



SAS Publishing



SAS/IntrNet[®] 9.1

SAS[®] Design-Time Controls

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS/IntrNet*® 9.1: SAS® *Design-Time Controls*. Cary, NC: SAS Institute Inc.

SAS/IntrNet® 9.1: SAS® **Design-Time Controls**

Copyright © 2004, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice: Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

Published July 2002, July 2004, December 2004

SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at support.sas.com/pubs or call 1-800-727-3228.

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 trademarks of their respective companies.

Table of Contents

About SAS Design–Time Controls.....	1
What Are SAS Design–Time Controls?.....	2
Who Uses SAS Design–Time Controls?.....	3
How SAS Design–Time Controls Work with SAS/IntrNet Software.....	4
Requirements for the SAS Design–Time Controls.....	6
Default Connection Information.....	8
Components in the SAS Design–Time Controls Package.....	9
Testing the Installation.....	12
Making Data Available.....	15
Overview of Controls.....	16
Common Properties.....	18
Inserting a Design–Time Control.....	21
Copying a Control.....	22
Previewing Output.....	23
Modifying Output from the Controls.....	24
Scheduled Publishing.....	26
Example Scenarios.....	28

About SAS Design–Time Controls

SAS Design–Time Controls are a powerful and exciting addition to SAS Web technologies. These add–in software components integrate back–end SAS functionality with your desktop HTML editor. You can now create Web pages that contain SAS content in a WYSIWYG, point–and–click fashion. In most cases, you don't have to write a single line of code!

What Are SAS Design–Time Controls?

SAS Design–Time Controls are add–in components for your HTML editor that help you to easily add SAS content to your Web pages. Design–Time Controls act like page–component wizards that help you to build parts of your Web page. The controls present a user–friendly, intuitive interface that insulates you from much of the complexity that comes along with sophisticated Web content. You get to control the look and feel of your Web pages in a WYSIWYG editor and at the same time access and surface the power of SAS software on your Web page. Our Design–Time Controls are very powerful and can generate many forms of Web page content including HTML, JavaScript, Java applets, ActiveX controls, and ASP and JSP code.

With the Design–Time Controls, you use the user interface of your HTML editor to insert a 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 writes the appropriate SAS content onto your Web page. After you publish your Web page, anyone can browse the content that you created. Users who view your page through a Web browser do not have to have Design–Time Controls installed on their machines in order to see the pages that you created.

This following controls are available with this release of our software:

Control Name	Generated Web Page Content
SAS Critical Success Factor	Critical success factor Java applet
SAS MDDB Report	Multidimensional database report in HTML form
SAS Stored Program	Output from your SAS/IntrNet program
SAS Table	HTML table
SAS Tabular Report	HTML table from PROC TABULATE
SAS Thin–Client Graphics	SAS graphic Java applet or ActiveX control
SAS Treeview	Java applet to visualize hierarchical data

Who Uses SAS Design–Time Controls?

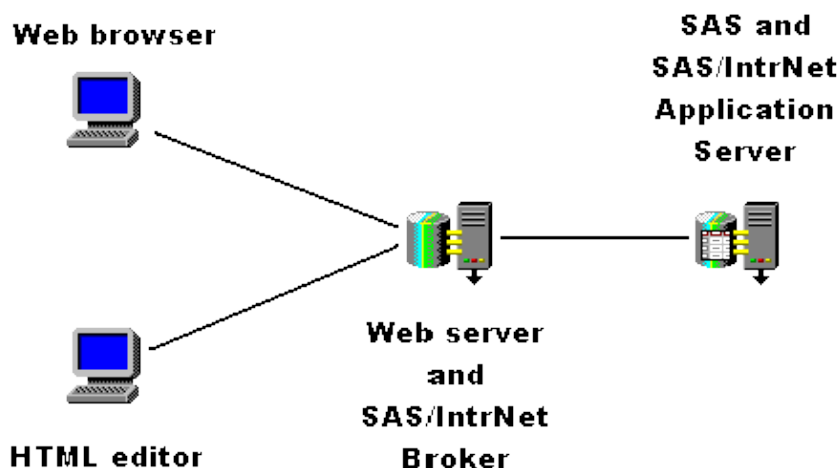
If your task is to analyze data, construct reports, and surface that content to the Web, then SAS Design–Time Controls can make life easier for you. The Design–Time Controls are for users who understand both the requirements for the desired Web pages and the underlying data structures upon which the pages are based. When used in conjunction with the SAS/ACCESS products, the Design–Time Controls also enable you to do Web–based reporting on data that resides in external data sources such as Oracle, DB2, and Microsoft Excel.

In the past, creating Web pages with a SAS server often meant having to write HTML and SAS code by hand. The Design–Time Controls make both tasks easier by enabling you to use a WYSIWYG editor to write the HTML and by providing you with dialogs that automatically generate the SAS code.

The Design–Time Controls can be of benefit to you regardless of whether your Web pages are created by one or more than one person. A designer can lay out the backgrounds, color schemes, and graphics for a Web page, and a content provider can, using the Design–Time Controls, insert SAS content in the appropriate locations on the page. Each control encapsulates its content, making it safe for anyone to edit the Web page and to avoid disturbing either the layout or SAS content.

How SAS Design–Time Controls Work with SAS/IntrNet Software

You do not have to have SAS installed on your client machine in order to use SAS Design–Time Controls. However, you must install and configure a SAS/IntrNet Application Server on a machine on your network. You must also have a Web server installed on a machine on your network. The Web server must be accessible from both the client machine on which the Web pages are created and from any Web browser machine on which the pages are viewed. The following diagram illustrates the required infrastructure.



The client, Web server, and SAS server machines can be three different machines, two different machines, or a single machine. Often, the client machine (with the HTML editor) also functions 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. However, 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.

If the control is set to generate static content, the events occur in the following manner.

When the control is saved

- Control property settings are serialized

When the Web page is browsed

- Static content is returned to the browser each time

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 that is 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.

the page is viewed. No SAS processing is performed at this point.

If the control is set to generate dynamic content, then you have a choice of generating Active Server Page (ASP) code or JavaServer Page (JSP) code. In either case, the events occur in the following manner.

When the control is saved

- 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.

When the Web page is browsed

- ASP or JSP code connects to SAS server and sends the current property settings.
- SAS server runs a program that is 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.

In both the static and dynamic cases, the resulting page should *initially* look identical. However, over time, the dynamic output might change if the underlying data changes.

In the static case, the SAS processing is performed once when you construct the page. The advantage of processing just once is low cost. The disadvantage is that you do not get to see the latest data values if your data source changes.

In the dynamic case, the SAS processing is performed for each user that visits the page. Users can see up-to-the-minute data values, an important advantage when the data sources are volatile. However, frequent processing can become prohibitive if the processing takes a long time or if the page gets a lot of visits. We recommend that you consider the pros and cons of dynamic publishing before deploying your Web pages.

Sometimes, you might find it necessary to blend static and dynamic publishing. For more information about how to get the best of both techniques, see [Scheduled Publishing](#).

Requirements for the SAS Design–Time Controls

The following sections detail the software prerequisites for using the SAS Design–Time Controls:

- Client requirements
 - Web browser requirements
 - Web server requirements
 - SAS server requirements
-

Client Requirements

These client machine requirements apply to the machine on which the Web pages are created and not to the machine on which the pages are viewed. See our Web browser requirements for information about viewing pages that are created with SAS Design–Time Controls.

The SAS Design–Time Control client components must be installed on a machine that is running a Windows operating system with Microsoft Internet Explorer Version 4.00 or later.

Note: If you are using Microsoft Internet Explorer Version 5.5, then you must install Internet Explorer 5.5 Service Pack 2 (SP2).

To use the controls, you must also have an HTML editor that supports them. The following list contains some editors that support our Design–Time Controls.

- Microsoft FrontPage 2000 and 2002
- Microsoft FrontPage 98*
- Microsoft Visual InterDev
- Softquad HoTMetaL Pro 5 and 6
- Macromedia Dreamweaver 2, 3, and 4
- Macromedia Dreamweaver UltraDev
- Macromedia Drumbeat 2000
- AppDev Studio 2.0: webAF 2.0

***Important note to FrontPage 98 users:** If you want to use the SAS Design–Time Controls with FrontPage 98, then you must install the Microsoft FrontPage 98b patch (fp98bupd.exe) in order for the controls to work properly.

Web Browser Requirements

To view Web pages that are created with SAS Design–Time Controls, you must have a Web browser. The Web browser requirements vary from control to control. Some of the controls produce only HTML. This HTML output can be viewed in the majority of browsers on a variety of hosts. Other controls produce tags that invoke Java applets or ActiveX controls. To view this output, you must use a browser that supports Java applets or ActiveX controls.

Web Server Requirements

To use the dynamic processing features of the SAS Design–Time Controls you must have

- a Web server environment that supports Active Server Pages (ASPs) or JavaServer Pages (JSPs)

- SAS Design–Time Control Web server components installed on the Web server machine.

Also, the Web server you are using must not require authentication when running the Application Broker. Your Web server can require authentication for other files or directories but not for the Broker CGI file.

To use the SAS Thin Client Graphics Control, the SAS TreeView Control, or graphics with the SAS MDDB Report Control, you must have the SAS/GRAPH client components installed on your Web server.

Note: If you installed Version 8 or later of the SAS/IntrNet CGI Tools for the Web Server package, you already have the SAS/GRAPH client components on your Web server. These components are also available on the Client–Side Components CD that is included with your Version 8 SAS System.

SAS Server Requirements

To use the SAS Design–Time Controls, you must have

- SAS/IntrNet software installed on a machine in your network, and you must set up an Application Server.

Note that

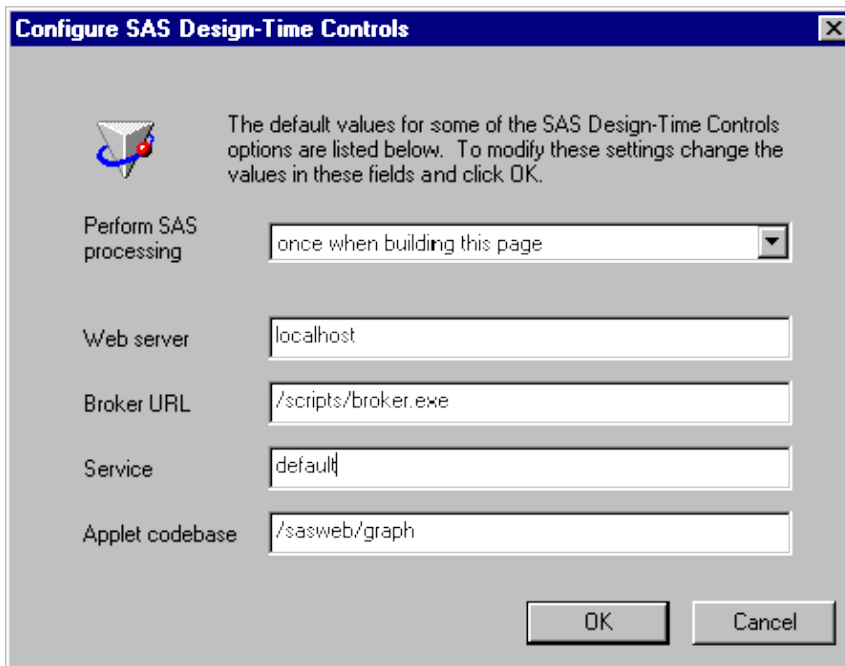
- ◆ The Tabular Report Control works only with SAS/IntrNet software, Version 8 or later because it uses the Output Delivery System (ODS).
- ◆ The TreeView Control works only with SAS/IntrNet software, Release 8.2 or later because it uses a Java applet that is not available in earlier releases.
- Release 8.2 or later of any SAS software that the control you are using requires.

Default Connection Information

When you install the SAS Design–Time Controls you are required to provide some information about things such as your Web server and Application Broker location. The controls use this information to connect to the SAS server and to query metadata such as lists of data sets and lists of variables. During the installation of the controls, this information is collected and placed in the Windows registry. The information is stored separately for each user on the machine.

You can use the SAS Design–Time Controls configuration program to view or to modify your default connection information. To run the configuration program

1. From the Windows **Start** menu, select **Programs** ➔ **SAS** ➔ **Design–Time Controls** ➔ **Configure SAS Design–Time Controls**
2. The following window appears:



3. Modify the values in this window and then select the **OK** button. The values you set in this window will be the initial values for any new Design–Time Controls that you insert into your Web page. The settings on this window do not affect controls that are already created; you can modify the settings for those controls individually.

Components in the SAS Design–Time Controls Package

The software components in the SAS Design–Time Controls package can be divided into four categories: client components, Dreamweaver components, Web server components, and SAS server components. This section describes the function of these components, lists the files that comprise them, and outlines the installation process for the components.

- Client components
 - Dreamweaver components
 - Web server components
 - SAS server components
-

Client Components

The client components in the SAS Design–Time Controls package are the controls themselves. The controls are ActiveX controls that are designed to function inside of an HTML editor. An icon represents the control in the editor window or on the Web page that you are constructing. A tabbed dialog provides a way to view and set control properties. The controls use the HTTP protocol to connect to your Web server, run the SAS/IntrNet Application Broker, and connect to the SAS/IntrNet Application Server.

All of the controls in this release of SAS Design–Time Controls are bound into a single file named `sasdtc.ocx`. You do not need to perform any manual steps to install the controls. The `sasdtc.ocx` file is placed on your hard drive and registered when you select **Client Component** from the **Components** list box that is presented during the installation of the SAS Design–Time Controls package.

Dreamweaver Components

The Dreamweaver components are client components that provide SAS Design–Time Control support for the Macromedia products Dreamweaver and Dreamweaver UltraDev. If you use either of these tools, then you must install the Dreamweaver components *after* installing Dreamweaver. After you install the Dreamweaver components, the Design–Time Controls are then available as selections from the Dreamweaver **Insert** menu or via the SAS panel of the Dreamweaver object palette.

You do not need to perform any manual steps to install the Dreamweaver components. They are installed when you select **Dreamweaver components** from the **Components** list box that is presented during the installation of the SAS Design–Time Controls package.

Web Server Components

If you use the Design–Time Controls to create static Web pages, then the only Web server component that you must have is the Application Broker. The Broker is part of the SAS/IntrNet product. For information about the Application Broker, see the documentation for the Application Dispatcher component of SAS/IntrNet software.

If you use the controls to create dynamic Web pages, then you need to install the Application Broker on your Web server, *and* you also need to install support for Active Server Pages (ASP) or Java Server Pages (JSP). The controls output ASP or JSP code instead of HTML in order to create dynamic Web pages. When the ASP or JSP page is served, the code makes a connection via HTTP to the Application Broker. The Broker contacts

the Application Server, runs the requested program, and returns the HTML. This HTML is deposited into the Web page stream by the ASP or JSP code.

Enabling ASP Support

To enable ASP support you need

- a Web server machine that runs the Microsoft Windows operating system and Microsoft Internet Information Server.
- the file named AppServer.dll that is provided in the SAS Design–Time Controls package. The ASP script that is generated by the controls calls AppServer.dll to connect to the Application Broker when the page is served. You do not need to perform any manual steps to install the AppServer.dll file. It is placed on your hard drive and registered when you select **Web Server Component** from the **Components** list box that is presented during the installation of the SAS Design–Time Controls package.

You can install Microsoft Internet Information Server either before or after you install the AppServer.dll file.

Enabling JSP Support

To enable JSP support you need to

1. Set up a JSP server environment on your Web server machine. This is additional software that is not provided by SAS. JSP environments exist from various vendors for many different operating systems. Install and configure your JSP server environment *before* proceeding with this part of the installation process.
2. Install the file named AppServer.jar that is provided in the SAS Design–Time Controls package. The JSP code that is generated by the controls calls the Java classes that are in AppServer.jar in order to connect to the Application Broker when the page is served. To install the AppServer.jar file, you must
 1. Select **Web Server Component** from the **Components** list box that is presented during the installation of the SAS Design–Time Controls package. The installation program places the AppServer.jar file on your hard drive.
 2. Copy the AppServer.jar file to the machine where your JSP server environment is installed. If you transfer the AppServer.jar file to a different machine, then make sure you perform a binary copy or a binary transfer. Transferring the AppServer.jar file in text mode corrupts the file.
 3. Find the CLASSPATH setting for your JSP server environment and add the full path for it to the AppServer.jar file. For example, if you are using JRun from Macromedia to serve your JSP pages, then you might have a properties file that contains something similar to the following line:

```
java.classpath=c:/jrun/classes
```

After adding the path for AppServer.jar, the line would look like the following (single line is wrapped for readability):

```
java.classpath=c:/jrun/classes;c:/Program Files/SAS/  
SAS Design-Time Controls/AppServer.jar
```

After changing the CLASSPATH setting, you will probably need to restart your JSP server.

SAS Server Components

The SAS Design–Time Controls require that some additional programs be made available from your Application Server. When you access the controls from within the HTML editor, they connect to the Application Server and run various programs. These programs return lists of data sets, lists of variables, and the control output. All of the required programs are contained in a SAS catalog named WEBSDK1.

The WEBSDK1 catalog is shipped with Release 8.2 and later of the SAS System.

Beginning with Version 8 of the SAS System, the Application Server will only run programs if they are listed in a PROGLIBS statement in the APPSRV procedure that starts the Application Server. You must find the SAS program file that starts your Application Server and edit it.

1. Locate the SAS/IntrNet service directory that is created when you set up the Application Server:

On Windows

From the Windows **Start** menu, select **Programs** ➔ **SAS** ➔ **IntrNet** ➔ **Service Directories**. A view opens that displays the folders that contain your SAS/IntrNet Application Server files.

On UNIX or VAX/VMS

The service directory is typically a subdirectory of the intrnet directory that is in your home directory.

On z/OS

The service directory is a partitioned data set with the name *prefix.WEB.service*.

2. Among the files in the service directory is a SAS program file that starts the Application Server (the default name for this file is appstart.sas). This file contains code for the APPSRV procedure. Insert the following code fragment somewhere after the PROC APPSRV statement and before the RUN statement:

```
proglibs sashelp.websdk1;
```

3. Restart the Application Server.
4. Repeat these steps for each Application Server service that you want to use with SAS Design–Time Controls.

Note: z/OS is the successor to the OS/390 and MVS operating systems. SAS/IntrNet 9.1 for z/OS is supported on the MVS, OS/390, and z/OS operating systems and, throughout this document, any reference to z/OS also applies to OS/390 and MVS, unless otherwise stated.

Testing the Installation

After you install the client and server components of the SAS Design–Time Controls, you can test the installation.

1. Testing the SAS server components
 2. Testing the client components
 3. Testing the Web server components
-

Testing the SAS Server Components

1. Test to see if your SAS/IntrNet Application Server is running. Invoke your Web browser and enter a URL like the following:

```
http://myserver/cgi-bin/broker.exe?_service=default&_program=status
```

A Web page that indicates that the server is functioning properly should be returned.

2. Test to see if the WEBSDK1 catalog is installed and accessible. Enter a URL like the following:

```
http://myserver/cgi-bin/broker.exe?_service=default&_program=sashelp.websdk1.member1.
```

A Web page that consists of a single selection list box should be returned. The list box should contain a list of data set names. If you see this list, then the WEBSDK1 catalog is installed and functioning correctly. If you see the following message

```
The program sashelp.websdk1.member1.scl is not available:  
ERROR: Catalog SASHELP.WEBSDK1 does not exist.
```

then you might have installed the WEBSDK1 catalog incorrectly. You will need to repeat the steps for setting up the SAS server components.

If you see the following message

```
sashelp.websdk1.member1.scl is unavailable, sashelp was not found in a  
proglibs or adminlibs statement.
```

then you need to modify your PROC APPSRV code to allow access to the WEBSDK1 catalog.

Testing the Client Components

1. Make sure that your SAS/IntrNet Application Server is running.
2. Start your HTML editor and begin editing a new HTML page.

Note to FrontPage 98 users: FrontPage 98 hides Design–Time Controls unless you turn on paragraph tags. Find the toolbar icon for paragraph tags and make sure that it is selected before you proceed. In addition, FrontPage 98 requires a patch in order to make SAS Design–Time Controls function properly. For more information about the patch, see the Client Requirements for the SAS Design–Time Controls

3. Access the list of Design–Time Controls. To access the list of controls in FrontPage 2000

1. Select **Insert** ➔ **Advanced** ➔ **Design–Time Control** from the menu bar. The Insert

- Design–Time Control dialog box is displayed with a list box that initially contains no items.
2. Select the **Customize** button and you should see the list of the controls that are installed with the SAS Design–Time Controls package.
 3. Check all of the check boxes that are next to the SAS controls and then select the **OK** button. The controls should now appear in the initial list box.
 4. Insert a SAS Table Control by selecting it from the list of available controls. At this point, you should see a SAS table icon inserted into the Web page or appearing in a floating dialog. If an icon does not appear, refer to the troubleshooting documentation.
 5. Access the properties dialog for the Table Control. To access the Properties dialog in FrontPage 2000, double–click on the icon. Other editors might require that you to select a properties button or select an option on the icon's pop–up menu. A slight delay occurs before the Properties dialog appears because the control connects to SAS and loads a list of available data sets.
 6. Examine the drop–down list that is provided for the **Data Set Name** field on the Dataset tab in the Properties dialog. This drop–down list should display several data set names. If it does, then your client components are functioning properly.
 7. If the drop–down list of data set names is empty, then look at the values for the fields on the Connection tab of the Properties dialog. Make sure that these values are correct. If these fields contain incorrect values or blanks
 1. Close down the Properties dialog and exit the HTML editor.
 2. Follow the steps on the Default Connection Information page to set the correct connection information.
 3. Repeat the steps for testing the client components.

Testing the Web Server Components

The next step is to test the Web server components that enable JSP and ASP support.

1. Make sure that your SAS/IntrNet Application Server is running.
2. Start your HTML editor and edit a new HTML page.
3. Insert a SAS Table Control. Invoke the Properties dialog and choose the data set **SASHELP.RETAIL** from the **Data Set Name** drop–down list on the Dataset tab.
4. Close the Properties dialog and save the HTML page.
5. Preview the HTML page in your Web browser to ensure that the HTML table is properly generated. You should see a table that displays the SASHELP.RETAIL data set.
6. Look at the address or location line in your Web browser and see if the URL begins with `http`. If the URL does not begin with `http`, then you are not viewing the Web page from a Web server; you are viewing a page that is a local file. If you are not viewing the page from a Web server, then publish this page to your Web server. After you are able to view the page in the browser and the URL begins with `http`, you can proceed to the next step.
7. Replace the static display of the RETAIL data set with a dynamic display using an ASP or JSP. Edit the same HTML file and open the properties for the SAS Table Control.
8. Select the Connection tab in the Properties dialog.
9. In the **Perform SAS processing** field, the default value is `once when building this page`.
 - ◆ If you want to test ASP support, select when Active Server Page is invoked from the drop–down list for this field.
 - ◆ If you want to test JSP support, select when Java Server Page is invoked from the drop–down list for this field.
10. Save the property changes.
11. Inspect the HTML source code for the page you are editing to verify that the output from the control is ASP or JSP code and not HTML.

- ◆ If you specified to perform processing when an Active Server Page is invoked, then you should see a block of code that starts like the following:

```
<%sub displayURL(URL)
  Dim AppServer, HTML
```

- ◆ If you specified to perform processing when a JavaServer Page is invoked, then you should see a block of code that starts like the following:

```
<%{
  appserver.AppServer IappServer = new appserver.AppServer();
```

12. Save the Web page with an extension of .asp to test Active Server Page support and an extension of .jsp to test JavaServer Page support. Make sure that the extension you choose matches the processing mode you select on the Connection tab in the Properties dialog.
13. View the ASP or JSP in your Web browser. Make sure that the URL in the address or location line begins with `http` or else the page will not display properly. If the URL does not begin with `http`, then publish the page to your Web server.

If you see the same table that you saw in the static HTML page, then ASP and JSP support is properly enabled. If you do not see the same table displayed, then you might not have installed the Web server components properly. For information on installing the Web server components, see [Web Server Components](#).

Making Data Available

The SAS Design–Time Controls enable you to easily browse and choose from lists of data sets, variables, catalogs, and other types of SAS metadata. The following sections provide details about making data available for SAS Design–Time Controls and the associated security exposure.

Note: The access control information on this page applies to all of the Design–Time Controls except for the MDDB Report Control. Beginning with Release 8.2 of the SAS System, the MDDB Report Control uses the access control features of SAS/EIS software. For general information, see Support for Access Control Features in the MDDB Report Viewer documentation. Detailed information can be found in the *SAS OLAP Server Administrator's Guide, Release 8.1*.

General Security Information

All of the data that you make available to the Design–Time Controls is available for read access from a user's Web browser. The SAS server components that are installed with the Design–Time Controls serve both the lists of metadata that you see at design time and the HTML output that the user sees at browse time. Because these SAS server components are not restricted to a single data set or catalog at design time, the Web browser user is not restricted to the choices that you made when you constructed the page. If you want to completely avoid the security exposure that is described, then see the section about disabling access.

Using the AUTHLIB Data Set

Beginning with Release 8.2 of the SAS System, use the AUTHLIB data set to control which SAS library entities are accessible through the Design–Time Controls. To make data available to the Design–Time Controls at the SAS library level

1. Modify the APPSRV procedure that starts the Application Server.
 1. Add ALLOCATE LIBRARY statements for each of the libraries that you want to make available.
 2. Then add those librefs to a DATALIBS statement.
2. Set up the AUTHLIB data set to control access to the libraries. The use of this data set is described in Controlling Access to Data Sources with the AUTHLIB Data Set in the Application Dispatcher documentation.

Disabling Access

If you are using Release 8.2 or later of the SAS System and you want to completely disable the data access that is allowed by the Design–Time Controls, add a record to the AUTHLIB data set that excludes all data.

Overview of Controls

The following controls are available with SAS Design–Time Controls. This section describes each of the controls.

- SAS Critical Success Factor Control
 - SAS MDDB Report Control
 - SAS Stored Program Control
 - SAS Table Control
 - SAS Tabular Report Control
 - SAS Thin Client Graphics Control
 - SAS TreeView Control
-

SAS Critical Success Factor Control

The SAS Critical Success Factor Control is a point–and–click interface to the Rangeview HTML Generator (DS2CSF macro). Many, but not all of the features of the Rangeview HTML Generator are implemented in the Critical Success Factor Control. Although no one–to–one correspondence exists between the Critical Success Factor Control properties and the Rangeview HTML Generator arguments, you can still use the syntax reference for the Rangeview HTML Generator (for Version 9 documentation, see the Web macros section of *SAS/GRAPH Software: Reference, Volumes 1 and 2.*) to determine the values to specify for the Critical Success Factor Control properties.

SAS MDDB Report Control

The SAS MDDB Report Control is modeled after the Layout page of the MDDB Report Viewer. Unlike the other controls, the MDDB Report Control cannot coexist with other page content. If you place the MDDB Report Control on a page along with other content, you will only see the MDDB report when the page is viewed in a browser. The MDDB report has special startup Javascript code that causes it to replace the contents of the current frame when it is displayed.

If you want a Web page that contains the MDDB report as well as other content, you need to create an frame–based HTML page. Dedicate one of the frames to the MDDB Report Control and place the rest of the content in the other frames.

Beginning with Release 8.2 of SAS/IntrNet Software, if you have access control enabled for SAS/EIS software, then you need to supply a valid username and password in order to use the MDDB Report Control.

SAS Stored Program Control

The SAS Stored Program Control is used to run any standard SAS/IntrNet program. For information about SAS/IntrNet programs, see The Program Component section of the SAS/IntrNet Application Dispatcher documentation.

Note: Your SAS/IntrNet program should not generate tags such as <HTML>, <HEAD>, and <BODY> because these tags are generated by your HTML editor.

SAS Table Control

The SAS Table Control is a point-and-click interface to the HTML Data Set Formatter. Many, but not all of the features of the HTML Data Set Formatter are implemented in the SAS Table Control. Although no one-to-one correspondence exists between the Table Control properties and the HTML Data Set Formatter arguments, you can still use the syntax reference for the HTML Data Set Formatter (support.sas.com/rnd/web/intrnet/format/) to determine the values to specify for the Table Control properties.

SAS Tabular Report Control

The SAS Tabular Report Control surfaces some of the functionality of the SAS TABULATE procedure. For more information about the TABULATE procedure, see the *SAS Procedures Guide*.

Note: This control requires Version 8 or later of SAS/IntrNet Software.

SAS Thin Client Graphics Control

The SAS Thin Client Graphics Control is a point-and-click interface to the Graph Applet HTML Generator (DS2GRAF macro). Many, but not all of the features of the Graph Applet HTML Generator are implemented in the Thin Client Graphics Control. Although no one-to-one correspondence exists between the Thin Client Graphics Control properties and the Graph Applet HTML Generator arguments, you can still use the syntax reference for the Graph Applet HTML Generator (support.sas.com/rnd/web/intrnet/format/) to determine the values to specify for the Thin Client Graphics Control properties.

The Thin Client Graphics Control also supports ActiveX graphics. If the SAS/GRAPH Control for ActiveX is installed on the machine from which you view your Web page, then you can use the Thin Client Graphics Control to generate ActiveX graphics instead of Java graphics.

1. On the Applet tab of the Properties dialog, select `ActiveX Control` in the **Graphics device** field.
2. On the Applet tab, either delete the value in the **Codebase** field or supply an appropriate value.

SAS TreeView Control

The SAS TreeView Control is a point-and-click interface to the Treeview HTML Generator. Many, but not all of the features of the Treeview HTML Generator are implemented in the TreeView Control. Although no one-to-one correspondence exists between the TreeView Control properties and the Treeview HTML Generator arguments, you can still use the syntax reference for the Treeview HTML Generator (for Version 9 documentation, see the Web macros section of *SAS/GRAPH Software: Reference, Volumes 1 and 2.*) to determine the values to specify for the TreeView Control properties.

Note: This control requires Release 8.2 or later of SAS/IntrNet Software.

Common Properties

Each of the SAS Design–Time Controls has properties that you can set. Certain properties are common for all controls; the following section describes the common properties, which are represented as fields on tabs in the Properties dialog box.

- Advanced tab
 - ◆ Program
 - ◆ Extra parameters
- Connection tab
 - ◆ Perform SAS processing
 - ◆ Show MDDB Report
 - ◆ Web server
 - ◆ Broker URL
 - ◆ Service
 - ◆ Debug

For more information about any of the properties, except those that are represented by the **Perform SAS processing** and **Show MDDB Report** fields, please refer to the Application Dispatcher documentation.

The Advanced Tab

The Advanced tab contains the following fields:

Program

Specifies the name of the program that runs on the SAS server. Generally, you do not need to modify this field unless you are using the SAS Stored Program Control.

If you are using the Stored Program Control, then specify the name of the SAS/IntrNet program that you want to run. The value for this field should be of the same format as the value for the `_PROGRAM` variable in a typical Application Dispatcher request.

Extra parameters

Specifies any extra parameters that are required by the program that is specified in the **Program** field. Generally, you do not need to modify this field unless you are using the SAS Stored Program Control.

If you are using the Stored Program Control, then specify any parameters that are required by the program. The format for the extra parameters is:

`name-1=value-1< . . . &name-n=value-n>`

If your extra parameter values contain spaces or special characters, you must URL–encode the characters. For information about encoding values, see the documentation for the `URLENCODE` function at support.sas.com/rnd/web/intrnet/apputil/index.html. You can use the `URLENCODE` function in a `DATA` step to produce an encoded string and then specify that string in the **Extra parameters** field.

The Connection Tab

The Connection tab contains the following fields:

Perform SAS processing

Specifies the SAS Design–Time Controls processing mode. You can choose from the following options:

- ◇ once when building this page
- ◇ when Java Server Page is invoked
- ◇ when Active Server Page is invoked.

These processing modes produce either static output or dynamic output.

Note: The MDDDB Report Control does not have this field. It has a **Show MDDDB Report** field.

When you select the **once when building this page** option, you are selecting the static processing mode. Selecting this option means that when you save the control properties or save the Web page, a connection to SAS is made, 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 and does not change until the next time you modify the control or save the page (depending upon the behavior of your HTML editor).

You can use static processing when you are constructing ASPs, JSPs, or HTML pages. Some points to remember about static processing:

- ◇ You do *not* incur the cost of running a SAS program each time a user browses a static page on your Web site.
- ◇ With static processing, the Web page might not display the most current data. Therefore, static processing is good for those instances when you want to display data that does not change (such as last year's revenue for your company). However, if you want to display a report that is based on a rapidly changing data source, 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 uses both JSPs and ASPs in the same manner. Therefore, the following discussion applies equally to both JSPs and ASPs.

To use dynamic processing you must

1. Make sure that your Web server is ASP– or JSP–enabled.
2. Make sure that the filename for the Web page that you are creating ends in .asp if you intend to use an ASP or .jsp if you intend to use a JSP.
3. From the **Perform SAS processing** field on the Connection tab, select the dynamic processing mode that corresponds with the filename extension.

The dynamic processing modes are powerful because they enable you to display up–to–date data on your Web page without you having to update the page. When the Web server serves the dynamic page, the code that is generated by the SAS Design–Time Control connects to SAS and runs a program. This program returns HTML output that represents the very latest data values.

Editing Web pages with dynamic content is similar to editing those with static content. One main difference is that you will probably be unable to use the preview feature of the HTML editor when you are in a dynamic processing mode. However, you can use the following steps to give yourself access to the preview feature:

1. Construct the page using the static processing mode.
2. Switch the processing mode for the control to a dynamic processing mode.
3. Change the filename extension after you finish creating the file.
4. Make sure to view the Web page using a URL that starts with `http` to see the result of the

dynamic control.

Show MDDB Report

Specifies how the MDDB report is generated and displayed. The MDDB Report Control does not have a **Perform SAS processing** field because the content for this control is always dynamic. Also, this dynamic behavior does not require the use of ASP or JSP. Instead, this control has a **Show MDDB Report** field with the following options:

- ◇ automatically
- ◇ by calling the showMRV function

In most cases, select **automatically**, which causes the MDDB report to be generated when the page is accessed via the Web browser.

If you want to run a preprocessing script before the MDDB report is generated, then select **by calling the showMRV function**. You must also include a function named showMRV in the Web page. This function must accept a single string argument. The value of this argument is the URL for the MDDB report. The function should perform any preprocessing you want and then execute the URL in order to generate the MDDB report. The following example shows a JavaScript function named showMRV that executes the URL:

```
function showMRV(mrvURL) {  
  /* Perform your preprocessing here */  
  location.replace(mrvURL);  
}
```

Web server

Specifies the domain of the Web server where you installed the Application Broker program. The default value for this field is the name of the domain that you specified when you installed the SAS Design–Time Controls. Supplying a value of `localhost` in this field is recommended if you intend to develop Web pages on one server and publish them on another server.

Broker URL

Specifies the path to the location of the Application Broker program. The combination of the value in this field and the value in the **Web server** field provides the full path to the Broker program. The default value for this field is the name of the Broker URL that you specified when you installed the SAS Design–Time Controls.

Service

Contains the name of the SAS/IntrNet service you want to use. The available service names are defined in the Application Broker configuration file. See the Application Dispatcher documentation for information about the `_SERVICE` variable and available services.

Debug

Specifies a debug option. The debug options help you to resolve problems that you might encounter when running the Design–Time Controls. The default value of 0 (zero), suppresses debug information. See the Application Dispatcher documentation for a list of valid debug values.

Inserting a Design–Time Control

The HTML editor you use determines the steps that are necessary for inserting a control.

- In most editors, you would find the correct menu option, such as **Insert** and then select the correct submenu option, such as **Design–Time Control**.
- In Microsoft FrontPage 98 and Microsoft FrontPage 2000, you would select **Insert** ➔ **Advanced** ➔ **Design–Time Control**.
- In Macromedia Dreamweaver and Macromedia Dreamweaver UltraDev, you can insert Design–Time Controls through either the **Insert** menu or the SAS panel of the object palette.

Refer to the documentation for your HTML editor for more information about how to insert controls.

Accessing Control Properties

After you insert a control into the page, you need to access the Properties dialog in order to set the properties for the control.

- Some HTML editors might automatically display the Properties dialog.
- Others editors will require you to invoke the dialog by double–clicking on the control or selecting an option from the control's pop–up menu.
- 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 refer to the documentation for your HTML editor.

FrontPage 98 Information

- If you are using FrontPage 98, you must make sure that your paragraph tags are turned on; otherwise, the editor hides the Design–Time Control. Find the toolbar icon for paragraph tags and make sure that it is selected before you proceed. You must also apply a patch before you can use Design–Time Controls with FrontPage 98. For information about acquiring this patch, see the client requirements for Design–Time Controls.
- When you bring up the pop–up menu for a Design–Time Control in FrontPage 98, you will see the option **Properties** listed twice. To access the preferred Properties dialog, choose the second of the two menu options. Choosing the first option displays a rudimentary properties list that is not as easy to understand as the tabbed properties dialog.

Copying a Control

If you need to insert two or more of the same type of control into a page and if many of the property settings are the same for these multiple controls, then you can copy a control instead of creating one from scratch. Most HTML editors will allow you to copy and paste a control into another spot on your page. Certain editors will also allow you to copy and paste across Web pages.

When you copy a control, the copy is of the same type as the original and contains all of the property settings of the original. If the properties for the new control need to be different from the original, then make those changes in the Properties dialog; be sure to change the properties that must be unique for each control.

Previewing Output

Some HTML editors have a feature that enables you to preview the Web page that you are creating. Depending upon the implementation of the preview feature, you might be able to also preview Design–Time Control output in a preview window. If you attempt to use a preview window and do not see Design–Time Control output, then check the source code for the page.

1. Verify that the control actually creates some output.
2. Determine the processing mode of the control. Open the Properties dialog and select the Connection tab. In the **Perform SAS processing** field (the MDDB Report Control does not have this field), you should see one of the following options:
 - ◆ once when building this page
 - ◆ when Java Server Page is invoked
 - ◆ when Active Server Page is invoked.

The first option specifies a static processing mode and generates static HTML. The generated output does not require a connection to SAS when the page is served up to a Web browser. The second and third options specify dynamic processing modes. With dynamic processing, the generated output is either Java code or VBScript that is run on the Web server. When the Web server serves the page, this code is executed and a connection to SAS is made. A connection to SAS is required in order to view output that is generated in dynamic mode.

Because most HTML editors do not utilize the Web server for previewing a page, you cannot run dynamic code and therefore will probably not be able to preview your output. In order to ensure the ability to preview your file, use the static processing option while you are constructing your page and then switch to a dynamic option after you finish creating the page. The visual appearance of the page should be the same for both the static and dynamic processing modes.

3. If your control generates either Java applets or ActiveX controls, then you must specify values for the **Codebase** and **Archive** fields on the Applet tab in the Properties dialog. Java applets and ActiveX controls require codebase and archive values. The Critical Success Factor Control and Thin Client Graphics Control output either Java applets or ActiveX controls.

The SAS Design–Time Controls installation suggests a default value of **/sasweb/graph** for the **Codebase** field. This value is a relative path that points to a folder on your Web server. If you are using the preview feature of your HTML editor, then this relative path might not point to the correct download location for the applet because the preview feature probably does not view the Web page as served from your Web server. Your options are to either

- ◆ specify a fully–qualified codebase path, such as `http://myserver/sasweb/graph`
- ◆ or view the Web page in your Web browser and make sure that the page is being viewed from its location on your Web server. To verify that the page is being served from your Web server and not as a local file, examine the URL in the Web browser and make sure that it begins with `http`.

In addition to a built–in preview feature, most HTML editors have an option that enables you to preview in the Web browser. Selecting this option invokes your Web browser and views the page as you last saved it. If you can set up your editing environment so that the preview–in–browser feature displays a Web page URL that starts with `http`, then this is probably the best technique for viewing the pages that you create. A Web page with a URL that starts with `http` is served by your Web server; therefore, relative paths such as `/sasweb/graph` can be used and both static and dynamic output can be viewed.

Modifying Output from the Controls

You can manually modify the Web page content that the Design-Time Control generates. For example, if you use the SAS Table Control to generate a SAS table and then want to add text to one of the column labels, you can either write a custom stored program to produce the desired output or you can manually change the output of the control.

If you choose to manually modify the control output, then you need to know a few things about the structure of the control in your Web page. The following example contains the Web page source code for a SAS Table Control.

```
<!--metadata type="DesignerControl" startspan
<object classid="clsid:6ED5010A-D596-11D3-87D7-00C04F2C0BF6" id="Table1"
dttcid="1" width="48" height="48">
  <param name="_Version" value="65536">
  <param name="_ExtentX" value="1270">
  <param name="_ExtentY" value="1270">
  <param name="_StockProps" value="0">
  <param name="BrokerURL" value="/cgi-bin/broker.exe">
  <param name="DataSet" value="SASHELP.RETAIL">
  <param name="Debug" value="0">
  <param name="ProcessingMode" value="build">
  <param name="Program" value="sashelp.websdk1.ds2htm2.scl">
  <param name="Service" value="default">
  <param name="WebServer" value="mydomain.com">
</object>
-->

<P>

<TABLE BORDER="1" WIDTH="0%" CELLPADDING="1" CELLSPACING="1">
  <TR>
    <TH ALIGN="CENTER" VALIGN="MIDDLE"><FONT
FACE="Arial,Helvetica,sans-serif">Retail sales in millions of $</FONT></TH>
    <TH ALIGN="CENTER" VALIGN="MIDDLE"><FONT
FACE="Arial,Helvetica,sans-serif">DATE</FONT></TH>
    <TH ALIGN="CENTER" VALIGN="MIDDLE"><FONT
FACE="Arial,Helvetica,sans-serif">YEAR</FONT></TH>
    <TH ALIGN="CENTER" VALIGN="MIDDLE"><FONT
FACE="Arial,Helvetica,sans-serif">MONTH</FONT></TH>
    <TH ALIGN="CENTER" VALIGN="MIDDLE"><FONT
FACE="Arial,Helvetica,sans-serif">DAY</FONT></TH>
  </TR>
</TABLE>

<P>

<!--metadata type="DesignerControl" ends-->
```

The control's source code begins and ends with HTML comments (highlighted in green). The starting comment is marked by the word `startspan`. The ending comment is marked by the word `endspan`. Within the starting comment is an OBJECT tag. The HTML code in the OBJECT tag specifies the stored parameter values for the control. These values represent the selections that were made in the Properties dialog for this control. Between the starting and ending comments is the table that the control generates (highlighted in blue).

Making Changes Permanent

In most HTML editors, you will need to remove the starting and ending comments in order to make your changes permanent. Otherwise, when you modify the control's properties or save the page, the control will regenerate its output and overwrite your changes.

In most HTML editors, you have to remove the starting and ending comments by hand. If you can access the Web page's HTML source code and find the control, delete the entire starting and ending HTML comment sections and then save the page.

If you use Microsoft Visual InterDev, then the editor helps you to remove the comments.

1. Switch to Source mode by selecting the Source tab at the bottom of the document window.
2. Bring up the pop-up menu for the control and select **Convert to Run-time Text**. Selecting this option cause the starting and ending comments to be removed and converts the control so that the output is all that remains. Now you can edit the content and your changes will remain.

Note: After you remove the starting and ending comments, you can no longer easily change the content back into a Design-Time Control.

Scheduled Publishing

The dynamic publishing features of the SAS Design–Time Controls enable you to run dynamic pages (ASPs and JSPs) periodically and store the generated HTML as static pages on your Web server. This function is useful if you have data that changes on a known basis and you want to refresh the HTML pages that contain the time–sensitive content.

The following example shows you how to use SAS to perform scheduled publishing. In addition to SAS, you also need to have scheduling software that is capable of running a program at a specified time. Examples include cron on UNIX systems and Control–M on IBM mainframe systems. You can also use the SAS/Warehouse Administrator to schedule your job.

Example

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

1. Use the SAS Design–Time Controls to generate either ASPs or JSPs that represent your report and store them on your Web server.
2. Modify the SAS program below and schedule it to run sometime after 6:00 a.m (for example, 6:10 a.m.). The sample code uses the SAS URL access method to retrieve the dynamic page and then stores it on a Web server using the SAS 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 and 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: If the SAS log contains either of the following error messages

ERROR: Service httpd not found.

ERROR: Service ftp not found

refer to our troubleshooting documentation.

Example Scenarios

This section provides example scenarios for working with the SAS Table Control and the SAS Thin Client Graphics Control. The scenarios assume that you know how to use your HTML editor to insert a Design-Time Control into an HTML page.

SAS Table Control

1. Insert a Table Control and access its properties. From the **Data Set Name** drop-down list, select **SASHELP.RETAIL**.
2. Go to the Variables tab.

Note: All of the fields on this tab are type-in fields, so if you know a variable's name, you can type it into the field instead of opening the Select dialog.

1. Select the **Select** button that is next to the **Display variables** field.
 2. On the Select dialog, select the **MONTH**, **DAY** and **SALES** variables, in that order, from the **Available** list box. Select the **OK** button.
 3. Select the **Select** button that is next to the **ID variables** field.
 4. On the Select dialog, select the **DATE** variable from the **Available** list box and then select the **OK** button.
 5. Type **SALES** in the **Sum variables** field to indicate that you want the **SALES** column summed.
3. Select the **OK** button to close the Properties dialog. At this point, a SAS server program executes and returns an HTML page that displays the columns **DATE**, **MONTH**, **DAY**, and **SALES** from the **SASHELP.RETAIL** data set. Save this HTML page and view it in your Web browser or via the preview window of your HTML editor.
 4. Set some more properties to make the generated table more attractive.

1. Access the properties for the Table Control and go to the Appearance tab. Specify a value of **10** for the **Padding** field and a value of **0** for the **Spacing** field.
2. Go to the Color tab and supply the following colors for the associated fields:

Field	Foreground Color	Background Color
Variables	Teal	Silver
Sum variables	White	Red
Column labels	White	Navy
ID variables	White	Navy

3. Select the **OK** button to close the Properties dialog. To see the enhanced table, save the HTML page and view the page in your Web browser or via the preview window of your HTML editor.

SAS Thin Client Graphics Control

Note: In order to use the Thin Client Graphics Control, you must license the SAS/GRAPH product.

1. Insert a Thin Client Graphics Control and access its properties. From the **Data Set Name** drop-down list, select **SASHELP.RETAIL**.
2. On the Variables tab and select
 - ◆ **YEAR** for the **Independent** field
 - ◆ **SALES** for the **Dependent** field
 - ◆ **MONTH** for the **Subgroup** field.

Note: All of the fields on this tab are type-in fields, so if you know a variable's name, you can type it into the field instead of accessing the drop-down list.

3. Select the **OK** button to close the Properties dialog. At this point, a SAS server program executes and returns a Java graphic that uses the variables that you selected from the SASHELP.RETAIL data set. Save this HTML page and view it in your Web browser or via the preview window of your HTML editor.