

## CHAPTER

## 2

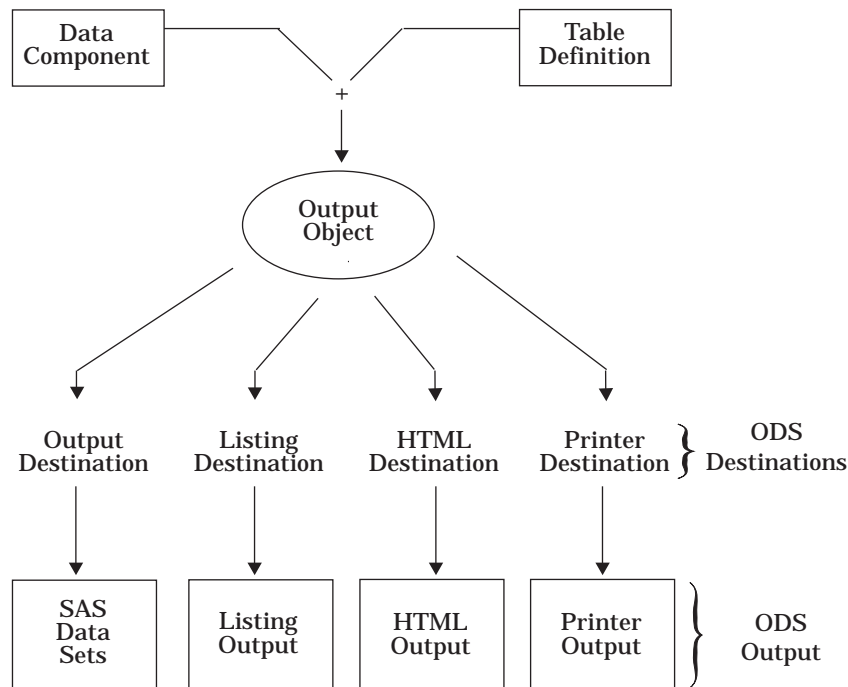
## Basic Concepts of the Output Delivery System

<i>Creating Output with ODS</i>	21
<i>ODS Destinations</i>	23
<i>Using Selection and Exclusion Lists</i>	23
<i>How to Specify an Output Object</i>	24
<i>How ODS Determines the Destinations for an Output Object</i>	25
<i>Modifying Selection and Exclusion Lists</i>	25
<i>Automatic Modification of Lists for All Destinations except the Output Destination</i>	26
<i>Explicit Modification of Lists All Destinations except the Output Destination</i>	26
<i>Automatic Modification of the List for the Output Destination</i>	27
<i>Explicit Modification of the List for the Output Destination</i>	28
<i>Resetting Lists when RUN-Group Processing Is in Effect</i>	28
<i>Understanding Some HTML Basics</i>	30
<i>Using HTML Tags</i>	30
<i>Using HTML Attributes</i>	30
<i>Using URLs and Filenames</i>	30
<i>HTML Links and References</i>	31
<i>What Are Links and References?</i>	31
<i>Implementing HTML Links and References</i>	31
<i>How ODS Constructs Links and References</i>	34
<i>Files Produced by the HTML Destination</i>	36
<i>The Body File</i>	36
<i>The Contents File</i>	39
<i>The Page File</i>	39
<i>The Frame File</i>	39
<i>Customizing ODS Output</i>	41
<i>Style Definitions, Style Elements, and Style Attributes</i>	41
<i>Table Definitions, Table Elements, and Table Attributes</i>	42
<i>Customizing Presentation Aspects of ODS Output</i>	42
<i>Customizing Output for a Tabular Output Object</i>	43

### Creating Output with ODS

In the past, the term “output” has generally referred to the outcome of a SAS procedure step. You could send this output to the Output window if you were working in a windowing environment, to an output device if you were working in line mode, or to a file if you used PROC PRINTTO or SAS system options.

With the advent of the Output Delivery System, “output” takes on a much broader meaning. The following figure illustrates the concept of output for Version 8 SAS. Definitions of the terms in the figure follow.

**Figure 2.1** Model of the Production of ODS Output*data component*

Each procedure that supports ODS and each DATA step produces one or more data components. Each data component contains results (numbers and characters) of the step in a form similar to a SAS data set.

*table definition*

A table definition is a description of how to render a tabular data component. (Almost all ODS output is tabular.) This description includes but is not limited to

- the order of the columns
- text and order of column headers
- formats for data
- font sizes and font faces.

*Note:* Not all procedures use a table definition.

Procedures like PROC CHART and PROC TIMEPLOT need to use a monospace font to correctly align their results. Such procedures do not use a table definition. The HTML and Printer output that they produce do not substantially differ from the Listing output. You cannot alter their HTML or Printer output except by specifying a different style definition. See the discussion of STYLE= on page 58 (for the HTML destination) or the discussion of STYLE= on page 68 (for the Printer destination).

Procedures like PROC PRINT, PROC REPORT, and PROC TABULATE produce an endless variety of results, depending on how you use the procedures. These procedures don't use a table definition either. However, both PROC REPORT and PROC TABULATE provide ways for you to customize their HTML and Printer output. For more information, consult "Customizing the Style Definition that ODS Uses" in "Fundamental Concepts for Using Base SAS Procedures" in *SAS Procedures Guide*, as well as the sections on PROC REPORT and PROC TABULATE procedures in the same document.  $\Delta$

*output object*

ODS binds a table definition to a data component in order to produce an output object. The output object, therefore, contains both the results of the procedure or DATA step and information about how to render the results. An output object has a name and a label.

*Note:* Although many output objects include a table definition, not all do. In some cases the output object is no more than the data component.  $\Delta$

*ODS destinations*

ODS currently supports four destinations:

- The Listing destination produces monospace output, which is formatted like traditional SAS procedure output.
- The HTML destination produces output that is formatted in Hypertext Markup Language.
- The Printer destination produces output that is formatted for high-resolution printers.
- The Output destination produces SAS data sets.

*ODS output*

ODS output consists of formatted output objects from any of the ODS destinations. The Output destination produces SAS data sets. The Listing, HTML, and Printer destinations produce Listing output, HTML output, and Printer output.

Traditional SAS output is, then, one kind of ODS output (Listing output), but it is no longer the only kind.

---

## ODS Destinations

ODS currently supports four destinations: the HTML destination, the Listing destination, the Output destination, and the Printer destination. ODS destinations can be open or closed. When a destination is open, ODS can send output objects to it. When a destination is closed, ODS cannot send output objects to it. An open destination uses system resources even if you use the selection and exclusion features of ODS to exclude all output objects from the destination. (See “Using Selection and Exclusion Lists” on page 23.) Therefore, to conserve resources, close all unnecessary destinations.

You open and close a destination with the appropriate ODS statement (ODS HTML, ODS LISTING, ODS PRINTER, or ODS OUTPUT). Refer to Chapter 3, “The ODS Statements,” on page 47 for more information on using each destination.

By default, the Listing destination is open, and all other destinations are closed. Consequently, if you do nothing, your SAS programs run and produce Listing output, just as they did in previous releases of the SAS System before ODS was available.

---

## Using Selection and Exclusion Lists

For each ODS destination, ODS maintains either a selection list (a list of output objects to send to the destination) or an exclusion list (a list of output objects to exclude from the destination). ODS also maintains an overall selection list or an overall exclusion list. You can use these lists to control which output objects go to which ODS destinations.

To see the contents of the lists use the ODS SHOW statement, which writes the lists to the SAS log (see “ODS SHOW Statement” on page 70). The following table shows the default lists:

**Table 2.1** Default List for Each ODS Destination

ODS Destination	Default List
HTML	SELECT ALL
Listing	SELECT ALL
Output	EXCLUDE ALL
Printer	SELECT ALL
Overall	SELECT ALL

## How to Specify an Output Object

In order to work with selection and exclusion lists, you must know how to specify an output object.

To specify an output object, you need to know what output objects your SAS program produces. The ODS TRACE statement writes to the SAS log a trace record that includes the path, the label, and other information about each output object that is produced. (See “ODS TRACE Statement” on page 71.) You can specify an output object as

- a full path (shown as the path in the trace record).
- a partial path. A partial path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For instance, if the full path is

```
Univariate.City_Pop_90.TestsForLocation
```

partial paths are

```
City_Pop_90.TestsForLocation
```

```
TestsForLocation
```

- a label (surrounded by quotation marks).
- a label path. (The trace record shows the label path only if you specify the LABEL option in the ODS TRACE statement.)
- a partial label path. A partial label path consists of any part of the label that begins immediately after a period (.) and continues to the end of the label. For instance, if the label path is

```
"The UNIVARIATE Procedure"."CityPop_90"  
."Tests For Location"
```

partial label paths are

```
"CityPop_90"."Tests For Location"
```

```
"Tests For Location"
```

- a mixture of labels and paths.

- any of these specifications followed by a pound sign (#) and a number. For example, TestsForLocation#3 refers to the third output object named TestsForLocation.

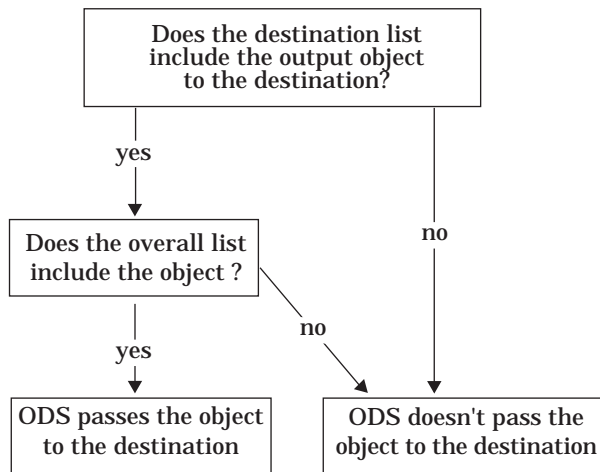
---

## How ODS Determines the Destinations for an Output Object

As each output object is produced, ODS uses the selection and exclusion lists to determine which destination or destinations to send it to. The following figure illustrates this process:

**Figure 2.2** Directing an Output Object to a Destination

For each destination, ODS first asks if the list for that destination includes the object. If it does not, ODS does not send the output object to that destination. If the list for that destination does include the object, ODS reads the overall list. If the overall list includes the object, ODS sends it to the destination. If the overall list does not include the object, ODS does not send it to the destination.



*Note:* Although you can maintain a selection list for one destination and an exclusion list for another, it is easier to understand the results if you maintain the same types of lists for all the destinations that you route output to.  $\Delta$

---

## Modifying Selection and Exclusion Lists

There are two ways to modify a selection or exclusion list:

- ODS automatically modifies lists at certain points in your SAS session.
- You explicitly modify a list when you submit a statement, like the ODS SELECT statement, that is designed to modify a list.

The Output destination behaves differently from the other destinations. Its selection list associates output objects with SAS data sets. As the objects are created, ODS sends them to the Output destination and creates the data sets. Whenever ODS closes a data set, it must remove the data set and the corresponding output objects from the selection list to avoid writing over the data set. In addition, in order to accommodate BY-group processing and RUN-group processing, ODS must sometimes modify the list for the output destination at the end of a run-group as well as at the end of a procedure or

DATA step. Therefore, the maintenance of the lists for the Output destination differs from the maintenance of the lists for other destinations.

This section explains automatic and explicit modifications of the lists for each destination.

## Automatic Modification of Lists for All Destinations except the Output Destination

At the end of each DATA step that uses ODS and at the end of each procedure step, ODS removes from the list all output objects that were not specified with the PERSIST option (see “ODS EXCLUDE Statement” on page 48 and “ODS SELECT Statement” on page 68). If ODS removes all objects from the list, it sets the list to its default of SELECT ALL. If any items in the list are specified with PERSIST, they remain in the list, and the resulting list is used.

If the list contains the argument ALL or NONE, it is not modified at the end of a procedure or DATA step. You must explicitly modify such a list (see “Explicit Modification of Lists All Destinations except the Output Destination” on page 26).

The following table illustrates how ODS modifies several different lists for the overall destination when a DATA step that uses ODS or a procedure step ends.

**Table 2.2** Automatically Modifying the Overall List at the End of a DATA or Procedure Step

Initial list	List after a DATA or procedure step ends
select moments quantiles	select all
exclude moments quantiles	select all
select moments(persist) quantiles	select moments(persist)
exclude moments(persist) quantiles	exclude moments(persist)
exclude all	exclude all

## Explicit Modification of Lists All Destinations except the Output Destination

There are three ways to explicitly modify the list for the Overall, HTML, Listing, or Printer destination with the ODS EXCLUDE and ODS SELECT statements:

- Change the list from a selection list to an exclusion list or from an exclusion list to a selection list. In this case, only the ODS statement that you use determines the contents of the new list.
- Change the scope of the list with the ALL or the NONE argument. In this case, only the ODS statement that you use determines the contents of the list.
- Specify individual objects in the ODS SELECT or the ODS EXCLUDE statement without changing the list from one type to another. In this case, you add new items to the list or change whether or not the PERSIST option is used with an item.

*Note:* You cannot remove items from the list this way.  $\Delta$

In addition, the following statements not only explicitly reset the overall list as specified, but they also set the list for each individual destination to its default.

- ODS SELECT ALL
- ODS SELECT NONE
- ODS EXCLUDE ALL

ODS EXCLUDE NONE

The following table illustrates these ways of modifying the HTML list:

**Table 2.3** Explicitly Modifying the List for the HTML Destination

If the initial list for the HTML destination is ...	And you specify ...	The new list for the HTML destination is ...
select moments(persist) quantiles	ods html exclude moments;	exclude moments
select moments(persist) quantiles	ods html select all;	select all
select moments(persist) quantiles	ods html select moments extremeobs(persist);	select moments quantiles extremeobs(persist)
select moments(persist) quantiles	ods select all;	select all

### Automatic Modification of the List for the Output Destination

ODS automatically removes an object from the Output selection list when it closes the data set that it is making from that object. The time at which the data set is closed depends on

- whether BY-group processing is in effect
- whether the PERSIST= option is in effect.

The following table shows when the data sets are automatically closed and the list is modified.

**Table 2.4** Automatically Modifying the List for the Output Destination

BY-group processing?	PERSIST=	How ODS Modifies the List
No	Not used	Closes the data set and removes the output object from the list as soon as it has made the data set. Removes all objects from the list when a DATA step that uses ODS or a procedure ends.
Yes	Not used	Closes all open data sets and removes all objects from the list when a run-group ends or when a DATA step that uses ODS or a procedure ends. This configuration lets you put data from multiple BY groups into one data set.

Table 2.4 (continued)

BY-group processing?	PERSIST=	How ODS Modifies the List
Yes or no	RUN	Closes all open data sets and removes all objects from the list when a DATA step that uses ODS or a procedure ends. This configuration lets you put data from multiple run groups into one data set.
Yes or no	PROC	Closes all open data sets and removes all objects from the list at the end of the SAS session.

### Explicit Modification of the List for the Output Destination

You modify the list for the Output destination with the ODS OUTPUT statement (see “ODS OUTPUT Statement” on page 61). Use ODS OUTPUT with one or more *data-set-specifications* to add objects to the list or to modify the associations between output objects and data sets.

The following statements always reset the list for the Output destination to EXCLUDE ALL.

- ODS OUTPUT CLEAR
- ODS SELECT ALL
- ODS SELECT NONE
- ODS EXCLUDE ALL
- ODS EXCLUDE NONE

### Resetting Lists when RUN-Group Processing Is in Effect

Remember that for procedures that support RUN-group processing (such as DATASETS, GLM, and REG), the RUN statement does not end the procedure. A QUIT statement explicitly ends such a procedure. If you omit a QUIT statement, a PROC or DATA statement implicitly ends such a procedure step.

When you are using ODS, it is wise to specify a QUIT statement at the end of every procedure that supports run-group processing. If you end every such procedure step explicitly, rather than waiting for the next PROC or DATA step to end it for you, the QUIT statement resets the selection list. The following figure illustrates the difference that a QUIT statement makes:



**Figure 2.3** What Happens When You Leave Off the QUIT Statement?

Assume that the overall list is set to the default of SELECT ALL.  
Set the Listing selection list to SELECT ANOVA.

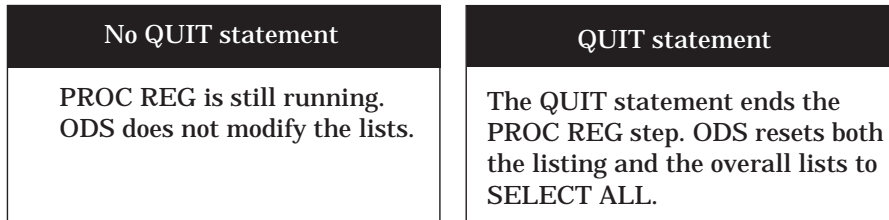
```
ods listing select anova;
```

Run PROC REG.

```
proc reg data=iron;
  model loss=fe;
run;
```

PROC REG sends one output object to the Listing destination.

ANOVA



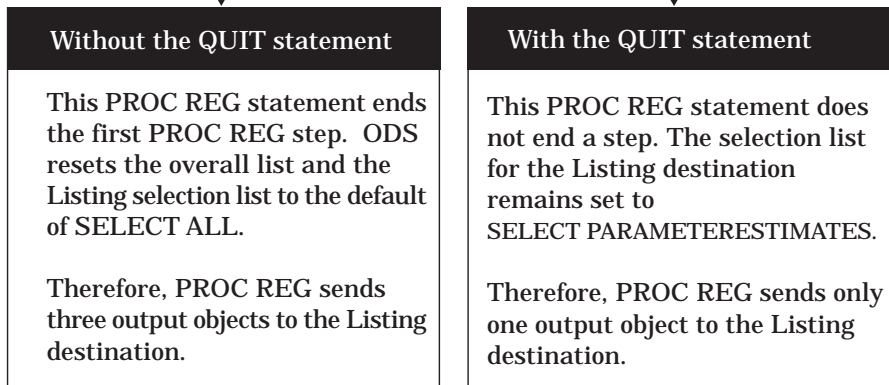
```
quit;
```

Modify the listing selection list to SELECT PARAMETERESTIMATES.

```
ods listing select parameterestimates;
```

Execute the PROC REG step.

```
proc reg data=iron;
  model loss=fe;
run;
```



ANOVA

FitStatistics

ParameterEstimates

ParameterEstimates

For an example of a SAS program that illustrates how the lists are written and cleared, see Example 4 on page 86.

---

## Understanding Some HTML Basics

Hypertext Markup Language (HTML) is a formatting language that is widely used, particularly on the World Wide Web. Browsers such as Netscape and Internet Explorer read HTML. In order to take full advantage of the features of ODS, it is helpful to understand something about HTML.

---

### Using HTML Tags

A browser that reads HTML knows how to format the text in the file because *tags* in the file provide the necessary instructions. In general, tags come in pairs: a begin tag and an end tag. The browser applies the tagging instructions to all the text between the begin and end tags. In the ASCII file that contains the HTML, each begin tag is the name of the tag inside angle brackets. The end tag is the same as the begin tag except that a forward slash (/) precedes the name of the tag. For example, the following example of HTML contains two sets of tags: the paragraph tags (<P> and </P>) and the italic tags (<I> and </I>).

```
<P> For information on the REPORT procedure,
see <I>SAS Procedures Guide, Version 7, First
Edition</I>.</P>
```

Different browsers may implement tags differently. The way that paragraphs and italics appear on your display, therefore, depends on the browser and on the fonts available to you.

---

### Using HTML Attributes

HTML attributes modify a tag. They may provide formatting information, or they may provide information that the browser needs to implement an HTML feature like a link. For example, the following FONT tag includes attributes that specify the face, size, and color to use:

```
<FONT face="Arial, Helv" size="3" color="#002288">
```

For information on the attributes that you can specify through ODS, see “Style Attributes” on page 192.

---

### Using URLs and Filenames

Any HTML file that you create has a location in the file structure of your operating system. Anyone at your site who has access to the file system can read such a file with an HTML browser.

However, for such a file to be available on the World Wide Web, it must be available on a web server. In order for the web server to be able to locate the file, the file must have a *uniform resource locator* (URL). A URL provides an address on a web server that users who do not have access to the file system can find. Although the URL and the operating system’s file name both point to the same file, the URL does so in a way that protects the integrity of the file system and limits access to selected files. At the same

time, the URL can potentially provide access to many more people, both on and off your site, than the filename can.

For example, suppose that you have this special location in your file system where you can place any HTML files that you wish to make public:

```
/rnd/users/smith/myhtmlfiles
```

Users with access to the file system can use this name to browse the files with a web browser. However, people accessing the files through the Web must use a URL.

Suppose, for example, that your company has a URL of `http://www.your-company.com`. Your personal files have their own local URL, preceded by the company's URL.

If your files in `/rnd/users/smith/myhtmlfiles` have a local URL of `smithhtml`, the URL that you supply for web access to a file in this location is

```
http://www.your-company.com/smithhtml/filename
```

*Note:* The URL and the file system name usually differ somewhat. This difference to some extent masks the structure of the file system from outside users. In addition, most sites have a *firewall* around their internal files. A firewall allows only those users whom it recognizes to access web files. Thus some HTML files on your system may be available to the public while others may be protected by a firewall.  $\Delta$

It is important to understand a little about URLs when you use the HTML destination because the implementations of links and cross references often use URLs (see "HTML Links and References" on page 31).

---

## HTML Links and References

---

### What Are Links and References?

An HTML link is a place in a document that allows you to jump to another specific place in the same document or in another document. A browser typically highlights the text that is between the tags that begin and end the link. When you click on the highlighted text, the browser displays the text at the link target. The browser might then display the contents of the target in the active window, or it might open another browser window that displays the contents of the target.

An HTML reference names a file for the browser to display. When a browser reads a reference, it displays the referenced file as if it were part of the file that it is displaying. You can't tell by looking at the browser's display that some of the material is in the file that you are actually viewing and that some is referenced.

When you use ODS, the software automatically creates the links and references that you need. You can, however, customize these links to some extent. If you wish to do so, you will need to understand how HTML implements links and references.

---

### Implementing HTML Links and References

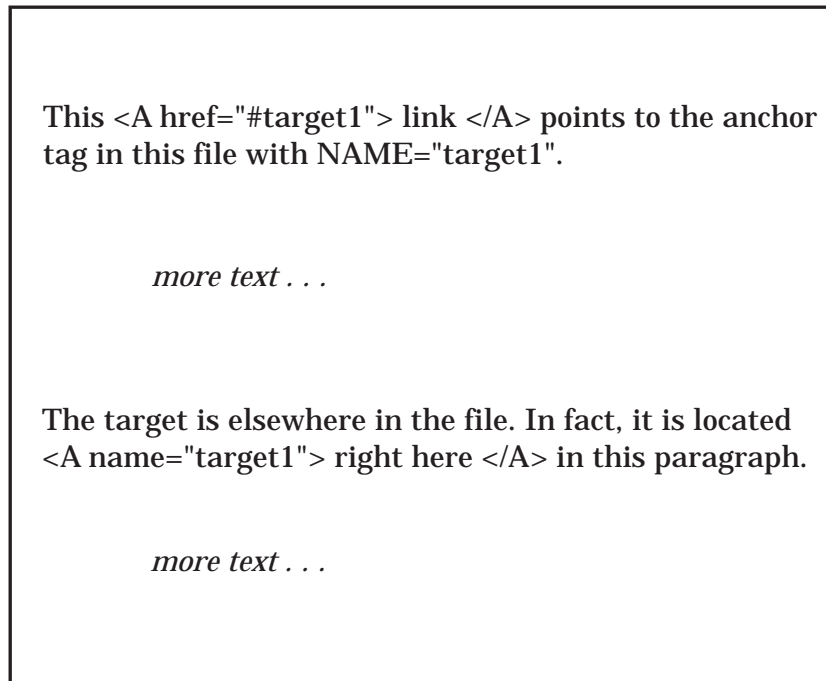
*Note:* This simplified discussion of HTML links and references is designed to provide information that will help you understand what ODS does when it builds links and references for you. For a complete discussion of HTML tagging, consult one of the many reference books that are available on the subject.  $\Delta$

Each link in HTML is implemented with a combination of two sets of `<A>` (anchor) tags. One anchor tag, which is the starting point of the link, has an `HREF` attribute that identifies the anchor tag to link to. The other anchor tag, which is the target of the

link, has a NAME attribute. This NAME attribute is what the HREF attribute in the first anchor tag points to. The value of each NAME attribute in a file must be unique so that each value of HREF points to a single, unambiguous location. The following figure illustrates linking within a file:

**Figure 2.4** Linking within a File

The browser highlights the word **link**. When you click on **link**, the browser positions the target **right here** in the active window.



The important features at the starting point of this link are

- The `<A>` and `</A>` tags surround the text that the browser will highlight.
- The HREF attribute points to the link's target. The target is an anchor tag whose NAME attribute matches the text that follows the pound sign in the HREF attribute. Because no text precedes the pound sign (`#`), the browser knows that the target is in the same file as the anchor.

When a link points to a target outside the file that is being displayed, the HREF attribute must include the path to that file. The path can be the path within the file system or the uniform resource locator (URL) of the file. The following figure illustrates a link from one file to another file that is specified with a URL:

**Figure 2.5** Linking to Another File

The browser highlights the word **link**. When you click on **link**, the browser positions the target **right here** in the active window or opens another window that displays the target.

File: /users/brown/documents/file1

URL: *http://www.company-url*  
*/local-url/file1*

This

`<A href="http://www.company-url/local-url/file2#target1">link</A>`  
points to an anchor tag in the file with the specified URL. The  
NAME attribute on the target anchor tag is "target1".

File: /users/brown/documents/file2

URL: *http://www.company-url*  
*/local-url/file2*

The target is in this file. In fact, it is located  
`<A name="target1">right here</A>` ←  
in this sentence.

The important features at the starting point (the anchor) of the link are

- The `<A>` and `</A>` tags surround the text that the browser will highlight.
- The HREF attribute points to the link's target. The text that precedes the pound sign (#) identifies the file that contains the target.

ODS provides features that enable you to customize the text that precedes the pound sign and the text that follows the pound sign. For information on how to do this, see the discussions of HTML-file-specification on page 52, ANCHOR= on page 54, BASE= on page 55, PATH= on page 57, and GPATH= on page 55 as well as "How ODS Constructs Links and References" on page 34.

HTML implements references in much the same way as it implements links. The main difference is that a link points to a particular location within a file and that a reference points to the file itself. HTML uses the SRC attribute to identify a file to reference. The value of the SRC attribute is constructed the same way that the value of the HREF attribute is constructed except that there is no pound sign and no text following it.

## How ODS Constructs Links and References

Several options in the ODS HTML statement affect how ODS constructs the links and references that point from the frame to the table of contents, table of pages, and body file and from the table of contents or table of pages to the body file. Links are made as HREF attributes on <A> (anchor) tags inside the HTML files. Each HREF attribute points to the NAME attribute on another <A> tag. The HREF must identify both the file that contains the target and the name of the anchor within that file. The value of HREF must be a valid target in a valid URL. It uses the following form:

```
<A href="URL#anchor-name">
```

ODS constructs the value of an HREF attribute based on information that you provide in the ODS HTML statement.

*Note:* HTML references to files use other tags, but the logic for creating the string that identifies the file is the same as the logic for creating an HREF attribute (see “Implementing HTML Links and References” on page 31).  $\Delta$

The URL in an HREF attribute is composed of information from three options in the ODS HTML statement: the BASE= option; the GPATH= or the PATH= option; and the BODY=, the CONTENTS=, or the PAGE= option.

- 1 If you specify BASE=, the value of that option is the first part of the URL for every HREF attribute that ODS writes.
- 2 If you specify GPATH= or PATH=, the next part of the URL in an HREF attribute comes from that option.

If the file that you are linking to is a high-resolution graphic, ODS uses information from the GPATH= option as the next part of the HREF. (For information on these options, see the discussion of GPATH= on page 55 and the discussion of PATH= on page 57.) The following table shows how ODS uses information from the GPATH= option in the URL in HREF attributes:

**Table 2.5** Building an HREF Attribute from the GPATH= Option

If the <i>file-specification</i> in GPATH= is ...	And the URL= suboption is ...	ODS uses this information in the second part of the URL in the HREF attribute* ...
an <i>external-file</i> or <i>libref.catalog</i>	not specified	the name of the file
an <i>external-file</i> or <i>libref.catalog</i>	specified, but not NONE	the value of the URL= suboption
an <i>external-file</i> or <i>libref.catalog</i>	NONE	No information from GPATH=
a <i>fileref</i>	specified or not specified	No information from GPATH=

\* If you do not specify GPATH=, ODS uses the value of PATH= to create this part of the HREF.

If the file that you are linking to is not a high-resolution graphic, ODS uses information from the PATH= option as the next part of the HREF. The following table shows how ODS uses information from the PATH= option in the URL in HREF attributes:

**Table 2.6** Building an HREF Attribute from the PATH= Option

If the <i>file-specification</i> in PATH= is ...	And the URL= suboption is...	ODS uses this information in the second part of the URL in the HREF attribute ...
an <i>external-file</i> or <i>libref.catalog</i>	not specified	the name of the file
an <i>external-file</i> or <i>libref.catalog</i>	specified, but not NONE	the value of the URL= suboption
an <i>external-file</i> or <i>libref.catalog</i>	NONE	No information from PATH=
a <i>fileref</i>	specified or not specified	No information from PATH=

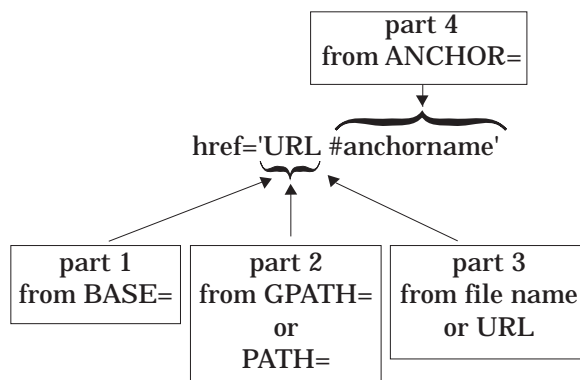
*Note:* If you use a *fileref* as the file specification in the BODY=, CONTENTS=, or PAGE= option in the ODS HTML statement, and you do not use the URL= suboption in that option, ODS does not use information from GPATH= or PATH= when it creates the complete URL for any corresponding HREF attributes.  $\Delta$

- 3 The last part of the URL that is used in an HREF attribute is, by default, the name of the file that contains the target. ODS determines the name of the file from the *file-specification* that you use in the BODY=, CONTENTS=, or PAGE= option. (ODS does not create links or references to frame files.) For more information on these options, see the discussion of HTML-file-specification on page 52.)

If you specify the URL= suboption in one of these options, ODS uses the string that you specify instead of the file name.

*Note:* If you use a *fileref* as the file specification and do not use the URL= suboption, ODS does not use information from GPATH= or PATH= when it creates the complete URL for the HREF attribute.  $\Delta$

The *anchor-name* comes from the value of the ANCHOR= option. The following figure illustrates the creation of the HREF:

**Figure 2.6** Creating the Value of an HREF Attribute

---

## Files Produced by the HTML Destination

The HTML destination can produce four kinds of files: body, contents, frame, and page files. You create these files with options in the ODS HTML statement (see “ODS HTML Statement” on page 50 for details).

---

### The Body File

The body file contains HTML output that is generated from the output objects that your SAS job creates. The style definition and the table definition that the job uses determine the appearance and content of the tables and the cells within them.

Typically, when you route an output object that does not contain graphics to the HTML destination, ODS places the results within TABLE tags, rendering them as one or more HTML tables.

Graphics output is produced according to the SAS code that generates it. Instead of using <TABLE> tags, the body file contains an <IMG> (image) tag that references the graphic. When you view the body file in a browser, you cannot tell that the graphic is not part of the body file because the <IMG> tag displays it in the browser.

*Note:* A very few procedures produce output objects that are neither tabular nor graphics. In these cases, the output is not tagged as an HTML table.  $\Delta$

Titles and footnotes in the body file are rendered as HTML tables of their own near the top and bottom of each page of HTML output.

*Note:* For graphics output, titles and footnotes are, by default, part of the graphics file. You can use the NOGTITLE and NOGFOOTNOTE options to place them in the body file instead. See the discussion of GTITLE on page 56 and GFOOTNOTE on page 55 for more information.  $\Delta$

All <TABLE> tags and all <IMG> tags are potential targets for links or references (see “HTML Links and References” on page 31). Therefore, ODS must provide an <A> tag with a NAME attribute close to each <TABLE> and <IMG> tag for links and references to point to. The NAME attribute on the anchor tag becomes the final part of any reference or link to the table. ODS inserts anchor tags in its HTML output as follows:

- ODS places an anchor tag near the top of each page, before all tables on the page (including the table that holds the titles) and before all images. This anchor is the target for links to the first table (excluding any titles) or to the first image on the page.

*Note:* Each procedure or DATA step starts a new page. In addition, ODS produces a new page of output whenever the SAS program explicitly asks for a new page. For example, if you use the page dimension in PROC TABULATE, you create a page for each value of the variable that defines the pages. In this context, the word page has nothing to do with the PAGESIZE= setting in your SAS session.  $\Delta$

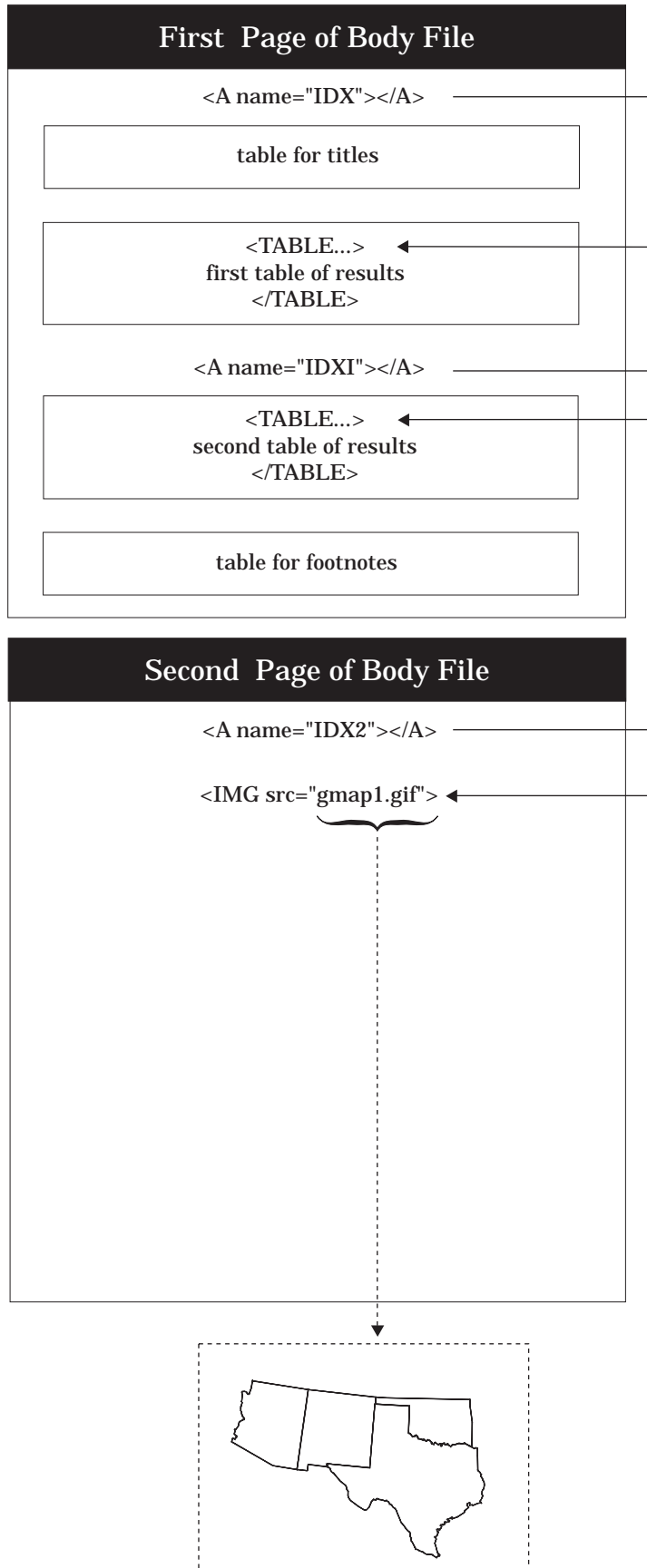
- ODS places an anchor tag slightly before each <TABLE> tag, provided that the table contains results (not titles or footnotes) and that it is not the first table or image on the page.
- ODS places an anchor tag slightly before each <IMG> tag, provided that it is not the first table or image on a page.

The following figure illustrates the placement of anchor tags from a SAS job that executes two procedures. The first procedure creates two HTML tables of results on a single page. The page also includes an HTML table for the title and one for the



footnote. Solid arrows indicate which `<A>` tag ODS uses as a target for each table. The second procedure creates a GIF file. The titles for this procedure are part of the GIF file (the default behavior). Again, the solid arrow indicates which anchor tag ODS uses as a target when it creates a link to the image. The dashed arrow points to the file that the `<IMG>` tag references.

Figure 2.7 Placement of <A> (anchor) Tags in HTML Output



For a view of this same file through a browser, see Display 2.1 on page 41.

---

## The Contents File

The contents file contains a link to the body file for each HTML table that ODS creates from procedure or DATA step results. The targets for these links are the values of the NAME attributes on the anchor tags that are in the body file (see “The Body File” on page 36). For example, an anchor tag that links to the second HTML table of results in Figure 2.7 on page 38 looks like this:

```
<A href="pop-body.htm#IDX1">
```

In this anchor tag

- pop-body.htm identifies the file that contains the target
- #IDX1 provides the name of the target.

You can view the contents file directly in the browser, or, if you make a frame file, you can see the contents file as part of the frame file (see “The Frame File” on page 39).

---

## The Page File

The page file contains a link to the body file for each page of HTML output that ODS creates from procedure or DATA step results. The targets for these links are the values of the NAME attributes on the anchor tags that are in the body file (see “The Body File” on page 36). For example, an anchor tag that links to the second page of results in Figure 2.7 on page 38 looks like this:

```
<A href="pop-body.htm#IDX2">
```

In this anchor tag

- pop-body.htm identifies the file that contains the target.
- #IDX2 provides the name of the target

You can view the page file directly in the browser, or, if you make a frame file, you can see the page file as part of the frame file (see “The Frame File” on page 39).

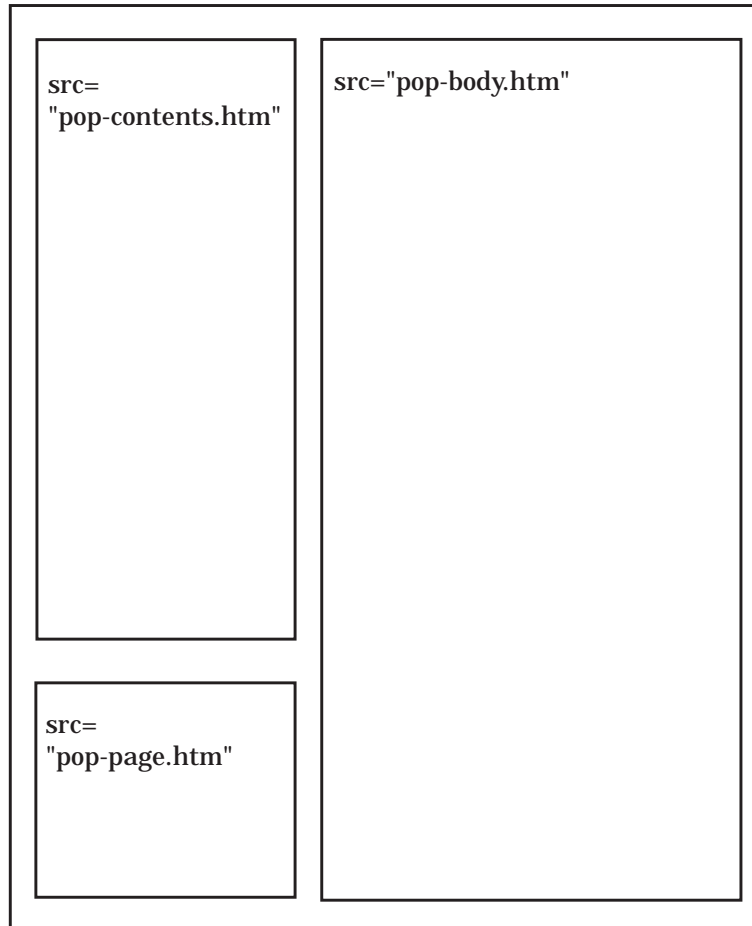
---

## The Frame File

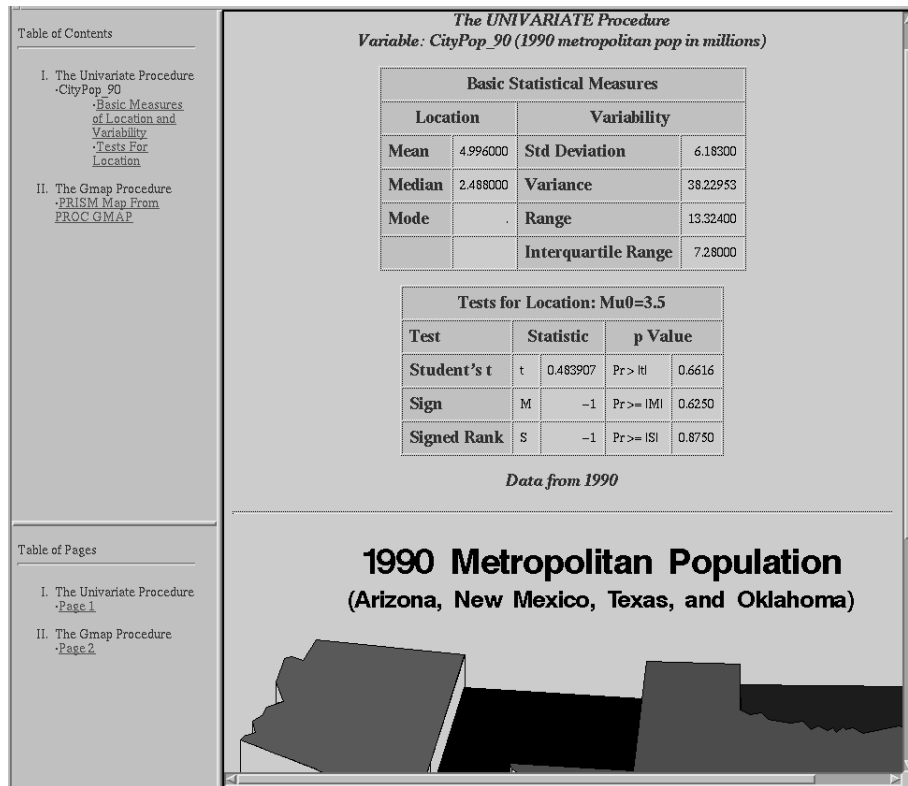
The frame file provides a simultaneous view of the body file and the contents file, the page file, or both. The following figure illustrates how a frame that references both the contents and page files looks (in part) to an ASCII editor. The SRC attribute identifies a file to display in the browser. ODS constructs the value for the SRC attribute the same way that it constructs the value for an HREF attribute in a page or contents file (see “The Contents File” on page 39).

**Figure 2.8** Schematic of an HTML Frame File

HTML Frame File: pop-frame.htm



Display 2.1 on page 41 shows the same frame file viewed from a browser.

**Display 2.1** Browser View of HTML Frame File

## Customizing ODS Output

By default, ODS output is formatted according to instructions that the procedure or DATA step points to. However, ODS provides ways for you to customize the output. You can customize the output for an entire SAS job, or you can customize the output for a single output object.

## Style Definitions, Style Elements, and Style Attributes

To customize the output at the level of the entire SAS job, you specify a *style definition*. A style definition describes how to render the presentation aspects (color, font face, font size, and so forth) of an entire SAS job. A style definition determines the overall look of the documents that use it.

Each style definition is composed of *style elements*. A style element is a collection of style attributes that apply to a particular part of the output. For instance, a style element may contain instructions for the presentation of column headers or for the presentation of the data inside cells. Style elements may also specify default colors and fonts for output that uses the style definition.

Each *style attribute* specifies a value for one aspect of the presentation. For instance, the BACKGROUND= attribute specifies the color for the background of an HTML table or for a colored table in Printer output. The FONT\_STYLE= attribute specifies whether to use a Roman, a slant, or an italic font. For information on style attributes, see “Style Attributes” on page 192.

*Note:* Because style definitions control the presentation of the data, they have no effect on output objects that go to the Listing or Output destination.  $\triangle$

---

## Table Definitions, Table Elements, and Table Attributes

A *table definition* describes how to render the output for a tabular output object. (Almost all ODS output is tabular.) A table definition determines the order of column headers and the order of variables, as well the overall look of the output object that uses it.

In addition to the parts of the table definition that order the headers and columns, each table definition contains or references *table elements*. A table element is a collection of table attributes that apply to a particular header, footer, or column. Typically, a *table attribute* specifies something about the data rather than about its presentation. For instance FORMAT specifies the SAS format to use in a column. However, some table attributes describe presentation aspects of the data.

*Note:* The parts of table definitions that control the presentation of the data have no effect on output objects that go to the Listing or Output destination. However, the parts that control the structure of the table and the data values do affect Listing output.  $\triangle$

For information on table attributes, see “Table Attributes” on page 210.

---

## Customizing Presentation Aspects of ODS Output

To specify a style definition for your ODS output, use the STYLE= option in the ODS statement for the destination or destinations that you are using (see Example 7 on page 100). Note that you must specify the option for each destination. Your site may have its own customized style definitions that are used for all HTML and Printer output to give the output a consistent appearance.

Some of the standard style definitions that are currently shipped with the SAS System include

default  
 beige  
 brick  
 brown  
 D3D  
 minimal  
 printer  
 statdoc

To find an up-to-date list of the style definitions:

- 1 Select



from the Explorer.

- 2 In the Results window, select the Results folder. With your cursor on this folder, use your right mouse button to open the Templates window.
- 3 In the Templates window, select and open `Sashe1p.tmplmst`.

- 4 Select **styles**, and use your right mouse button to open this folder, which contains a list of available style definitions. If you want to view the underlying SAS code for a style definition, select and open it.

*Operating Environment Information:* For information on navigating in the Explorer window without a mouse, see the section on “Window Controls and General Navigation” in the SAS documentation for your operating environment.  $\Delta$

You can also display a list of the available styles by submitting the following PROC TEMPLATE step:

```
proc template;
  list styles;
run;
```

You can create your own style definitions or modify existing ones with PROC TEMPLATE (see Chapter 5, “The TEMPLATE Procedure,” on page 131).

## Customizing Output for a Tabular Output Object

For a procedure, the name of the table definition that is used for an output object comes from the procedure code. The DATA step uses a default table definition unless you specify an alternative with the TEMPLATE= suboption in the ODS option in the FILE statement (see the discussion of TEMPLATE= on page 110).

To find out which table definitions a procedure or the DATA step uses for the output objects that it creates, look at a trace record. To produce a trace record in your SAS log, submit the following SAS statements:

```
ods trace on;
your-proc-or-DATA-step
ods trace off;
```

The trace record refers to the table definition as a template. For a detailed explanation of the trace record, see “Contents of the Trace Record” on page 72.

*Note:* Remember that not all procedures use a table definition. If you produce a trace record for one of these procedures, no definition appears in the trace record.  $\Delta$

You can use PROC TEMPLATE to modify a table definition as a whole, but, in general, you can’t directly specify a table element for your procedure or DATA step to use without modifying the definition itself. When a procedure or DATA step uses a table definition, it uses the table elements that that definition defines or references.

*Note:* Two base procedures, PROC REPORT and PROC TABULATE, do provide a way for you to access table elements from the procedure step itself. Accessing the table elements enables you to do such things as specify background colors for certain cells, change the font face for column headers, and so forth. For more information, consult “Customizing the Style Definition that ODS Uses” in “Fundamental Concepts for Using Base SAS Procedures” in *SAS Procedures Guide*, as well as the documentation for these two procedures in the same document.  $\Delta$





The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *The Complete Guide to the SAS® Output Delivery System, Version 8*, Cary, NC: SAS Institute Inc., 1999. 310 pp.

**The Complete Guide to the SAS® Output Delivery System, Version 8**

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-425-X

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 the 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.

1st printing, October 1999

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.

The Institute is a private company devoted to the support and further development of its software and related services.