SAS/STAT® 13.1 User’s Guide
Customizing the Kaplan-Meier Survival Plot
Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore
for additional books and resources.
Chapter 23
Customizing the Kaplan-Meier Survival Plot

Contents

Overview ................................................................. 780
Controlling the Survival Plot by Specifying Procedure Options ........................................ 781
  Enabling ODS Graphics and the Default Kaplan-Meier Plot ........................................ 781
  Individual Survival Plots ........................................... 783
  Hall-Wellner Confidence Bands and Homogeneity Test .................................................. 785
  Equal-Precision Bands ............................................. 786
  Displaying the Patients-at-Risk Table inside the Plot ...................................................... 788
  Displaying the Patients-at-Risk Table outside the Plot ..................................................... 790
  Modifying At-Risk Table Times ...................................................................................... 791
  Reordering the Groups .................................................................................................... 794
  Suppressing the Censored Observations ......................................................................... 797
  Failure Plots ................................................................................................................... 798

Controlling the Survival Plot by Modifying Graph Templates ............................................. 799
  The Modularized Templates ............................................................................................. 799
  Changing the Plot Title .................................................................................................... 801
  Modifying the Axis .......................................................................................................... 803
  Changing the Line Thickness ............................................................................................ 805
  Changing the Group Color ............................................................................................... 806
  Changing the Line Pattern ............................................................................................... 807
  Changing the Font ............................................................................................................ 808
  Changing the Legend and Inset Position ........................................................................... 810
  Changing How the Censored Points Are Displayed .......................................................... 812
  Adding a Y-Axis Reference Line ....................................................................................... 813
  Changing the Homogeneity Test Inset .............................................................................. 815
  Suppressing the Second Title and Adding a Footnote ..................................................... 817
  Adding a Small Inset Table with Event Information ......................................................... 818
  Adding an External Table with Event Information ............................................................. 820
  Suppressing the Legend .................................................................................................... 822
  Kaplan-Meier Plot with Event Table and Other Customizations ....................................... 823
  Compiled Template Cleanup ............................................................................................. 824

Graph Templates, Macros, and Macro Variables ................................................................. 825
  The Macro Variables ........................................................................................................ 827
  The Smaller Macros ......................................................................................................... 830
  The Larger Macros .......................................................................................................... 830
  Event Table Macros ........................................................................................................ 835
Overview

The LIFETEST procedure is a nonparametric procedure for analyzing survival data. You can use PROC LIFETEST to compute the Kaplan-Meier curve (1958), which is a nonparametric maximum likelihood estimate of the survivor function. The Kaplan-Meier plot (also called the product-limit survival plot) is a popular tool in medical, pharmaceutical, and life sciences research. The Kaplan-Meier plot contains step functions that represent the Kaplan-Meier curves of different samples (strata). The Kaplan-Meier plot has many other features that you can add or change through procedure options, graph templates, and style templates. This chapter explores these features in detail but does not explain how to interpret the graphs or the underlying analysis. For more information about PROC LIFETEST and the Kaplan-Meier plot, see Chapter 56, “The LIFETEST Procedure.”

This chapter shows you how to modify the Kaplan-Meier plot through a series of examples. It discusses four types of examples: specifying procedure options, modifying graph templates by using macro variables, modifying graph templates by using macros, and changing styles. Most examples do not go into detail about the tools that underlie the template changes. Each example is designed to be small, simple, self-contained, and easy to copy and use “as is” or with minor modifications. Subsequent sections provide more details about the macro variables and macros that are used to modify the graph templates. You can use the simple examples to make a wide variety of changes without reading or understanding the detailed descriptions at the end of this chapter.

Statistical procedures produce tables by using the Output Delivery System (ODS) and produce graphs by using ODS Graphics. Procedures produce graphs as automatically as they produce tables, and graphs and tables are integrated in the ODS output. Graphs that are produced by ODS Graphics are controlled by options, the data object (the matrix of information that is graphed), a style template, and a graph template. A style template is a SAS program that controls the overall appearance of graphs, including colors, line and marker styles, sizes, fonts, and so on. A graph template is a SAS program, written in the Graph Template Language
(GTL), that provides a detailed specification of the layout and contents of each graph. Each graph that is created when ODS Graphics is enabled is controlled by a graph template.\footnote{ODS Graphics might or might not be enabled by default. ODS Graphics is usually enabled by default in the SAS windowing environment and disabled when you invoke SAS in other ways. However, these defaults can be changed in a number of ways. ODS Graphics is enabled in the first example in this chapter by the ODS GRAPHICS ON statement and remains enabled throughout the chapter.}

If you want to modify a graph template, you usually use the TEMPLATE procedure to display the template of interest, and then you copy it into your editor, modify it, and submit it to SAS to compile. Then, when you run your procedure, it uses the new template. The PROC LIFETEST survival plot is the only plot in SAS for which you have another alternative available for template modification. SAS provides the survival plot templates in a series of macros and macro variables that are modular and easier to modify than the original templates. This chapter provides numerous examples of using these macros and macro variables.

The data that are used in this chapter come from 137 bone marrow transplant patients in a study by Klein and Moeschberger (1997) and are available in the BMT data set in the Sashelp library. At the time of transplant, each patient is classified in one of three risk categories: ALL (acute lymphoblastic leukemia), AML (acute myelocytic leukemia)–Low Risk, and AML–High Risk. The endpoint of interest is the disease-free survival time, which is the time in days until death, relapse, or the end of the study. The variable Group represents the patient’s risk category, the variable T represents the disease-free survival time, and the variable Status is the censoring indicator. A status of 1 indicates an event time, and a status of 0 indicates a censored time.

Controlling the Survival Plot by Specifying Procedure Options

This section provides a series of examples that use ODS Graphics and the PLOTS= option in the PROC LIFETEST statement to control the appearance of the survival plot. Other examples use formats and the ORDER= option to control the order of the groups.

Enabling ODS Graphics and the Default Kaplan-Meier Plot

You can use the following statements to enable ODS Graphics and run PROC LIFETEST:

```sas
ods graphics on;
proc lifetest data=sashelp.BMT;
time T * Status(0);
strata Group;
run;
```

ODS Graphics is enabled for this step and all subsequent steps until it is disabled. ODS Graphics remains enabled throughout the examples in this chapter.

You specify in the TIME statement that the disease-free survival time is recorded in the variable T. You further specify that the variable Status indicates censoring and 0 indicates a censored time. Separate survivor functions are displayed for each group in the Group variable, which you specify in the STRATA statement.

The plot in Figure 23.1 consists of three step functions, one for each of the three groups of patients. The plot shows that patients in the AML–Low Risk group have longer disease-free survival than patients in the ALL and AML–High Risk groups.
The following step, which explicitly specifies the default PLOTS=SURVIVAL option, is equivalent to the preceding step:

```
proc lifetest data=sashelp.BMT plots=survival;
  time T * Status(0);
  strata Group;
run;
```

The PLOTS= option enables you to control the graphs that a procedure produces. You can use it to request nondefault graphs and specify options for some graphs. You can specify graph names (PLOTS=SURVIVAL), graph options (PLOTS=SURVIVAL(ATRISK OUTSIDE)), and suboptions (PLOTS=SURVIVAL(ATRISK OUTSIDE(0.15))). The PLOTS= option is described in the section “PROC LIFETEST Statement” on page 4332 in Chapter 56, “The LIFETEST Procedure.”
Individual Survival Plots

You can use the STRATA=INDIVIDUAL option to request individual survival plots. By default, the STRATA=OVERLAY option produces the plot of overlaid step functions displayed in Figure 23.1. You can run the same analysis but request the results in three separate graphs, one per patient group, as follows:

```
proc lifetest data=sashelp.BMT plots=survival(strata=individual);
  time T * Status(0);
  strata Group;
run;
```

The first of the three survival plots is displayed in Figure 23.2. To conserve space, the other graphs are not displayed.

**Figure 23.2** One of Three Individual Plots

You can use the STRATA=PANEL option as follows to display the results in separate panels of a single graphical display:

```
proc lifetest data=sashelp.BMT plots=survival(strata=panel);
  time T * Status(0);
  strata Group;
run;
```

The results are displayed in Figure 23.3.
The rest of this chapter discusses overlaid plots such as the one displayed in Figure 23.1.
Hall-Wellner Confidence Bands and Homogeneity Test

You can use the following statements to add Hall-Wellner confidence bands (Hall and Wellner 1980) to Figure 23.1 and display the $p$-value from a test that the strata are homogeneous:

```
proc lifetest data=sashelp.BMT plots=survival(cb=hw test);
    time T * Status(0);
    strata Group;
run;
```

The results are displayed in Figure 23.4. The Hall-Wellner confidence bands extend to the last event times. The small $p$-value supports rejecting the hypothesis that the groups are homogeneous.

**Figure 23.4** Confidence Bands and Homogeneity Test
Equal-Precision Bands

You can use the following statements to add equal-precision bands to the plot:

```
proc lifetest data=sashelp.BMT plots=survival(cb=ep test);
  time T * Status(0);
  strata Group;
run;
```

The results are displayed in Figure 23.5.

![Figure 23.5 Equal-Precision Bands](image)

You can use the following statements to add both Hall-Wellner and equal-precision bands to the plot:

```
proc lifetest data=sashelp.BMT plots=survival(cb=all test);
  time T * Status(0);
  strata Group;
run;
```

The results are displayed in Figure 23.6.
Figure 23.6  Hall-Wellner and Equal-Precision Bands

Product-Limit Survival Estimates
With 95% Hall-Wellner and Equal Precision Bands

+ Censored
Logrank p=0.0010

Disease Group  ALL  AML-High Risk  AML-Low Risk
Displaying the Patients-at-Risk Table inside the Plot

You can add the patients-at-risk table to the Kaplan-Meier plot as follows:

```sas
proc lifetest data=sashelp.BMT plots=survival(cb=hw test atrisk);
   time T * Status(0);
   strata Group;
run;
```

The results are displayed in Figure 23.7. By default, the at-risk table is displayed inside the body of the plot. This table shows the number of patients who are at risk for each group for each of the different times. For these data, the default survival times at which at-risk values are displayed are 0 to 2500 by 500. You will see how to specify other values in subsequent examples.

Figure 23.7 At-Risk Table inside the Plot
The group labels for the at-risk table are group numbers, and these numbers appear in the legend. Numbers are used rather than the actual labels because the length of the longest label (13) is greater than the default that is set by the maximum label length option (MAXLEN=12). You can display labels rather than the group numbers by specifying a MAXLEN= value equal to the maximum group label length as follows:

```sas
proc lifetest data=sashelp.BMT plots=survival(cb=hw test atrisk(maxlen=13));
  time T * Status(0);
  strata Group;
run;
```

The results are displayed in Figure 23.8. The legend entries and the order of the rows in the at-risk table correspond to the sort order of the values of the Group variable.

**Figure 23.8** At-Risk Table with Labels

![At-Risk Table with Labels](image-url)
Displaying the Patients-at-Risk Table outside the Plot

You can use the PLOTS=SURVIVAL(OUTSIDE) option to display the at-risk table outside the body of the plot. The option OUTSIDE(0.15) reserves 15% of the vertical graph window for the at-risk table. This example illustrates that the PLOTS= option has options nested within options and options nested within those nested options. The following step produces the plot in Figure 23.9:

```sas
proc lifetest data=sashelp.BMT
   plots=survival(atrisk(maxlen=13 outside(0.15)));
   time T * Status(0);
   strata Group;
run;
```

**Figure 23.9** Controlling Legend Order
Modifying At-Risk Table Times

The following step explicitly controls the time values at which the at-risk values are displayed by using the PLOTS=SURVIVAL(ARISK=0 TO 3000 BY 1000) option:

```
proc lifetest data=sashelp.BMT
   plots=survival(atrisk(maxlen=13 outside)=0 to 3000 by 1000);
   time T * Status(0);
   strata Group;
run;
```

The results are displayed in Figure 23.10.

**Figure 23.10** Specifying At-Risk Values
You can specify at-risk values that do not correspond to the original time axis tick marks. You can use the PLOTS=SURVIVAL(ATRISK(ATRISK(TICK))) option to add tick marks that correspond to the specified at-risk values:

```sas
proc lifetest data=sashelp.BMT plots=survival(atrisk
    (atrisktick maxlen=13 outside)=0 500 750 1000 1250 1500 1750 2000 2500);
    time T * Status(0);
    strata Group;
run;
```

The results are displayed in Figure 23.11.

**Figure 23.11** Controlling At-Risk Tick Marks
You can display tick values only at those times that are given in the ATRISK= list:

```latex
proc lifetest data=sashelp.BMT plots=survival(atrisk
   (atrisktickonly maxlen=13 outside)=0 1250 2500);
   time T * Status(0);
   strata Group;
run;
```

The results are displayed in Figure 23.12.

**Figure 23.12** Controlling At-Risk Tick Marks
Reordering the Groups

You can change the order of the legend entries by first changing each original group value to a new value in the desired order and then running the analysis with a FORMAT statement to provide the original values. In this example, the order is changed to AML–Low Risk (the top function), followed by ALL (the middle function), followed by AML–High Risk. With this ordering, there is a clearer correspondence between the functions, the at-risk table, and the legend. The following steps illustrate this reordering:

```
proc format;
  invalvalue bmtnum 'AML-Low Risk' = 1 'ALL' = 2 'AML-High Risk' = 3;
  value bmtfmt 1 = 'AML-Low Risk' 2 = 'ALL' 3 = 'AML-High Risk';
run;

data BMT(drop=g);
  set sashelp.BMT(rename=(group=g));
  Group = input(g, bmtnum.);
run;

proc lifetest data=BMT plots=survival(cl test atrisk(maxlen=13));
  time T * Status(0);
  strata Group / order=internal;
  format group bmtfmt.;
run;
```

The PROC FORMAT step has two statements. The INVALUE statement creates an informat that maps the values of the original Group variable into integers that have the correct order. The VALUE statement creates a format that maps the integers back to the original values. The informat is used with the INPUT function in the DATA step to create a new integer Group variable. The FORMAT statement assigns the BMTFMT format to the Group variable so that the actual risk groups are displayed in the analysis. You specify the ORDER=INTERNAL option in the STRATA statement to sort the Group values based on internal order (the order specified by the integers, which are the internal unformatted values). This example also illustrates the CL option, which displays pointwise confidence limits for the survival curve (instead of the Hall-Wellner confidence bands). The results are displayed in Figure 23.13.
Figure 23.13 Controlling Legend Order

Product-Limit Survival Estimates
With Number of Subjects at Risk and 95% Confidence Limits

Survival Probability

<table>
<thead>
<tr>
<th>Group</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>AML-Low Risk</td>
<td>54</td>
<td>36</td>
<td>27</td>
<td>18</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>ALL</td>
<td>36</td>
<td>16</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>AML-High Risk</td>
<td>45</td>
<td>13</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Disease-Free Survival Time

+ Censored
Logrank p=0.0010

Group: AML-Low Risk, ALL, AML-High Risk
You can submit the following steps to display ALL first, followed by AML–Low Risk and then AML–High Risk:

```sas
proc format;
invalue bmtnum 'ALL' = 1 'AML-Low Risk' = 2 'AML-High Risk' = 3;
value bmtfmt 1 = 'ALL' 2 = 'AML-Low Risk' 3 = 'AML-High Risk';
run;

data BMT(drop=g);
  set sashelp.BMT(rename=(group=g));
  Group = input(g, bmtnum.);
run;

proc lifetest data=BMT plots=survival(cl test atrisk(maxlen=13));
time T * Status(0);
strata Group / order=internal;
format group bmtfmt.;
run;
```

The results are displayed in Figure 23.14.

**Figure 23.14** Controlling Legend Order
Suppressing the Censored Observations

You can use the PLOTS=SURVIVAL(NOCENSOR) option to suppress the display of censored observations as follows:

```plaintext
proc lifetest data=sashelp.BMT
   plots=survival(nocensor test atrisk(maxlen=13));
   time T * Status(0);
   strata Group;
run;
```

The results are displayed in Figure 23.15.

**Figure 23.15** Censored Values Not Displayed
Failure Plots

All the discussion up to this point has been about survival plots. You can instead plot failure probabilities by using the PLOTS=SURVIVAL(FAILURE) option as follows:

```sas
proc lifetest data=sashelp.BMT
   plots=survival(cb=hw failure test atrisk(maxlen=13));
   time T * Status(0);
   strata Group;
run;
```

The results are displayed in Figure 23.16.

![Figure 23.16 Failure Plot](image-url)
Controlling the Survival Plot by Modifying Graph Templates

The preceding section illustrates the PLOTS= options for controlling the survival plot. If you need to make modifications that are not shown in that section, this section shows how to modify the survival plot by using macros and macro variables to modify graph templates.

The Modularized Templates

SAS provides the two templates that are used to make the survival plot in a modularized form. The modularized version of the two survival plot templates is available in the SAS sample library and is on the Web at http://support.sas.com/documentation/onlinedoc/stat/ex_code/131/templft.html.

The file defines the macro %ProvideSurvivalMacros, which defines a series of macros and macro variables. The %ProvideSurvivalMacros macro contains a %GLOBAL statement, a series of %LET statements, and several macro definitions. It ends with a call to the %CompileSurvivalTemplates macro (which is defined inside the %ProvideSurvivalMacros macro), which compiles the two survival plot templates. By using these macros and macro variables, you can easily specify single changes that modify both templates. All the statements in this file are displayed and explained in more detail in the section “Graph Templates, Macros, and Macro Variables” on page 825.

The %ProvideSurvivalMacros macro provides a way to provide (and in subsequent steps restore) the default macros and macro variables. The macros and macro variables are designed so that you can make most changes by submitting just a few lines of SAS code. Hence, you should not modify any of the statements while they are inside the %ProvideSurvivalMacros macro. Rather, you should use this macro only to provide all the default macros and macro variables. You should modify the individual macros and macro variables outside the context of the %ProvideSurvivalMacros macro. The reasons for this will become clearer as you work through the examples. Before you modify anything, you must submit the %ProvideSurvivalMacros macro definition from the sample library to SAS. You can both store the macros in a temporary file and submit them to SAS by submitting the following statements:

```sas
data _null_;%let url = //support.sas.com/documentation/onlinedoc/stat/ex_code/131;
infile "http:\&url/templft.html" device=url;
file 'macros.tmp';
retain pre 0;
input;
  if index(_infile_, '</pre>') then pre = 0;
  if pre then put _infile_;%inc 'macros.tmp' / nosource;
  if index(_infile_, '<pre>') then pre = 1;
run;
```

3You might wonder why these macros are not simply made available in the SAS autocall library. The autocall library provides macros that you can run. In this context, you do not need to simply run a macro. You need to copy it, extract parts of it, modify those parts, and submit the modified statements. That is not convenient with the autocall library.
4However, there might be something that you always want to change. For example, if you always want the survival plot to be entitled ‘Kaplan-Meier Plot’, then you can modify the title once inside the %ProvideSurvivalMacros macro. This is not illustrated in this chapter. All examples illustrate ad hoc changes that are made outside the context of the %ProvideSurvivalMacros macro.
Submitting these statements only defines the %ProvideSurvivalMacros macro. It does not make any of its component macros and macro variables available. The URL macro variable is used to avoid an overly long INFILE statement.

You can provide the default macros and macro variables by running the following macro:

```
%ProvideSurvivalMacros
```

Running this macro provides the default macros and macro variables (or restores them if you have previously submitted the %ProvideSurvivalMacros macro). The %ProvideSurvivalMacros macro also runs the %CompileSurvivalTemplates macro and hence replaces any compiled survival plot templates that you might have created in the past. You can recompile the templates by submitting the following macro:

```
%CompileSurvivalTemplates
```

This macro runs PROC TEMPLATE and compiles the templates from all the macros and macro variables in the %ProvideSurvivalMacros macro along with any that you modified. Running this macro produces two compiled templates that are stored in a special SAS data file called an item store. For more information about SAS item stores, see the section “SAS Item Stores” on page 851. Assuming that you have not modified your ODS path by using an ODS PATH statement, compiled templates are stored in an item store in the Sasuser library. Files in the Sasuser library persist across SAS sessions until they are deleted. When you are done with a modified template, it is wise to clean up all remnants of it by restoring the default macros and by deleting the modified templates from the Sasuser template item store. You can delete the modified templates (so that SAS can only find the original templates) by running the following step:

```
proc template;
run;
```

This step deletes the compiled templates from the item store sasuser.templat. You can omit the STORE= option if you are using the default ODS path, but it is good practice to explicitly control which templates are deleted. Deleting the compiled templates does not change any of the macros or macro variables. Only the compiled templates (not the macros or macro variables) affect the graph when you run PROC LIFETEST. For more information about compiled templates, item stores, and cleanup, see the section “SAS Item Stores” on page 851.

---

5Semicolons are not needed after a macro call like this one, so they are not used in these examples.
Changing the Plot Title

Here is a simple, complete program (except for retrieving the %ProvideSurvivalMacros macro from the sample library) with setup, macro variable modifications to change the title, and cleanup:

```sas
/*--- Original Macro Variable Definitions ----------------------------------
%let TitleText0 = METHOD " Survival Estimate";
%let TitleText1 = &titletext0 " for " STRATUMID;
%let TitleText2 = &titletext0 "s";
-------------------------------------------------------------------------*/

/* Make the macros and macro variables available. */
%ProvideSurvivalMacros

/* Change the title. */
%let TitleText0 = "Kaplan-Meier Plot";
%let TitleText1 = &titletext0 " for " STRATUMID;
%let TitleText2 = &titletext0;

/* Compile the templates with the new title. */
%CompileSurvivalTemplates

/* Perform the analysis and make the graph. */
proc lifetest data=sashelp.BMT /* Perform the analysis and make the graph. */
  plots=survival(cb=hw test); /* the graph. */
  time T * Status(0);
  strata Group;
run;

/* Optionally restore the default macros and macro variables. */
%ProvideSurvivalMacros

/* Delete the modified templates. */
proc template;
run;
```

The results are displayed in Figure 23.17. You can see that the graph title is now ‘Kaplan-Meier Plot’.

There are multiple title macro variables because two different types of plots are defined in the survival plot templates. The first macro variable, TitleText0, contains the text that is the same for both types of plots. The second macro variable, TitleText1, contains the title for the single-stratum case. The third macro variable, TitleText2, contains the title for the multiple-strata case. Both TitleText1 and TitleText2 use the common text defined in TitleText0. Both TitleText0 and TitleText2 were changed from their original definition; the definition of TitleText1 was copied from the %ProvideSurvivalMacros macro. You must provide all relevant %LET statements when you modify TitleText0. In this case it is TitleText0 and TitleText2, but it is easy to copy all three and then just modify what you need. Alternatively, when you know the number of strata, you can modify only TitleText1 or TitleText2.
Figure 23.17 Kaplan-Meier Plot Title Modification

Kaplan-Meier Plot
With 95% Hall-Wellner Bands

+ Censored
Logrank p=0.0010

Survival Probability vs. Disease-Free Survival Time

Disease Group: ALL, AML-High Risk, AML-Low Risk
Modifying the Axis

The following statements modify the default tick value list for the Y axis from the default increment of 0.2 to have an increment of 0.25 and also change the Y-axis label to ‘Survival’:

```sas
/*--- Original Macro Variable Definitions ----------------------------------
%let yOptions = label="Survival Probability" shortlabel="Survival"
   linearopts=(viewmin=0 viewmax=1
              tickvaluelist=(0 .2 .4 .6 .8 1.0));
-------------------------------------------------------------------------*/

%ProvideSurvivalMacros

%let yOptions = label="Survival"
   linearopts=(viewmin=0 viewmax=1
              tickvaluelist=(0 .25 .5 .75 1));

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT plots=survival(cb=hw test);
  time T * Status(0);
  strata Group;
run;

The results are displayed in Figure 23.18.

**Figure 23.18** Y-Axis Modification
Chapter 23: Customizing the Kaplan-Meier Survival Plot

The following statements modify the Y axis so that tick marks start at 0.2:

```sas
%ProvideSurvivalMacros
%let yOptions = label="Survival"
   linearopts=(viewmin=0.2 viewmax=1
              tickvaluelist=(0 .2 .4 .6 .8 1.0));

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT plots=survival(cb=hw test);
   time T * Status(0);
   strata Group;
run;
```

You only need to change the value of the VIEWMIN= option, in this case from 0 to 0.2. You do not need to modify the tick value list. The VIEWMIN= option (not the tick value list) controls the smallest value shown on the axis. The results are displayed in Figure 23.19.

**Figure 23.19** Y Axis, First Tick Change
The following steps modify the line thickness for the step functions in the survival plot:

```sas
%ProvideSurvivalMacros

%let StepOpts = lineattrs=(thickness=2.5);

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT plots=survival(cb=hw test);
  time T * Status(0);
  strata Group;
run;
```

The results are displayed in Figure 23.20.

By default, the `StepOpts` macro variable is null.
Changing the Group Color

SAS styles control the colors displayed in graphs. The style elements `GraphData1, GraphData2, ..., GraphData12` control the appearance of groups of observations such as the survival step plots in the Kaplan-Meier plot. You can override these colors by using the `GraphOpts` macro variable (which is null by default). By default, the colors for the first three groups in the HMTLBlue style are shades of blue, red, and green. You can change them to a pure green, red, and blue as follows:

```sas
%ProvideSurvivalMacros

%let GraphOpts = DataContrastColors=(green red blue)
    DataColors=(green red blue);

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT
    plots=survival(cb=hw test atrisk(outside maxlen=13));
    time T * Status(0);
    strata Group;
run;
```

The results are displayed in Figure 23.21. The `DATACONTRASTCOLORS=` option specifies the contrast colors, which are used for markers and lines. The `DATACOLORS=` option specifies the colors, which are used for shaded areas such as confidence bands.

**Figure 23.21** Named Color Specifications

![Product-Limit Survival Estimates](image-url)
The original colors (as shown in Figure 23.33) are more subtle than those shown in Figure 23.21. If you want to change the order of the original colors by using this approach, then you need to know what they are so that you can specify them. The graph colors for the HTMLBlue and Statistical styles are extracted from the style in the section “Displaying a Style and Extracting Color Lists” on page 842 and displayed in Figure 23.36. The section “Modifying Color Lists” on page 845 shows you how to change the graph template to specify the original colors in a different order. The section “Swapping Colors among Style Elements” on page 846 shows you how to use a macro to change a style template to specify the original colors in a different order (without having to extract and specify the color names).

---

Changing the Line Pattern

You can change the line patterns as follows:

```sas
%ProvideSurvivalMacros

%let GraphOpts = attrpriority=none
   DataLinePatterns=(ShortDash MediumDash LongDash);

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT
   plots=survival(cb=hw test atrisk(outside maxlen=13));
   time T * Status(0);
   strata Group;
run;
```

The results are displayed in Figure 23.22.
Other values for the DATALINEPATTERNS= option are provided in the section “The Macro Variables” on page 827. You must use the option ATTRPRIORITY=NONE when you want to have varying line patterns in an ATTRPRIORITY=COLOR style like HTMLBlue or Pearl. In an ATTRPRIORITY=COLOR style, groups are not distinguished by line patterns, and the line patterns for second and subsequent groups match the line pattern for the first group.

### Changing the Font

You can change the Y-axis, X-axis, and title fonts as follows:

```sas
%ProvideSurvivalMacros

/*--- Original Macro Variable Definitions ----------------------------------*/
%let TitleText0 = METHOD " Survival Estimate";
%let TitleText1 = &titletext0 " for " STRATUMID;
%let TitleText2 = &titletext0 "s";
%let yOptions = label="Survival Probability"
    shortlabel="Survival"
    linearopts=(viewmin=0 viewmax=1
                tickvaluelist=(0 .2 .4 .6 .8 1.0));
%let xOptions = shortlabel=XNAME
    offsetmin=.05
    linearopts=(viewmax=MAXTIME tickvaluelist=XTICKVALS
                tickvaluefitpolicy=XTICKVALFITPOL);
-------------------------------------------------------------------------*/
%let tatters = textattrs=(size=12pt weight=bold family='arial');
%let TitleText0 = METHOD " Survival Estimate";
%let TitleText1 = &titletext0 " for " STRATUMID / &tatters;
%let TitleText2 = &titletext0 "s" / &tatters;

%let yOptions = label="Survival Probability"
    shortlabel="Survival"
    labelattrs=(size=10pt weight=bold)
    tickvalueattrs=(size=8pt)
    linearopts=(viewmin=0 viewmax=1
                tickvaluelist=(0 .2 .4 .6 .8 1.0));
%let xOptions = shortlabel=XNAME
    offsetmin=.05
    labelattrs=(size=10pt weight=bold)
    tickvalueattrs=(size=8pt)
    linearopts=(viewmax=MAXTIME tickvaluelist=XTICKVALS
                tickvaluefitpolicy=XTICKVALFITPOL);

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT plots=survival(cb=hw test);
    time T * Status(0);
    strata Group;
run;
```
The results are displayed in Figure 23.23.

**Figure 23.23** Changing the Line Patterns

Font options include the following:

- **COLOR**=style-reference | color
- **FAMILY**=style-reference | ‘string’
- **SIZE**=style-reference | dimension
- **STYLE**=style-reference | NORMAL | ITALIC
- **WEIGHT**=style-reference | NORMAL | BOLD

Fonts vary from installation to installation. Sample font strings include: ‘Times New Roman’, ‘Courier New’, ‘Arial’, and ‘Calibri’. For more information about text and label attribute options, see *SAS Graph Template Language: Reference*. For information about changing fonts in ODS styles, see the section “Displaying a Style and Extracting Font Information” on page 848. ODS Graphics can use a single style element in more than one place in a graph; this example shows how to change individual graph components.
Chapter 23: Customizing the Kaplan-Meier Survival Plot

Changing the Legend and Inset Position

This example shows you how to move the legend inside the plot (to the top right) and move the homogeneity test and censored value legend to the bottom right of the plot:

```sas
%ProvideSurvivalMacros

/*/--- Original Macro Variable Definitions ----------------------------------
%let InsetOpts = autoalign=(TOPRIGHT BOTTOMLEFT TOP BOTTOM)
   border=true BackgroundColor=GraphWalls:Color Opaque=true;
%let LegendOpts = title=GROUPNAME location=outside;
-------------------------------------------------------------------------*/

%let InsetOpts = autoalign=(BottomRight)
   border=true BackgroundColor=GraphWalls:Color Opaque=true;
%let LegendOpts = title=GROUPNAME location=inside across=1 autoalign=(TopRight);

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT
   plots=survival(cb=hw test atrisk(outside maxlen=13));
   time T * Status(0);
   strata Group;
run;
```

This example shows you how to replace the AUTOALIGN=(TOPRIGHT BOTTOMLEFT TOP BOTTOM) option in the macro variable InsetOpts with AUTOALIGN=(BOTTOMRIGHT) and add the AUTOALIGN=(TOPRIGHT) option to the LegendOpts macro variable. You can also add the option ACROSS=1 to the LegendOpts macro variable to stack all legend entries vertically (with just one element in each row).

The results are displayed in Figure 23.24.
Figure 23.24 Controlling Legend Placement

Product-Limit Survival Estimates
With Number of Subjects at Risk and 95% Hall-Wellner Bands

Disease Group
- ALL
- AML-High Risk
- AML-Low Risk

Survival Probability

Disease-Free Survival Time

<table>
<thead>
<tr>
<th>Disease Group</th>
<th>0</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
<th>2500</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>38</td>
<td>16</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>AML-High Risk</td>
<td>45</td>
<td>13</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>AML-Low Risk</td>
<td>54</td>
<td>36</td>
<td>27</td>
<td>18</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>
Changing How the Censored Points Are Displayed

By default, PROC LIFETEST displays a plus sign to indicate censoring. This example illustrates how to change the plus sign to a small filled circle in both the step plots and the inset box. The following steps change the template and create Output 23.25:

```plaintext
/*--- Original Macro Variable Definitions ----------------------------------*/
%let Censored = markerattrs=(symbol=plus);
%let CensorStr = "+ Censored";
-------------------------------------------------------------------------*/

%ProvideSurvivalMacros

%let censored = markerattrs=(symbol=circlefilled size=3px);
%let censorstr = "(*ESC*){Unicode '25cf'x} Censored"
   / textattrs=GraphValueText(family=GraphUnicodeText:FontFamily);

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT plots=survival(cb=hw atrisk(outside maxlen=13));
   time T * Status(0);
   strata Group;
run;
```

Figure 23.25  Survival Plot with a Modified Display of Censoring

![Survival Plot with a Modified Display of Censoring](image)
The Unicode Consortium (http://unicode.org/) provides a list of character codes. Also see Figure 22.2.7 in Chapter 22, “ODS Graphics Template Modification,” for information about the Unicode specification for other markers. Although some Unicode characters are supported in some fonts, you should always specify a Unicode font when using special characters.

---

**Adding a Y-Axis Reference Line**

You can add a horizontal reference line to the survival plot by adding the following statement to the template:

```
referenceline y=0.5;
```

You can do this by using the `%StmtsTop` macro. By default, this macro is empty. You can use the `%StmtsTop` macro to add new statements to the beginning of the block of statements that define the appearance of the graph. In contrast, you can use the `%StmtsBottom` macro to provide statements at the end of the statement block. ODS Graphics draws statements in the order in which they appear; therefore, reference lines should be drawn first so they do not obscure other parts of the graph.

The following step creates the plot in Figure 23.26:

```
%ProvideSurvivalMacros

%macro StmtsTop;
   referenceline y=0.5;
%mend;

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT plots=survival(cb=hw test);
   time T * Status(0);
   strata Group;
run;
```
**Figure 23.26** Horizontal Reference Line
Changing the Homogeneity Test Inset

This example modifies the contents of the %pValue macro. The original %pValue macro definition is as follows:

``` SAS
%macro pValue;
  if (PVALUE < .0001)
    entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
  else
    entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
  endif;
%mend;
```

The following example directly specifies the test name (replacing the internal name ‘Logrank’ with ‘Log Rank’) and adds blank spaces around the equal sign:

``` SAS
%ProvideSurvivalMacros

%macro pValue;
  if (PVALUE < .0001)
    entry "Log Rank p " eval (PUT(PVALUE, PVALUE6.4));
  else
    entry "Log Rank p = " eval (PUT(PVALUE, PVALUE6.4));
  endif;
%mend;

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT plots=survival(cb=hw test);
  time T * Status(0);
  strata Group;
run;
```

The results are displayed in Figure 23.27.
Because this template modification replaces a character string that is more appropriately set by PROC LIFETEST, you should clean up afterward as follows:

```sas
%ProvideSurvivalMacros

proc template;
run;
```
Suppressing the Second Title and Adding a Footnote

The following steps add an ENTRYFOOTNOTE statement to the %StmtsBeginGraph macro and suppress the second title:

```
%ProvideSurvivalMacros

%let ntitles = 1;
%macro StmtsBeginGraph;
   entryfootnote halign=left "Acme Company %sysfunc(date(),worddate.)" /
       textattrs=GraphDataText;
%mend;

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT
   plots=survival(cb=hw test);
   time T * Status(0);
   strata Group;
run;
```

The results are displayed in Figure 23.28.

**Figure 23.28** Footnote but No Second Title
By default, the nTitles macro variable is 2, and all titles are displayed. Setting nTitles to 1 suppresses the second title. You can add titles or footnotes to the plot by adding them to the %StmtsBeginGraph macro. This example adds a footnote that consists of a company name followed by the current date, formatted by using the WORDDATE format. The GraphDataText style element is used; it has a smaller font than the default style element, GraphFootnoteText.

---

### Adding a Small Inset Table with Event Information

This example shows you how to modify the template to produce the plot displayed in Output 23.29. This new plot has an inset table in the top right corner that shows the number of observations and the number of events in each stratum. The legend has been moved inside the plot and combined with the old inset table that shows the marker for censored observations. Also, the title is changed to ‘Kaplan-Meier Plot’.

```sas
%ProvideSurvivalMacros

%let TitleText2 = "Kaplan-Meier Plot";
%let LegendOpts = title="+ Censored"
   location=inside autoalign=(Bottom);
%let InsetOpts = ;

%macro StmtsBottom;
   dynamic %do i = 1 %to 3; StrVal&i NObs&i NEvent&i %end;;
   layout gridded / columns=3 border=TRUE autoalign=(TopRight);
      entry ""; entry "Event"; entry "Total";
      %do i = 1 %to 3;
         %let t = / textattrs=GraphData&i;
         entry halign=right Strval&i &t; entry NEvent&i &t; entry NObs&i &t;
      %end;
   endlayout;
%mend;

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT plots=survival(cb=hw atrisk(outside maxlen=13));
   time T * Status(0);
   strata Group;
run;
```

The results are displayed in Figure 23.29.

---

6This legend is wide and might not be displayed if your graph is small. If the legend is not displayed, try increasing the size of the graph by specifying the WIDTH= or HEIGHT= option in the ODS GRAPHICS statement.
The macro variable TitleText2, which controls the title for the multiple-strata plot, is changed. You can change all three title macro variables, as is done in the construction of Figure 23.17, or you can change only TitleText2 when you have multiple overlaid strata, as in this example. The LegendOpts macro variable value was changed from TITLE=GROUPNAME LOCATION=OUTSIDE to display the censored value legend in place of the legend title and to display the legend inside the bottom of the plot. When the InsetOpts macro variable is null, the usual inset that contains the censored value and p-value is not displayed.

The %StmtsBottom macro (null by default) is replaced with a macro that creates the new inset table. This macro adds statements to the bottom of the templates. If you ignore for a moment most of the options, the core of the generated statements is as follows:

```laconic
dynamic StrVal1 NObs1 NEvent1 StrVal2 NObs2 NEvent2 StrVal3 NObs3 NEvent3;
layout gridded / columns=3;
  entry ""; entry "Event"; entry "Total";
  entry Strval1; entry NEvent1; entry NObs1;
  entry Strval2; entry NEvent2; entry NObs2;
  entry Strval3; entry NEvent3; entry NObs3;
endlayout;
```

The macro first constructs a DYNAMIC statement that includes the names of the dynamic variables that contain some of the results. PROC LIFETEST creates these dynamic variables and sets them to values, but you must declare them in your template before using them. For more information about these dynamic variables, see the section “Additional Dynamic Variables” on page 838. The macro then constructs a 4 × 3 grid that contains a table consisting of a title line and a row for each stratum (which consists of the stratum
Chapter 23: Customizing the Kaplan-Meier Survival Plot

Adding an External Table with Event Information

This example adds a table to the plot that displays a summary of event information. The following statements create Figure 23.30:

%ProvideSurvivalMacros

%let GraphOpts = DesignHeight=DefaultDesignWidth;

%SurvivalSummaryTable

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT
    plots=survival(cb=hw atrisk(outside maxlen=13));
    time T * Status(0);
    strata Group;
run;
The GraphOpts macro variable specifies the option DESIGNHEIGHT=DEFAULTDESIGNWIDTH. At the default graph size (the size at which the graph is designed), this option sets the graph height to the default graph width of 640 pixels. The macro %SurvivalSummaryTable adds new statements to the graph templates that display the number of subjects, number of events, number of censored observations, median survival time, and 95% confidence limits for the median survival time. For more information about the %SurvivalSummaryTable macro, see the section “Event Table Macros” on page 835.
Suppressing the Legend

The plot in Figure 23.30 has a legend. However, the plot displays values in the tables by using colors that match the colors of the step functions, so you do not need the legend. The next statements show how to remove the legend:

```latex
%ProvideSurvivalMacros

%let GraphOpts = DesignHeight=500px;
%let LegendOpts = ;

%SurvivalSummaryTable

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT
  plots=survival(cb=hw atrisk(maxlen=13));
  time T * Status(0);
  strata Group;
run;
```

**Figure 23.31** Plot with Legend Removed

The legend is suppressed when the `LegendOpts` macro variable is null. This example also illustrates changing the design height to 500 pixels and moving the at-risk table back inside the body of the plot.
Kaplan-Meier Plot with Event Table and Other Customizations

This example combines a number of features from previous examples. The order of the strata levels in the tables is ALL, AML–Low Risk, and AML–High Risk (see the section “Reordering the Groups” on page 794). The title is set to ‘Kaplan-Meier Plot’ (see the section “Changing the Plot Title” on page 801). The second title line is suppressed (see the section “Suppressing the Second Title and Adding a Footnote” on page 817). The graph height is set to 500 pixels (see the section “Suppressing the Legend” on page 822). The legend and the inset box that contains the legend for censored values are both suppressed (see the sections “Suppressing the Legend” on page 822 and “Adding a Small Inset Table with Event Information” on page 818). The event table is displayed outside the plot (see the section “Adding an External Table with Event Information” on page 820) and the at-risk table is displayed inside the plot (see the section “Displaying the Patients-at-Risk Table inside the Plot” on page 788).

```
proc format;
   invalue bmtnum 'ALL' = 1 'AML-Low Risk' = 2 'AML-High Risk' = 3;
   value bmtfmt 1 = 'ALL' 2 = 'AML-Low Risk' 3 = 'AML-High Risk';
run;

data BMT(drop=g);
   set sashelp.BMT(rename=(group=g));
   Group = input(g, bmtnum.);
run;

%ProvideSurvivalMacros

%let TitleText2 = "Kaplan-Meier Plot";
%let nTitles = 1;
%let GraphOpts = DesignHeight=500px;
%let LegendOpts = ;
%let InsetOpts = ;

%SurvivalSummaryTable

%CompileSurvivalTemplates

proc lifetest data=BMT plots=survival(cb=hw atrisk(maxlen=13));
   time T * Status(0);
   strata Group / order=internal;
   format group bmtfmt .;
run;
```

The results are displayed in Figure 23.32.
Figure 23.32 Kaplan-Meier Plot with Extensive Customizations

# Compiled Template Cleanup

The following step restores all the default macros and macro variables and deletes the modified templates:

```sas
%ProvideSurvivalMacros

proc template;
run;
```

For more information about deleting compiled templates, see the section “SAS Item Stores” on page 851.
Graph Templates, Macros, and Macro Variables

The %ProvideSurvivalMacros macro and the macros and macro variables that it provides have the following properties:

- Many options, including most of the options that are specified in multiple places in the templates, are extracted to macro variables.
- The %CompileSurvivalTemplates macro provides the main body of the two templates. You can call it to compile the templates after making changes.
  - The template \texttt{Stat.Lifetest.Graphics.ProductLimitSurvival} provides the survival template when the at-risk table is inside the body of the plot.
  - The template \texttt{Stat.Lifetest.Graphics.ProductLimitSurvival2} provides the survival template when the at-risk table is outside the body of the plot.\footnote{The macros do not affect any graph that uses graph templates other than the two templates that are modified here. The macros do not affect the STRATA=PANEL plot that uses the template \texttt{Stat.Lifetest.Graphics.ProductLimitSurvivalPanel} or the failure plot that uses the template \texttt{Stat.Lifetest.Graphics.ProductLimitFailure}.}

The two templates share many statements, and a macro %DO loop creates both versions.

- The portion of the templates for the table for the $p$-values is stored in the macro %pValue.
- The portion of the templates for the single-stratum case is stored in the macro %SingleStratum.
- The portion of the templates for the multiple-strata case is stored in the macro %MultipleStrata.
- The macro %AtRiskLatticeStart begins the two-cell lattice that contains the plot above the table when the at-risk table is outside the body of the plot.
- The macro %AtRiskLatticeEnd ends the two-cell lattice that contains the plot and the table when the at-risk table is outside the body of the plot.
- Some empty macros (%StmtsBeginGraph, %StmtsTop, and %StmtsBottom) are provided to enable you to add statements and options to strategic places in the templates.
- The %SurvTabHeader, %SurvivalTable, and %SurvivalSummaryTable macros enable you to easily add more GTL statements to the Kaplan-Meier plot templates to display event information for each stratum.
This organization makes it easy to identify the relevant parts of the templates, modify these parts, and recompile the templates. A small portion of the `%ProvideSurvivalMacros` macro follows:

```
%macro ProvideSurvivalMacros;

%global atriskopts bandopts censored censorstr classopts
graphopts groups insetopts legendopts ntitles stepopts tiplabel
tips titletext0 titletext1 titletext2 xoptions yoptions;

%let TitleText0 = METHOD " Survival Estimate";
%let TitleText1 = &titletext0 " for " STRATUMID;
%let TitleText2 = &titletext0 "s"; /* plural: Survival Estimates */

%let yOptions = label="Survival Probability" shortlabel="Survival"
   linearopts=(viewmin=0 viewmax=1
               tickvaluelist=(0 .2 .4 .6 .8 1.0));

%let xOptions = shortlabel=XNAME offsetmin=.05
   linearopts=(viewmax=MAXTIME tickvaluelist=XTICKVALS
               tickvaluefitpolicy=XTICKVALFITPOL);

%macro CompileSurvivalTemplates; . . . %mend;
%macro pValue; . . . %mend;
%macro SingleStratum; . . . %mend;
%macro MultipleStrata; . . . %mend;

%CompileSurvivalTemplates

%mend;
```
The Macro Variables

The macros and macro variables are designed so that most of the time you need to modify only the macro variables and not the larger macros. However, you have the full flexibility to modify both. You can modify any of the following macro variables:

```sas
%let TitleText0 = METHOD " Survival Estimate";
%let TitleText1 = &titletext0 " for " STRATUMID;
%let TitleText2 = &titletext0 "s"; /* plural: Survival Estimates */
%let nTitles = 2;
%let yOptions = label="Survival Probability" shortlabel="Survival"
  linearopts=(viewmin=0 viewmax=1
              tickvaluelist=(0 .2 .4 .6 .8 1.0));
%let xOptions = shortlabel=XNAME offsetmin=.05
  linearopts=(viewmax=MAXTIME tickvaluelist=XTICKVALS
              tickvaluefitpolicy=XTICKVALFITPOL);
%let Tips = rolename=(_tip1= ATRISK _tip2=EVENT)
  tiplabel=(_tip1="Number at Risk" _tip2="Observed Events")
  tip=(x y _tip1 _tip2);
%let TipLabel = tiplabel=(y="Survival Probability");
%let StepOpts = ;
%let Groups = group=STRATUM index=STRATUMNUM;
%let BandOpts = &groups modelname="Survival";
%let InsetOpts = autoalign=(TOPRIGHT BOTTOMLEFT TOP BOTTOM)
  border=true BackgroundColor=GraphWalls:Color Opaque=true;
%let LegendOpts = title=GROUPNAME location=outside;
%let AtRiskOpts = display=(label) valueattrs=(size=7pt);
%let ClassOpts = class=CLASSATRISK colorgroup=CLASSATRISK;
%let Censored = markerattrs=(symbol=plus);
%let CensorStr = "+ Censored";
%let GraphOpts = ;
```
The %ProvideSurvivalMacros macro declares that these macro variables are global in scope, so you can assign values to them in your programs and have them affect the internal macros. These macro variables specify a variety of GTL options; for more information, see SAS Graph Template Language: Reference. The macro variables are as follows.

TitleText0 provides the common text that is used in the title for the single-stratum and multiple-strata cases. METHOD is a dynamic variable that PROC LIFETEST sets. In these examples, the value of METHOD is ‘Product-Limit’; the product-limit method is also known as the Kaplan-Meier (1958) method.

TitleText1 provides the title text for the single-stratum title (relying on TitleText0).

TitleText2 provides the title text for the multiple-strata title (relying on TitleText0).

nTitles specifies the number of titles. Set the macro variable nTitles to 1 to suppress the second title line or 0 to suppress all title lines. You can add titles to the plot by adding ENTRYTITLE statements to the top of the %StmtsBeginGraph macro even when you suppress the usual titles by setting the nTitles macro variable to 0 or 1. By default, nTitles equals 2.

yOptions provides the Y-axis options. The LABEL= option provides the axis label. The SHORTLABEL= option provides the axis label for small plots when the LABEL= option label is too long. The LINEAROPTS= option specifies linear axis options. This and most other axes are linear axes; alternatives include log-scale axes. The VIEWMIN=0 and VIEWMAX=1 options ensure that the axis goes from 0 to 1 even when the actual results have a more restricted range. The TICKVALUELIST= option provides the tick values. Standard SAS number list abbreviations like 0 TO 1 BY 0.2 are not valid in the GTL.

xOptions provides the X-axis options. The LABEL= option is not provided, so the axis label comes from the column label in the ODS data object. You can add a LABEL= option or other axis options if you want. The SHORTLABEL= option provides the axis label for small plots when the label is too long. The short label comes from a dynamic variable that PROC LIFETEST provides. The OFFSETMIN= option ensures that there is extra space between the axis and the minimum tick mark. The LINEAROPTS= option specifies linear axis options. The VIEWMAX= option ensures that the axis goes to the value in the MAXTIME dynamic variable set by PROC LIFETEST. The TICKVALUELIST= option provides the tick values in a dynamic variable. The TICKVALUEFITPOLICY= option provides, in a dynamic variable, the approach for handling dense tick marks. Approaches include rotation, staggering, and thinning.

Tips provides options for tooltips for the step plots. Tooltips are text boxes that appear in HTML output when you rest your mouse pointer over part of the plot when the IMAGEMAP=ON option is specified in the ODS GRAPHICS statement. Tooltips are provided for the X- and Y-axis columns. Additional columns that are assigned roles (and hence are eligible to use as tooltips) include the at-risk and event columns. These columns are given the tooltip labels ‘Number at Risk’ and ‘Observed Events’. Unless you are specifically interested in tooltips, you probably do not need to modify this macro variable.

TipLabel provides a label for the Y-axis tooltip. Unless you are specifically interested in tooltips, you do not need to modify this macro variable.

StepOpts provides options for the step functions. This macro variable is null by default. You can use this option to control the line thickness (for example, LINEATTRS=(THICKNESS=2.5)) and other aspects of the step functions.
The Macro Variables provides the name of the data object columns that provide group names and the index that provides the order of the group names. You will probably never need to modify this macro variable.

BandOpts provides the group information for band plots. You will probably never need to modify this macro variable.

InsetOpts provides options for the inset table that provides the censored value legend and the homogeneity test p-value. The AUTOALIGN= option specifies the places in the plot where the inset table can be positioned. If your preferred placement is somewhere other than the top right corner, you can modify the automatic placement list. The BORDER= option displays a border around this table. The BACKGROUNDCOLOR= option controls the table background. By default, it matches the background color for the rest of the plot by using the GraphWalls:Color style reference. The OPAQUE=TRUE option specifies an opaque table that hides any graphical elements that are behind the table. You can set the InsetOpts macro variable to null to suppress the usual inset that contains the censored value and p-value.

LegendOpts provides options for the external legend that identifies the strata. The title comes from a dynamic variable GroupName that the procedure sets. By default, the legend is outside the plot. Specify LOCATION=INSIDE and an AUTOALIGN= option such as the one provided in the InsetOpts macro variable if you want the legend to appear inside the plot. You can set the LegendOpts macro variable to null to suppress the legend.

AtRiskOpts provides options for the at-risk table. The option DISPLAY=(LABEL) limits the display to labels. VALUEATTRS=(SIZE=7PT) specifies a font size of seven points.

ClassOpts provides the options that are used in the at-risk table to distinguish groups of observations.

Censored provides the marker (a plus sign) that is displayed in the plot to indicate censored observations.

CensorStr provides the character for the inset table that shows how censored observations appear in the plot.

GraphOpts provides options for the template BEGINGRAPH statement. By default, the GraphOpts macro variable is null. The following options are particularly useful:

- **ATTRPRIORITY=** AUTO | NONE | COLOR specifies the priority for varying the attributes that distinguish groups of observations. AUTO honors the setting that is otherwise in effect. COLOR varies only the color attribute. NONE simultaneously varies colors, markers, and lines. Styles such as HMTLBlue and Pearl are ATTRPRIORITY=COLOR styles, whereas styles such as DEFAULT, Statistical, Listing, and RTF are ATTRPRIORITY=NONE styles.

- **DATACOLORS=(color-list)** specifies the list of colors (which control confidence bands) to replace the graph data colors from the GraphData1–GraphDataN style elements.

- **DATACONTRASTCOLORS=(color-list)** specifies the list of contrast colors (which control markers and lines) to replace the graph data contrast colors from the GraphData1–GraphDataN style elements.

- **DATALINEPATTERNS=(line-pattern-list)** specifies the list of line patterns to replace the graph data line patterns from the GraphData1–GraphDataN style elements. There are 46 line patterns, and you can specify each pattern by using an integer in the range 1 to 46. Some patterns have names associated with them. You can specify either the name or the number for the following number/name pairs: 1 Solid, 2 ShortDash, 4
Chapter 23: Customizing the Kaplan-Meier Survival Plot


- **DESIGNHEIGHT=height** sets the graph height. You can set the graph height to the default graph width of 640 pixels by specifying the option DESIGNHEIGHT=DEFAULTDESIGNWIDTH. Or you can specify a size in pixels, such as DESIGNHEIGHT=500PX. Although the graph is designed at the specified height, you can resize it for the actual display by using the WIDTH= and HEIGHT= options in the ODS GRAPHICS statement. By default, DESIGNHEIGHT=480PX.

- **DESIGNWIDTH=width** sets the graph width. You can set the graph width to the default graph height of 480 pixels by specifying the option DESIGNWIDTH=DEFAULTDESIGNHEIGHT. Or you can specify a size in pixels, such as DESIGNWIDTH=600PX. Although the graph is designed at the specified width, you can resize it for the actual display by using the WIDTH= and HEIGHT= options in the ODS GRAPHICS statement. By default, DESIGNWIDTH=640PX.

---

The Smaller Macros

The `%ProvideSurvivalMacros` macro provides four small macros that are easy for you to modify:

```latex
%macro StmtsBeginGraph; %mend;
%macro StmtsTop; %mend;
%macro StmtsBottom; %mend;
```

```latex
%macro pValue;
  if (PVALUE < .0001)
    entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
  else
    entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
  endif;
%mend;
```

By default, the `%StmtsBeginGraph`, `%StmtsTop`, and `%StmtsBottom` macros are empty. You can use them to add new statements to the BEGINGRAPH block or to the beginning or end of the block of statements that define the appearance of the graph.

The `%pValue` macro is used to control the display of the p-value from the homogeneity test.

---

The Larger Macros

The examples and information up to this point have illustrated how you can make simple changes to the survival plot. It is unlikely that you will ever have to do more than that. If you need to make changes to the overall layout of the graph, then you must modify one of the other macros. The `%CompileSurvivalTemplates` macro, which is the macro that compiles all the pieces that you modified, is as follows:
%macro CompileSurvivalTemplates;
  %local outside;
  proc template;
    %do outside = 0 %to 1;
      define statgraph
       Stat.Lifetest.Graphics.ProductLimitSurvival%scan(2,2-&outside);
        dynamic NStrata xName plotAtRisk
         %if %nrbquote(&censored) ne %then plotCensored;
         plotCL plotHW plotEP labelCL labelHW labelEP maxTime xtickVals
         xtickValFitPol rowWeights method StratumID classAtRisk
         plotTest GroupName Transparency SecondTitle TestName pValue
            _byline_ _bytitle_ _byfootnote_;
            BeginGraph %if %nrbquote(&graphopts) ne %then / &graphopts;;
            if (NSTRATA=1)
               %if &ntitles %then %do;
                 if (EXISTS(STRATUMID)) entrytitle &titletext1;
                 else entrytitle &titletext0;
               %end;
            %end;
            %if &ntitles gt 1 %then %do;
              if (PLOTATRISK=1);
                entrytitle "With Number of Subjects at Risk" /
                    textattrs=GRAPHVALUETEXT;
                if (PLOTATRISK=1);
                  entrytitle SECONDTITLE / textattrs=GRAPHVALUETEXT;
                  %if not outside %then endif; %end;
            %end;
            %StmtsBeginGraph
            %AtRiskLatticeStart
            layout overlay / xaxisopts=(&xoptions) yaxisopts=(&yoptions);
            %StmtsTop
            %SingleStratum
            %StmtsBottom
            endlayout;
            %AtRiskLatticeEnd(class)
            %StmtsEnd;
            %if &ntitles %then %do; %end;
            %if &ntitles gt 1 %then %do; %end;
            %if not outside %then if (PLOTATRISK=1);
              entrytitle "With Number of Subjects at Risk" /
                textattrs=GRAPHVALUETEXT;
              %if not outside %then endif; %end;
            %end;
            %StmtsBeginGraph
            %AtRiskLatticeStart
            layout overlay / xaxisopts=(&xoptions) yaxisopts=(&yoptions);
            %StmtsTop
            %MultipleStrata
            %StmtsBottom
            endlayout;
            %AtRiskLatticeEnd(class)
            %StmtsEnd;
    endif;
end;
The macro `%DO loop compiles the following two templates:

- **Stat.Lifetest.Graphics.ProductLimitSurvival** when the macro variable `Outside` is 0 and `%SCAN(2,2-&OUTSIDE)` is null
- **Stat.Lifetest.Graphics.ProductLimitSurvival2** when the macro variable `Outside` is 1 and `%SCAN(2,2-&OUTSIDE)` is 2

The primary difference between these templates is that when the macro variable `Outside` is 1, a `LAYOUT LATTICE` statement block is used to place the at-risk table outside the graph. When `Outside` is 1, the macros `%AtRiskLatticeStart` and `%AtRiskLatticeEnd` provide the `LAYOUT LATTICE` statement block (two cells, plot above and at-risk table below) and the `LAYOUT OVERLAY` statement block for the at-risk table. The `%AtRiskLatticeStart` and `%AtRiskLatticeEnd` macros are defined as follows:

```
%macro AtRiskLatticeStart;
    %if &outside %then %do;
        layout lattice / rows=2 rowweights=ROWWEIGHTS
columndatarange=union rowgutter=10;
        cell;
    %end;
%mend;
%macro AtRiskLatticeEnd(useclassopts);
    %if &outside %then %do;
        endcell;
        cell;
        layout overlay / walldisplay=none xaxisopts=(display=none);
        axistable x=TATRISK value=ATRISK / &atriskopts
        %if &useclassopts ne %then &classopts;
        endlayout;
    endcell;
endlayout;
%mend;
```

The `%CompileSurvivalTemplates` macro relies on two other macros: `%SingleStratum` for the single-stratum case and `%MultipleStrata` for the multiple-strata case. The `%SingleStratum` macro is as follows:
%macro SingleStratum;
   if (PLOTHW=1 AND PLOTEP=0)
      bandplot LimitUpper=HW_UCL LimitLower=HW_LCL x=TIME /
         modelname="Survival" fillattrs=GRAPHCONFIDENCE
         name="HW" legendlabel=LABELHW;
   endif;
   if (PLOTHW=0 AND PLOTEP=1)
      bandplot LimitUpper=EP_UCL LimitLower=EP_LCL x=TIME /
         modelname="Survival" fillattrs=GRAPHCONFIDENCE
         name="EP" legendlabel=LABELEP;
   endif;
   if (PLOTHW=1 AND PLOTEP=1)
      bandplot LimitUpper=HW_UCL LimitLower=HW_LCL x=TIME /
         modelname="Survival" fillattrs=GRAPHDATA1 datatransparency=.55
         name="HW" legendlabel=LABELHW;
      bandplot LimitUpper=EP_UCL LimitLower=EP_LCL x=TIME /
         modelname="Survival" fillattrs=GRAPHDATA2
         datatransparency=.55 name="EP" legendlabel=LABELEP;
   endif;
   if (PLOTCL=1)
      if (PLOTHW=1 OR PLOTEP=1)
         bandplot LimitUpper=SDF_UCL LimitLower=SDF_LCL x=TIME /
            modelname="Survival" display=(outline)
            outlineattrs=GRAPHPREDICTIONLIMITS name="CL" legendlabel=LABELCL;
      else
         bandplot LimitUpper=SDF_UCL LimitLower=SDF_LCL x=TIME /
            modelname="Survival" fillattrs=GRAPHCONFIDENCE name="CL"
            legendlabel=LABELCL;
      endif;
   endif;
stepplot y=SURVIVAL x=TIME / name="Survival" &tips legendlabel="Survival"
   &stepopts;
   if (PLOTCENSORED=1)
      scatterplot y=CENSORED x=TIME / &censored &tiplabel
         name="Censored" legendlabel="Censored";
   endif;
   if (PLOTCL=1 OR PLOTCL=1 OR PLOTEP=1)
      discretelegend "Censored" "CL" "HW" "EP" / location=outside
         halign=center;
   else
      if (PLOTCENSORED=1)
         discretelegend "Censored" / location=inside
            autoalign=(topright bottomleft);
      endif;
   endif;
%if not &outside %then %do;
   if (PLOTATRISK=1)
      innermargin / align=bottom;
      axistable x=TATRISK value=ATRISK / &atriskopts;
      endinnermargin;
   %end;
%mend;
The \%MultipleStrata macro is as follows:

\%macro MultipleStrata;
    if (PLOTHW=1)
        bandplot LimitUpper=HW_UCL LimitLower=HW_LCL x=TIME / &bandopts
datatransparency=Transparency;
    endif;
    if (PLOTEP=1)
        bandplot LimitUpper=EP_UCL LimitLower=EP_LCL x=TIME / &bandopts
datatransparency=Transparency;
    endif;
    if (PLOTCL=1)
        if (PLOTHW=1 OR PLOTEP=1)
            bandplot LimitUpper=SDF_UCL LimitLower=SDF_LCL x=TIME / &bandopts
display=(outline) outlineattrs=(pattern=ShortDash);
        else
            bandplot LimitUpper=SDF_UCL LimitLower=SDF_LCL x=TIME / &bandopts
datatransparency=Transparency;
        endif;
    endif;
stepplot y=SURVIVAL x=TIME / &groups name="Survival" &tips &stepopts;
if (PLOTTEST=1)
    scatterplot y=CENSORED x=TIME / &groups &tiplabel &censored;
endif;
%if not &outside %then %do;
    innermargin / align=bottom;
    axistable x=TATRISK value=ATRISK / &atriskopts &classopts;
    endinnermargin;
%end;
%if %nrbquote(&legendopts) ne %then %do;
    DiscreteLegend "Survival" / &legendopts;
%end;
%if %nrbquote(&insetopts) ne %then %do;
    if (PLOTTEST=1)
        layout gridded / rows=2 &insetopts;
        entry &censorstr
        if (%pValue)
            endlayout;
        else
            layout gridded / rows=1 &insetopts;
            entry &censorstr;
            endlayout;
        endif;
    else
        if (PLOTTEST=1)

\%end;
Event Table Macros

All the macros and macro variables that have been described up to this point are used in defining the two survival plot graph templates. Some macros (\%StmtsTop and \%StmtsBottom) and macro variables (StepOpts and GraphOpts) are null and do not affect the generated template code, but all are resolved somewhere in the process of producing the templates. In contrast, the macros \%SurvTabHeader, \%SurvivalTable, and \%SurvivalSummaryTable are not used by default. They are available for you to use to add more statements to the templates.

The \%SurvTabHeader macro provides the headings for the event table:

```
%macro SurvTabHeader(multiple);
  %if &multiple %then %do; entry ""; %end;
  entry "";
  entry &r "Median";
  entry "";
  %if &multiple %then %do; entry ""; %end;
  entry &r "Subjects";
  entry &r "Event";
  entry &r "Censored";
  entry &r "Survival";
  entry &r PctMedianConfid;
  entry halign=left "CL";
%mend;
```

This table is not displayed by default. There are two types of headings: one for multiple strata and one for a single stratum.

The \%SurvivalTable macro provides the body of the event table:

```
%macro SurvivalTable;
  %local fmt r i t;
  %let fmt = bestd6.;
  %let r = halign = right;
  columnheaders;
  layout overlay / pad=(top=5);
  if (NSTRATA=1)
    layout gridded / columns=6 border=TRUE;
    dynamic PctMedianConfid NObs NEvent Median LowerMedian UpperMedian;
    %SurvTabHeader(0)
    entry &r NObs;
```
The %SurvivalSummaryTable macro redefines the %AtRiskLatticeStart and %AtRiskLatticeEnd macros so that they provide the body of the event table:

%macro SurvivalSummaryTable;
  %macro AtRiskLatticeStart;
    layout lattice / columnndatarange=union rowgutter=10
    rows=%if &outside %then 2 rowweights=ROWWEIGHTS;
    %else 1;;
    %if &outside %then %do; cell; %end; %end;
  endcolumnheaders;
%mend;

%macro AtRiskLatticeEnd(useclassopts);
  %if &outside %then %do;
    endcell;
    layout overlay / walldisplay=none xaxisopts=(display=none);
    axistable x=TATRISK value=ATRISK / &atriskopts
    %if &useclassopts ne %then &classopts;;
    endlayout;
  endcell;
%mend;

%SurvivalTable
endlayout;
%mend;
If you want to create an event table like the one displayed in Figure 23.30, you only need to call the %SurvivalSummaryTable macro. If you want to modify the table, then you need to modify the %SurvTabHeader and %SurvivalTable macros.

Dynamic Variables

Graph templates consist of instructions, written by SAS developers, in conjunction with SAS procedure code. However, SAS developers cannot fully provide some instructions when the template is written, because some elements of some graphs cannot be known until the procedure runs. For example, the legend title in a graph that has multiple strata corresponds to the label or name of the stratification variable, and the procedure calculates the $p$-value for the homogeneity test. SAS procedures create dynamic variables to provide some run-time information to graphs. Some dynamic variables are set by the procedure and are declared in the template. Other dynamic variables are also set by the procedure, but you must declare them directly or through the template modification macros before you can use them.

Dynamic Variables That Are Automatically Declared

The primary dynamic variables are as follows:

- `_ByFootNote_` is a binary variable that, when true, displays the BY-group BY line as a footnote.
- `_ByLine_` is a character variable that provides the BY-group BY line.
- `_ByTitle_` is a binary variable that, when true, displays the BY-group BY line as a title.
- `ClassAtRisk` is a character variable that names the data object column that contains the classification (stratification) values.
- `GroupName` is a character variable that contains the stratification legend title.
- `LabelCL` is a character variable that contains the label for the confidence limits legend entry (including the percent sign).
- `LabelEP` is a character variable that contains the label for the equal-precision band legend entry (including the percent sign).
- `LabelHW` is a character variable that contains the label for the Hall-Wellner band legend entry (including the percent sign).
- `MaxTime` is a numeric variable that contains the maximum value to display on the X axis.
- `Method` is a character variable that contains the method for the plot title.
- `NStrata` is an integer variable that contains the number of strata.
- `PValue` is a numeric variable that contains the $p$-value for the homogeneity test.
- `PlotAtRisk` is a binary variable that, when true, is used to display the at-risk table.
- `PlotCensored` is a binary variable that, when true, displays the censored values on the step functions.

Axis labels can be set directly in the template or at run time through dynamic variables or through data object column labels.
PlotCL is a binary variable that, when true, displays the pointwise confidence limits.
PlotEP is a binary variable that, when true, displays the equal-precision band.
PlotHW is a binary variable that, when true, displays the Hall-Wellner confidence band.
PlotTest is a binary variable that, when true, displays the $p$-value for the homogeneity test.
RowWeights is a pair of relative heights of the plot and the external at-risk table.
SecondTitle is a character variable that provides the second title line.
StratumID is a character variable that provides the value of the stratification variable for the single stratum case.
TestName is a character variable that provides the name of the homogeneity test (for example, ‘logrank’).
Transparency is a numeric variable that provides the transparency for the confidence bands in the multiple strata case.
XName is a character variable that contains a short label for the X axis, which might be used in place of the ordinary X-axis label when the ordinary label is long or the plot is small.
XtickValFitPol is a character variable that contains the option for handling dense tick values on the X axis.
XtickVals is a list of X-axis tick values.

**Additional Dynamic Variables**

PROC LIFETEST passes to the survival plots a number of dynamic variables that contain summary statistics. Some of these dynamic variables are used when you call the `%SurvivalSummaryTable` macro. Table 23.1 and Table 23.2 list these additional dynamic variables for the Kaplan-Meier curves and the life-table curves, respectively. These dynamic variables are not declared in the templates for the survival curves, but you can declare them and use them to enhance the default plots. The names of the dynamic variables depend on the `STRATA=` suboption of the `PLOTS=SURVIVAL` option: `STRATA=INDIVIDUAL` produces a separate plot for each stratum, and `STRATA=OVERALL` produces one plot that has overlaid curves.

<table>
<thead>
<tr>
<th>STRATA=</th>
<th>Dynamic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERLAY</td>
<td>StrVal$j$</td>
<td>Label for the $j$th stratum</td>
</tr>
<tr>
<td></td>
<td>NObs$j$</td>
<td>Number of observations in the $j$th stratum</td>
</tr>
<tr>
<td></td>
<td>NEvent$j$</td>
<td>Number of events in the $j$th stratum</td>
</tr>
<tr>
<td></td>
<td>Median$j$</td>
<td>Median survival time of the $j$th stratum</td>
</tr>
<tr>
<td></td>
<td>LowerMedian$j$</td>
<td>Lower median survival time of the $j$th stratum</td>
</tr>
<tr>
<td></td>
<td>UpperMedian$j$</td>
<td>Upper median survival time of the $j$th stratum</td>
</tr>
<tr>
<td></td>
<td>PctMedianConfid</td>
<td>Confidence of the median intervals, in percentage</td>
</tr>
<tr>
<td>INDIVIDUAL</td>
<td>NObs</td>
<td>Number of observations</td>
</tr>
<tr>
<td></td>
<td>NEvent</td>
<td>Number of events</td>
</tr>
</tbody>
</table>

9Because the number of dynamic variables is a function of the number of strata, the template definition cannot automatically contain the correct number of dynamic variables.


### Table 23.1  continued

<table>
<thead>
<tr>
<th>STRATA=</th>
<th>Dynamic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Median</td>
<td>Median survival time</td>
</tr>
<tr>
<td></td>
<td>LowerMedian</td>
<td>Lower median survival time</td>
</tr>
<tr>
<td></td>
<td>UpperMedian</td>
<td>Upper median survival time</td>
</tr>
<tr>
<td></td>
<td>PctMedianConfid</td>
<td>Confidence of the median intervals, in percentage</td>
</tr>
</tbody>
</table>

### Table 23.2  Additional Dynamic Variables for `Stat.Graphics.LifetableSurvival`

<table>
<thead>
<tr>
<th>STRATA=</th>
<th>Dynamic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERLAY</td>
<td>StrVal&lt;sub&gt;j&lt;/sub&gt;</td>
<td>Label for the &lt;sup&gt;j&lt;/sup&gt;th stratum</td>
</tr>
<tr>
<td></td>
<td>NObs&lt;sub&gt;j&lt;/sub&gt;</td>
<td>Number of observations in the &lt;sup&gt;j&lt;/sup&gt;th stratum</td>
</tr>
<tr>
<td></td>
<td>NEvent&lt;sub&gt;j&lt;/sub&gt;</td>
<td>Number of events in the &lt;sup&gt;j&lt;/sup&gt;th stratum</td>
</tr>
<tr>
<td>INDIVIDUAL</td>
<td>NObs</td>
<td>Number of observations</td>
</tr>
<tr>
<td></td>
<td>NEvent</td>
<td>Number of events</td>
</tr>
</tbody>
</table>

---

**Style Templates**

Graphs that are produced by ODS Graphics are controlled by the data object (the matrix of information that is graphed), the graph template (the program that controls how a specific graph is constructed), and a style template (a program that controls the overall appearance of graphs, including colors, line and marker styles, sizes, fonts, and so on). Although it is rarely necessary, you can use different styles or modify styles to change the appearance of all graphs, including the survival plot. In the past, you could make certain Kaplan-Meier plot modifications only through style modifications. However, with the addition of the DATACOLORS=, DATACONTRASTCOLORS=, and DATALINEPATTERNS= options in the GTL, you no longer have to modify styles in order to modify how groups of observations are displayed. This section shows you how to change styles, extract group color and other information from styles, and modify styles.
Changing the Style

The graphs that have been displayed up to this point were all produced by using the HTMLBlue style, which is the default style for the HTML destination in the SAS windowing environment. This is an all-color style (because of the ATTRPRIORITY=’Color’ option); it does not rely on line style or marker changes to differentiate groups. You can switch to a style that varies colors, markers, and lines by specifying the STYLE= option in an ODS destination statement. You can use the HTMLBlueCML style as follows to make a graph whose line patterns differ:

```sas
ods html style=htmlbluecml image_dpi=300;
proc lifetest data=sashelp.BMT
   plots=survival(cb=hw test atrisk(outside maxlen=13));
   time T * Status(0);
   strata Group;
run;
ods html close;
```

The results are displayed in Figure 23.33. This example also illustrates specifying the IMAGE_DPI= option to control the resolution (measured in dots per inch, or DPI) of the image. All images in this chapter are created at 300 DPI. The default setting for the HTML destination is 100 DPI. Images that are created at 300 DPI are clearer than images created at 100 DPI, but they require about nine times as much disk space.

Figure 23.33 Line and Color Group Differentiation
Color Priority Styles

You can use the HTMLBlue or Pearl style when you want to distinguish groups only by color. Alternatively, you can easily modify any other style to be an all-color style like HTMLBlue or Pearl by using the ATTRPRIORITY=’Color’ option:10

```
proc template;
  define style styles.ListingColor;
    parent = styles.Listing;
    style Graph from Graph / attrpriority = "Color";
  end;
run;
```

You need to specify the new style name in an ODS destination statement, as in the following:

```
ods html style=ListingColor image_dpi=300;
proc lifetest data=sashelp.BMT
  plots=survival(cb=hw test atrisk(outside maxlen=13));
  time T * Status(0);
  strata Group;
run;
ods html close;
```

The results are displayed in Figure 23.34.

---
10The style option is ATTRPRIORITY=quoted-string, whereas the GTL option is ATTRPRIORITY=keyword.
Displaying a Style and Extracting Color Lists

You can use PROC TEMPLATE with the SOURCE statement to display a style as follows:

```latex
proc template;
   source styles.htmlblue;
run;
```

The results of this step, which are not shown, include the option PARENT=STYLES.STATISTICAL and do not include definitions of the colors (gData1, gData2, ..., gData12) and contrast colors (gcData1, gcData2, ..., gcData12). These are the color definitions that are used in the style elements GraphData1, GraphData2, ..., GraphData12. You can examine the parent Statistical style as follows:

```latex
proc template;
   source styles.statistical;
run;
```

The results of this step are not shown because they are hard to interpret in their raw form, but the desired color definitions are included. You can submit the following statements to display the colors for the Statistical (and hence HTMLBlue) style in a more understandable form:

```latex
proc template;
   source styles.statistical / file='style.tmp';
run;

data colors;
   infile 'style.tmp';
   input;
   if index(_infile_, 'data') then do;
      element = scan(_infile_, 1, ' ');
      Color = scan(_infile_, 3, ' ;');
      Type = ifc(index(element, 'gc'), 'Line', 'Fill') || ' Colors';
      i = input(compress(element, 'gcdat';'), ?? 2.);
      if i then output;
   end;
run;
```

```latex
proc sort; by descending type i; run;
```

```latex
proc print; id type; by descending type; var color; run;
```

The results are displayed in Figure 23.35. All colors are specified in values of the form $\text{CXrrggbb}$, where the last six characters specify RGB (red, green, blue) values on the hexadecimal scale of 00 to FF (0 to 255, base 10).
**Figure 23.35** HMTLBlue Style Colors List

<table>
<thead>
<tr>
<th>Type</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Colors</td>
<td>cx445694</td>
</tr>
<tr>
<td></td>
<td>cxA23A2E</td>
</tr>
<tr>
<td></td>
<td>cx01665E</td>
</tr>
<tr>
<td></td>
<td>cx543005</td>
</tr>
<tr>
<td></td>
<td>cx9D3CDB</td>
</tr>
<tr>
<td></td>
<td>cx7F8E1F</td>
</tr>
<tr>
<td></td>
<td>cx2597FA</td>
</tr>
<tr>
<td></td>
<td>cxB26084</td>
</tr>
<tr>
<td></td>
<td>cxD17800</td>
</tr>
<tr>
<td></td>
<td>cx47A82A</td>
</tr>
<tr>
<td></td>
<td>cxB38EF3</td>
</tr>
<tr>
<td></td>
<td>cxF9DA04</td>
</tr>
<tr>
<td>Fill Colors</td>
<td>cx6F7EB3</td>
</tr>
<tr>
<td></td>
<td>cxD05B5B</td>
</tr>
<tr>
<td></td>
<td>cx66A5A0</td>
</tr>
<tr>
<td></td>
<td>cxA9865B</td>
</tr>
<tr>
<td></td>
<td>cxB689CD</td>
</tr>
<tr>
<td></td>
<td>cxBABC5C</td>
</tr>
<tr>
<td></td>
<td>cx94BDE1</td>
</tr>
<tr>
<td></td>
<td>cxC0D7BA1</td>
</tr>
<tr>
<td></td>
<td>cxCF974B</td>
</tr>
<tr>
<td></td>
<td>cx87C873</td>
</tr>
<tr>
<td></td>
<td>cxBABEF1</td>
</tr>
<tr>
<td></td>
<td>cxDDDA17E</td>
</tr>
</tbody>
</table>

You can use the following steps to display the `GraphData1 – GraphData12` line and fill colors (contrast colors and colors, respectively):

```plaintext
data display;
  array y[12] y1 - y12;
  do i = 1 to 12; y[i] = i; end;
  do x = 1 to 10; output; end;
  do i = 1 to 12; y[i] = i + .5; end;
  do x = 1 to 10; output; end;
run;

data _null_;  
  set colors;
  call symputx(compress(type || put(i, 2.)), color);
run;
```
Chapter 23: Customizing the Kaplan-Meier Survival Plot

```
proc sgplot noautolegend data=display;
  %macro reg;
    title 'Line and Fill Colors, Respectively';
    %do i = 1 %to 12;
      reg y=y%eval(13-&i) x=x / lineattrs=GraphData&i clmattrs=GraphData&i
       nomarkers clm curvelabelpos=max
       curvelabel=" GraphData&i &&LineColors&i &&FillColors&i";
    %end;
  %mend;
  %reg
  xaxis display=none;
  yaxis display=none;
run;
```

The results are displayed in Figure 23.36. The colors in Figure 23.36 are richer than the colors in the bands in the survival plots because of the DATATRANSPARENCY= options in the BANDPLOT statements.

**Figure 23.36** HTMLBlue Style Colors Display

<table>
<thead>
<tr>
<th>Line and Fill Colors, Respectively</th>
<th>GraphData1 cx445694 cx6F7EB3</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphData2 cxA23A2E cxD05B5B</td>
<td></td>
</tr>
<tr>
<td>GraphData3 cx01665E cx66A5A0</td>
<td></td>
</tr>
<tr>
<td>GraphData4 cx543005 cx9865B</td>
<td></td>
</tr>
<tr>
<td>GraphData5 cx9D3CDB cx689CD</td>
<td></td>
</tr>
<tr>
<td>GraphData6 cx7F8E1F cxBABC5C</td>
<td></td>
</tr>
<tr>
<td>GraphData7 cx2597FA cx94BDE1</td>
<td></td>
</tr>
<tr>
<td>GraphData8 cxB26084 cxCD7BA1</td>
<td></td>
</tr>
<tr>
<td>GraphData9 cxD17800 cxCF974B</td>
<td></td>
</tr>
<tr>
<td>GraphData10 cx47A82A cx87C873</td>
<td></td>
</tr>
<tr>
<td>GraphData11 cxB38EF3 cxB7AEF1</td>
<td></td>
</tr>
<tr>
<td>GraphData12 cxF9DA04 cxDDD17E</td>
<td></td>
</tr>
</tbody>
</table>
You can use the information in Figure 23.36 to specify the desired colors in the graph template. You can copy the third, second, and first colors from each list and switch the colors as follows:

```sas
%ProvideSurvivalMacros

%let GraphOpts = DataContrastColors=(cx01665E cxA23A2E cx445694)
   DataColors=(cx66A5A0 cxD05B5B cx6F7EB3);

%CompileSurvivalTemplates

proc lifetest data=sashelp.BMT
   plots=survival(cb=hw test atrisk(outside maxlen=13));
      time T * Status(0);
      strata Group;
run;
```

The results are displayed in Figure 23.37. The familiar colors are used, but they are now in a different order. The next section shows you how to modify a style template to change the color order without having to extract the original color names.

You can use the information in Figure 23.36 to modify the style template, but the next example shows an easier way.
Swapping Colors among Style Elements

You can modify the colors in a style as follows:

```sas
%macro reorder(from, to, list);
    proc template;
        define style styles.&to;
            parent=styles.&from;
            %do i = 1 %to 12;
                %let s = %scan(&list, &i);
                %if &s ne %then %do;
                    style GraphData&i from GraphData&i /
                        contrastcolor = GraphColors("gcdata&s")
                        color = GraphColors("gdata&s");
                %end;
            %end;
        %end;
    end;
%mend;

%reorder(htmlblue, MyStyle, 3 2 1) / /* Parent style. */
       /* New style to create. Specify it in an ODS */
       /* destination statement. */
       /* Replace the first few GraphData colors */
       /* (1 2 3) with the colors from the specified */
       /* GraphData style elements (3 2 1). */
       /* You can specify up to 12 integers in the */
       /* range 1 - 12. */
```

The `%Reorder` macro creates a new style called MyStyle that inherits most of its attributes from the HTMLBlue style. However, in the new style, the colors for groups 1, 2, and 3 have been replaced by the colors for groups 3, 2, and 1. In other words, the colors for `GraphData1` and `GraphData3` have been switched.

The following step creates the plot:

```sas
ods html style=mystyle;
proc lifetest data=sashelp.BMT
    plots=survival(cb=hw test atrisk(outside maxlen=13));
    time T * Status(0);
    strata Group;
run;
ods html close;
```

The survival plot is not shown, but it matches the plot in Figure 23.37.

The rest of this section is optional. It explains how you can directly modify colors in a style template when the `%Reorder` macro or the technique illustrated in the section “Changing the Group Color” on page 806 is not sufficient.
The source code for the MyStyle style (as generated by the %Reorder macro) is as follows:

```sas
proc template;
   define style Styles.MyStyle;
      parent = styles.htmlblue;
      style GraphData1 from GraphData1 /
         color = GraphColors('gdata3')
         contrastcolor = GraphColors('gcdata3');
      style GraphData2 from GraphData2 /
         color = GraphColors('gdata2')
         contrastcolor = GraphColors('gcdata2');
      style GraphData3 from GraphData3 /
         color = GraphColors('gdata1')
         contrastcolor = GraphColors('gcdata1');
   end;
end;
run;
```

You can create a modified style that has direct color specifications by using the colors in Figure 23.35 as follows:

```sas
proc template;
   define style Styles.MyStyle;
      parent = styles.htmlblue;
      style GraphData1 from GraphData1 /
         color = cx66A5A0
         contrastcolor = cx01665E;
      style GraphData2 from GraphData2 /
         color = cxD05B5B
         contrastcolor = cxA23A2E;
      style GraphData3 from GraphData3 /
         color = cx6F7EB3
         contrastcolor = cx445694;
   end;
end;
run;
```

You can define additional GraphDataN style elements as well. For more information about how to define style elements, see the section “Displaying Other Style Elements” on page 850.

You can delete the new style template as follows:

```sas
proc template;
   delete Styles.MyStyle / store=sasuser.templat;
run;
```
Displaying a Style and Extracting Font Information

You can use PROC TEMPLATE with the SOURCE statement to display a style and its parent styles in the SAS log:

```sas
proc template;
  source styles.htmlblue / expand;
run;
```

The results of this step are long and are not shown. You can write a copy of the style templates to a file as follows:

```sas
proc template;
  source styles.htmlblue / expand file='style.tmp';
run;
```

The EXPAND option writes the specified style (HTMLBlue), followed by its parent (Statistical), and followed by the parent’s parent (DEFAULT) to the file. The following step extracts and displays the first place in the file that the graph fonts are defined (which make the final decision in the style template):

```sas
data _null_;
  infile 'style.tmp' pad;
  input line $char80.;
  file print;
  if index(lowcase(line), ' graphfonts ') then y + 1;
  if y then put line $char80.;
  if y and index(line, ';') then stop;
run;
```

The results are displayed in Figure 23.38.

![Figure 23.38 Graph Font Definition](image)

If the `GraphFonts` style element is defined in the HTMLBlue style, then it will appear first in the file, followed by the definitions from the Statistical style and then the DEFAULT style. In this case, the `GraphFonts` style element is defined in the DEFAULT style (last in the file), which is overridden by a definition in the Statistical style (closer to the top of the file); that is the definition that is inherited by the HTMLBlue style and displayed in Figure 23.38.
The following step creates a new style, BigFont, that changes the `GraphLabelFont` style element from a regular 10-point font to a bold 12-point font and changes the `GraphValueFont` style element from a regular 9-point font to a bold 8-point font:

```sas
proc template;
  define style Styles.BigFont;
  parent = Styles.HTMLBlue;
  style graphfonts from graphfonts /
    'GraphLabelFont' = ("<sans-serif>, <MTsans-serif>",12pt,bold)
    'GraphValueFont' = ("<sans-serif>, <MTsans-serif>",8pt,bold);
  end;
run;
```

The following step creates the plot that is displayed in Output 23.39:

```sas
ods html style=BigFont;
proc lifetest data=sashelp.BMT plots=survival(maxlen=13 atrisk);
  time T * Status(0);
  strata Group;
run;
ods html close;
```

**Figure 23.39** Font Modifications
You can delete the new style template as follows:

```
proc template;
   delete Styles.BigFont / store=sasuser.templat;
run;
```

For information about making ad hoc font changes in the graph templates rather than making more global font changes in style templates, see the section “Changing the Font” on page 808.

---

## Displaying Other Style Elements

You can use the approach from the previous example to display other style elements:

```
proc template;
   source styles.htmlblue / expand file='style.tmp';
run;

data _null_;
   infile 'style.tmp' pad;
   input line $char80.;
   file print;
   if index(lowcase(line), ' graphdata1 ') then y + 1;
   if y then put line $char80.;
   if y and index(line, ';') then stop;
run;
```

This example displays the `GraphData1` style element. The results are displayed in Figure 23.40.

![Figure 23.40 GraphData1 Style Element](image)

The following steps display all the `GraphFonts` style elements from all the styles:

```
proc template;
   source styles / file='style.tmp';
run;

data _null_;
   infile 'style.tmp' pad;
   length style $ 80;
   retain style;
   input line $char80.;
   file print;
   if index(lowcase(line), 'define style') then style = line;
```
The results of this step are not displayed. You can use this approach to help you better understand the options that are available for modifying styles. The SOURCE statement specifies a single-level value of STYLES rather than a specific style name such as STYLES.HTMLBLUE, so all templates that begin with STYLES as the first level (all style templates) are written to the file. The DATA step displays all definitions of GraphFonts and the names of all styles that define the GraphFonts style element.

You can insert the name of another style element (in lowercase with a leading and trailing blank) in the preceding programs in place of ‘graphdata1’ or ‘graphfonts’. After you display a style element, you can modify the definition and create a new style that uses the modified definition, as in the example in the section “Displaying a Style and Extracting Font Information” on page 848. Some of the style elements that you might want to display and modify are listed in Table 23.3.

---

**SAS Item Stores**

In other sections of this chapter, you submit PROC TEMPLATE statements (either directly or through macros) to compile and save graph and style templates. Compiled templates are stored in special SAS files called item stores. Assuming that you have not modified your ODS path with an ODS PATH statement, the templates that you compile are stored in an item store in the Sasuser library. By default, all templates that SAS provides are stored in an item store in the Sashelp library. By default, the Sashelp item store has read access only; you cannot write to it. By default, the Sasuser item store has update access; you can both read and write to it. **CAUTION:** Never set the Sashelp item store to update or write access. If you do, and if you have administrator privileges on your computer, you could corrupt the Sashelp item store.

Assuming that the default ODS path is used, ODS first looks for a template in the Sasuser item store. If it does not find the template there, ODS next looks for the template in the Sashelp item store. Files in the Sasuser library persist across SAS sessions until you delete them. You can run the following step to delete the entire Sasuser item store (including all compiled graph and style templates that you added or modified) so that ODS uses only the templates the SAS System supplies:

```sas
  ods path sashelp.tmplmst(read);
  proc datasets library=sasuser nolist;
    delete templat(memtype=itemstor);
  run;
  ods path reset;
```

For more information about the ODS path and SAS item stores, see Chapter 21, “Statistical Graphics Using ODS.”
Table 23.3 Style Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Class</th>
<th>Description</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>AfterCaption</td>
<td>GraphAnnoText</td>
<td>GraphHeaderBackground</td>
<td>List3</td>
</tr>
<tr>
<td>Batch</td>
<td>GraphAxisLines</td>
<td>GraphHistogram</td>
<td>ListItem</td>
</tr>
<tr>
<td>Body</td>
<td>GraphBackground</td>
<td>GraphInitial</td>
<td>Note</td>
</tr>
<tr>
<td>BodyDate</td>
<td>GraphBand</td>
<td>GraphLabel2Text</td>
<td>NoteBanner</td>
</tr>
<tr>
<td>ByContentFolder</td>
<td>GraphBar</td>
<td>GraphLabelText</td>
<td>NoteContentFixed</td>
</tr>
<tr>
<td>Byline</td>
<td>GraphBlock</td>
<td>GraphLegendBackground</td>
<td>Output</td>
</tr>
<tr>
<td>BylineContainer</td>
<td>GraphBlockHeader</td>
<td>GraphMissing</td>
<td>PageNo</td>
</tr>
<tr>
<td>Caption</td>
<td>GraphBorderLines</td>
<td>GraphOther</td>
<td>Pages</td>
</tr>
<tr>
<td>Color_list</td>
<td>GraphBox</td>
<td>GraphOutlier</td>
<td>PagesProcLabel</td>
</tr>
<tr>
<td>Colors</td>
<td>GraphBoxMean</td>
<td>GraphOutlines</td>
<td>PagesTitle</td>
</tr>
<tr>
<td>Container</td>
<td>GraphBoxMedian</td>
<td>GraphOverflow</td>
<td>Paragraph</td>
</tr>
<tr>
<td>ContentFolder</td>
<td>GraphBoxWhisker</td>
<td>GraphPhaseBox</td>
<td>Parskip</td>
</tr>
<tr>
<td>ContentProcLabel</td>
<td>GraphClipping</td>
<td>GraphPrediction</td>
<td>PrePage</td>
</tr>
<tr>
<td>ContentTitle</td>
<td>GraphColors</td>
<td>GraphPredictionLimits</td>
<td>ProcTitle</td>
</tr>
<tr>
<td>Contents</td>
<td>GraphConfidence</td>
<td>GraphReference</td>
<td>ProcTitleFixed</td>
</tr>
<tr>
<td>Continued</td>
<td>GraphConfidence2</td>
<td>GraphRunTest</td>
<td>RowFooter</td>
</tr>
<tr>
<td>Data</td>
<td>GraphConnectLine</td>
<td>GraphSelection</td>
<td>RowFooterEmphasis</td>
</tr>
<tr>
<td>DataEmphasis</td>
<td>GraphContour</td>
<td>GraphStars</td>
<td>RowFooterEmphasisFixed</td>
</tr>
<tr>
<td>DataEmphasisFixed</td>
<td>GraphControlLimits</td>
<td>GraphTitle1Text</td>
<td>RowFooterFixed</td>
</tr>
<tr>
<td>DataFixed</td>
<td>GraphData1</td>
<td>GraphTitleText</td>
<td>RowFooterStrong</td>
</tr>
<tr>
<td>DataStrong</td>
<td>GraphData2</td>
<td>GraphUnderflow</td>
<td>RowFooterStrongFixed</td>
</tr>
<tr>
<td>DataStrongFixed</td>
<td>GraphData3</td>
<td>GraphUnicodeText</td>
<td>RowHeader</td>
</tr>
<tr>
<td>Date</td>
<td>GraphData4</td>
<td>GraphValueText</td>
<td>RowHeaderEmphasis</td>
</tr>
<tr>
<td>Document</td>
<td>GraphData5</td>
<td>GraphWalls</td>
<td>RowHeaderEmphasisFixed</td>
</tr>
<tr>
<td>DropShadowStyle</td>
<td>GraphData6</td>
<td>GraphZoneA</td>
<td>RowHeaderFixed</td>
</tr>
<tr>
<td>ErrorBanner</td>
<td>GraphData7</td>
<td>GraphZoneB</td>
<td>RowHeaderStrong</td>
</tr>
<tr>
<td>ErrorContentFixed</td>
<td>GraphData8</td>
<td>GraphZoneC</td>
<td>RowHeaderStrongFixed</td>
</tr>
<tr>
<td>ExtendedPage</td>
<td>GraphData9</td>
<td>Header</td>
<td>SysTitleAndFooterContainer</td>
</tr>
<tr>
<td>FatalBanner</td>
<td>GraphData10</td>
<td>HeaderEmphasis</td>
<td>SystemFooter</td>
</tr>
<tr>
<td>FatalContentFixed</td>
<td>GraphData11</td>
<td>HeaderEmphasisFixed</td>
<td>SystemTitle</td>
</tr>
<tr>
<td>Fonts</td>
<td>GraphData12</td>
<td>HeaderFixed</td>
<td>Table</td>
</tr>
<tr>
<td>Footer</td>
<td>GraphDataDefault</td>
<td>HeaderStrong</td>
<td>ThreeColorAltRamp</td>
</tr>
<tr>
<td>FooterEmphasis</td>
<td>GraphDataText</td>
<td>HeaderStrongFixed</td>
<td>ThreeColorRamp</td>
</tr>
<tr>
<td>FooterEmphasisFixed</td>
<td>GraphEllipse</td>
<td>HeadersAndFooters</td>
<td>TitleAndNoteContainer</td>
</tr>
<tr>
<td>FooterFixed</td>
<td>GraphError</td>
<td>Index</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>FooterStrong</td>
<td>GraphFinal</td>
<td>Item</td>
<td>TwoColorAltRamp</td>
</tr>
<tr>
<td>FooterStrongFixed</td>
<td>GraphFit</td>
<td>IndexProcName</td>
<td>TwoColorRamp</td>
</tr>
<tr>
<td>Frame</td>
<td>GraphFit2</td>
<td>IndexTitle</td>
<td>UserText</td>
</tr>
<tr>
<td>Graph</td>
<td>GraphFloor</td>
<td>LayoutContainer</td>
<td>WarnBanner</td>
</tr>
<tr>
<td>GraphAltBlock</td>
<td>GraphFonts</td>
<td>LineContent</td>
<td>WarnContentFixed</td>
</tr>
<tr>
<td>GraphAnnoLine</td>
<td>GraphFootnoteText</td>
<td>List</td>
<td></td>
</tr>
<tr>
<td>GraphAnnoShape</td>
<td>GraphGridLines</td>
<td>List2</td>
<td></td>
</tr>
</tbody>
</table>
References


Index

p value
   LIFETEST procedure, 815

ACROSS= GTL option
   LIFETEST procedure, 810

at-risk table (inside)
   LIFETEST procedure, 788

at-risk table (outside)
   LIFETEST procedure, 790

at-risk value specification
   LIFETEST procedure, 791–793

ATRISK survival-plot option
   LIFETEST procedure, 788

ATRISK(MAXLEN=13) survival-plot option
   LIFETEST procedure, 789

ATRISK(OUTSIDE) survival-plot option
   LIFETEST procedure, 790

ATRISK= survival-plot option
   LIFETEST procedure, 791

%AtRiskLatticeEnd macro
   LIFETEST procedure, 832

%AtRiskLatticeStart macro
   LIFETEST procedure, 832

ATRISK_TICK survival-plot option
   LIFETEST procedure, 792

ATRISK_TICKONLY survival-plot option
   LIFETEST procedure, 793

ATTRPRIORITY='Color' style
   LIFETEST procedure, 841

ATTRPRIORITY=NONE GTL option
   LIFETEST procedure, 807

AUTOALIGN= GTL option
   LIFETEST procedure, 807

axis label modification
   LIFETEST procedure, 803

CB=ALL survival-plot option
   LIFETEST procedure, 786

CB=EP survival-plot option
   LIFETEST procedure, 786

CB=HW survival-plot option
   LIFETEST procedure, 785

Censored macro variable
   LIFETEST procedure, 812

censored value legend
   LIFETEST procedure, 797

CensorStr macro variable
   LIFETEST procedure, 812

%CompileSurvivalTemplates macro
   LIFETEST procedure, 801, 831

DATACOLORS= GTL option
   LIFETEST procedure, 806, 845

DATACONTRASTCOLORS= GTL option
   LIFETEST procedure, 806, 845

DATALINEPATTERNs= GTL option
   LIFETEST procedure, 807

date (displaying in a footnote)
   LIFETEST procedure, 817

DESIGNHEIGHT= GTL option
   LIFETEST procedure, 823

DESIGNHEIGHT=500PX GTL option
   LIFETEST procedure, 822

DESIGNHEIGHT=DEFAULTDESIGNWIDTH GTL option
   LIFETEST procedure, 820

ENTRYFOOTNOTE GTL statement
   LIFETEST procedure, 817

equal-precision bands
   LIFETEST procedure, 786

event summary table
   LIFETEST procedure, 820, 822

events (number of inset table)
   LIFETEST procedure, 818

EXPAND option
   TEMPLATE procedure, 848

FAILURE survival-plot option
   LIFETEST procedure, 798

FAMILY=GRAPHUNICODETEXT GTL option
   LIFETEST procedure, 812

font changes
   LIFETEST procedure, 808

fonts
   LIFETEST procedure, 848–850

footnote
   LIFETEST procedure, 817

format
   LIFETEST procedure, 794, 796, 823

GraphDataN style element modification
   LIFETEST procedure, 847

GraphOpts macro variable
   LIFETEST procedure, 806, 807, 820, 822, 823, 845

group reordering
   LIFETEST procedure, 794, 796
Hall-Wellner confidence bands
LIFETEST procedure, 785, 786
HTMLBlueCML style
LIFETEST procedure, 840

informat
LIFETEST procedure, 794, 796, 823
inset
LIFETEST procedure, 810
inset table
LIFETEST procedure, 818
InsetOpts macro variable
LIFETEST procedure, 810, 818, 823

Kaplan-Meier plot
LIFETEST procedure, 780, 781, 801

label modification
LIFETEST procedure, 803
legend
LIFETEST procedure, 810
legend suppression
LIFETEST procedure, 822
LegendOpts macro variable
LIFETEST procedure, 810, 818, 822

LIFETEST procedure
p value, 815
_ByFootNote_ dynamic variable, 837
_ByLine_ dynamic variable, 837
_ByTitle_ dynamic variable, 837
ACROSS= GTL option, 810
at-risk table (inside), 788
at-risk table (outside), 790
at-risk value specification, 791–793
ATRISK survival-plot option, 788
ATRISK(MAXLEN=13) survival-plot option, 789
ATRISK(OUTSIDE) survival-plot option, 790
ATRISK= survival-plot option, 791
%AtRiskLatticeEnd macro, 832
%AtRiskLatticeStart macro, 832
AtRiskOpts macro variable defined, 829
ATRISKTICK survival-plot option, 792
ATRISKTICKONLY survival-plot option, 793
ATTRPRIORITY= GTL option, 829
ATTRPRIORITY='Color' style, 841
ATTRPRIORITY=NONE GTL option, 807
AUTOALIGN= GTL option, 810, 818
axis label modification, 803
BandOpts macro variable defined, 829
CB=ALL survival-plot option, 786
CB=EP survival-plot option, 786
CB=HW survival-plot option, 785
Censored macro variable, 812
Censored macro variable defined, 829
censored value legend, 797
CensorStr macro variable, 812
CensorStr macro variable defined, 829
ClassAtRisk dynamic variable, 837
ClassOpts macro variable defined, 829
%CompileSurvivalTemplates macro, 801, 831
DATACOLORS= GTL option, 806, 829, 845
DATACONTRASTCOLORS= GTL option, 806, 829, 845
DATALINEPATTERNS= GTL option, 807, 829
date (displaying in a footnote), 817
DESIGNHEIGHT= GTL option, 823, 830
DESIGNHEIGHT=500PX GTL option, 822
DESIGNHEIGHT=DEFAULTDESIGNWIDTH GTL option, 820
DESIGNWIDTH= GTL option, 830
ENTRYFOOTNOTE GTL statement, 817
equal-precision bands, 786
event summary table, 820, 822
events (number of inset table), 818
FAILURE survival-plot option, 798
FAMILY=GRAPHUNICODETEXT GTL option, 812
font changes, 808
fonts, 848–850
footnote, 817
format, 794, 796, 823
GraphDataN style element modification, 847
GraphOpts macro variable, 806, 807, 820, 822, 823, 845
GraphOpts macro variable defined, 829
group reordering, 794, 796
GroupName dynamic variable, 837
Groups macro variable defined, 829
Hall-Wellner confidence bands, 785, 786
HTMLBlueCML style, 840
informat, 794, 796, 823
inset, 810
inset table, 818
InsetOpts macro variable defined, 829
InsetOpts macro variable, 810, 818, 823
InsetOpts macro variable defined, 829
Kaplan-Meier plot, 780, 781, 801
label modification, 803
label modification, 817
LegendOpts macro variable, 806, 807, 820, 822, 823, 845
LegendOpts macro variable defined, 829
LegendOpts macro variable
LIFETEST procedure, 810, 818, 822
Log rank p value, 815
Log rank p value, 815
LowerMedian dynamic variable, 838
macro variables, 827–830
macros, 799
MARKERATTRS= GTL option, 812
MAXLEN=13 survival-plot option, 789
MaxTime dynamic variable, 837
Median dynamic variable, 838
Method dynamic variable, 837
%MultipleStrata macro, 834
NAME= GTL option, 822, 823
NEvent dynamic variable, 838
NOb dynamic variable, 838
NOCENSOR survival-plot option, 797
NStrata dynamic variable, 837
nTitles macro variable, 817, 823
nTitles macro variable defined, 828
ORDER= option, 794, 796
OUTSIDE survival-plot option, 790
patients-at-risk table (inside), 788
patients-at-risk table (outside), 790
PctMedianConfid dynamic variable, 838
PlotAtRisk dynamic variable, 837
PlotCensored dynamic variable, 837
PlotCL dynamic variable, 838
PlotEP dynamic variable, 838
PlotHW dynamic variable, 838
PLOTS=SURVIVAL, 782
PlotTest dynamic variable, 838
product-limit survival plot, 781
%ProvideSurvivalMacros macro, 801
PValue dynamic variable, 837
%pValue macro, 830
reference line, 813
REFERENCELINE GTL statement, 813
%Reorder macro, 846
reordering groups, 794, 796
RowWeights dynamic variable, 838
sample library, 799
SecondTitle dynamic variable, 838
%SingleStratum macro, 833
StepOpt macro variable, 805, 823
StepOpt macro variable defined, 828
%StmtsBeginGraph macro, 817, 830
%StmtsBottom macro, 830
%StmtsTop macro, 830
STRATA=INDIVIDUAL survival-plot option, 783
STRATA=PANEL survival-plot option, 783
StratumID dynamic variable, 838
StrVal dynamic variable, 838
style change, 840, 846
style cleanup, 847
style color list modification, 845, 846
style colors, 842, 843
summary of events table, 820, 822
survival plot, 781
%SurvivalSummaryTable macro, 836
%SurvivalTable macro, 835
%SurvTabHeader macro, 835
template cleanup, 800, 816, 824, 851
TEST survival-plot option, 785
testName dynamic variable, 838
TEXTATTRS= GTL option, 808, 812
tick value modification, 803
TICKVALUELIST= GTL option, 803
tipLabel macro variable defined, 828
tips macro variable defined, 828
title change, 801
TitleText0 macro variable, 808
TitleText0 macro variable defined, 828
TitleText1 macro variable, 808
TitleText1 macro variable defined, 828
TitleText2 macro variable, 808, 818, 823
TitleText2 macro variable defined, 828
Transparency dynamic variable, 838
Unicode, 812
UpperMedian dynamic variable, 838
VIEWMIN= GTL option, 804
XName dynamic variable, 838
xOptions macro variable, 808
xOptions macro variable defined, 828
XtickValFitPol dynamic variable, 838
XtickVals dynamic variable, 838
yOptions macro variable, 803, 804, 808
yOptions macro variable defined, 828
line patterns
LIFETEST procedure, 807
LINEATTRS=(THICKNESS=2.5) GTL option
LIFETEST procedure, 805
LOCATION=INSIDE GTL option
LIFETEST procedure, 810, 818
Log rank \( p \) value
LIFETEST procedure, 815

macro variables
LIFETEST procedure, 827–830
macros
LIFETEST procedure, 799
MARKERATTRS= GTL option
LIFETEST procedure, 812
MAXLEN=13 survival-plot option
LIFETEST procedure, 789
%MultipleStrata macro
LIFETEST procedure, 834
NAME= GTL option
LIFETEST procedure, 822, 823
NOCENSOR survival-plot option
LIFETEST procedure, 797
nTitles macro variable
LIFETEST procedure, 817, 823

ORDER= option
LIFETEST procedure, 794, 796

OUTSIDE survival-plot option
LIFETEST procedure, 790

patients-at-risk table (inside)
LIFETEST procedure, 788

patients-at-risk table (outside)
LIFETEST procedure, 790

PLOTS=SURVIVAL
LIFETEST procedure, 782

product-limit survival plot
LIFETEST procedure, 781

%ProvideSurvivalMacros macro
LIFETEST procedure, 801

%pValue macro
LIFETEST procedure, 830

reference line
LIFETEST procedure, 813

REFERENCELINE GTL statement
LIFETEST procedure, 813

%Reorder macro
LIFETEST procedure, 846

reordering groups
LIFETEST procedure, 794, 796

sample library
LIFETEST procedure, 799

%SingleStratum macro
LIFETEST procedure, 833

StepOpts macro variable
LIFETEST procedure, 805, 823

%StmtsBeginGraph macro
LIFETEST procedure, 817, 830

%StmtsBottom macro
LIFETEST procedure, 830

%StmtsTop macro
LIFETEST procedure, 830

STRATA=INDIVIDUAL survival-plot option
LIFETEST procedure, 783

STRATA=PANEL survival-plot option
LIFETEST procedure, 783

style change
LIFETEST procedure, 840, 846

style cleanup
LIFETEST procedure, 847

style color list modification
LIFETEST procedure, 845, 846

style colors
LIFETEST procedure, 842, 843

summary of events table
LIFETEST procedure, 820, 822

survival plot
LIFETEST procedure, 781

%SurvivalSummaryTable macro
LIFETEST procedure, 836

%SurvivalTable macro
LIFETEST procedure, 835

%SurvTabHeader macro
LIFETEST procedure, 835

template cleanup
LIFETEST procedure, 800, 816, 824, 851

TEMPLATE procedure
EXPAND option, 848

TEST survival-plot option
LIFETEST procedure, 785

TEXTATTRS= GTL option
LIFETEST procedure, 808, 812

tick value modification
LIFETEST procedure, 803

TICKVALUЕLIST= GTL option
LIFETEST procedure, 803

title change
LIFETEST procedure, 801

TitleText0 macro variable
LIFETEST procedure, 808

TitleText1 macro variable
LIFETEST procedure, 808

TitleText2 macro variable
LIFETEST procedure, 808, 818, 823

Unicode
LIFETEST procedure, 812

VIEWMIN= GTL option
LIFETEST procedure, 804

xOptions macro variable
LIFETEST procedure, 808

yOptions macro variable
LIFETEST procedure, 803, 804, 808