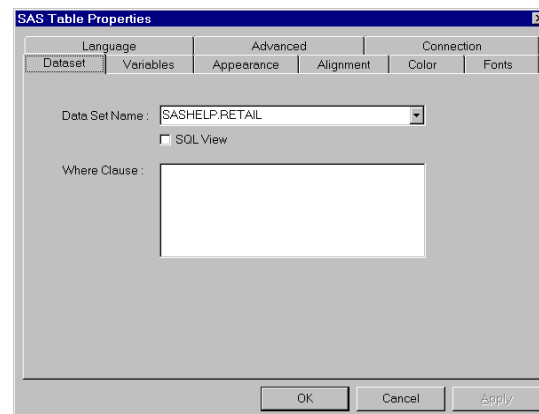


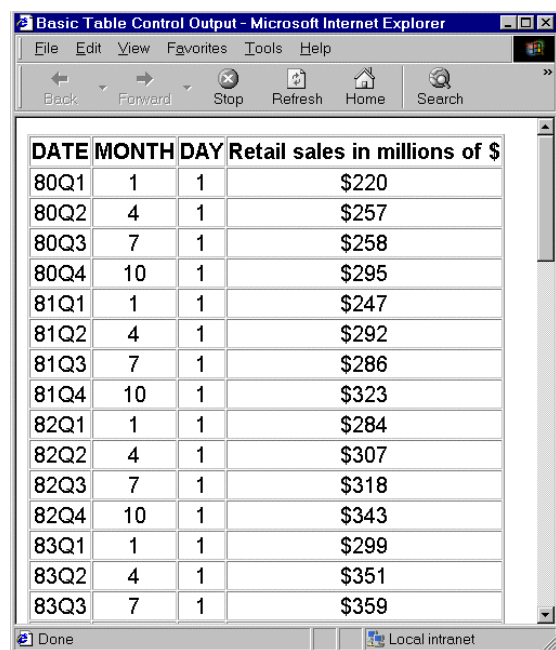
Figure 3. Dataset tab of the SAS Table control properties dialog

Select the data set SASHELP.RETAIL from the **Data Set Name** dropdown list.

You should now choose the variables that you wish to display in the HTML table. Go to the Variables tab and click the **Select** button next to the **Display variables** field. When the variable selector dialog opens, select the variables MONTH, DAY and SALES, in that order. Then click OK to close the Variable Selector dialog. Click the **Select** button next to the **ID variables** field. Select the variable DATE and click OK to close the variable

selector dialog. Type SALES in the **Sum variables** field to indicate that you want to sum the SALES column. Note that all of the fields on the **Variables** tab are text boxes. If you know the name of a variable you can type it in the box instead of opening the Variable Selector dialog.

Click OK to close the Table control properties dialog. At this point a SAS server program executes and returns an HTML table that contains the columns DATE, MONTH, DAY and SALES from the data set SASHELP.RETAIL. Save this HTML page and view it using your Web browser or using the preview window of your HTML editor. Figure 4 shows that the HTML table that was generated is pretty bland looking, so we will now set some other properties to make the table more attractive.



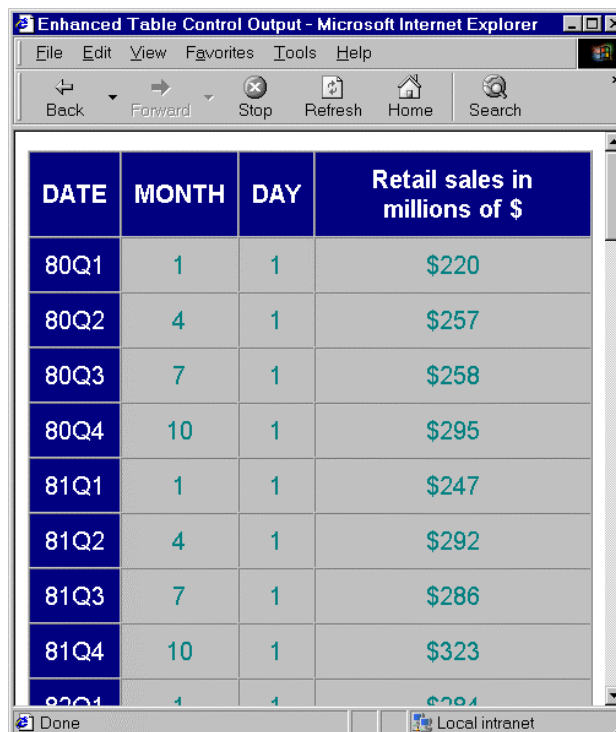
DATE	MONTH	DAY	Retail sales in millions of \$
80Q1	1	1	\$220
80Q2	4	1	\$257
80Q3	7	1	\$258
80Q4	10	1	\$295
81Q1	1	1	\$247
81Q2	4	1	\$292
81Q3	7	1	\$286
81Q4	10	1	\$323
82Q1	1	1	\$284
82Q2	4	1	\$307
82Q3	7	1	\$318
82Q4	10	1	\$343
83Q1	1	1	\$299
83Q2	4	1	\$351
83Q3	7	1	\$359

Figure 4. Basic HTML table generated by the SAS Table control

Access the properties dialog of the Table control and go to the **Appearance** tab. Supply a value of 10 for **Cell Padding** and a value of 0 for **Cell Spacing**. Go to the **Color** tab and supply the following colors for the various elements.

Element	Foreground Color	Background Color
Variables	Teal	Silver
Sum Variables	White	Red
Column Labels	White	Navy
ID Variables	White	Navy

Click OK to close the Table control properties dialog and save the HTML page. To see the enhanced table appearance (shown in Figure 5), view the page using your Web browser or using the preview window of your HTML editor.



DATE	MONTH	DAY	Retail sales in millions of \$
80Q1	1	1	\$220
80Q2	4	1	\$257
80Q3	7	1	\$258
80Q4	10	1	\$295
81Q1	1	1	\$247
81Q2	4	1	\$292
81Q3	7	1	\$286
81Q4	10	1	\$323
82Q1	1	1	\$284
82Q2	4	1	\$307
82Q3	7	1	\$318
82Q4	10	1	\$343
83Q1	1	1	\$299
83Q2	4	1	\$351
83Q3	7	1	\$359

Figure 5. Enhanced HTML table generated by the SAS Table control

STATIC PROCESSING VS. DYNAMIC PROCESSING

The SAS Design-Time Controls can operate in a few different processing modes. These processing modes produce either static output or dynamic output. If you examine the **Connection** tab in the properties dialog for each control you will see that the first field has the following options.

- Perform SAS processing once when building this page.
- Perform SAS processing when Java Server Page is invoked.
- Perform SAS processing when Active Server Page is invoked.

Note: The MDDB Report control does not have this field, but rather has a field named "Show MDDB Report." Refer to the Design-Time Control documentation for information on this field.

The first option "**once when building this page**" is the static processing mode. If you select this option, the DTC connects to SAS when you save the control properties or when you save the Web page. A program runs inside the Application Server, and HTML is deposited into the Web page that you are editing. The output from the control is static when you use this mode. It will not change until the next time you modify the control or save the page (depending upon your HTML editor behavior).

Static processing mode may be appropriate for a lot of pages. This mode can be used to construct static content for HTML, ASP, or JSP pages. It also means that each time a user browses this page on your Web site you are not incurring the cost of running a SAS program. However, if you are using static processing mode, the Web page may not display the latest data. This may be just fine if you are displaying a table of last year's revenue. Since that data will not change, it makes sense to use static processing mode. However, if you want to display a report based on some underlying data source that is rapidly changing then you probably want to consider a dynamic processing mode.

The second and third options, “**when Java Server Page is invoked**” and “**when Active Server Page is invoked**,” are dynamic processing modes. SAS Design-Time Controls use JSP and ASP in a similar manner. Therefore, consider statements about dynamic processing to apply equally to either ASP or JSP.

To use dynamic processing you must first make sure that your Web server is ASP or JSP enabled. Then, you must make sure that the file name for the Web page you are creating ends in .asp, if you intend to use ASP for dynamic processing, or .jsp if you intend to use JSP. Next, set the processing mode on the **Connection** tab to match the file name extension you are using.

Dynamic processing mode is powerful, but the differences from static mode can be subtle. The power of dynamic processing mode surfaces when you want to display the latest data in your Web page. When the Web server serves the dynamic page, the code generated by the SAS Design-Time Control will connect to SAS and run a program. This program will return HTML output representing the very latest data values. This is invaluable if you have a data source that is changing frequently. By using dynamic processing your Web page evolves and stays current without your intervention. From the perspective of editing the Web page content, dynamic seems almost exactly like static processing mode. However, there are some differences to consider. It is worth noting that you will probably be unable to use the preview feature of the HTML editor with dynamic processing mode. The best technique is to construct the page using static processing mode. Then, switch the processing mode on the control and change the file name extension when you are finished creating the page. When you are ready to view the Web pages, you must make sure to use an address that starts with “http” to see the result of the dynamic controls.

BUILDING A DYNAMIC WEB PAGE USING ASP OR JSP

The process of building dynamic Web pages is very similar to the process of creating static pages. The major difference is that for dynamic pages, you defer the execution of the Application Dispatcher program, and hence, generation of the SAS content, until you view the page in your browser. For the sake of brevity we will start with the static Web page that was constructed earlier, and shown in Figure 5.

Make a copy of the HTML file and open it in your HTML editor. Access the Design-Time Control properties dialog and go to the **Connection** tab. Note that the value specified in the **Perform SAS processing** dropdown list is **once when building this page**. This indicates static processing mode. To convert to a dynamic processing mode, select either **when Java Server Page is invoked** or **when Active Server Page is invoked**, depending on the type of dynamic page you wish to generate. Close the properties dialog and save the Web page.

Using your HTML editor, view the HTML code that was generated. You will see that the Design-Time Control did not create static HTML. Rather you will see VB Script or Java code, depending on the processing mode you chose. This code will run when a Web browser views the page. Thus, if the data that are in the SAS data set SASHELP.RETAIL change over time, the Web page will reflect that change when viewed in a Web browser. This is the difference between static and dynamic processing.

You must now give your Web page the correct file extension based on the dynamic processing mode that you chose. Use an extension of .asp for Active Server Pages and .jsp for Java Server Pages. You are now ready to view the page via your Web browser. Note that as mentioned earlier, you must use a Web browser to view dynamic pages, because the preview feature of your HTML editor will likely fail.

The result you see should be exactly the same as the page shown in Figure 5. Again, if the data had changed, then what you are viewing now would look different from the static page.

ADVANCED TOPIC: SCHEDULED PUBLISHING

If you take advantage of the dynamic publishing features of the SAS Design-Time controls, the ASP or JSP pages you create can be run on a scheduled basis. That is, the dynamic pages can be run periodically and the HTML that is generated can be stored as static pages on your Web server. This is useful if you have data that changes on a periodic basis, and you want to refresh the HTML pages that contain the time-sensitive content.

The remainder of this section discusses how you can accomplish this using the SAS System. Additionally, you will need scheduling software that is capable of running a program at a specified time. Examples include **cron** on Unix systems, **at** on Windows systems and **Control-M** on IBM mainframe systems. Alternatively, the SAS/Warehouse Administrator® software can be used to schedule your job.

Suppose you have data that is updated at 6:00 a.m. each day. You would like to generate reports that reflect the updated data and then store those reports on your Web server.

First, use the SAS Design-Time Controls to generate either ASP or JSP pages that represent your report, and store the pages on your Web server. Then, modify the following SAS program and schedule it to run at a specific time, for example, 6:10 a.m. This sample code uses the SAS software URL access method to retrieve the dynamic DTC page and store it as a static page on a Web server using the SAS software FTP access method.

```

*;
* Provide the path to the dynamic DTC page
* using the SAS URL access method.
*;
filename dtcpage url "http://myserver/path-
to-dynamic-dtc-page";

*;
* Provide the name of the static HTML page
* to be created, the domain of the Web
* server machine where the static page is to
* be stored, the user name a password to be
* used to log into the Web server machine
* and any other host commands using the SAS
* URL access method.
*;
filename static ftp "path-to-static-page"
host="myserver"
user="userid" pass="password"
rcmd="ascii"
rcmd="site umask 002";

*;
* Run the dynamic DTC page by accessing it
* via the SAS URL access method. Then
* publish the resulting HTML to the
* Web server using the SAS FTP access
* method.
*;
data _null_;
infile dtcpage;
file static;
input;
put _infile_;
run;

```

Note that if the SAS log contains the error message "ERROR: Service httpd not found" or "ERROR: Service ftp not found", please refer to the Troubleshooting section of the SAS Design-Time Controls documentation.

CONCLUSION

SAS Design-Time Controls are a powerful and easy way to add SAS-based content to your Web pages. They allow you to build attractive Web pages with a minimal amount of effort. SAS/ACCESS software also provides powerful data access capabilities in this easy-to-use environment. By supporting Active Server Pages and Java Server Pages, Design-Time Controls bring a new dimension to the types of applications you can develop with the SAS Application Dispatcher.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® Indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

REFERENCES

SAS/IntrNet Web Site, SAS Institute Inc., Cary, NC.
<http://www.sas.com/rnd/web/intrnet/>

SAS Application Dispatcher documentation, SAS Institute Inc., Cary, NC. <http://www.sas.com/rnd/web/intrnet/dispatch.html>

SAS HTML Formatting Tools Web Site, SAS Institute Inc., Cary, NC. <http://www.sas.com/rnd/web/intrnet/format>

SAS MDDb Report Viewer Web Site, SAS Institute Inc., Cary, NC. <http://www.sas.com/rnd/web/intrnet/mddbapp.html>

SAS/IntrNet Papers and Presentations, SAS Institute Inc., Cary, NC. <http://www.sas.com/rnd/web/intrnet/papers/>

SAS Design-Time Controls Web Site, SAS Institute Inc., Cary, NC. <http://www.sas.com/rnd/web/dtc/>

"SAS Design-Time Controls Now Available as Downloadable Component for SAS/IntrNet® Software," SAS Institute Inc., Cary, NC, 2000.
http://www.sas.com/service/news/feature/03jul00/sas_dtc.html

"SAS Design-Time Controls Add Support for Macromedia Dreamweaver," SAS Institute Inc., Cary, NC, 2000.
<http://www.sas.com/service/news/feature/18dec00/dtcs2.html>

"SAS Design Time Controls: New Tools for Information Delivery in the Enterprise," SAS Institute Inc., Cary, NC, 2000.
<http://www.sas.com/rnd/web/intrnet/papers/dtcDec00.pdf>

"Microsoft Design-Time Controls Overview," Microsoft Corp., Redmond, WA, 1997.
<http://msdn.microsoft.com/workshop/components/dtctrl/doc/overview.asp>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. If you have specific needs for controls and/or use an HTML editor or other application that does not support DTCs, let me know and I will evaluate your request as a future enhancement.

Contact the author at:

Vincent DelGobbo
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Phone: (919) 677-8000
Fax: (919) 677-4444
Email: Vincent.DelGobbo@sas.com
Web: <http://www.sas.com/rnd/web/dtc/>

