SAS/OR® 14.1 User's Guide: Project Management

The GANTT Procedure

This document is an individual chapter from *SAS/OR® 14.1 User's Guide: Project Management*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS/OR® 14.1 User's Guide: Project Management*. Cary, NC: SAS Institute Inc.

SAS/OR® 14.1 User's Guide: Project Management

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

July 2015

Chapter 8
The GANTT Procedure

Contents

Overview: GANTT Procedure

The GANTT procedure produces a Gantt chart, which is a graphical scheduling tool for the planning and control of a project. In its most basic form, a Gantt chart is a bar chart that plots the tasks of a project versus time. PROC GANTT displays a Gantt chart that corresponds to a project schedule such as that produced by the CPM procedure or one that is input directly to the procedure. PROC GANTT offers several options and statements for tailoring the chart to your needs.

Using PROC GANTT, you can plot the predicted early and late schedules and identify critical, supercritical, and slack activities. In addition, you can visually monitor a project in progress with the actual schedule and compare the actual schedule against a target baseline schedule. You can also graphically view the effects of scheduling a project subject to resource limitations. Any combination of these schedules can be viewed simultaneously (provided the relevant data exist) together with any user-specified variables of interest, such as project deadlines and other important dates. PROC GANTT enables you to display the early, late, and actual schedules in a single bar to produce a more meaningful schedule for tracking an activity in progress.

PROC GANTT can display the project logic on the Gantt chart by exhibiting dependencies between tasks by using directed arcs to link the related activities. You can use either the activity-on-arc (AOA) or Activity-on-Node (AON) style of input for defining the project network. In addition, the GANTT procedure recognizes nonstandard precedence types. With PROC GANTT, you can display weekends, holidays, and multiple calendars, and you can depict milestones, reference lines, and a timenow line on the chart. PROC GANTT

enables you to annotate text and graphics on the Gantt chart and provides you with a wide variety of options to control and customize the graphical appearance of the chart.

The GANTT procedure also supports an automatic text annotation facility that is designed specifically for labeling Gantt charts independently of the SAS/GRAPH Annotate facility. It enables you to display label strings with a minimum of effort and data entry while providing the capability for more complex chart labeling situations. An important feature of this facility is the ability to link label coordinates and text strings to variables in the Schedule data set. This means that you can preserve the Label data set even though the schedule dates may change. Several options enable you to customize the annotation, such as the clipping of text strings that run off the page or the chart and the specification of a split character to split labels that are too long.

Using the GANTT procedure, you can produce a wide variety of Gantt charts. You can generate zoned Gantt charts with several options to control its appearance. You can display a zone variable column as well as draw a line demarcating the different zones. You can also control the bar height and bar offset of each type of schedule bar. This enables you to change the display order of the schedules as well as giving you the capability to produce a Gantt chart with embedded bars. You can override the default schedule bar pattern assignments at the activity level. In addition, you can restrict the schedule types to which the specified pattern is to be applied. You can also override the text color for selected columns of activity text at the activity level. These features facilitate the production of multiproject and multiprocess Gantt charts. Finally, you can also associate HTML pages with activity bars and create Web-enabled Gantt charts.

The GANTT procedure enables you to control the number of pages output by the procedure in both horizontal and vertical directions. In addition, you can control the number of jobs displayed per page as well as the number of tickmarks displayed per page. You can display ID variables on every page and even let the procedure display the maximum number of ID variables that can fit on one page. You can number the pages, justify the Gantt chart in the horizontal and vertical directions with respect to the page boundaries, and maintain the original aspect ratio of the Gantt chart on each page.

PROC GANTT gives you the option of displaying the Gantt chart in one of three modes: line-printer, full-screen, or graphics mode. The default mode is graphics mode, which enables you to produce charts of high resolution quality. Graphics mode requires SAS/GRAPH software. See the section "Graphics Version" on page 539 for more information on producing high-quality Gantt charts. You can also produce line-printer quality Gantt charts by specifying the LINEPRINTER option in the PROC GANTT statement. In addition to submitting the output to either a plotter or printer, you can view the Gantt chart at the terminal in full-screen mode by specifying the FULLSCREEN option in the PROC GANTT statement. See the section "Full-Screen Version" on page 535 for more information on viewing Gantt charts in full-screen mode. The GANTT procedure also produces a macro variable that indicates the status of the invocation and also contains other useful statistics about the Gantt charts generated by the invocation.

There are several distinctive features that characterize the appearance of the chart produced by the GANTT procedure:

- The horizontal axis represents time, and the vertical axis represents the sequence of observations in the data set.
- Both the time axis and the activity axis can be plotted across more than one page.
- The procedure automatically provides extensive labeling of the time axis, enabling you to determine easily the exact time of events plotted on the chart. The labels are determined on the basis of the formats of the times being plotted. You can also specify user-defined formats for the labeling.

- In graphics mode, the COMPRESS option in the CHART statement enables you to produce the entire Gantt chart on one page. The PCOMPRESS option enables you to produce the entire Gantt chart on one page while maintaining the original aspect ratio of the Gantt chart. Both these options work in conjunction with the HPAGES= and VPAGES= options, which specify the number of pages in the horizontal and vertical directions for the chart.

Project information is communicated into PROC GANTT using SAS data sets. The input data sets used by PROC GANTT are as follows:

- **The Schedule data set** contains the early, late, actual, resource-constrained, and baseline schedules and any other activity-related information. The activity-related information can include precedence information, calendar used by the activity, special dates, and any other information that you want to identify with each activity. This data set can be the same as the Schedule data set produced by PROC CPM, or it can be created separately by a DATA step. Each observation in the Schedule data set represents an activity and is plotted on a separate row of the chart unless activity splitting during resource-constrained scheduling has caused an activity to split into disjoint segments. For details regarding the output format in this case, see the section "Displayed Output" on page 562.

- **The Precedence (Logic) data set** contains the precedence information of the project in AON format in order to draw a Logic Gantt chart of the project. Specifying this data set is not necessary if the precedence information exists in the Schedule data set. If the data set is specified, however, the ACTIVITY variable must exist in both the Schedule and Precedence data sets.

  Typically you would use this feature when scheduling in PROC CPM with nonstandard precedence constraints where the LAG variables are not transferred to the Schedule data set or with the COLLAPSE option. Setting the Precedence data set for PROC GANTT to be the Activity data set (used in PROC CPM) establishes the required precedence relationships. This is also a convenient feature when drawing several Gantt charts for the same project with different schedule information (such as when monitoring a project in progress). Specifying a Precedence data set avoids having to duplicate the precedence information in every Schedule data set.

- **The Label data set** contains the label information of the project that enables you to draw labeled Gantt charts independently of the SAS/GRAPH Annotate facility. It requires a minimum of effort and provides you with a convenient mechanism to link label strings and their coordinates to variables in the Schedule data set. Another convenient feature is its ability to replicate labels across all activities. Both these features facilitate reuse of the Label data set.

- **The Workday and the Calendar data sets** together enable you to represent any type of work pattern, during a week and within each day of the week, on the Gantt chart. The same Workday and Calendar data sets used by PROC CPM can also be passed to PROC GANTT.

- **The Holiday data set** enables you to associate standard holidays and vacation periods with each calendar and represent them on the Gantt chart. Like the Workday and Calendar data sets, the same Holiday data set used by PROC CPM can also be used by PROC GANTT.

- **The Annotate data set** contains the graphics and text that are to be annotated on the Gantt chart. This data set is used by the GANTT procedure in conjunction with the Annotate facility in SAS/GRAPH software.

The GANTT procedure produces one output data set.

- **The Imagemap data set** contains the outline coordinates for the schedule bars used in the Gantt chart that can be used to generate HTML MAP tags.

When displaying the precedence relationships between activities on the Gantt chart, bear in mind the following facts with regard to data sets used by PROC GANTT:

- The Schedule data set (and optionally the Precedence data set) contains the variables that define the precedence relationships between activities in the project.
- You can handle nonstandard precedence constraints in PROC GANTT when using AON format by identifying the LAG variables in the CHART statement.
- When you use PROC CPM to produce the schedule for a project with nonstandard precedence relationships, the LAG variables are not automatically included in the Schedule data set. Use an ID statement or the XFERVARS option in the PROC CPM statement to add them.
- When you generate the schedule using PROC CPM with the COLLAPSE option, it is recommended that you use the Activity data set to define the precedence relationships for the Gantt procedure by specifying the PRECDATA= option in the PROC GANTT statement. This ensures that all the relevant precedence information is extracted.

Each option and statement available in the GANTT procedure is explained in the section "Syntax: GANTT Procedure" on page 496. The section "Examples: GANTT Procedure" on page 570 illustrates most of these options and statements.

Getting Started: GANTT Procedure

In order to draw a Gantt chart, at the very minimum you need a Schedule data set. This data set is expected to be *similar* to the OUT= Schedule data set produced by PROC CPM, with each observation representing an activity in the project. It is possible to obtain a detailed Gantt chart by specifying the following single statement:

PROC GANTT *DATA= SAS-data-set* **;**

The data set specified is the Schedule data set produced by PROC CPM.

As an example of this, consider the software development project in the "Getting Started" section in Chapter 4, "The CPM Procedure." The output schedule for this example is saved in a data set, INTRO1, which is displayed in Figure 8.1.

Figure 8.1 Software Project Plan

Project Schedule

Obs	activity	succesr1	succesr2	duration	descrpt
1	TESTING	RECODE		20	Initial Testing
2	PRELDOC	DOCEDREV	QATEST	15	Prel. Documentation
3	MEETMKT	RECODE		1	Meet Marketing
4	RECODE	DOCEDREV	QATEST	5	Recoding
5	QATEST	PROD		10	QA Test Approve
6	DOCEDREV	PROD		10	Doc. Edit and Revise
7	PROD			1	Production

Obs	E_START	E_FINISH	L_START	L_FINISH	T_FLOAT	F_FLOAT
1	01MAR04	20MAR04	01MAR04	20MAR04	0	0
2	01MAR04	15MAR04	11MAR04	25MAR04	10	10
3	01MAR04	01MAR04	20MAR04	20MAR04	19	19
4	21MAR04	25MAR04	21MAR04	25MAR04	0	0
5	26MAR04	04APR04	26MAR04	04APR04	0	0
6	26MAR04	04APR04	26MAR04	04APR04	0	0
7	05APR04	05APR04	05APR04	05APR04	0	0

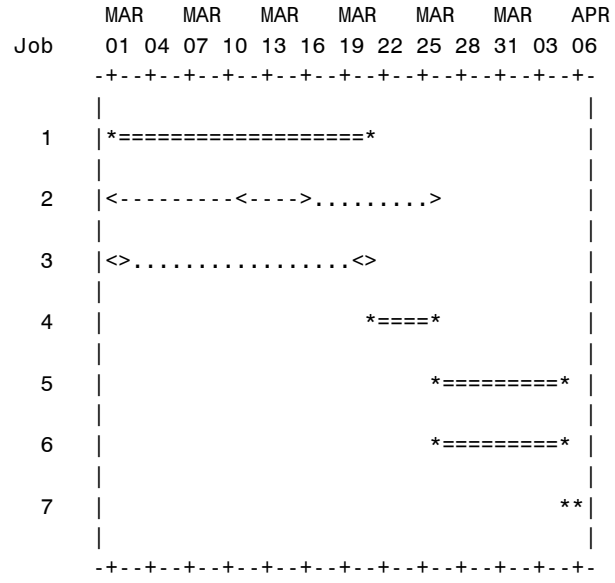
The following code produces the Gantt chart shown in Figure 8.2.

```
title 'Line-Printer Gantt Chart';
proc gantt lineprinter data=intro1;
run;
```

The DATA= option could be omitted if the INTRO1 data set is the most recent data set created; by default, PROC GANTT uses the _LAST_ data set.

Figure 8.2 Line-Printer Gantt Chart

Line-Printer Gantt Chart



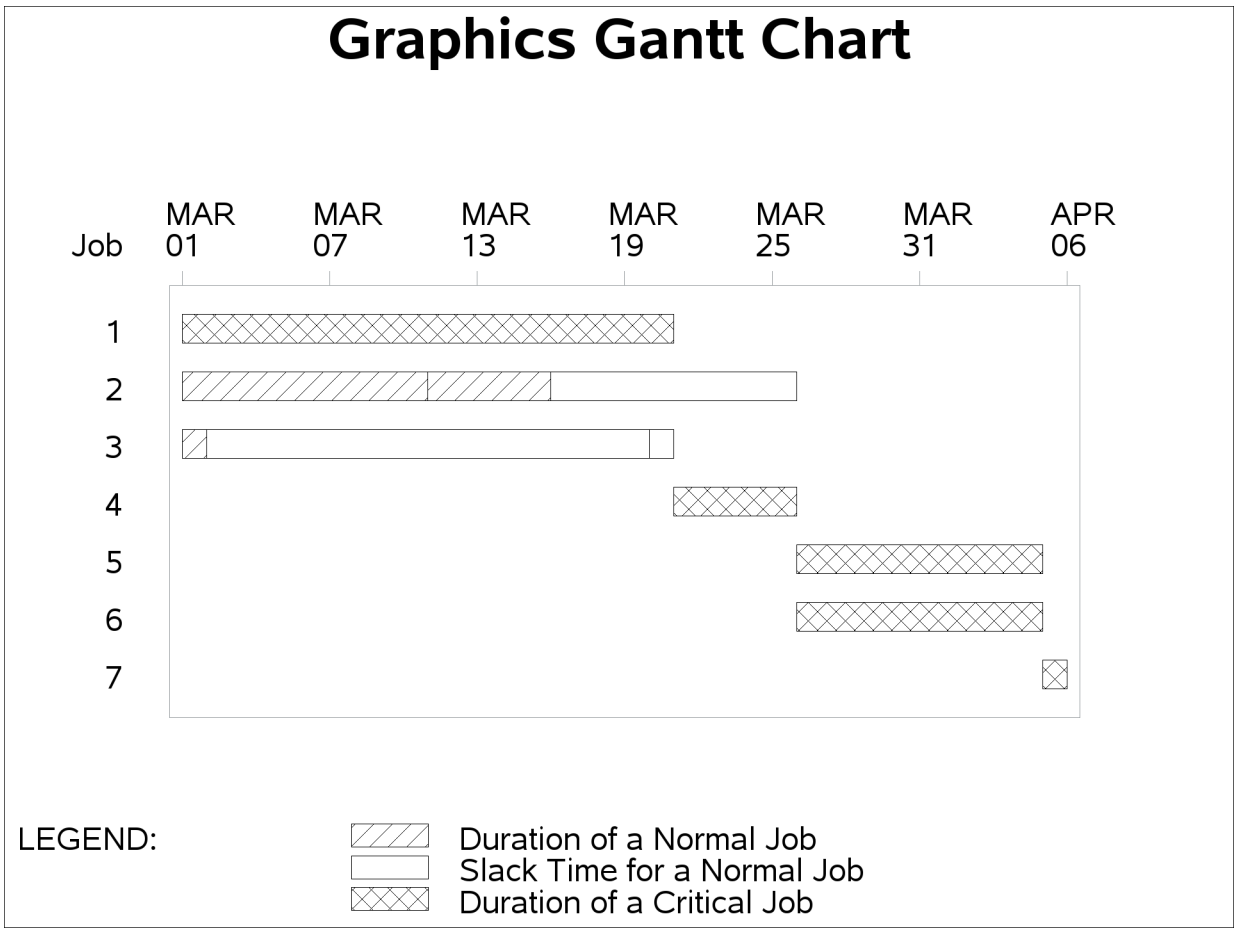
LEGEND

Symbol	Explanation
<---->	Duration of a Normal Job
>....>	Slack Time for a Normal Job
====	Duration of a Critical Job

You can produce a high-resolution graphics quality Gantt chart by specifying the GRAPHICS option instead of the LINEPRINTER option in the PROC GANTT statement. Graphics mode is also the default display mode. The resulting Gantt chart is shown in Figure 8.3.

```
proc gantt graphics data=intro1;
run;
```

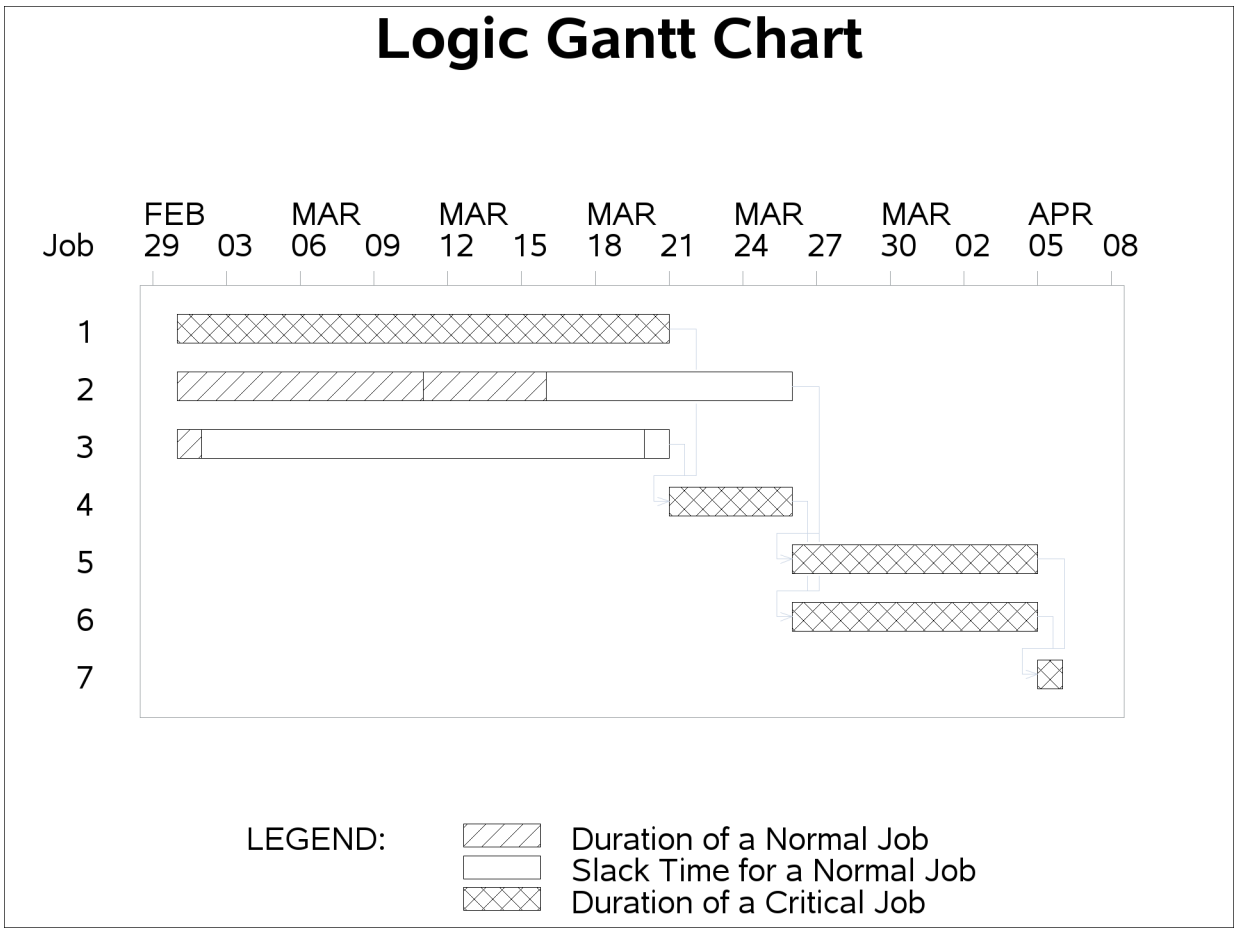
Figure 8.3 Graphics Gantt Chart



Finally, you can draw a Logic Gantt chart by defining the precedence information to PROC GANTT in AON format using the ACTIVITY= and SUCCESSOR= options in the CHART statement. The Logic Gantt chart is shown in Figure 8.4.

```
proc gantt data=intro1;
   chart / activity=activity successor=(succesr1-succesr2);
run;
```

Figure 8.4 Logic Gantt Chart



For further examples illustrating typical invocations of the GANTT procedure when managing projects, see Chapter 3, "Introduction to Project Management."

Syntax: GANTT Procedure

The following statements are used in PROC GANTT:

PROC GANTT *options* **;**
BY *variables* **;**
CHART *specifications / options* **;**
ID *variables* **;**

Functional Summary

Table 8.1 outlines the options available for the GANTT procedure, classified by function.

Table 8.1 Functional Summary

Description	Statement	Option
Axis Formatting Options		
Specifies the increment for labeling axis	CHART	INCREMENT=
Specifies the ending time for axis	CHART	MAXDATE=
Specifies the starting time for axis	CHART	MINDATE=
Specifies the smallest interval identified on chart	CHART	MININTERVAL=
Suppresses time portion of datetime tickmark	CHART	NOTMTIME
Specifies the number of columns per mininterval	CHART	SCALE=
Specifies the format of time axis labels	CHART	TIMEAXISFORMAT=
Uses first plot variable format for tickmarks	CHART	USEFORMAT
Bar Enhancement Options		
Specifies the actual bar height	CHART	ABARHT=
Specifies the actual bar offset	CHART	ABAROFF=
Specifies the default bar height	CHART	BARHT=
Specifies the default bar offset	CHART	BAROFF=
Specifies the baseline bar height	CHART	BBARHT=
Specifies the baseline bar offset	CHART	BBAROFF=
Specifies the color of connect line	CHART	CHCON=
Specifies the early/late bar height	CHART	EBARHT=
Specifies the early/late bar offset	CHART	EBAROFF=
Specifies the holiday bar height	CHART	HBARHT=
Specifies the holiday bar offset	CHART	HBAROFF=
Specifies the character for drawing connect line	CHART	HCONCHAR=
Draws a horizontal connect line	CHART	HCONNECT
Specifies the characters for drawing schedule	CHART	JOINCHAR=
Specifies the line style of connect line	CHART	LHCON=
Suppresses PATTERN variable for bar fills	CHART	NOPATBAR
Specifies the overprint character for schedule variables	CHART	OVERLAPCH=
Specifies the overprint character for CHART variables	CHART	OVPCHAR=
Specifies the schedule types that use the PATTERN variable	CHART	PATLEVEL=
Specifies the PATTERN variable for bar fills and text color	CHART	PATTERN=

Table 8.1 *continued*

Description	Statement	Option
Specifies the resource bar height	CHART	RBARHT=
Specifies the resource bar offset	CHART	RBAROFF=
Specifies the characters for plotting times	CHART	SYMCHAR=
Calendar Options		
Specifies the calendar identifier	CHART	CALID=
Specifies the length of the workday	CHART	DAYLENGTH=
Specifies the beginning of the workday	CHART	DAYSTART=
Marks all breaks in a day	CHART	MARKBREAK
Marks all non-working days	CHART	MARKWKND
Data Set Options		
Specifies the Annotate data set	GANTT	ANNOTATE=
	CHART	ANNOTATE=
Specifies the Calendar data set	GANTT	CALEDATA=
Specifies the Schedule data set	GANTT	DATA=
Specifies the Holiday data set	GANTT	HOLIDATA=
Specifies the Imagemap output data set	GANTT	IMAGEMAP=
Specifies the Label data set	GANTT	LABDATA=
Specifies the Precedence (Logic) data set	GANTT	PRECDATA=
Specifies the Work pattern data set	GANTT	WORKDATA=
Graphics Catalog Options		
Specifies the description of the catalog entry	CHART	DESCRIPTION=
Specifies the name of graphics catalog	GANTT	GOUT=
Specifies the name of catalog entry	CHART	NAME=
Holiday Options		
Specifies the character for plotting holidays	CHART	HOLICHAR=
Specifies the holiday start variable	CHART	HOLIDAY=
Specifies the holiday duration variable	CHART	HOLIDUR=
Specifies the holiday finish variable	CHART	HOLIFIN=
Specifies the holiday duration units	CHART	INTERVAL=
ID Variable Options		
Specifies the number of columns between ID variables	CHART	BETWEEN=
Marks critical activities	CHART	CRITFLAG
Specifies the activity text columns that use pattern color	CHART	CTEXTCOLS=
Allows duplicate ID values	CHART	DUPOK
Displays ID variables on every page	CHART	IDPAGES
Maximizes number of ID variables on page	CHART	MAXIDS
Suppresses job number	CHART	NOJOBNUM
Specifies the split character for dividing ID labels	GANTT	SPLIT=
Strips leading blanks from character variables	GANTT	STRIPIDBLANKS

Table 8.1 *continued*

Description	Statement	Option
Labeling Options		
Specifies the label variable linking to Schedule data set	CHART	LABVAR=
Specifies the rules for label layout	CHART	LABRULE=
Specifies the split character for labels	CHART	LABSPLIT=
Specifies the maximum number of digits in integer label	GANTT	LABMAXINT=
Logic Options		
Specifies the activity variable for AON format	CHART	ACTIVITY=
Uses AOA precedence specifications	CHART	AOA
Specifies the color of precedence connections	CHART	CPREC=
Specifies the headnode variable for AOA format	CHART	HEAD=
Specifies the lag variable for AON format	CHART	LAG=
Specifies the schedule bar connected to precedence lines	CHART	LEVEL=
Specifies the line style of precedence connections	CHART	LPREC=
Specifies the maximum displacement of local vertical	CHART	MAXDISLV=
Specifies the minimum interdistance of global verticals	CHART	MININTGV=
Specifies the minimum offset of global vertical	CHART	MINOFFGV=
Specifies the minimum offset of local vertical	CHART	MINOFFLV=
Suppresses drawing arrow head	CHART	NOARROWHEAD
Suppresses automatic range extension	CHART	NOEXTRANGE
Terminates procedure if bad precedence data	CHART	SHOWPREC
Specifies the successor variable for AON format	CHART	SUCCESSOR=
Specifies the tailnode variable for AOA format	CHART	TAIL=
Specifies the width of precedence connections	CHART	WPREC=
Milestone Options		
Specifies the milestone color	CHART	CMILE=
Specifies the duration variable	CHART	DUR=
Specifies the font for the milestone symbol	CHART	FMILE=
Specifies the milestone height	CHART	HMILE=
Specifies the milestone character	CHART	MILECHAR=
Specifies the value for the milestone symbol	CHART	VMILE=
Miscellaneous Options		
Invokes full-screen version	GANTT	FS
Invokes graphics version	GANTT	GRAPHICS
Invokes line-printer version	GANTT	LP
Specifies the maximum number of decimals for a number	GANTT	MAXDEC=
Specifies the unit for padding finish times	CHART	PADDING=
Specifies the upper limit on number of pages	CHART	PAGES=
Displays summary of symbols and patterns	CHART	SUMMARY

Table 8.1 *continued*

Description	Statement	Option
Page Layout Options		
Positions chart at bottom of page	CHART	BOTTOM
Specifies the axis color	CHART	CAXIS=
Specifies the frame fill color	CHART	CFRAME=
Specifies the width of the chart axis area	CHART	CHARTWIDTH=
Draws chart on one page in graphics mode	CHART	COMPRESS
Fills each page as much as possible	CHART	FILL
Specifies the characters for table outlines and dividers	CHART	FORMCHAR=
Specifies the number of pages spanning time axis	CHART	HPAGES=
Left justifies chart	CHART	LEFT
Specifies the line width	CHART	LWIDTH=
Specifies the number of activities on each page	CHART	NJOBS=
Suppresses frame	CHART	NOFRAME
Suppresses legend	CHART	NOLEGEND
Suppresses page number at upper right corner	CHART	NOPAGENUM
Specifies the number of tickmarks on each page	CHART	NTICKS=
Displays page number at upper right corner	CHART	PAGENUM
Draws chart proportionally on one page	CHART	PCOMPRESS
Right justifies chart	CHART	RIGHT
Specifies the number of rows between consecutive activities	CHART	SKIP=
Positions chart at top of page	CHART	TOP
Specifies the number of pages spanning activity axis	CHART	VPAGES=
Reference Line Options		
Specifies the reference line color	CHART	CREF=
Specifies the reference line style	CHART	LREF=
Specifies the placement of the reference lines	CHART	REF=
Specifies the reference line character	CHART	REFCHAR=
Specifies that reference lines should be labeled	CHART	REFLABEL
Schedule Selection Options		
Specifies the actual start variable	CHART	A_START=
Specifies the actual finish variable	CHART	A_FINISH=
Specifies the baseline start variable	CHART	B_START=
Specifies the baseline finish variable	CHART	B_FINISH=
Concatenates early/late and actual schedules	CHART	COMBINE
Specifies the early start variable	CHART	E_START=
Specifies the early finish variable	CHART	E_FINISH=
Specifies the late start variable	CHART	L_START=
Specifies the late finish variable	CHART	L_FINISH=
Specifies the resource-constrained start variable	CHART	S_START=
Specifies the resource-constrained finish variable	CHART	S_FINISH=

Table 8.1 *continued*

Description	Statement	Option
Timenow Line Options		
Specifies the timenow line color	CHART	CTNOW=
Specifies the timenow line style	CHART	LTNOW=
Suppresses the timenow label	CHART	NOTNLABEL
Specifies the placement of the timenow line	CHART	TIMENOW=
Specifies the timenow line character	CHART	TNCHAR=
Specifies the timenow line width	CHART	WTNOW=
Text Formatting Options		
Specifies the text color	CHART	CTEXT=
Specifies the text font	CHART	FONT=
Specifies the text height multiplier	CHART	HEIGHT=
Specifies the vertical offset for the activity text	CHART	HTOFF=
Web Options		
Specifies the Imagemap output data set	GANTT	IMAGEMAP=
Specifies the HTML variable	CHART	WEB=
Zone Options		
Specifies the zone line color	CHART	CZONE=
Specifies the zone line style	CHART	LZONE=
Suppresses the zone column	CHART	NOZONECOL
Displays only new zone values	CHART	ONEZONEVAL
Specifies the zone line width	CHART	WZONE=
Specifies the zone variable	CHART	ZONE=
Specifies the zone line offset	CHART	ZONEOFF=
Specifies the zone line span	CHART	ZONESPAN=

PROC GANTT Statement

PROC GANTT *options* **;**

The following options can appear in the PROC GANTT statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies the input data set that contains the appropriate Annotate variables for the purpose of adding text and graphics to the Gantt chart. The data set specified must be an Annotate-type data set. See the section "Annotate Processing" on page 541 for information specifically on annotate processing with the GANTT procedure.

The data set specified with the ANNOTATE= option in the PROC GANTT statement is a "global" ANNOTATE= data set, in the sense that the information in this data set is displayed on every Gantt chart produced in the current invocation of PROC GANTT. This option is available only in graphics mode.

See Example 8.21, "Using the SAS/GRAPH ANNOTATE= Option," for further illustration of this option.

CALEDATA=*SAS-data-set*

CALENDAR=*SAS-data-set*

identifies a SAS data set that specifies the work pattern during a standard week for *each* of the calendars that is to be used in the project. Each observation of this data set (also referred to as the Calendar data set) contains the name or the number of the calendar being defined in that observation, the names of the shifts or work patterns used each day, and, optionally, a standard workday length in hours. For details on the structure of this data set, see the section "Multiple Calendars and Holidays" on page 534. The work shifts referred to in the CALEDATA data set are defined in the WORKDATA data set.

DATA=*SAS-data-set*

names the SAS data set that carries the schedule information to be used by PROC GANTT. If the DATA= option is omitted, the most recently created SAS data set is used. This data set, also known as the Schedule data set, contains all the time variables (early, late, actual, resource-constrained, and baseline start and finish times, and any other variables to be specified in a CHART statement) that are to be plotted on the chart. For projects that use multiple calendars, this data set also identifies the calendar that is used by each activity. The Schedule data set also contains precedence information when drawing a Logic Gantt chart in graphics mode. See the section "Schedule Data Set" on page 529 for more details.

FULLSCREEN

FS

indicates that the Gantt chart be drawn in full-screen mode. This mode enables you to scroll horizontally and vertically through the output using commands, pull-down menus, or function keys. See the section "Full-Screen Version" on page 535 for more information.

GOUT=*graphics catalog*

specifies the name of the graphics catalog used to save the output produced by PROC GANTT for later replay. This option is available only in graphics mode.

GRAPHICS

indicates that the Gantt chart produced be of high-resolution quality. This is the default mode of display. If you invoke the GANTT procedure in Graphics mode, but you do not have SAS/GRAPH software licensed at your site, the procedure stops and issues an error message. See the section "Graphics Version" on page 539 for more information.

HOLIDATA=*SAS-data-set*

names the SAS data set that specifies holidays. These holidays can be associated with specific calendars that are also identified in the HOLIDATA data set (also referred to as the Holiday data set). The HOLIDATA= option must be used with the HOLIDAY= option in the CHART statement, which specifies the variable in the SAS data set that contains the start time of holidays. Optionally, the data set can include a variable that specifies the length of each holiday or a variable that identifies the finish time of each holiday (if the holidays are longer than one unit of the INTERVAL= option). For projects involving multiple calendars, this data set can also include the variable named by the CALID= option that identifies the calendar to be associated with each holiday.

IMAGEMAP=*SAS-data-set*

names the SAS data set that receives a description of the areas of a graph and a link for each area. This information is for the construction of HTML image maps. You use a SAS DATA step to process the output file and generate your own HTML files. The graph areas correspond to the link information that comes from the WEB variable in the schedule data set. This gives you complete control over the appearance and structure of your HTML pages.

LABDATA=*SAS-data-set*

LABELDATA=*SAS-data-set*

LABEL=*SAS-data-set*

specifies the input data set that contains the label specific information. This option is required to initiate the automatic text annotation of the Gantt chart. See the section "Label Data Set" on page 557 for information on the variables it can contain. This option is available only in graphics mode.

LABMAXINT=*n*

LMI=*n*

specifies the maximum number of digits in the integer part when displaying an unformatted numeric as a string. The default value is 16. The maximum number of decimal positions is specified using the MAXDEC= option in the PROC GANTT statement. This option is applicable only to labels defined with the Label data set.

LINEPRINTER

LP

indicates that the Gantt chart be drawn in line-printer mode.

MAXDEC=*n*

M=*n*

indicates the maximum number of decimal positions displayed for a number. A decimal specification in a format overrides a MAXDEC= specification. The default value of MAXDEC= is 2.

PRECDATA=*SAS-data-set*

names the SAS data set that contains the variables that define the precedence constraints in AON format. This data set is required if the Schedule data set does not contain the required precedence information as, for example, when the COLLAPSE option in PROC CPM causes some observations to be excluded from the Schedule data set. When this option is specified, it is mandatory that the ACTIVITY variable exist in both data sets and be identical in both type and length. This option is available only in graphics mode.

SPLIT=*'character'*

S=*'character'*

splits labels used as column headings where the split character appears. When you define the value of the split character, you must enclose it in single quotes. In PROC GANTT, column headings for ID variables consist of either variable labels (if they are present and space permits) or variable names. If the variable label is used as the column heading, then the split character determines where the column heading is to be split.

WORKDATA=*SAS-data-set*

WORKDAY=*SAS-data-set*

identifies a SAS data set that defines the work pattern during a standard working day. Each numeric variable in this data set (also referred to as the Workday data set) is assumed to denote a unique shift pattern during one working day. The variables must be formatted as SAS time values, and the observations are assumed to specify, alternately, the times when consecutive shifts start and end.

BY Statement

BY *variables* **;**

A BY statement can be used with PROC GANTT to obtain separate Gantt charts for observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the schedule data to be sorted in order of the BY variables. If your Schedule data set is not sorted, use the SORT procedure with a similar BY statement to sort the data. The chart for each BY group is formatted separately based only on the observations within that group.

CHART Statement

CHART *specifications / options* **;**

The options that can appear in the CHART statement are listed below. The options are classified under appropriate headings: first, all options that are valid for all modes of the procedure are listed, followed by the options classified according to the mode (line-printer, full-screen, or graphics) of invocation of the procedure. Most of the options in line-printer and full-screen modes are also valid in graphics mode with similar interpretations. The differences and similarities in interpretation of the options are documented under the section "Mode-Specific Differences" on page 561.

General Options

The CHART statement controls the format of the Gantt chart and specifies additional variables (other than early, late, actual, resource-constrained, and baseline start and finish times) to be plotted on the chart. For example, suppose a variable that you want to specify in the CHART statement is one that contains the target finish date for each activity in a project; that is, if FDATE is a variable in the Schedule data set containing the desired finish date for each activity, the CHART statement can be used to mark the value of FDATE on the chart for each activity. A CHART specification can be one of the following types:

variable_1 . . . variable_n

variable_1='symbol_1' . . . variable_n='symbol_n'

(variables)='symbol_1' . . . (variables)='symbol_n'

variable_1 . . . variable_n

indicates that each variable is to be plotted using the default symbol, the first character of the variable name. For example, the following statement

```
CHART SDATE FDATE;
```

causes the values of SDATE to be plotted with an 'S' and the values of FDATE with an 'F.'

variable_1='symbol_1' . . . variable_n='symbol_n'
indicates that each variable is to be plotted using the symbol specified. The symbol must be a single character enclosed in quotes.

(variables)='symbol_1' . . . (variables)='symbol_n'
indicates that each variable within the parentheses is to be plotted using the symbol associated with that group. The symbol must be a single character enclosed in single quotes. For example, the following statement

```
CHART (ED SD)='*'
      (FD LD)='+';
```

plots the values of the variables in the first group using an asterisk ('*') and the values of the variables in the second group using a plus sign ('+').

A single CHART statement can contain specifications in more than one of these forms. Also, each CHART statement produces a separate Gantt chart.

NOTE: It is not necessary to specify a CHART statement if default values are to be used to draw the Gantt chart.

The following options can appear in the CHART statement.

A_FINISH=*variable*
AF=*variable*
specifies the variable that contains the actual finish time of each activity in the Schedule data set. This option is not required if the default variable name A_FINISH is used.

A_START=*variable*
AS=*variable*
specifies the variable that contains the actual start time of each activity in the Schedule data set. This option is not required if the default variable name A_START is used.

B_FINISH=*variable*
BF=*variable*
specifies the variable that contains the baseline finish time of each activity in the Schedule data set. This option is not required if the default variable name B_FINISH is used.

B_START=*variable*
BS=*variable*
specifies the variable that contains the baseline start time of each activity in the Schedule data set. This option is not required if the default variable name B_START is used.

BETWEEN=*number*
specifies the number of columns between two consecutive ID variable columns. This option gives you greater flexibility in spacing the ID columns. The default value of the BETWEEN= option is 3.

CALID=*variable*

specifies the variable in the Schedule, Holiday, and Calendar data sets that is used to identify the name or number of the calendar to which each observation refers. This variable can be either numeric or character depending on whether the different calendars are identified by unique numbers or names, respectively. If this variable is not found in any of the three data sets, PROC GANTT looks for a default variable named _CAL_ in that data set (a warning message is issued to the log). For each activity in the Schedule data set, this variable identifies the calendar that is used to mark the appropriate holidays and weekends for the activity. For further details, see the section "Multiple Calendars and Holidays" on page 534.

COMBINE

concatenates the early/late and actual schedule bars of an activity into a single bar and draws a timenow line on the Gantt chart. The COMBINE option does not affect the resource-constrained or baseline schedule bars. If the TIMENOW= option is not specified, it is implicitly assumed to exist and set to missing. The computation of TIMENOW is then carried out as described in the TIMENOW= option. Since the timenow line represents the instant at which a "snapshot" of the project is taken, values less than TIMENOW can be regarded as the "past" and values greater or equal to TIMENOW can be regarded as the "future." The GANTT procedure uses this property of the timenow line to partition the chart into two regions; the region to the left of the timenow line reporting only the actual schedule (events that have already taken place), and the region to the right (including the timenow line) reporting only the predicted early/late schedule.

CRITFLAG

FLAG

indicates that critical jobs be flagged as being critical or super-critical. An activity is critical if its total float is zero. If the total float is negative, the activity is super-critical. Critical activities are marked 'CR,' and super-critical activities are marked 'SC' on the left side of the chart.

DAYLENGTH=*daylength*

specifies the length of the workday. Each workday is plotted starting at the beginning of the day as specified in the DAYSTART= option and ending *daylength* hours later. The value of *daylength* should be a SAS time value. If the INTERVAL= option is specified as DTSECOND, DTMINUTE, DTHOUR, or DTDAY, the default value of *daylength* is 24 hours. If the INTERVAL= option is specified as WORKDAY or DTWRKDAY, the default value of *daylength* is 8 hours. For other values of the INTERVAL= option, the DAYLENGTH= option is ignored.

NOTE: The DAYLENGTH= option is needed to mark the non-worked periods within a day correctly (if the MARKBREAK option is in effect). The DAYLENGTH= option is also used to determine the start and end of a weekend precisely (to the nearest second). This accuracy is needed if you want to depict on a Gantt chart the exact time (for example, to within the nearest hour) for the start and finish of holidays or weekends. This option is used only if the times being plotted are SAS datetime values.

DAYSTART=*daystart*

specifies the start of the workday. The end of the day, *dayend*, is computed as *daylength* seconds after *daystart*. The value of *daystart* should be a SAS time value. This option is to be specified only when the value of the INTERVAL= option is one of the following: WORKDAY, DTSECOND, DTMINUTE, DTHOUR, DTDAY, or DTWRKDAY. For purposes of denoting on the Gantt chart, the weekend is assumed to start at *dayend* on Friday and end at *daystart* on Monday morning. Of course, if the SCALE= and MININTERVAL= values are such that the resolution is not very high, you will be unable

to discern the start and end of holidays and weekends to the nearest hour. The default value of *daystart* is 9:00 a.m. if INTERVAL=WORKDAY or INTERVAL=DTWRKDAY, and midnight otherwise.

DUPOK

causes duplicate values of ID variables *not to be skipped*. As described later in the ID Statement section, if two or more consecutive observations have the same combination of values for all the ID variables, only the first of these observations is plotted. The DUPOK option overrides this behavior and causes *all* the observations to be plotted.

DURATION=*variable*

DUR=*variable*

identifies a variable in the Schedule data set that determines whether or not an activity is to be regarded as a milestone with respect to a specific schedule. This option is not required if the default variable name _DUR_ is used. A value of 0 for this variable indicates that if the start and finish times of the activity with respect to a given schedule are identical (a schedule taken to mean early, late, actual, resource-constrained or baseline), then the activity is represented by a milestone with respect to the given schedule. A nonzero value treats identical start and finish times in the default manner by implicitly padding the finish times as specified by the PADDING= option. The milestone symbol is defined by the MILECHAR= option in line-printer and full-screen modes and by the CMILE= , FMILE= , HMILE= , and VMILE= options in graphics mode; these four options represent the color, font, height, and value of the symbol, respectively. See the descriptions of these options for their default values. To illustrate, suppose that the observations for activities **A** and **B** from the Schedule data set are as follows:

ACTIVITY	E_START	E_FINISH	A_START	A_FINISH	_DUR_
A	27JUL04	27JUL04	31JUL04	31JUL04	1
B	31JUL04	31JUL04	01AUG04	02AUG04	0

In this example, the actual schedule for activity **A** begins on '31JUL04' and finishes at the end of the day, as explained in the section "Schedule Data Set" on page 529. PROC GANTT uses the _DUR_ variable to recognize that activity **A** has nonzero duration, pads the finish time by a PADDING= unit, and displays a bar representing one day. In contrast, the value of '0' for _DUR_ in activity **A** alerts PROC GANTT that padding be ignored for any schedule with identical start and finish times. Consequently, the early schedule for activity **B** is represented on the chart by the milestone symbol at '31JUL04.' The actual schedule, however, not having identical start and finish times, is padded as usual and plotted as starting on '01AUG04' and finishing at the end of '02AUG04.'

E_FINISH=*variable*

EF=*variable*

specifies the variable that contains the early finish time of each activity in the Schedule data set. This option is not required if the default variable name E_FINISH is used.

E_START=*variable*

ES=*variable*

specifies the variable that contains the early start time of each activity in the Schedule data set. This option is not required if the default variable name E_START is used.

FILL

causes each page of the Gantt chart to be filled as completely as possible before a new page is started (when the size of the project requires the Gantt chart to be split across several pages). If the FILL option is not specified, the pages are constrained to contain an approximately equal number of activities. The FILL option is not valid in full-screen mode because all of the activities are plotted on one logical page.

HCONNECT

causes a line to be drawn for each activity from the left boundary of the chart to the beginning of the bar for the activity. This feature is particularly useful when the Gantt chart is drawn on a large page. In this case, the schedule bars for some of the activities may not start close enough to the left boundary of the chart; the connecting lines help to identify the activity associated with each bar.

HOLIDAY=(*variable***)**

HOLIDAYS=(*variable***)**

specifies the date or datetime variable in the Holiday data set that identifies holidays to be marked on the schedule. If there is no end time nor duration specified for the holiday, it is assumed to start at the time specified by the HOLIDAY variable and last one unit of *interval*, where *interval* is the value of the INTERVAL= option.

HOLIDUR=(*variable***)**

HDURATION=(*variable***)**

specifies the variable in the Holiday data set that identifies the durations of the holidays that are to be marked on the schedule.

HOLIFIN=(*variable***)**

HOLIEND=(*variable***)**

specifies the date or datetime variable in the Holiday data set that identifies the finish times of the holidays that are to be marked on the schedule.

IDPAGES

displays ID variables on every page. By default, the ID variables are displayed only on the first page.

INCREMENT=*increment*

specifies the number of *mininterval*s between time axis labels on the Gantt chart. If the INCREMENT= option is not specified, a value is chosen that provides the maximum possible labeling.

INTERVAL=*interval*

HOLINTERVAL=*interval*

specifies the units for the values of the HOLIDUR variables. Valid values for this option are DAY, WEEKDAY, WORKDAY, DTSECOND, DTMINUTE, DTHOUR, DTDAY, or DTWRKDAY. If the value for the INTERVAL= option has been specified as WEEKDAY, WORKDAY, or DTWRKDAY, weekends are also marked on the Gantt chart with the same symbol as holidays for line-printer quality charts. Graphics-quality Gantt charts use the same PATTERN statement as the one used for marking holidays. The default value of the INTERVAL= option is DAY if the times being plotted are SAS date values and is DTDAY if the times being plotted are SAS datetime values. See the section "Specifying the INTERVAL= Option" on page 535 for further information regarding this option.

L_FINISH=*variable*

LF=*variable*

specifies the variable that contains the late finish time of each activity in the Schedule data set. This option is not required if the default variable name L_FINISH is used.

L_START=*variable*

LS=*variable*

specifies the variable that contains the late start time of each activity in the Schedule data set. This option is not required if the default variable name L_START is used.

MARKBREAK

causes all breaks (non-worked periods) during a day to be marked on the Gantt chart. The symbol used for marking the breaks is the same as the HOLICHAR= symbol. This option may not be of much use unless the chart has been plotted with a scale that enables you to discern the different hours within a day on the Gantt chart. For instance, if the chart is in terms of days, there is no point in trying to show the breaks within a day; on the other hand, if it is in terms of hours or seconds, you may want to see the start and end of the various shifts within a day. This option turns on the MARKWKND option.

MARKWKND

causes all weekends (or non-worked days during a week) to be marked on the Gantt chart. The symbol used for marking weekends is the same as the HOLICHAR= symbol. Note that weekends are also marked on the chart if the value of the INTERVAL= option is WEEKDAY, WORKDAY, or DTWRKDAY.

MAXDATE=*maxdate*

specifies the end time for the time axis of the chart. The default value is the largest value of the times being plotted unless the logic options are invoked without the NOEXTRANGE option in the CHART statement. For a discussion of the default behavior in this instance, see the section "Formatting the Axis" on page 552.

MAXIDS

displays as many consecutive ID variables as possible in the presence of an ID statement. In the absence of this option, the default displays *all* of the variables or *none* if this is not possible.

MINDATE=*mindate*

specifies the starting time for the time axis of the chart. The default value is the smallest value of the times being plotted unless the logic options are invoked without the NOEXTRANGE option in the CHART statement. For a discussion of the default behavior in this instance, see the section "Formatting the Axis" on page 552.

MININTERVAL=*mininterval*

specifies the smallest interval to be identified on the chart. For example, if MININTERVAL=DAY, then one day is represented on the chart by *scale* (see the SCALE= option) number of columns. The default value of the MININTERVAL= option is chosen on the basis of the formats of the times being plotted, as explained in the section "Specifying the MININTERVAL= Option" on page 533. See also the section "Page Format" on page 532 for a further explanation of how to use the MININTERVAL= option in conjunction with the SCALE= option.

NOJOBNUM
suppresses displaying an identifying job number for each activity. By default, the job number is displayed to the left of the Gantt chart.

NOLEGEND
suppresses displaying the concise default legend at the bottom of each page of the Gantt chart. The NOLEGEND option is not effective in full-screen mode.

NOTNLABEL
suppresses displaying the timenow label. By default, the label is displayed on the bottom border of the chart.

PADDING=*padding*

FINPAD=*padding*
requests that finish times on the chart be increased by one *padding* unit. An exception to this is when a milestone is to be plotted. See the DUR= option for further information regarding this. The PADDING= option enables the procedure to mark the finish times as the end of the last time period instead of the beginning. Possible values for *padding* are NONE, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QTR, YEAR, DTSECOND, DTMINUTE, DTHOUR, DTWEEK, DTMONTH, DTQTR, or DTYEAR. The default value is chosen on the basis of the format of the times being plotted. See the section "Specifying the PADDING= Option" on page 532 for further explanation of this option.

PAGELIMIT=*pages*

PAGES=*pages*
specifies an upper limit on the number of pages allowed for the Gantt chart. The default value of *pages* is 100. This option is useful for preventing a voluminous amount of output from being generated by a wrong specification of the MININTERVAL= or SCALE= option. This option is ignored in full-screen mode.

REF=*values*
indicates the position of one or more vertical reference lines on the Gantt chart. The values allowed are constant values. Only those reference lines that fall within the scope of the chart are displayed.

In line-printer and full-screen modes, the reference lines are displayed using the character specified in the REFCHAR= option. In graphics mode, use the CREF=, LREF=, and LWIDTH= options to specify the color, style, and width of the reference lines.

REFLABEL
specifies that the reference lines are to be labeled. The labels are formatted in the same way as the time axis labels and are placed along the bottom border of the Gantt chart at the appropriate points. If the reference lines are too numerous and the scale does not allow all the labels to be nonoverlapping, then some of the labels are dropped.

S_FINISH=*variable*

SF=*variable*
specifies the variable that contains the resource-constrained finish time of each activity in the Schedule data set. This option is not required if the default variable name S_FINISH is used.

S_START=*variable*

SS=*variable*

specifies the variable that contains the resource-constrained start time of each activity in the Schedule data set. This option is not required if the default variable name S_START is used.

SCALE=*scale*

requests that *scale* number of columns on the chart represent one unit of *mininterval* where *mininterval* is the value of the MININTERVAL= option. In line-printer and graphics modes, the default value of the SCALE= option is 1 if the time axis of the chart is too wide to fit on one page. If the time axis fits on less than one page, then a default value is chosen that expands the time axis as much as possible but still fits the time axis on one page. In full-screen mode, the default value of the SCALE= option is always 1.

SKIP=*skip*

S=*skip*

requests that *skip* number of lines be skipped between the plots of the schedules of two activities. The SKIP= option can take integer values between 0 and 4, inclusive. In graphics mode, 0 is not a valid value. The default value of the SKIP= option is 1.

STRIPIDBLANKS

STRIPID

strips all leading blanks from character ID variables. The default behavior is to preserve any leading blanks.

SUMMARY

requests that a detailed description of all symbols and patterns that are used in the Gantt chart be displayed before the first page of the chart. In line-printer mode, this description includes examples of some strings that could occur in the body of the Gantt chart. The SUMMARY option is not supported in full-screen mode.

TIMEAXISFORMAT=*format|(format_1 < . . . , format_3>)*

TAFORMAT=*format|(format_1 < . . . , format_3>)*

specifies formats for up to three rows of time-axis labeling. One time-axis row is displayed for each format specified. The formats control the rows of the time-axis from top to bottom. Missing formats yield a blank row.

TIMENOW=*value*

specifies the position for the timenow line on the chart. If the value is invalid or set to missing, TIMENOW is set to be the time period that follows the maximum of all specified actual times. If there are no actual times, TIMENOW is set to be equal to the current date. The value of TIMENOW is written to the log.

The timenow line has precedence over all other variables and reference lines and is drawn only if it falls within the range of the chart axis. If TIMENOW is based on the maximum of the actual times, and the MAXDATE= option is not specified, then the range of the chart axis is increased, if necessary, to display the timenow line. The timenow line is labeled by default; the label is formatted in the same way as the time axis and is placed along the bottom border of the chart. The timenow line is displayed in line-printer and full-screen modes using the character specified by the TNCHAR= option (or `T`, if none is specified) in the CHART statement. In graphics mode, use the CTNOW=, LTNOW=, and

WTNOW= options in the CHART statement to specify the color, style, and width of the timenow line. In the presence of a timenow line, the actual schedule for an activity with an actual start less than TIMENOW and a missing actual finish time is represented on the Gantt chart by a bar that begins at the actual start and ends at TIMENOW to indicate that the activity is in progress at TIMENOW. This behavior is consistent with the convention used by PROC CPM. A warning is also issued to the log in this case.

USEFORMAT

specifies that the tickmark labels of the Gantt chart axis are to be displayed using the format associated with the first plot variable appearing in the order E_START=, E_FINISH=, L_START=, L_FINISH=, A_START=, A_FINISH=, S_START=, S_FINISH=, B_START=, B_FINISH=. This format is also used for labeling any reference lines and the timenow line.

NOTE: An INFORMAT statement might be necessary to identify user-defined formats. This enables the GANTT procedure to recognize the data type of the start and finish times specified in the input data set.

Full-Screen and Line-Printer Options

The following options can appear in the CHART statement and are specifically for the purpose of producing Gantt charts in line-printer and full-screen modes.

FORMCHAR[*index list* **]=***'string'*

defines the characters to be used for constructing the chart outlines and dividers. The value is an 11-character string: the first character represents the vertical bar, the second character represents the horizontal bar, and the remaining characters represent the upper left, upper middle, upper right, middle left, middle middle (cross), middle right, lower left, lower middle, and lower right, respectively. The default value of the FORMCHAR= option is `'|----|+|---'`. You can substitute any character or hexadecimal string to customize the chart appearance. Use an index list to specify which default form character each supplied character replaces, or replace the entire default string by specifying the full 11 character replacement string with no index list. For example, change the four corners to asterisks by using

```
formchar(3 5 9 11)= '****'
```

Specifying the following produces charts with no outlines or dividers.

```
formchar='           '    (11 blanks)
```

If you route your output to an IBM 6670 printer that uses an extended font (typestyle 27 or 225) with input character set 216, it is recommended that you specify

```
formchar='FABFACCCBCEB8FECABCBBB'X
```

If you use a printer with a TN (text) print train, it is recommended that you specify the following:

```
formchar='4FBFACBFBC4F8F4FABBFBB'X
```

HCONCHAR=*'character'*

specifies the symbol to be used for drawing the connecting line described in the HCONNECT option. The default character is `-` . This is a line-printer option and is not valid in conjunction with the GRAPHICS option. For corresponding graphics options, see the LHCON= and CHCON= options described later in this section under "Graphics Options."

HOLICHAR=*'character'*

indicates the character to display for holidays. Note that PROC GANTT displays only those holidays that fall within the duration or the slack time of an activity. The default character used for representing holidays is `!` .

JOINCHAR=*'string'*

defines a string eight characters long that identifies nonblank characters to be used for drawing the schedule. The first two symbols are used to plot the schedule of an activity with positive total float. The first symbol denotes the duration of such an activity while the second symbol denotes the slack present in the activity's schedule. The third symbol is used to plot the duration of a *critical* activity (with zero total float). The next two symbols are used to plot the schedule of a *supercritical* activity (one with negative float). Thus, the fourth symbol is used to plot the negative slack of such an activity starting from the late start time (to early start time), and the fifth symbol is used to plot the duration of the activity (from early start to early finish). The sixth symbol is used to plot the actual schedule of an activity if the A_START and A_FINISH variables are specified. The seventh symbol is used to plot the resource-constrained schedule of an activity if the S_START and S_FINISH variables are specified. The eighth symbol is used to plot the baseline schedule of an activity if the B_START and B_FINISH variables are specified. The default value of the JOINCHAR= option is `'-.=-*-*_'` .

MILECHAR=*'character'*

indicates the character to display for the milestone symbol. If this option is not used, the letter `M` is used. In the event that another milestone or a character representing a start or finish time is to be plotted in this column, the OVERLAPCH= character is used.

OVERLAPCH=*'character'*

OVLPCHAR=*'character'*

indicates the overprint character to be displayed when more than one of the symbols in SYMCHAR=*'string'* or MILECHAR=*'character'* are to be plotted in the same column. The default character is `*` .

OVPCHAR=*'character'*

indicates the character to be displayed if one of the variables specified in the CHART statement is to be plotted in the same column as one of the start or finish times. If no OVPCHAR= option is given, the 'at' symbol (`@`) is used. Note that if one of the E_START, E_FINISH, L_START, L_FINISH, A_START, A_FINISH, S_START, S_FINISH, B_START, or B_FINISH times coincides with another, the overprint character to be displayed can be specified separately using the OVERLAPCH= option.

REFCHAR=*'character'*

indicates the character to display for reference lines. If no REFCHAR= option is given, the vertical bar (|) is used. If a time variable value is to be displayed in the column where a REF= value goes, the plotting symbol for the time variable is displayed instead of the REFCHAR= value. Similarly, the HOLICHAR= symbol has precedence over the REFCHAR= value.

SYMCHAR=*'string'*

defines the symbols to be used for plotting the early start, late start, early finish, late finish, actual start and finish, resource-constrained start and finish, and baseline start and finish times, in that order. The default value is `'<<>>**<>[]'`. If any of the preceding symbols coincide with one another or with the milestone symbol, the symbol plotted is the one specified in the OVERLAPCH= option (or `*`, if none is specified). If the actual, resource-constrained, and baseline schedules are not plotted on the chart, you can specify only the first four symbols. If fewer than the required number of symbols are specified, nonspecified symbols are obtained from the default string.

TNCHAR=*'character'*

indicates the character to display for the timenow line. If this option is not used, the letter `T` is used.

Graphics Options

The following describes the interpretation of the CHART specification in graphics mode. Note that the GANTT procedure is not supported with the ActiveX or Java series of devices on the GOPTIONS statement.

As before, the CHART statement controls the format of the Gantt chart and specifies additional variables (other than the early, late, actual, resource-constrained, and baseline start and finish times) to be plotted on the chart. The same forms for the specification of CHART variables (as in the line-printer and full-screen version) are allowed, although the interpretation is somewhat different. Each form of specification is repeated here with a corresponding description of the interpretation. Note that the symbols for any activity are plotted on a line above the one corresponding to that activity. In addition to plotting the required symbol, PROC GANTT draws a vertical line below the symbol in the same color as the symbol. The length of the line is the same as the height of the bars (referred to as bar height) that represent the durations of the activities on the Gantt chart. This line helps identify the exact position of the plotted value. See also the section "Special Fonts for Project Management and Decision Analysis" on page 547 for information on a special set of symbols that are suitable for representing CHART variables on a Gantt chart.

variable1 . . . variablen

indicates that each variable is to be plotted using symbols specified in SYMBOL statements. The *i*th variable in the list is plotted using the plot symbol, color, and font specified in the *i*th SYMBOL statement. The height specified in the SYMBOL statement is multiplied by the bar height to obtain the height of the symbol that is plotted. Thus, if `H=0.5` in the first SYMBOL statement, and the bar height is 5% of the screen area, then the first symbol is plotted with a height of 2.5%. For example, suppose the following two SYMBOL statements are in effect:

```
SYMBOL1 V=STAR C=RED   H=1;
SYMBOL2 V=V    C=GREEN H=0.5 F=GREEK;
```

Then, the following statement

```
CHART SDATE FDATE;
```

causes values of SDATE to be plotted with a red star that is as high as each bar and the values of FDATE with an inverted green triangle that is half as high as the bar height. See the section "Using SYMBOL Statements" on page 545 for further information on using the SYMBOL statement.

variable1='symbol1' . . . variablen='symboln'
indicates that each variable is to be plotted using the symbol specified. The symbol must be a single character enclosed in quotes. The font used for the symbol is the same as the font used for the text.

(variables)='symbol1' . . . (variables)='symboln'
indicates that each variable in parentheses is to be plotted using the symbol associated with that group. The symbol must be a single character enclosed in single quotes. For example, the following statement

```
CHART (ED SD)='*'
      (FD LD)='+';
```

plots the values of variables in the first group using an asterisk (*) and the values of variables in the second group using a plus sign (+).

A single CHART statement can contain requests in more than one of these forms.

NOTE: It is not necessary to specify a CHART statement if only default values are used to draw the Gantt chart.

The following options can appear in the CHART statement specifically for the production of high-resolution graphics quality Gantt charts.

ABARHT=*h*
specifies that the height of the actual schedule bar be *h* cellheights. The value of *h* is restricted to be a positive real number. The default bar height is one cellheight. This specification will override a BARHT= specification. In the event that the actual schedule bar corresponds to the logic bar (using the LEVEL= option), the value is ignored and the default value is used instead. Any non-working days corresponding to this schedule bar are also drawn using the same height as the schedule bar unless the HBARHT= option is specified.

ABAROFF=*d*
specifies that the actual schedule bar be offset *d* cellheights from its default position. A value of zero corresponds to the default position. The direction of increase is from top to bottom. This specification will override a BAROFF= specification. In the event that the actual schedule bar corresponds to the logic bar (specified using the LEVEL= option), the value is ignored and the default value is used instead. Any non-working days corresponding to this schedule bar are drawn using the offset of the schedule bar unless the HBAROFF= option is specified.

ACTIVITY=*variable*

ACT=*variable*
specifies the variable identifying the names of the nodes representing activities in the Schedule data set. This option is required when the precedence information is specified using the AON format. The variable can be either numeric or character in type. If the PRECDATA= option is specified, then this variable must also exist in the Precedence data set and have identical type and length.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*
specifies the input data set that contains the appropriate Annotate variables for the purpose of adding text and graphics to the Gantt chart. The data set specified must be an Annotate-type data set. See also the section "Annotate Processing" on page 541 for information specifically on annotate processing with the GANTT procedure.

The ANNOTATE= data set specified in a CHART statement is used only for the Gantt chart created by that particular CHART statement. You can also specify an ANNOTATE= data set in the PROC GANTT statement, which provides "global" Annotate information to be used for all Gantt charts created by the procedure.

AOA

causes PROC GANTT to use the specification for the AOA format for producing a Logic Gantt chart when the precedence information has been specified in both AOA format (TAIL= and HEAD= options) and AON format (ACTIVITY=, SUCCESSOR=, and, optionally, LAG= options). The default behavior is to use the AON format.

BARHT=*h*

specifies that the height of all the schedule bars be *h* cellheights. The value of *h* is restricted to be a positive real number. The default value is one cellheight. This specification can be overridden for each schedule type by specifying the bar height option appropriate for that schedule type. If a Logic Gantt chart is produced, the specified bar height is ignored for the logic bar (specified using the LEVEL= option) and the default bar height of one cellheight is used for it instead. All non-working days corresponding to a schedule bar are drawn using the height of the schedule bar unless the HBARHT= option is specified.

BAROFF=*d*

specifies that all the schedule bars be offset *d* cellheights from their default positions. A value of zero corresponds to the default positions. The direction of increase is from top to bottom. This specification can be overridden for each schedule type by specifying the bar offset option that is appropriate for that schedule type. If a Logic Gantt chart is produced, the specified bar offset is ignored for the logic bar (specified using the LEVEL= option) and the default bar offset of zero used instead.

BBARHT=*h*

specifies that the height of the baseline schedule bar be *h* cellheights. The value of *h* is restricted to be a positive real number. The default bar height is one cellheight. This specification overrides a BARHT= specification. In the event that the baseline schedule bar corresponds to the logic bar (using the LEVEL= option), the value is ignored and the default value is used instead. Any non-working days corresponding to this schedule bar are also drawn using the same height as the schedule bar unless the HBARHT= option is specified.

BBAROFF=*d*

specifies that the baseline schedule bar be offset *d* cellheights from its default position. A value of zero corresponds to the default position. The direction of increase is from top to bottom. This specification overrides a BAROFF= specification. In the event that the baseline schedule bar corresponds to the logic bar (specified using the LEVEL= option), the value is ignored and the default value is used instead. Any non-working days corresponding to this schedule bar are drawn using the offset of the schedule bar unless the HBAROFF= option is specified.

BOTTOM

BJUST

positions the bottom of the Gantt chart at the bottom of the page, just above the footnotes. This option is ignored if you specify the TOP or TJUST option.

CAXIS=*color*

CAXES=*color*

CA=*color*

specifies the color to use for displaying axes for the Gantt chart. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

CFRAME=*color*

CFR=*color*

specifies the color to use for filling the axis area. If the CFRAME= option is not specified and the GSTYLE system option is not in effect, then the axis area is not filled. If the GSTYLE system option is in effect, then the default color depends on the current ODS style template; see the section "ODS Style Templates" on page 565 for more information. The CFRAME= option is ignored if the NOFRAME option is specified.

CHARTWIDTH=*p*

CHARTPCT=*p*

specifies the width of the axis area as a percentage of the total Gantt chart width in the chart that would be produced if you had a page large enough to contain the entire chart without compression. The Gantt procedure rescales the chart so the axis area with is *p*% of the virtual chart width and the text area width is (100-*p*)% of the virtual chart width.

This option gives you the capability to generate Gantt charts that are consistent in their appearance. In the event that the chart fits on a single page, it is possible to get a smaller chart than had the CHARTWIDTH= option not been specified. You can use the FILL option in this case if you wish to use the entire page.

CHCON=*color*

specifies the color to use for drawing the horizontal connecting lines. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

CMILE=*color*

specifies the color to use for drawing the milestone symbol on the chart. If the CMILE= option is not specified, the default color of the milestones follows the rules for coloring the bars of the relevant schedule. For example, the milestone depicting a critical activity is drawn with the color of the fill pattern used for critical activities. For an activity with slack, the early start and late start milestone are drawn with the color of the fill pattern used for the duration and the slack time of a noncritical activity, respectively. You can also control the color at the activity level by using a PATTERN variable.

COMPRESS

specifies that the Gantt chart be drawn on the number of output pages determined by the HPAGES= and VPAGES= options. If the HPAGES= option is not specified, the procedure assumes a default of HPAGES=1. If the VPAGES= option is not specified, the procedure assumes a default of VPAGES=1. The COMPRESS option does not attempt to maintain the aspect ratio of the Gantt chart. To maintain the aspect ratio of the Gantt chart, use the PCOMPRESS option instead.

CPREC=*color*

specifies the color to use for drawing the precedence connections. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

CREF=*color*

specifies the color to use for drawing vertical reference lines on the chart. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

CTEXT=*color*

CT=*color*

specifies the color to use for displaying text that appears on the chart, including variable names or labels, tickmark values, values of ID variables, and so on. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

CTEXTCOLS=*name*

CTEXTCOLS=*(namelist)*

CPATTEXT=*name*

CPATTEXT=*(namelist)*

CACTTEXT=*name*

CACTTEXT=*(namelist)*

names the columns of activity text to be displayed using the color of the PATTERN variable when one exists or from the fill pattern from a particular schedule bar.

A missing value for a PATTERN variable results in the default text color being used. The default text color is the value of the CTEXT= option.

In the absence of a PATTERN variable, the activity text color is the color of the fill pattern indicating the duration of the schedule identified by the PATLEVEL= option. If PATLEVEL=EARLY or PATLEVEL=LATE, the color depends on the status of the activity. Colors for critical duration, supercritical duration, and normal duration are used depending on whether the activity is critical, supercritical, or noncritical, respectively. If more than one level is specified, the first in order of appearance on the Gantt chart is used, that is, in order EARLY, LATE, ACTUAL, RESOURCE, BASELINE.

Possible values for the CTEXTCOLS= option are shown in the following table.

Value	Interpretation
ZONE	ZONE variable column
JOBNUM	Job number column
ID	ID variable columns
FLAG	Status flag column
ALL	All of the above (default)

CTNOW=*color*

specifies the color to use for drawing the timenow line on the chart. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

CZONE=*color*

CZLINE=*color*

specifies the color to use for drawing the horizontal zone lines that demarcate the different zones on the chart. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

DESCRIPTION=*'string'*

DES=*'string'*

specifies a descriptive string, up to 40 characters in length, that appears in the description field of the master menu of PROC GREPLAY. If the DESCRIPTION= option is omitted, the description field contains a description assigned by PROC GANTT.

EBARHT=*h*

LBARHT=*h*

specifies that the height of the early/late schedule bar be *h* cellheights. The value of *h* is restricted to be a positive real number. The default bar height is one cellheight. This specification overrides a BARHT= specification. In the event that the early/late schedule bar corresponds to the logic bar (using the LEVEL= option), the value is ignored and the default value is used instead. Any non-working days corresponding to this schedule bar are also drawn using the same height as the schedule bar unless the HBARHT=option is specified.

EBAROFF=*d*

LBAROFF=*d*

specifies that the early/late schedule bar be offset *d* cellheights from its default position. A value of zero corresponds to the default position. The direction of increase is from top to bottom. This specification overrides a BAROFF= specification. In the event that the early/late schedule bar corresponds to the logic bar (specified using the LEVEL= option), the value is ignored and the default value is used instead. Any non-working days corresponding to this schedule bar are drawn using the offset of the schedule bar unless the HBAROFF= option is specified.

FONT=*font*

specifies the font to use for displaying job numbers, ID variables, legend, labels on the time axis, and so forth. The default font depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

FMILE=*font*

specifies the font to use for drawing the milestone symbol on the chart. To select a symbol from the special symbol table, set FMILE=NONE or leave it unspecified. If the FMILE= option is specified without a corresponding VMILE= option, the value of the FMILE= option is ignored, and the default milestone symbol, a filled diamond, is used instead. A warning is issued to the log in this instance.

See also the section "Special Fonts for Project Management and Decision Analysis" on page 547 for information on a special set of symbols that are suitable for representing milestones on a Gantt chart.

HBARHT=*h*

specifies that all non-working days be displayed with a bar which is *h* cellheights high. The default behavior is to use the same height as that of the schedule bar.

HBAROFF=*d*

specifies that the bars which represent non-working days be offset *d* cellheights from their default positions. The default behavior is to use the same offset as that of the schedule bar.

HEAD=*variable*

HEADNODE=*variable*

specifies the variable (either character or numeric) in the Schedule data set that contains the name of the node that represents the finish of the activity. This option is required when the precedence information is specified using the AOA format.

HEIGHT=*h*

specifies that the height for all text in PROC GANTT, excluding TITLE and FOOTNOTE statements, be *h* times the value of HTEXT=, the default text height specified in the GOPTIONS statement of SAS/GRAPH. The value of *h* is a positive real number; the default value is 1.0.

To illustrate, suppose you have the specification HEIGHT=0.6 in the CHART statement and the following GOPTIONS statement:

```
GOPTIONS htext = 2 in;
```

Then the height for all text in PROC GANTT is 0.6 x 2 in = 1.2 in.

If the value of HTEXT= is not specified in a GOPTIONS statement, then the text height is determined by the font size attribute of the GraphDataText element of the current ODS style template. See the section "ODS Style Templates" on page 565 for more information about ODS styles.

For each activity, all text corresponding to the JOB, FLAG, and ID variables is displayed at a depth of *d* cells from the top of the first bar corresponding to the activity, where *d* is the value of the HTOFF= option. The default value of *d* is 1.0. Furthermore, the text strings do not overwrite one another and *skip*, the value of the SKIP= option, is not increased to accommodate a large text height. Subject to the preceding restrictions, PROC GANTT calculates the maximum allowable value for text height as the height occupied by (*skip* + *the number of different schedule bars drawn per activity*) blank lines. Specifically, this is the height between like bars corresponding to consecutive activities. If the specified text height exceeds this value, the height is truncated to the maximum allowable value and a warning is issued to the log. This option enables you to enlarge the text to at least the height occupied by all of the schedule bars, making it easier to read. This is especially useful when the value of the VPOS= option is very large, and several schedule bars are plotted for each activity. It also provides easier identification of the activity corresponding to a given schedule bar.

HMILE=*height*

specifies the height in cells of the milestone symbol. The height is a positive real number; the default value is 1.0.

HPAGES=*h*

specifies that the Gantt chart is to be produced using *h* horizontal pages. This, however, may not be possible due to intrinsic constraints on the output. For example, the GANTT procedure requires that every horizontal page represent at least one activity. Thus, the number of horizontal pages can

never exceed the number of activities in the project. Subject to such inherent constraints, the GANTT procedure attempts to use the specified value for the HPAGES= option; if this fails, it uses *h* as an upper bound. The exact number of horizontal pages used by the Gantt chart is given in the _ORGANTT macro variable. See the section "Macro Variable _ORGANTT" on page 563 for further details.

The appearance of the chart with respect to the HPAGES= option is also influenced by the presence of other related procedure options. The HPAGES= option performs the task of determining the number of vertical pages in the absence of the VPAGES= option. If the COMPRESS or PCOMPRESS option is specified in this scenario, the chart uses one vertical page; if neither option is specified, the number of vertical pages is computed to display as much of the chart as possible in a proportional manner.

HTOFF=*d*

specifies that the line upon which all activity text rests, also referred to as the *font baseline*, is positioned at a depth of *d* cells below the top of the first bar. The default value of *d* is 1.0. The value of the HTOFF= option can be any nonnegative real number less than the (*skip* + *the number of different schedule bars per activity* - 1). A value of 0 positions text on the line corresponding to the top of the first bar. Assigning the maximum value corresponds to positioning text directly above the bar reserved for CHART variables of the next activity on the page. If a value larger than the maximum is specified, PROC GANTT truncates this value to the maximum and issues a warning to the log. Furthermore, if the HEIGHT= and HTOFF= values cause activity text to overwrite the text headings, PROC GANTT reduces the HTOFF= value accordingly and issues a warning to the log.

LABVAR=*variable*

specifies the variable that links observations in the Label data set (label definitions) to observations in the Schedule data set (activities). This variable must exist in both the Schedule data set and the Label data set and be identical in type and length. The variable can be either numeric or character in type. The linking can be a 1-1, 1-many, many-1, or many-many relationship. The linking can be used to extract positional information as well as the text string information from the Schedule data set for an observation in the Label data set when such information cannot be retrieved from the relevant variables in the Label data set.

If the _Y variable does not exist or its value is missing, the vertical coordinate for a label's placement position is determined from the activities that are linked to it and their relative positions on the activity axis of the Gantt chart. A value of -1 for _Y implies linking of the label to every activity (assuming data values are used). This is equivalent to specifying the LABVAR= option in the CHART statement and linking every activity to the label. Note that any Label data set observation with dual linkage definitions is ignored. That is, an observation with _Y equal to -1 and with a nonmissing value for the LABVAR= variable is ignored.

The following rules apply to label definitions in the Label data set that are linked to activities in the Schedule data set:

- If the _X variable does not exist in the Label data set or its value is missing, the horizontal coordinate is extracted from the Schedule data set using the _XVAR variable.
- If the _LABEL variable does not exist in the Label data set or its value is missing, the text string is determined from the Schedule data set using the _LVAR variable.

LABRULE=*rule*

LABFMT=*rule*

specifies the rule to use for laying out labels that are defined in the Label data set. Valid values for *rule* are PAGECLIP and FRAMCLIP. PAGECLIP displays a label at the specified location and clips any part of the label that runs off the page. A value of FRAMCLIP differs from PAGECLIP in that it clips all labels with data value coordinates that run off the frame of the Gantt chart. The default value for *rule* is PAGECLIP.

LABSPLIT=*'character'*

LABELSPLIT=*'character'*

splits labels that are defined in the Label data set wherever the split character appears. By default, if there are embedded blanks, the GANTT procedure attempts to split strings at suitable blanks so that the resulting lines are equal in length. To suppress the default splitting when using strings embedded with blanks, specify a dummy character not used in the labeling.

LAG=*variable*

LAG=(*variables***)**

specifies the variables identifying the lag types of the precedence relationships between an activity and its successors. Each SUCCESSOR variable is matched with the corresponding LAG variable; that is, for a given observation, the *i*th LAG variable defines the relationship between the activities specified by the ACTIVITY variable and the *i*th SUCCESSOR variable. The LAG variables must be character type and their values are expected to be specified as one of FS, SS, SF, FF, which denote 'Finish-to-Start', 'Start-to-Start', 'Start-to-Finish', 'Finish-to-Finish', respectively. You can also use the *keyword_duration_calendar* specification used by PROC CPM although PROC GANTT uses only the *keyword* information and ignores the lag *duration* and the lag *calendar*. If no LAG variables exist or if an unrecognized value is specified for a LAG variable, PROC GANTT interprets the lag as a 'Finish-to-Start' type. If the PRECDATA= option is specified, the LAG variables are assumed to exist in the Precedence data set; otherwise, they are assumed to exist in the Schedule data set.

LEFT

LJUST

displays the Gantt chart left-justified with the left edge of the page. This option has priority over the RIGHT or RJUST option. Note that when displaying a Gantt chart in graphics mode, the chart is centered in both horizontal and vertical directions in the space available after accounting for titles, footnotes, and notes. The chart justification feature enables you to justify the chart in the horizontal and vertical directions with the page boundaries.

LEVEL=*number*

specifies the schedule bar to use for drawing the precedence connections. The default value of LEVEL= is 1, which corresponds to the topmost bar.

LHCON=*linetype*

specifies the line style (1 – 46) to be used for drawing the horizontal connecting line produced by the HCONNECT option described earlier in this section. Possible values for *linetype* are

1	solid line (the default value when LHCON= is omitted)
2 – 46	various dashed lines. See Figure 8.5.

For the corresponding line-printer option, see the HCONCHAR= option described earlier in this section.

LPREC=*linetype*

specifies the line style (1 – 46) to use for drawing the precedence connections. The default line style is 1, a solid line. See Figure 8.5 for examples of the various line styles available.

LREF=*linetype*

specifies the line style (1 – 46) to use for drawing the reference lines. The default line style is 1, a solid line. See Figure 8.5 for examples of the various line styles available. For the corresponding line-printer option, see the REFCHAR= option described earlier.

LTNOW=*linetype*

specifies the line style (1 – 46) to use for drawing the timenow line. The default line style is 1, a solid line. See Figure 8.5 for examples of the various line styles available.

LWIDTH=*linewidth*

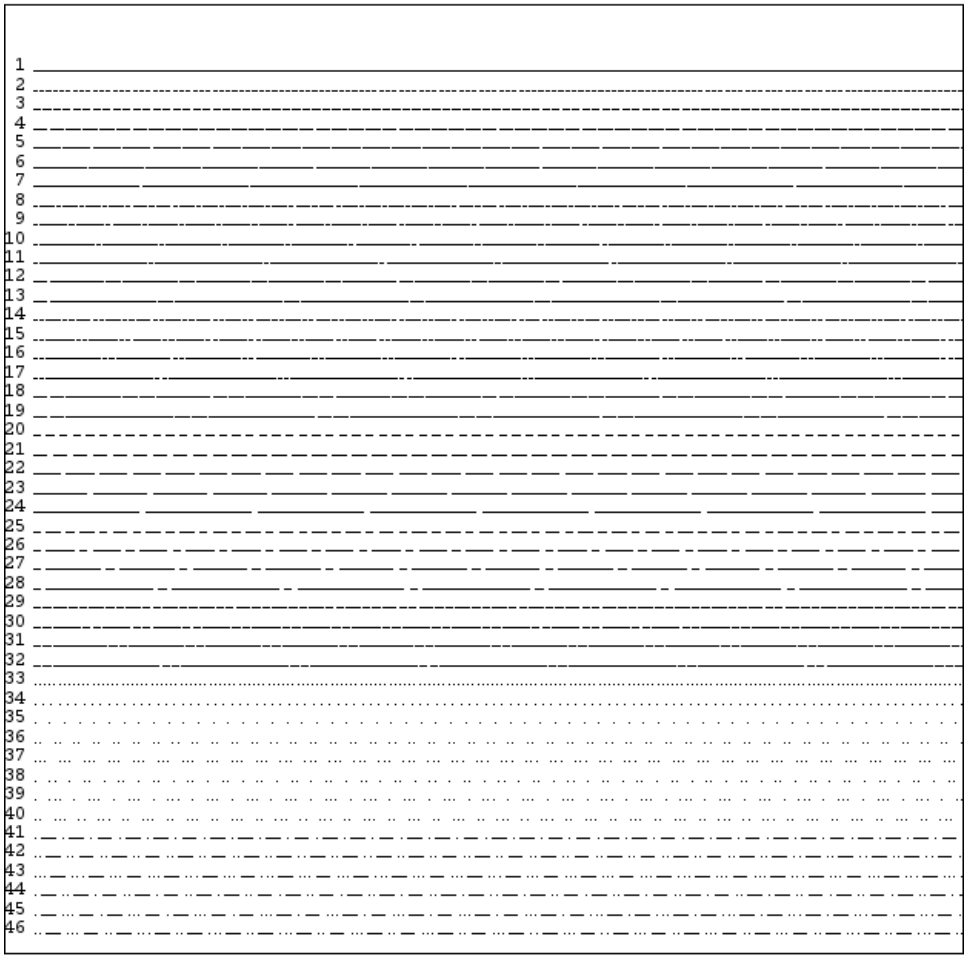
specifies the line width to be used for drawing lines, other than the timenow line and precedence connection lines, used in the Gantt chart. The default width is 1.

LZONE=*linetype*

LZLINE=*linetype*

specifies the line style (1 – 46) to use for drawing the horizontal zone lines which demarcate the different zones on the chart. The default line style is 1, a solid line.

Figure 8.5 Valid Line Styles



MAXDISLV=*columns*

specifies the maximum allowable distance, in number of columns, that a local vertical can be positioned from its minimum offset to avoid overlap with a global vertical. The value of the MAXDISLV= option must be greater than or equal to 0.1; the default value is 1. For the definitions of global and local verticals, see the section "Specifying the Logic Options" on page 549.

MININTGV=*columns*

specifies the minimum inter-distance, in number of columns, of any two global verticals to prevent overlap. The value of the MININTGV= option must be greater than or equal to 0.1; the default value is 0.75.

MINOFFGV=*columns*

specifies the minimum offset, in number of columns, of a global vertical from the end of the bar with which it is associated. The value of the MINOFFGV= option must be greater than or equal to 0.1; the default value is 1.

MINOFFLV=*columns*
specifies the minimum offset, in number of columns, of a local vertical from the end of the bar with which it is associated. The value of the MINOFFLV= option must be greater than or equal to 0.1; the default value is 1.

NAME=*'string'*
where *'string'* specifies a descriptive string, up to eight characters long, that appears in the name field of the master menu of the GREPLAY procedure. If you omit the NAME= option, the name field of the master menu contains the name of the procedure.

NJOBS=*number*

NACTS=*number*
specifies the number of jobs that should be displayed on a single page. This option overrides the VPAGES= option.

NOARROWHEAD

NOARRHD
suppresses the arrowhead when drawing the precedence connections.

NOEXTRANGE

NOXTRNG
suppresses the automatic extension of the chart axis range when drawing a Logic Gantt chart and neither the MINDATE= nor MAXDATE= option is specified.

NOFRAME

NOFR
suppresses drawing the vertical boundaries to the left and right of the Gantt chart; only the top axis and a parallel line at the bottom are drawn. If this option is not specified, the entire chart area is framed.

NOPAGENUM
suppresses numbering the pages of a multipage Gantt chart. This is the default behavior. To number the pages of a multipage Gantt chart on the upper right-hand corner of each page, use the PAGENUM option.

NOPATBAR
suppresses the use of the PATTERN variable for filling the schedule bars. The default fill patterns are used instead. Typically, this option is used when you want to color the activity text using the CTEXTCOLS= option but leave the bars unaffected by the PATTERN variable.

NOTMTIME
suppresses the display of the time portion of the axis tickmark label when the value of MININTERVAL is DTDAY. When MININTERVAL=DTDAY, the time axis tickmarks are labeled with three lines, the first indicating the month, the second indicating the day, and the third indicating the time. This option effectively lowers the first two lines by a line and drops the third line altogether.

NOZONECOL
suppresses displaying the ZONE variable column that is automatically done in the presence of a zone variable.

NTICKS=*number*

NINCRS=*number*

specifies the number of tickmarks that should be displayed on the first horizontal page of the Gantt chart. The number of tickmarks on the remaining horizontal pages is determined by the page width and the columns of text that are to be displayed (ZONE, IDs, flag, and so forth). The page width is determined to be the minimum width necessary to fit the first page. If the IDPAGES option is specified, the number of tickmarks is the same as that specified by the NTICKS= option. This option overrides the HPAGES= option.

ONEZONEVAL

displays the value of the ZONE variable in the ZONE variable column only for activities that begin a new zone. A blank string is displayed for all other activities.

PAGENUM

numbers the pages of the Gantt chart on the top right-hand corner of the page if the chart exceeds one page. The numbering scheme is from left to right, top to bottom.

PATLEVEL=*name*

PATLEVEL=*(namelist)*

specifies the different schedule bar levels to fill using the PATTERN variable. By default, all of the schedule bar levels for an activity are filled using the pattern defined by the PATTERN variable. Note that holiday and non-working days are not filled with this pattern.

Possible values for the PATLEVEL= option and their actions are shown in the following table.

Value	Interpretation
EARLY	Early/Late schedule durations
LATE	Early/Late schedule durations
ACTUAL	Actual schedule durations
RESOURCE	Resource schedule duration
BASELINE	Baseline schedule duration
ALL	All of the above (default)

In the absence of a PATTERN variable, this option defines the schedule type that determines the color for the activity text columns (ZONE variable, ID variable, Job number, Critical Flag), which are identified with the CTEXTCOLS= option. In this case, only one schedule type is used, namely the first one appearing in the order EARLY, LATE, ACTUAL, RESOURCE, BASELINE.

PATTERN=*variable*

PATVAR=*variable*

specifies an integer variable in the Schedule data set that identifies the pattern for filling the schedule bars and coloring the milestones. The default PATTERN variable name is _PATTERN. If the value of the PATTERN variable is missing for a particular activity, or if there is no PATTERN variable, the different schedule bars and milestones for the activity are drawn using the corresponding default patterns given in Table 8.7. The procedure uses the defined or default pattern to fill all the schedule bars and color all the milestones associated with the activity, except for holidays and non-working days. Use the PATLEVEL= option to restrict the application of the defined pattern to selected schedule bar levels.

When plotting split activities, you have the additional capability of overriding the defined pattern at the segment level by specifying a value for the PATTERN variable for the schedule data set observation representing the segment. Setting it to missing results in inheriting the PATTERN variable value from the observation for the same activity with a missing SEGMT_NO. For example, setting PATTERN=SEGMT_NO in the CHART statement when using split activities results in each segment using a different pattern.

Note that, if the value of the PATTERN variable is *n* for a particular activity, the GANTT procedure uses the specifications in the *n*th generated PATTERN definition, not the specifications in the PATTERN *n* statement.

The chart legend and summary, when displayed, indicate the default patterns that identify the different schedule types represented on the Gantt chart as listed in Table 8.7. Since the PATTERN variable overrides these values at the activity level, you must be careful in interpreting the summary and legend when using a PATTERN variable, especially if any of the specified pattern definitions overlap with one of the default patterns.

PCOMPRESS

specifies that every output page of the Gantt chart is to be produced maintaining the original aspect ratio of the Gantt chart. The number of output pages is determined by the HPAGES= and VPAGES= options. In the absence of the HPAGES= and VPAGES= options, the PCOMPRESS option displays the Gantt chart on a single page.

RBARHT=*h*

SBARHT=*h*

specifies that the height of the resource-constrained schedule bar be *h* cellheights. The value of *h* is restricted to be a positive real number. The default bar height is one cellheight. This specification overrides a BARHT= specification. In the event that the resource-constrained schedule bar corresponds to the logic bar (using the LEVEL= option), the value is ignored and the default value is used instead. Any non-working days corresponding to this schedule bar are also drawn using the same height as the schedule bar unless the HBARHT= option is specified.

RBAROFF=*d*

SBAROFF=*d*

specifies that the resource-constrained schedule bar be offset *d* cellheights from its default position. A value of zero corresponds to the default position. The direction of increase is from top to bottom. This specification overrides a BAROFF= specification. In the event that the resource-constrained schedule bar corresponds to the logic bar (specified using the LEVEL= option), the value is ignored and the default value is used instead. Any non-working days corresponding to this schedule bar are drawn using the offset of the schedule bar unless the HBAROFF= option is specified.

RIGHT

RJUST

displays the Gantt chart right-justified with the right edge of the page. This option is ignored in the presence of the LEFT or LJUST option.

SHOWPREC

causes PROC GANTT to terminate in the event that a valid AOA or AON specification exists, and an error occurs either in the logic system (memory allocation, data structure creation, and so on) or simply due to bad data (missing values for the ACTIVITY, TAIL, HEAD variables, and so on). The default behavior is to attempt drawing the chart without the precedence connections.

SUCCESSOR=*variable*

SUCC=*variable*

SUCCESSOR=(*variables***)**

SUCC=(*variables***)**

specifies the variables identifying the names of the immediate successors of the node specified by the ACTIVITY variable. This option is required when the precedence information is specified in the AON format. These variables must have the same type as the ACTIVITY variable. If the PRECDATA= option has been specified, the SUCCESSOR variables are assumed to exist in the Precedence data set; otherwise, they are assumed to exist in the Schedule data set.

TAIL=*variable*

TAILNODE=*variable*

specifies the variable in the Schedule data set that contains the name of the node that represents the start of the activity. This option is required when the precedence information is specified using the AOA format. The variable can be either numeric or character in type.

TOP

TJUST

positions the top of the Gantt chart at the top of the page, just below the titles. This option has priority over the BOTTOM or BJUST option.

VMILE=*value*

specifies a plot symbol from the font specified in the FMILE= option to be used as the milestone symbol on the chart. If the FMILE= option is set to NONE or is not specified, then the milestone symbol is the symbol specified by the VMILE= option in the special symbol table shown in Figure 8.7. The default milestone symbol is a filled diamond.

VPAGES=*v*

Specifies that the Gantt chart is to be produced using *v* vertical pages. This, however, may not be possible due to intrinsic constraints on the output. For example, the GANTT procedure requires that every vertical page represent at least one tickmark. Thus, the number of vertical pages can never exceed the number of tickmarks in the axis. Subject to such inherent constraints, the GANTT procedure attempts to use the specified value for the VPAGES= option; if this fails, it uses *v* as an upper bound. The exact number of vertical pages used by the Gantt chart is provided in the _ORGANTT macro variable. See the section "Macro Variable _ORGANTT" on page 563 for further details.

The appearance of the chart with respect to the VPAGES= option is also influenced by the presence of other related procedure options. The VPAGES= option performs the task of determining the number of horizontal pages in the absence of the HPAGES= option. If the COMPRESS or PCOMPRESS option is specified in this scenario, the chart uses one horizontal page. If neither the COMPRESS nor PCOMPRESS option is specified, the number of horizontal pages is computed in order to display as much of the chart as possible in a proportional manner.

WEB=*variable*

HTML=*variable*

specifies the character variable in the schedule data set that identifies an HTML page for each activity. The procedure generates an HTML image map using this information for all the schedule bars, milestones, and ID variables corresponding to an activity.

WPREC=*linewidth*
specifies the line width to use for drawing the precedence connections. The default width is 1.

WTNOW=*linewidth*
specifies the line width to use for drawing the timenow line. The default width is 4.

WZONE=*linewidth*

WZLINE=*linewidth*
specifies the line width to use for drawing the horizontal zone lines which demarcate the different zones on the chart. The default linewidth is 1.

ZONE=*variable*

ZONEVAR=*variable*
names the variable in the Schedule data set that is used to separate the Gantt chart into zones. This option enables you to produce a zoned Gantt chart. The GANTT procedure does not sort the Schedule data set and processes the data in the order it appears in the Schedule data set. A change in the value of the zone variable establishes a new zone. By default, the GANTT procedure displays a ZONE variable column before the ID variable columns. You can suppress this column using the NOZONECOL option. The GANTT procedure also draws a horizontal line demarcating zones. By default, the line spans the entire chart in the horizontal direction, both inside and outside the axis area. You can control the span of this line using the ZONESPAN= option. You can also adjust the vertical offset of the line from its default position by using the ZONEOFFSET= option. In addition, you can also control the graphical attributes associated with this line such as color, style, and width using the CZONE=, LZONE=, and WZONE= options, respectively.

ZONEOFF=*d*

ZONEOFFSET=*d*
specifies the offset in cellheights of the zone line from its default position of 0.5 cell height above the top of the first schedule bar for the first activity in the zone. The default value of *d* is 0. The direction of increase is from top to bottom.

ZONESPAN=*name*

ZONELINE=*name*
specifies the span of the horizontal zone line that is drawn at the beginning of each new zone. Valid values for 'name' are LEFT, RIGHT, ALL, and NONE. The value of LEFT draws a line that spans the width of the columns of text that appear on the left hand side of the Gantt chart. The value of RIGHT draws a line that spans the width of the axis area which appears on the right-hand side of the chart. The value of ALL draws a line spanning both the preceding regions while the value of NONE suppresses the line altogether. The default value is ALL.

ID Statement

ID *variables* **;**

The ID statement specifies the variables to be displayed that further identify each activity. If two or more consecutive observations have the same combination of values for all the ID variables, only the first of these observations is plotted unless the DUPOK option is specified in the CHART statement.

By default, if the ID variables do not all fit on one page, they are all omitted and a message explaining the omission is printed to the log. You can override this behavior and display the maximum number of consecutive ID variables that can fit on a page by specifying the MAXIDS option in the CHART statement.

If the time axis of a Gantt chart spans more than one page, the ID variables are displayed only on the first page of each activity. You can display the ID variables on every page by specifying the IDPAGES option in the CHART statement.

Details: GANTT Procedure

Schedule Data Set

Often, the Schedule data set input to PROC GANTT is the output data set (the OUT= data set) produced by PROC CPM, sometimes with additional variables. Typically, this data set contains

- the start and finish times for the early and late schedules (E_START, E_FINISH, L_START, and L_FINISH variables)
- the actual start and finish times (A_START and A_FINISH variables) of activities that have been completed or are in progress for projects that are in progress or completed
- the resource-constrained start and finish times of the activities (S_START and S_FINISH variables) for projects that have been scheduled subject to resource constraints
- the baseline start and finish times (B_START and B_FINISH variables) of activities when monitoring and comparing the progress of a project against a target schedule

When such a data set is input as the Schedule data set to PROC GANTT, the procedure draws a Gantt chart showing five different schedules for each activity: the predicted early/late schedules using E_START, E_FINISH, L_START, and L_FINISH on the first line for the activity, the actual schedule using A_START and A_FINISH on the second line, the resource-constrained schedule using S_START and S_FINISH on the third line, and the baseline schedule using B_START and B_FINISH on the fourth line.

The SEGMT_NO Variable

Normally, each observation of the Schedule data set causes one set of bars to be plotted corresponding to the activity in that observation. If activity splitting has occurred during resource-constrained scheduling, the Schedule data set produced by PROC CPM contains more than one observation for each activity. It also contains a variable named SEGMT_NO. For activities that are not split, this variable has a missing value. For split activities, the number of observations in the Schedule data set is equal to (1 + *the number of disjoint segments that the activity is split into*). The first observation corresponding to such an activity has SEGMT_NO equal to missing, and the S_START and S_FINISH variables are equal to the start and finish times, respectively, of the entire activity. Following this observation, there are as many observations as the number of disjoint segments in the activity. All values for these segments are the same as the first observation for this activity except SEGMT_NO, S_START, S_FINISH, and the duration. SEGMT_NO is the index of the segment, S_START and S_FINISH are the resource-constrained start and finish times for this segment,

and duration is the duration of this segment. See the section "Displayed Output" on page 562 for details on how PROC GANTT treats the observations in this case.

NOTE: For a given observation in the Schedule data set, the finish times (E_FINISH, L_FINISH, A_FINISH, S_FINISH, and B_FINISH) denote the last *day* of work when the variables are formatted as SAS *date* values; if they are formatted as SAS *time* or *datetime* values, they denote the last *second* of work. For instance, if an activity has E_START='2JUN04' and E_FINISH='4JUN04', then the earliest start time for the activity is the beginning of June 2, 2004, and the earliest finish time is the end of June 4, 2004. Thus, PROC GANTT assumes that the early, late, actual, resource-constrained, or baseline finish time of an activity is at the end of the time interval specified for the respective variable. The exceptions to this type of default behavior occur when either the DURATION= option or the PADDING= option is in effect. See the section "Specifying the PADDING= Option" on page 532 for further details.

All start and finish times, and additional variables specified in the CHART statement must be numeric and have the same formats. The ID and BY variables can be either numeric or character. Although the data set does not have to be sorted, the output may be more meaningful if the data are in order of increasing early start time. Further, if the data set contains segments of split activities, the data should also be sorted by SEGMT_NO for each activity.

A family of options, available only in graphics mode, enables you to display the precedence relationships between activities on the Gantt chart. The precedence relationships are established by specifying a set of variables in the CHART statement; this can be done in one of two ways. These variables must lie in the Schedule data set and, optionally, in the Precedence data set defined by the PRECDATA= option in the PROC GANTT statement. See the section "Specifying the Logic Options" on page 549 for more details on producing a Logic Gantt chart.

Also available in graphics mode is an automatic text annotation facility that enables you to annotate labels on the Gantt chart independently of the SAS/GRAPH Annotate facility. A useful property of this facility is the ability to link label coordinates and text strings to variables in the Schedule data set. You can create links of two types. An implicit link automatically links an observation in the Label data set to every observation in the Schedule data set. An explicit link uses a variable that must exist on both data sets and be identical in type and length. For more information on the linking variable in the automatic text annotation facility, see the section "Automatic Text Annotation" on page 556.

Missing Values in Input Data Sets

Table 8.2 summarizes the treatment of missing values for variables in the input data sets used by PROC GANTT.

Table 8.2 Treatment of Missing Values in PROC GANTT

Data Set	Variable	Action
CALEDATA	CALID	Default calendar (0 or "DEFAULT")
	SUN, . . . ,_SAT_	Corresponding shift for default calendar
	D_LENGTH	DAYLENGTH, if available; else, 8:00, if INTERVAL=WORKDAY or DTWRKDAY; 24:00, otherwise
DATA	A_FINISH	Value ignored
	A_START	Value ignored

Table 8.2 (continued)

Data Set	Variable	Action
	ACTIVITY	Input error: logic options are ignored
	B_FINISH	Value ignored
	B_START	Value ignored
	CALID	Default calendar (0 or "DEFAULT")
	CHART	Value ignored
	DUR	Nonzero
	E_FINISH	Value ignored
	E_START	Value ignored
	HEADNODE	Input error: logic options are ignored
	ID	Missing
	L_FINISH	Value ignored
	L_START	Value ignored
	LAG	FS
	S_FINISH	Value ignored
	S_START	Value ignored
	SEGMT_NO	See the section "Displayed Output" on page 562
	SUCCESSOR	Value ignored
	TAILNODE	Input error: logic options are ignored
	ZONE	Zone value
HOLIDATA	CALID	Holiday applies to all calendars defined
	HOLIDAY	Observation ignored
	HOLIDUR	Ignored, if HOLIFIN is not missing; else, 1.0
	HOLIFIN	Ignored, if HOLIDUR is not missing; else, HOLIDAY + (1 unit of INTERVAL)
LABDATA	_ALABEL	0
	_CLABEL	CTEXT=
	_FLABEL	FONT=
	_HLABEL	1
	_JLABEL	L
	_LABEL	Use _LVAR
	_LVAR	Value ignored
	_PAGEBRK	0
	_RLABEL	0
	_X	Use _XVAR
	_XOFFSET	0
	_XVAR	Value ignored
	_XSYS	DATA
	_Y	Use LABVAR=
	_YOFFSET	0
	_YSYS	DATA
	LABVAR	Value ignored
PRECDATA	ACTIVITY	Input error: logic options are ignored
	LAG	FS
	SUCCESSOR	value ignored
WORKDATA	any numeric variable	00:00, if first observation; 24:00, otherwise

Specifying the PADDING= Option

As explained in the section "Schedule Data Set" on page 529, the finish times in the Schedule data set denote the final time unit of an activity's duration; that is, the activity finishes at the end of the day/second specified as the finish time. A plot of the activity's duration should continue through the end of the final time unit. Thus, if the value of the E_FINISH variable is '4JUN04', the early finish time for the activity is plotted at the end of June 4, 2004 (or the beginning of June 5, 2004).

In other words, the finish times are *padded* by a day (second) if the finish time variables are formatted as SAS date (SAS time or datetime) values. This treatment is consistent with the meaning of the variables as output by PROC CPM. Default values of PADDING corresponding to different format types are shown in Table 8.3.

The PADDING= option is provided to override the default padding explained above. Valid values of this option are NONE, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QTR, YEAR, DTSECOND, DTMINUTE, DTHOUR, DTWEEK, DTMONTH, DTQTR, and DTYEAR. Use the value NONE if you do not want the finish times to be adjusted.

Table 8.3 Default Values of the PADDING= Option Corresponding to Format Type

Format	PADDING
SAS time value	SECOND
SAS date value	DAY
SAS datetime value	DTSECOND
Other	NONE

It is recommended that when plotting zero duration activities, you include a variable in the Schedule data set that has value zero if and only if the activity has zero duration. Defining this variable to the GANTT procedure using the DURATION= (or DUR=) option in the CHART statement ensures that a zero duration activity is represented on the chart by a Milestone. If this is not done, an activity with zero duration is shown on the chart as having a positive duration since finish times are padded to show the end of the last time unit.

Page Format

The GANTT procedure divides the observations (activities) into a number of subgroups of approximately equal numbers. The size of each group is determined by the PAGESIZE system option. Similarly, the time axis is divided into a number of approximately equal divisions depending on the LINESIZE system option.

If the FILL option is specified, however, each page is filled as completely as possible before plotting on a new page. If both axes are split, the pages are ordered with the chart for each group of activities being plotted completely (the time axis occupying several consecutive pages, if needed) before proceeding to the next group.

If a BY statement is used, each BY group is formatted separately.

Two options that control the format of the chart are the MININTERVAL= and SCALE= options. The value for the MININTERVAL= option, denoted by *mininterval*, is the smallest time interval unit to be identified on the chart. The value for the SCALE= option, denoted by *scale*, is the number of columns to be used to denote one unit of *mininterval*. For example, if MININTERVAL=MONTH and SCALE=10, the chart is formatted so that 10 columns denote the period of one month. The first of these 10 columns denotes the start of the month and the last denotes the end, with each column representing approximately three days. Further, the INCREMENT= option can be used to control the labeling. In this example, if INCREMENT=2, then the time axis would have labels for alternate months.

Specifying the MININTERVAL= Option

The value specified for the MININTERVAL= option is the smallest time interval unit to be identified on the chart. If the time values being plotted are SAS *date* values, the valid values for *mininterval* are DAY, WEEK, MONTH, QTR, or YEAR. If the values are SAS *datetime* values, valid values for *mininterval* are DTSECOND, DTMINUTE, DTHOUR, DTDAY, DTWEEK, DTMONTH, DTQTR, or DTYEAR. If they are SAS *time* values, then valid values are SECOND, MINUTE, or HOUR.

NOTE: If the times being plotted are SAS *datetime* values and *mininterval* is either DTSECOND, DTMINUTE, or DTHOUR, the output generated could run into several thousands of pages. Therefore, be careful when choosing a value for *mininterval*.

Table 8.4 shows the default values of *mininterval* corresponding to different formats of the times being plotted on the chart.

Table 8.4 Default Values of the MININTERVAL= Option

Format	MININTERVAL= Value
DATEw.	DAY
DATETIMEw.d	DTDAY
HHMMw.d	HOUR
MONYYw.	MONTH
TIMEw.d	HOUR
YYMMDDw.	MONTH
YYQw.	MONTH

Labeling on the Time Axis

If the variables being plotted in the chart are unformatted numeric values, the time axis is labeled by the corresponding numbers in increments specified by the INCREMENT= option. However, if the variables have *date*, *datetime*, or *time* formats, then the time axis is labeled with two or three lines. Each line is determined by the value of *mininterval*, which in turn is determined by the format of the plotted times (see Table 8.4). Table 8.5 illustrates the format of the label corresponding to different values of *mininterval*.

Table 8.5 Label Format Corresponding to MININTERVAL= Value

MININTERVAL= Value	First Line	Second Line	Third Line
DAY, WEEK, DTWEEK	Month	Day	
MONTH, QTR, YEAR, DTMONTH, DTQTR, DTYEAR	Year	Month	
DTSECOND, DTMINUTE, DTHOUR, DTDAY	Month	Day	Time
SECOND, MINUTE, HOUR	Time		

Multiple Calendars and Holidays

Work pertaining to a given activity is assumed to be done according to a particular *calendar*. A calendar is defined in terms of a *work pattern* for each day and a *workweek structure* for each week. In addition, each calendar may include holidays during the year. See the "Multiple Calendars" section in the PROC CPM chapter for details on how calendars are defined and how all the options work together. In this chapter, a less detailed description is provided. PROC GANTT uses the same structure as PROC CPM for defining calendars, but the options for using them differ in minor ways. The following are the differences in syntax:

- The CALID variable is specified as an option in the CHART statement and is not a separate statement as in PROC CPM.
- The HOLIDAY variable is specified as an option in the CHART statement and is not a separate statement as in PROC CPM.
- The HOLIDUR and HOLIFIN variables are specified as options in the CHART statement and not in a separate HOLIDAY statement.
- The INTERVAL= option is specified in the CHART statement and not in the procedure statement as in PROC CPM.

The WORKDATA (or Workday) data set specifies distinct shift patterns during a day. The CALEDATA (or Calendar) data set specifies a typical workweek for all the calendars in the project; for each day of a typical week, it specifies the shift pattern that is followed. The HOLIDATA (or Holiday) data set specifies a list of holidays and the calendars that they refer to; holidays are defined either by specifying the start of the holiday and its duration in *interval* units, where the INTERVAL= option has been specified as *interval*, or by specifying the start and end of the holiday period. If both the HOLIDUR and the HOLIFIN variables have missing values in a given observation, the holiday is assumed to start at the date and time specified for the HOLIDAY variable and last one unit of *interval*. If a given observation has valid values for both the HOLIDUR and the HOLIFIN variables, only the HOLIFIN variable is used so that the holiday is assumed to start and end as specified by the HOLIDAY and HOLIFIN variables, respectively. The Schedule data set (the DATA= data set) , specifies the calendar that is used by each activity in the project through the CALID variable (or a default variable _CAL_). Each of the three data sets used to define calendars is described in greater detail in the "Multiple Calendars" section in the PROC CPM chapter.

Each new value for the CALID variable in either the Calendar or the Holiday data set defines a new calendar. If a calendar value appears in the Calendar data set and not in the Holiday data set, it is assumed to have the same holidays as the default calendar (the default calendar is defined in the PROC CPM chapter). If a calendar value appears in the Holiday data set and not in the Calendar data set, it is assumed to have the same work pattern structures (for each week and within each day) as the default calendar. In the Schedule data set, valid values for the CALID variable are those that are defined in either the Calendar or the Holiday data set.

All the holiday, workday and workweek information is used by PROC GANTT for display only; in particular, the weekend and shift information is used only if the MARKWKND or MARKBREAK option is in effect. The value of the INTERVAL= option, which has a greater scope in PROC CPM, is used here only to determine the end of holiday periods appropriately. Further, the Workday, Calendar, and Holiday data sets and the processing of holidays and different calendars are supported only when *interval* is DAY, WEEKDAY, WORKDAY, DTSECOND, DTMINUTE, DTHOUR, DTDAY, or DTRKDAY.

Specifying the INTERVAL= Option

The INTERVAL= option is needed only if you want holidays or breaks or both during a week or day to be indicated on the Gantt chart. The value of INTERVAL= is used to compute the start and end of holiday periods to be compatible with the way they were computed and used by PROC CPM. Further, if the MARKWKND or MARKBREAK option is in effect, the INTERVAL= option, in conjunction with the DAYSTART= and DAYLENGTH= options and the Workday, Calendar, and Holiday data sets, helps identify the breaks during a standard week or day as well as the holidays that are to be marked on the chart. Valid values of *interval* are DAY, WEEKDAY, WORKDAY, DTSECOND, DTMINUTE, DTHOUR, DTDAY, and DTWRKDAY. If *interval* is WEEKDAY, WORKDAY, or DTWRKDAY, the MARKWKND option is in effect; otherwise, breaks during a week are indicated only if MARKWKND is specified and breaks within a day are marked only if MARKBREAK is specified.

Full-Screen Version

You can invoke PROC GANTT in full-screen mode by specifying FS (or FULLSCREEN) in the PROC GANTT statement. The full-screen mode offers you a convenient way to browse the Gantt chart for the project. For large projects, where the chart could span several pages, the full-screen mode is especially convenient because you can scroll around the output using commands on the command line, pull-down menus, or function keys. You can scroll vertically to a given job on the task axis by specifying a job number or scroll horizontally to a given point in time along the time axis by specifying a date. You can optionally display the title and the legend.

The specifications for the full-screen version of PROC GANTT and the output format are the same as those for the line-printer version. The following is a list of the few minor differences:

- The FILL option is not relevant in this case because all of the activities are plotted on one logical page.
- The NOLEGEND option is not effective. The screen always displays only the body of the chart along with the ID columns. To see what the symbols mean, you can use the SHOW LEGEND command, which causes the legend to be displayed at the bottom of the chart. To delete the legend, use the DELETE LEGEND command.
- The SUMMARY option is not supported in full-screen mode.

- The SCALE= option works the same way as in the line-printer version, except for its default behavior. The default value is always 1, unlike in the line-printer case where, if the time axis fits on less than one page, the default value is chosen so that the time axis fills as much of the page as possible.

Output Format

The output format is similar to the line-printer version of PROC GANTT. When PROC GANTT is invoked with the FS option, the screen is filled with a display of the Gantt chart. The display consists of column headings at the top and ID values (if an ID statement is used to specify ID variables) at the left. The body of the chart occupies the bottom right portion of the display. The column headings can be scrolled left or right, the ID values can be scrolled up or down, and the body of the chart can scroll along both directions. The display does not include the TITLES or LEGEND.

In addition to using the symbols and join characters as described for the line-printer version of PROC GANTT, the full-screen version also uses different colors to distinguish the types of activities and their associated bars.

You can use the FIND command to locate a particular job (by job number) or a particular time along the time axis. The format of the FIND command is FIND JOB *n* or FIND TIME *t*. All the commands that are specific to PROC GANTT are described as follows.

Local Commands

Table 8.6 lists the commands that can be used in the full-screen version of PROC GANTT.

Table 8.6 Full-Screen Commands and Their Purpose

Scrolling	Controlling Display	Exiting
BACKWARD	SHOW	END
FORWARD	DELETE	CANCEL
LEFT	FIND	
RIGHT		
TOP		
BOTTOM		
VSCROLL		
HSCROLL		

BACKWARD
scrolls towards the top of the Gantt chart by the VSCROLL amount. A specification of BACKWARD MAX scrolls to the top of the chart. You can also specify the vertical scroll amount for the current command as BACKWARD PAGE | HALF | *n*. Note that during vertical scrolling, the column headings are not scrolled.

BOTTOM
scrolls to the bottom of the Gantt chart.

DELETE LEGEND | TITLE
deletes the legend or the title from the screen. A specification of DELETE LEGEND deletes the legend from the current display; DELETE TITLE deletes the current title (titles) from the current display.

END
ends the current invocation of the procedure.

FIND
scrolls to the specified position on the chart. The format of the command is FIND JOB *n* or FIND TIME *t*.

A specification of FIND JOB *n* scrolls backward or forward, as necessary, in order to position the activity with job number *n* on the screen. The specified activity is positioned at the top of the screen, unless this would result in blank space at the bottom of the screen. In this instance, the chart is scrolled down to fit as many jobs as space permits.

A specification of FIND TIME *t* scrolls left or right, as necessary, in order to position the time *t* on the time axis to appear on the screen. The specified time is positioned at the left boundary of the displayed chart area unless this would result in blank space at the right of the screen. In this instance, the chart is scrolled to the right to fit as much of the time axis as space permits.

FORWARD
scrolls towards the bottom of the Gantt chart by the VSCROLL amount. A specification of FORWARD MAX scrolls to the bottom of the chart. You can also specify the vertical scroll amount for the current command as FORWARD PAGE | HALF | *n*. Note that during vertical scrolling, the column headings are not scrolled.

HELP
displays a HELP screen listing all the full-screen commands specific to PROC GANTT.

HOME
moves the cursor to the command line.

HSCROLL
sets the amount of information that scrolls horizontally when you execute the LEFT or RIGHT command. The format is HSCROLL PAGE | HALF | *n*. The specification is assumed to be in number of columns. A specification of HSCROLL PAGE sets the scroll amount to be the number of columns in the part of the screen displaying the plot of the schedules. A specification of HSCROLL HALF is half that amount; HSCROLL *n* sets the horizontal scroll amount to *n* columns. The default setting is PAGE.

KEYS
displays current function key settings.

LEFT
scrolls towards the left boundary of the Gantt chart by the HSCROLL amount. A specification of LEFT MAX scrolls to the left boundary. You can also specify the horizontal scroll amount for the current command as LEFT PAGE | HALF | *n*. Note that during horizontal scrolling, the ID columns are not scrolled.

RIGHT
scrolls towards the right boundary of the Gantt chart by the HSCROLL amount. A specification of RIGHT MAX scrolls to the right boundary. You can also specify the horizontal scroll amount for the current command as RIGHT PAGE | HALF | *n*. Note that during horizontal scrolling, the ID columns are not scrolled.

SHOW LEGEND | TITLE
displays the legend or the title on the screen. A specification of SHOW LEGEND displays the legend in the bottom portion of the current display; SHOW TITLE displays the current title (titles) in the top portion of the current display.

TOP
scrolls to the top of the Gantt chart.

VSCROLL
sets the amount of information that scrolls vertically when you execute the BACKWARD or FORWARD command. The format is VSCROLL PAGE | HALF | *n*. The specification is assumed to be in number of rows. A specification of VSCROLL PAGE sets the scroll amount to be the number of rows in the part of the screen displaying the plot of the schedules. A specification of VSCROLL HALF is half that amount; VSCROLL *n* sets the vertical scroll amount to *n* rows. The default setting is PAGE.

Global Commands

Most of the global commands used in SAS/FSP software are also valid with PROC GANTT. Some of the commands used for printing screens are described below.

SAS/FSP software provides you with a set of printing commands that enable you to take pictures of windows and to route those pictures to a printer or a file. Whether you choose to route these items directly to a printer queue or to a print file, SAS/FSP software provides you with a means of specifying printing instructions. The following is an overview of these related commands and their functions:

FREE
releases all items in the print queue to the printer. This includes pictures taken with the SPRINTcommand as well as items sent to the print queue with the SEND command. All items in the print queue are also automatically sent to the printer when you exit the procedure, send an item that uses a different form, or send an item to a print file. Items are also sent automatically when internal buffers have been filled.

Items sent to a file: If you have routed pictures taken with the SPRINT command to a file rather than to a printer, the file is closed when you execute a FREE command. It is also closed when you send an item that uses a different form, send items to a different print file or to the print queue, or exit the procedure.

NOTE: Any items sent to the same print file after it has been closed will replace the current contents.

PRTFILE *'filename'*

PRTFILE *fileref*

PRTFILE CLEAR

specifies a file to which the procedure sends pictures taken with the SPRINT command instead of sending them to the default printer. You can specify an actual filename or a previously assigned fileref.

Using a filename: To specify a file named *destination-file*, execute

`prtfile` '*destination-file*'

where *destination-file* follows your system's conventions. Note that quotes are required when you specify a filename rather than a fileref.

Using a fileref: You can also specify a previously assigned fileref.

Using the default: Specify PRTFILE CLEAR to prompt the procedure to route information once again to the queue for the default printer.

Identify the current print file: Specify PRTFILE to prompt the procedure to identify the current print file.

SPRINT [NOBORDER][NOCMD]

takes a picture of the current window exactly as you see it, including window contents, border, and command line. By default, the picture is sent to the queue for the default printer.

Border and command line: By default, both the window border and command line are included in the picture you take with the SPRINT command. You can capture a picture of the window contents that excludes either the window border, the command line, or both. Specify the NOBORDER option to exclude the border and the NOCMD option to exclude the command line. Taking a picture of the window contents without the border and command line is a convenient way to print text for a report.

Destination: The destination of the picture captured with the SPRINT command is determined by the PRTFILE command. By default, the picture goes to the default printer. Use the PRTFILE command if you want it sent to a file instead. Each time you execute the SPRINT command, the picture you take is appended to the current print file; it does not write over the current file. See the PRTFILE command for further explanation.

Graphics Version

Formatting the Chart

If necessary, PROC GANTT divides the Gantt chart into several pages. You can force the Gantt chart to fit on one page by specifying the COMPRESS option in the CHART statement. You can achieve a similar result using the PCOMPRESS option, which also maintains the aspect ratio. In addition, you can fit the chart into a prescribed number of horizontal and vertical pages by using the HPAGES= and VPAGES= options in the CHART statement.

The amount of information contained on each page is determined by the values of the graphics options HPOS= and VPOS= specified in a GOPTIONS statement. If any compression of the Gantt chart is performed, the values of HPOS and VPOS are increased, as necessary, to the number of rows and columns respectively,

that the entire chart occupies in uncompressed mode. The default height of each row of the Gantt chart is computed as (100/*v*)% of the screen height where VPOS=*v*. Thus, the larger the value of VPOS, the narrower the row. You can control the default bar height and default bar offset by using the BARHT= option and the BAROFF= option, respectively. You can further override these at the schedule level. For example, the ABARHT= option affects only the height of the actual schedule bars. The screen is assumed to be divided into *h* columns where HPOS=*h*; thus, each column is assumed to be as wide as (100/*h*)% of the screen width. Hence, the specifications SCALE=10 and MININTERVAL=WEEK imply that a duration of one week is denoted by a bar of length (1000/*h*)% of the screen width.

The height of the text characters is controlled by both the HEIGHT= option in the CHART statement and the HTEXT= option specified in a GOPTIONS statement. The text height is set equal to the product of the HEIGHT= and HTEXT= values. (If neither the HEIGHT= option nor the HTEXT= option has been specified, the text height is given by the font size attribute of the GraphDataText element of the current ODS style template. See the section "ODS Style Templates" on page 565 for more information about ODS styles.) The units in which the text height is measured are those of the HTEXT= option. By default, the value of HEIGHT= is 1, which sets the text height to be equal to the HTEXT= value. The default value of HTEXT= is 1 unit, where a unit is defined by the GUNIT= option in a GOPTIONS statement. Thus, in the absence of the HEIGHT=, HTEXT=, and GUNIT= options, and with no font size provided by the current ODS style template, the text height is the same as the bar height, namely one cell height. Increasing the value of HEIGHT= is useful when you use the COMPRESS option, particularly when you have a very large chart. Since the chart is scaled as appropriate to fit on one page, the text can be very hard to discern, or even illegible, and would benefit from enlargement. Relative positioning of the font baseline for activity text is controlled by the HTOFF= option in the CHART statement. By default, the font baseline for an activity is at the bottom of the first bar corresponding to the activity.

The color of the text characters is specified by using the CTEXT= option in the CHART statement. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information. You can override the text colors for selected columns of activity text at the activity level by using a PATTERN variable in the Schedule data set and specifying the CTEXTCOLS= option in the CHART statement.

The font used for the text characters is specified with the FONT= option in the CHART statement. The default font depends on the GOPTIONS statement and the GSTYLE system option; see the section "ODS Style Templates" on page 565 for more information.

Global PATTERN statements are used to specify the fill pattern for the different types of bars drawn on the Gantt chart. Each fill pattern can be associated with a color. Patterns can be used to reflect the status of an activity (normal, critical, supercritical) in the predicted early/late schedule, to indicate the different schedule types (actual, resource-constrained, baseline), and to represent weekends, holidays and breaks on the Gantt chart. See the section "Using PATTERN Statements" on page 541 for details. In addition, you can override these fill patterns for selected schedules at an activity level by using a PATTERN variable in the Schedule data set and specifying the PATLEVEL= option in the CHART statement.

You can use global SYMBOL statements to define the symbols that represent CHART variables in the Gantt chart. The SYMBOL statement enables you to select symbols from different fonts and modify their appearance to suit your requirements. You can specify a color and a height for the symbol in addition to a variety of other options. See the section "Using SYMBOL Statements" on page 545 for details.

Annotate Processing

The Annotate facility enables you to enhance graphics output produced by PROC GANTT. However, if the only items being annotated are symbols and text strings, it is recommended that you use the Automatic Text Annotation facility that is built into the Gantt procedure instead. This facility was developed specifically for labeling Gantt charts; it has some very useful features and requires a minimum of effort.

To use the SAS/GRAPH Annotate facility, you must create an Annotate data set that contains a set of graphics commands that can be superimposed on the Gantt chart. This data set has a specific format and must contain key variables. Each observation in the Annotate data set represents a command to draw a graphics element or perform an action. The values of the variables in the observation determine what is done and how it is done. The observations in an Annotate data set can be created by explicitly assigning values to the Annotate variables through a DATA step or SAS/FSP procedure or by implicitly assigning values with Annotate macros within a SAS DATA step. The process of creating Annotate observations is greatly simplified through the use of Annotate macros.

Coordinates specify where graphic elements are to be positioned. A coordinate system, in turn, determines how coordinates are interpreted. There are several different coordinate systems that are used by the Annotate facility. Typically, one of three major drawing areas can be associated with any coordinate system: data area, procedure output area, and graphics output area. This chapter explains the coordinate system that is based on the data area of PROC GANTT.

When annotating a graph produced by any of the graphics procedures, you may find it helpful to use data coordinates that refer to the data values corresponding to the graph that is being annotated. For example, if you want to label a particular activity of a Gantt chart with additional text, you can position the text accurately if you use data coordinates instead of screen coordinates. With respect to PROC GANTT, the Annotate facility uses the time axis and the activity axis of the Gantt chart as the basis for the data coordinate system. To use this feature, create a Annotate data set based on the Schedule data set that is input to the procedure, utilizing Annotate macros whenever possible to simplify the process.

NOTE: The data coordinate system enables you to annotate the graph even if it spans multiple pages. However, each annotation must be entirely contained within a given page. For example, you cannot annotate a line on the Gantt chart that runs from one page of the chart to another.

In addition to a coordinate system based on the data, you can select a coordinate system based on either the procedure output area or the Graphics output area. You would typically need to use one of these systems, for example, if you want to annotate text outside the chart area.

Using PATTERN Statements

PROC GANTT uses those patterns that are available with the GCHART procedure. PROC GANTT uses a maximum of nine different patterns to denote various phases in an activity's duration and the various types of schedules that are plotted. Patterns are specified in PATTERN statements that can be used anywhere in your SAS program. Table 8.7 lists the function of each of the first nine PATTERN statements that are used by PROC GANTT.

Any PATTERN statements that you specify are used. If more are needed, default PATTERN statements are used.

You can override any of these patterns at the activity level by using a PATTERN variable in the schedule data set. A PATTERN variable is identified by specifying the PATTERN= option in the CHART statement or by the presence of the default _PATTERN variable.

Table 8.7 PATTERN Statements used by PROC GANTT

PATTERN	Used to Denote
1	Duration of a noncritical activity
2	Slack time for a noncritical activity
3	Duration of a critical activity
4	Slack time for a supercritical activity
5	Duration of a supercritical activity
6	Actual duration of an activity
7	Break due to a holiday
8	Resource-constrained duration of an activity
9	Baseline duration of an activity

Refer to the SAS/GRAPH documentation for a detailed description of PATTERN statements. Most of the relevant information is reproduced here for the sake of completeness.

PATTERN Statement Syntax

The general form of a PATTERN statement is

PATTERN*n options;*

where

- *n* is a number ranging from 1 to 255. If you do not specify a number after the keyword PATTERN, PATTERN1 is assumed.
- *options* enables you to specify the colors and patterns used to fill the bars in your output.

PATTERN statements are additive; if you specify a C= or V= option in a PATTERN statement and then omit that option in a later PATTERN statement ending in the same number, the option remains in effect. To turn off options specified in a previous PATTERN*n* statement, either specify all options in a new PATTERN*n* statement, or use the keyword PATTERN*n* followed by a semicolon. For example, the following statement turns off any C= or V= option specified in previous PATTERN3 statements:

```
pattern3;
```

You can reset options in PATTERN statements to their default values by specifying a null value. A comma can be used (but is not required) to separate a null parameter from the next option.

For example, both of the following statements cause the C= option to assume its default value (the value of the CPATTERN= option or the first color in the COLORS= list):

```
pattern c=, v=solid;
```

or

```
pattern c= v=solid;
```

In the following statement, both options are reset to their default values:

```
pattern2 c= v=;
```

You can also turn off options by specifying the RESET= option in a GOPTIONS statement.

General options

You can specify the following options in a PATTERN statement.

COLOR= *color*

C= *color*

specifies the color to use for a bar or other area to be filled. If you do not specify the C= option in a PATTERN statement, the procedure uses the value you specified for the CPATTERN= option in a GOPTIONS statement. If you omitted the CPATTERN= option, the procedure uses the pattern specified by the V= option (see below) with each color in the COLORS= list before it uses the next PATTERN statement.

REPEAT= *n*

R= *n*

specifies the number of times the PATTERN statement is to be reused. For example, the following statement represents one pattern to be used by SAS/GRAPH software:

```
pattern1 v=x3 c=red;
```

You can use the REPEAT= option in the statement to repeat the pattern before going to the next pattern. For example, if you specify the following statements, PATTERN1 is repeated ten times before PATTERN2 is used:

```
pattern1 v=x3 c=red  r=10;
pattern2 v=s  c=blue r=10;
```

Remember that if you omit the COLOR= option in the PATTERN statement and you do not specify the CPATTERN= option, SAS/GRAPH software repeats the pattern for each color in the current COLORS= list. If you specify the R= option in a PATTERN statement from which the C= option is omitted, the statement cycles through the COLORS= list the number of times given by the value of the R= option.

For example, if the current device has seven colors, then the following statement results in 70 patterns because each group of seven patterns generated by cycling through the COLORS= list is repeated ten times:

```
pattern v=x3 r=10;
```

VALUE= *value*

V= *value*

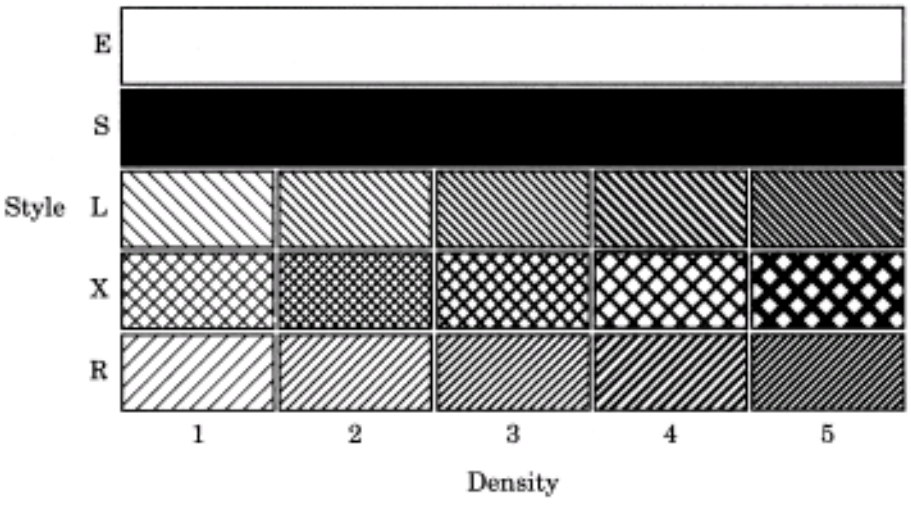
specifies the pattern to use for a bar or other area to be filled. The valid values you can use depend on what procedure you are using and the type of graph you are producing. In PROC GANTT, which produces bars, you must use one of the pattern values shown in Figure 8.6.

In a PATTERN statement, if you specify a value for the V= option but not for the C= option, the procedure uses the value you specified for the CPATTERN= option in a GOPTIONS statement. If you omitted the CPATTERN= option, the procedure uses the pattern specified for the V= option with each color in the COLORS= list before it uses the next PATTERN statement. Thus, if you specify the following statements, the PATTERN1 statement is used for the first type of bar, namely, for the duration of a noncritical activity:

```
pattern1 c=red   v=x3;
pattern2         v=s;
pattern3 c=blue  v=l3;
pattern4 c=green v=r4;

proc gantt data=sched;
```

The PATTERN2 statement is used for the second type of bar, namely, for the slack time of a noncritical activity. Because a C= value is not specified in the PATTERN2 statement, SAS/GRAPH software uses the PATTERN2 statement and cycles through the colors in the COLORS= list for the device to obtain as many patterns as there are colors in the list. If needed, the PATTERN3 and PATTERN4 values are then used for any remaining types of bars.

Figure 8.6 Pattern Selection Guide



Using SYMBOL Statements

You can specify a SYMBOL statement anywhere in your SAS program. SYMBOL statements give PROC GANTT information about the characters to be used for plotting the CHART variables.

See also the section "Special Fonts for Project Management and Decision Analysis" on page 547 for a description of some typically used Gantt chart symbols that can be specified using a SYMBOL statement.

Refer to the SAS/GRAPH documentation for a detailed description of SYMBOL statements. Most of the relevant information is reproduced here for the sake of completeness.

SYMBOL Statement Syntax

The general form of a SYMBOL statement is

SYMBOL*n options;*

where

- *n* is a number ranging from 1 to 255. Each SYMBOL statement remains in effect until you specify another SYMBOL statement ending in the same number. If you do not specify a number following the keyword SYMBOL, SYMBOL1 is assumed.
- *options* enables you to specify the plot characters and color.

SYMBOL statements are additive; that is, if you specify a given option in a SYMBOL statement and then omit that option in a later SYMBOL statement ending in the same number, the option remains in effect. To turn off all options specified in previous SYMBOL statements, you can specify all options in a new SYMBOL*n* statement, use the keyword SYMBOL*n* followed by a semicolon, or specify a null value. A comma can be used (but is not required) to separate a null parameter from the next option.

For example, both of the following statements cause the C= option to assume its default value (the value of the CSYMBOL= option or the first color in the COLORS= list):

```
symbol1 c=, v=plus;
```

and

```
symbol1 c= v=plus;
```

In the following statement, both options are reset to their default values:

```
symbol4 c= v=;
```

You can also turn off options by specifying the RESET= option in a GOPTIONS statement.

General options

You can specify the following options in the SYMBOL statement.

COLOR=*color*

C=*color*

specifies the color to use for the corresponding plot specification. If you do not specify the C= option in a SYMBOL statement, the procedure uses the value you specified for the CSYMBOL= option in a GOPTIONS statement. If you omit the CSYMBOL= option, the procedure uses the value specified by the V= option with each color in the COLORS= list before it uses the next SYMBOL statement.

FONT=*font*

F=*font*

specifies the font from which the symbol corresponding to the value specified with the V= option is to be drawn. If you do not specify a font, the V= option specifies the symbol from the special symbol table shown in Figure 8.7.

H=*height*

specifies the height of the symbol that is to be drawn.

For example, this SYMBOL statement

```
symbol1 c=green v=K f=special h=2;
```

indicates that the symbol at each data point is the letter K from the SPECIAL font (a filled square), drawn in green, the height being twice the bar height.

REPEAT=*n*

R=*n*

specifies the number of times the SYMBOL statement is to be reused.

V=*special-symbol*

V=*'string'*

identifies the symbols from the font specified by the FONT= option in the SYMBOL statement for the corresponding plot specifications. If the FONT= option is not specified, the plot symbol is the symbol corresponding to the value of V= in the special symbol table shown in Figure 8.7. Also permitted without a FONT= specification are the letters A through W and the numbers 0 through 9. If the font is a symbol font, such as MARKER, the string specified with the V= option is the character code for the symbol. If the font is a text font, such as SWISS, the string specified with the V= option is displayed as the plot symbol. By default, the value of V= is PLUS, which produces the plus symbol (+) from the special symbol table.

Note that if you use the special symbol comma (,) with the V= option, you must enclose the comma in quotes as illustrated in the following statement:

```
symbol1 v=',';
```

Figure 8.7 Special Symbol Table

VALUE=		Plot Symbol	VALUE=		Plot Symbol
PLUS			%	(percent)	
X			&	(ampersand)	
STAR			'	(single quote)	
SQUARE			=	(equals)	
DIAMOND			-	(hyphen)	
TRIANGLE			@	(at)	
HASH			*	(asterisk)	
Y			+	(plus)	
Z			>	(greater than)	
PAW			.	(period)	
POINT			<	(less than)	
DOT			,	(comma)	
CIRCLE			/	(slash)	
_	(underscore)		?	(question mark)	
"	(double quote)		(	(left parenthesis)	
#	(pound sign)		)	(right parenthesis)	
$	(dollar sign)		:	(colon)	

Note: The words or special characters in the VALUE= column are entered exactly as shown.

Special Fonts for Project Management and Decision Analysis

Two special marker fonts, ORFONT and ORFONTE, are available in versions 6.08 and later. These two fonts are meant to be used with SAS/OR software and provide a variety of symbols that are typically used in Project Management and Decision Analysis. The fonts ORFONT and ORFONTE are shown in Figure 8.8 and Figure 8.9, respectively. The fonts behave like any SAS/GRAPH font providing you with the capability to control attributes such as color and height.

Figure 8.8 ORFONT - A Filled Font

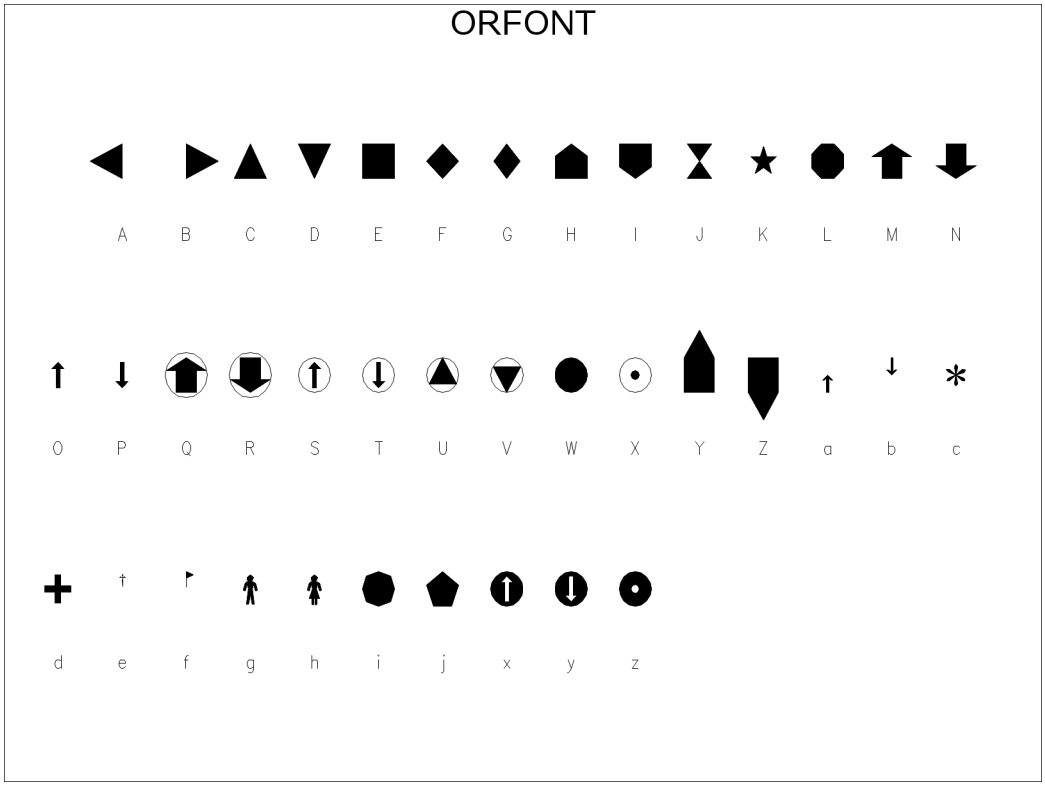
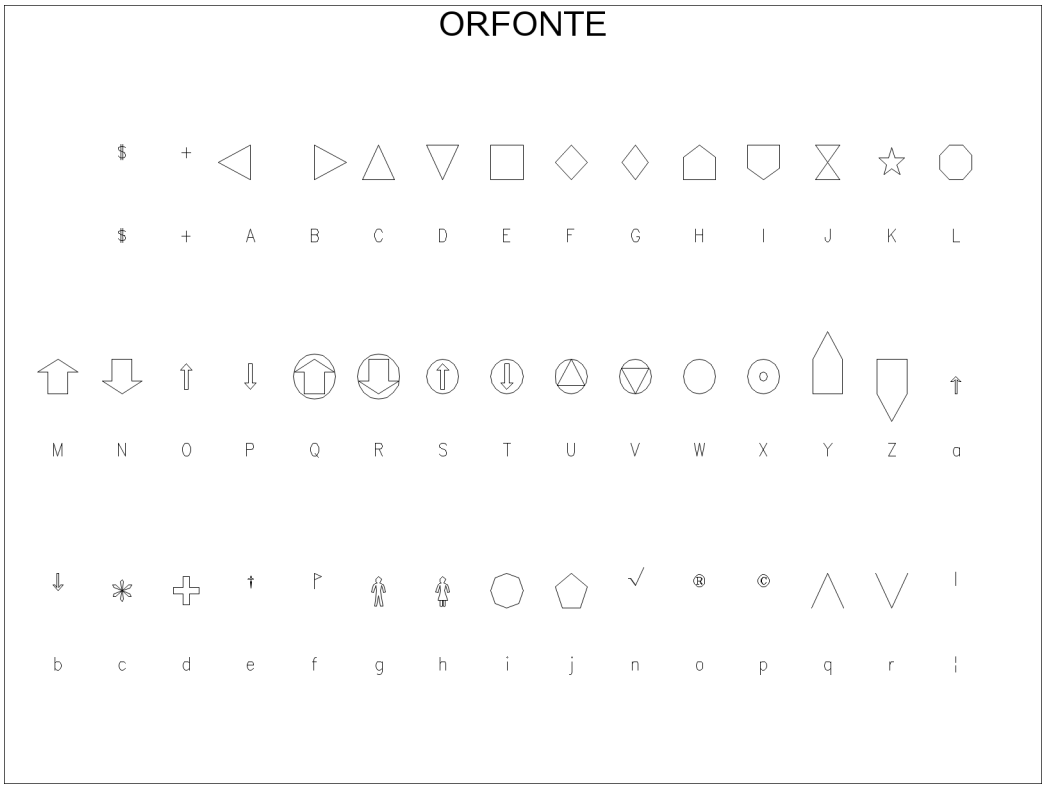


Figure 8.9 ORFONTE - An Empty Font



For example, to use a filled yellow “doghouse” symbol to represent milestones on the Gantt chart, specify the options

```
VMILE="H"  FMILE=ORFONT  CMILE=YELLOW
```

in the CHART statement.

If you wish to represent a CHART variable with an empty blue “circled arrow,” then specify the following options in the corresponding SYMBOL statement.

```
V="Q"  F=ORFONTE  C=BLUE;
```

Specifying the Logic Options

The Logic options are a family of options used with the GANTT procedure that enable you to view the precedence relationships between activities on the Gantt chart. The Logic options constitute a high-resolution graphics feature and, as such, are only valid with specification of the GRAPHICS option in the PROC GANTT statement. The Logic options can accommodate nonstandard precedence relationships. The Logic options enable you to control the color, line style, and width of the connecting arcs as well as their layout and positioning on the Gantt chart. You can specify the precedence information required to draw the connections in one of two formats and store it in a data set different from the Schedule data set. You can also use the Schedule data set produced by PROC CPM to provide the precedence information. When using the Schedule data set from PROC CPM, you can ensure that all the relevant precedence information exists in the data set by either specifying the XFERVARS option in the PROC CPM statement or by using an ID statement.

The Logic options are not valid with the specification of either a BY statement or the COMBINE option in the CHART statement.

In order to invoke the logic options, you need to, minimally, specify a set of variables that defines the precedence relationships between tasks. This can be done using one of two formats for defining project networks, the AOA specification or the AON specification.

Activity-on-Arc (AOA) Specification

In the AOA specification, each activity of the project is represented by an arc. The node at the tail of the arc represents the start of the activity, and the node at the head of the arc represents the finish of the activity. The relationship between an activity and its successor is represented by setting the tail node of the successor arc to be the head node of the activity arc. One of the disadvantages of using the AOA method is that it cannot accommodate nonstandard lag types; all lag types are of the Finish-to-Start (FS) type.

The variables required by PROC GANTT to establish a valid AOA specification are defined using the HEADNODE= and TAILNODE= options in the CHART statement.

Activity-on-Node (AON) Specification

In the AON specification, each activity is represented by a node. All arcs originating from an activity terminate at its successors. Consequently, all arcs terminating at an activity originate from its predecessors.

The variables required by PROC GANTT to establish a valid AON specification are defined by the ACTIVITY= and SUCCESSOR= options in the CHART statement.

Optionally, nonstandard precedence relationships can be specified using the LAG= option in the CHART statement to define a variable that defines the lag type of a relationship.

Precedence Data Set

When using the AON specification, you can specify the precedence information using a data set different from the Schedule data set. This is particularly useful when producing several Gantt charts for the same project with different schedule information as would typically be the case when monitoring a project in progress. It eliminates the requirement that the precedence information exist in each Schedule data set and enables for more compact data. This separate data set is specified by the PRECDATA= option in the PROC GANTT statement statement and is referred to as the *Precedence data set*.

In order to graphically represent the precedence relationships derived from the Precedence data set on the Gantt chart, you must link the Precedence data set with the Schedule data set by means of a common variable. This common variable is selected as the ACTIVITY variable by virtue of the fact that it always exists in the Precedence data set. Thus, when using the Precedence data set, you need to ensure that the ACTIVITY variable exists in the Schedule data set, too.

In the event that both a valid AOA and a valid AON specification exist, PROC GANTT uses the AON specification by default. To override the default, use the AOA option in the CHART statement.

Drawing the Precedence Connections

The relationship between an activity and its successor is represented on the Gantt chart by a series of horizontal and vertical line segments that connect their schedule bars corresponding to a specified type (early/late, actual, and so forth). For a given connection, the intersection of a horizontal segment with a vertical segment is called a *turning point* of the connection. The type of the schedule bar used for the connection, also called the *logic bar*, is determined by the LEVEL= option in the CHART statement.

Every connection is comprised of either three or five segments and is termed a 3-segment or a 5-segment connection, respectively. The segments are routed in the following sequence:

a) a horizontal segment that originates from the appropriate end of the logic bar corresponding to the activity. The length of this segment is controlled by the MINOFFGV= and MININTGV= options in the CHART statement.

b) a vertical segment traveling from activity to the successor

c) a horizontal segment traveling towards the appropriate end of the successor's logic bar. The length of this segment is determined by the MINOFFLV= and MAXDISLV= options in the CHART statement.

d) a vertical and horizontal segment into the logic bar of the successor

Every connection begins with a horizontal line segment originating from the activity's logic bar and ends with a a horizontal line segment terminating at the successor's logic bar. If the lag type of the relationship is SS or SF, the initial horizontal segment originates from the left end of the activity's logic bar, otherwise it originates from the right end of the logic bar. If the lag type of the relationship is SS or FS, the final horizontal segment terminates at the left end of the successor's logic bar, otherwise it terminates at the right end of the logic bar.

NOTE: The ends of the bars must be consistent with the lag type of the connection if it is to be drawn; that is, the left end of the activity's logic bar must represent a start time if an SS or SF lag type connection is to be drawn, and the right end of the activity's logic bar must represent a finish time if an FS or FF lag type connection is to be drawn.

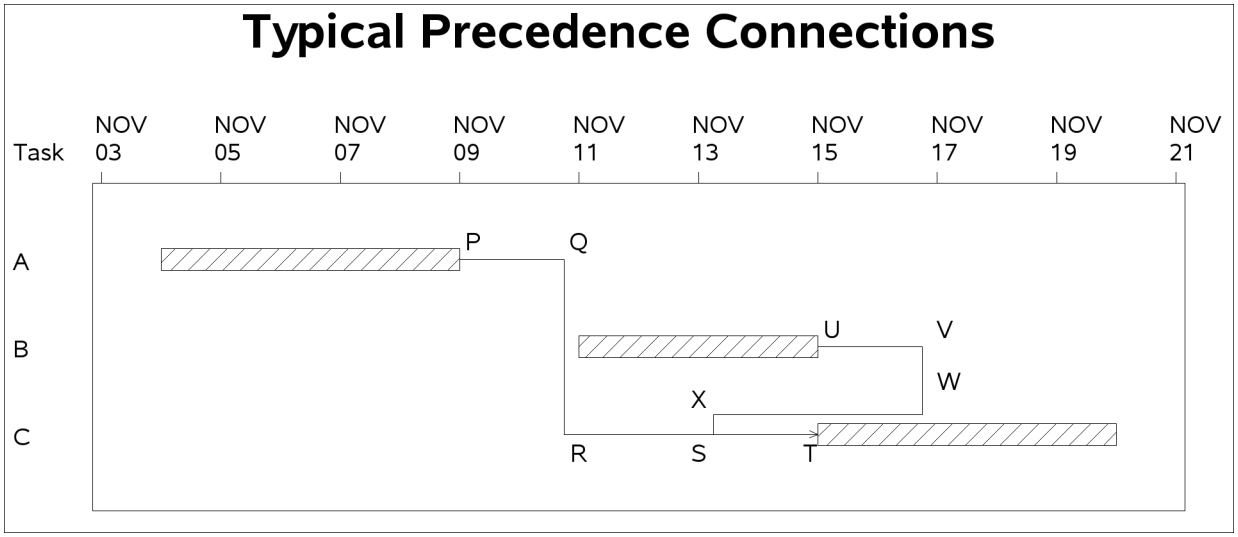
Violation of these conditions is unlikely when using the Schedule data set generated by PROC CPM. An example violating these conditions is a Schedule data set containing incorrect or invalid data. The following example illustrates two observations that are in violation of these conditions. The first observation is invalid data (E_START greater than E_FINISH) while the second observation is incomplete (missing E_START and L_FINISH times).

```
E_START   E_FINISH  L_START  L_FINISH
03MAR04   01MAR04      .         .
   .      05MAR04   07MAR04      .
```

The following figure illustrates two typical precedence connections between an activity and its successor.

```
ACTIVITY  SUCCESSOR   LAG
   A          C       FS
   B          C       FS
```

Figure 8.10 Typical Precedence Connections



The connection from activity **A** to activity **C** is comprised of three segments PQ, QR, and RT whereas the connection from activity **B** to activity **C** is made up of five segments UV, VW, WX, XS, and ST; the two additional segments correspond to the optional segments mentioned in item **d)** above. Points Q, R, V, W, X, and S are turning points.

Formatting the Axis

If neither MINDATE= nor MAXDATE= have been specified, the time axis of the Gantt chart is extended by a small amount in the appropriate direction or directions in an attempt to capture all of the relevant precedence connections on the chart. While this will succeed for the majority of Gantt charts, it is by no means guaranteed. If connection lines still tend to run off the chart, you can perform one or both of the following tasks.

- Use the MINDATE= or MAXDATE= options (or both) in the CHART statement to increase the chart range as necessary.
- Decrease the values of the MINOFFGV=, MININTGV=, MAXDISLV=, and MINOFFLV= options to reduce the horizontal range spanned by the vertical segments so that they will lie within the range of the time axis.

On the other hand, if the automatic extension supplied by PROC GANTT is excessive, you can suppress it by specifying the NOEXTRANGE option in the CHART statement.

The following section, "Controlling the Layout," addresses the CHART statement options MINOFFGV=, MININTGV=, MINOFFLV=, and MAXDISLV= which control placement of the vertical segments that make up a connection. For most Gantt charts, default values of these options will suffice since their usage is typically reserved for "fine tuning" chart appearance. This section can be skipped unless you want to control the layout of the connection. The description of the layout methodology and concepts is also useful to help you understand the routing of the connections in a complex network with several connections of different types.

Controlling the Layout

The concepts of global and local verticals are first introduced in order to describe the function of the segment placement controls.

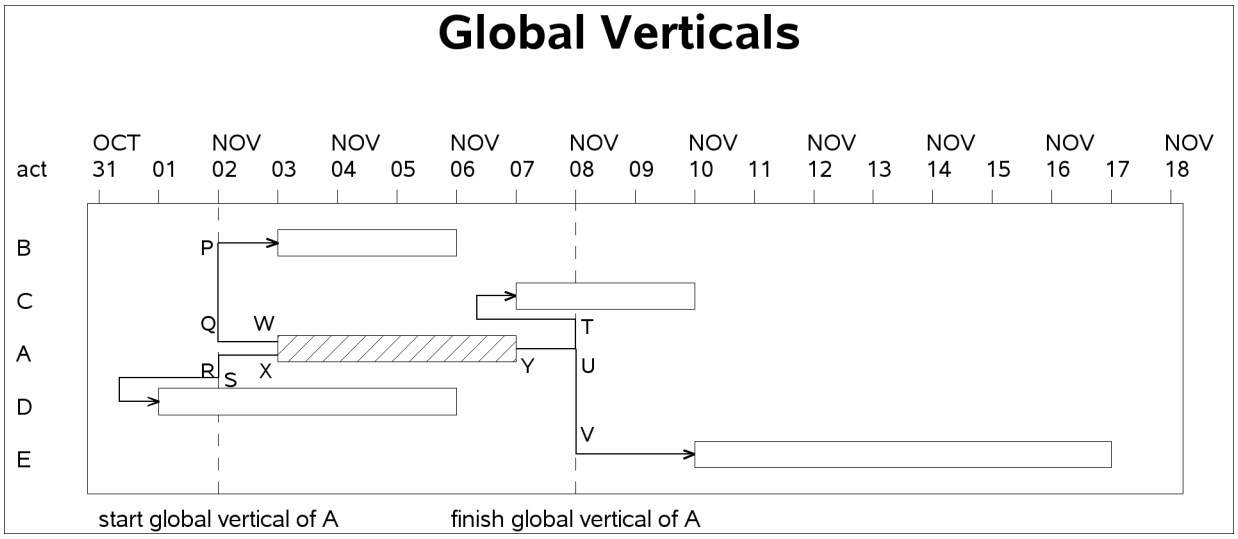
Global Verticals

In the interest of minimizing clutter on the chart, each activity is assigned a maximum of two vertical tracks for *placement* of the vertical segment described in item **b)** above. One vertical track is maintained for SS and SF lag type connections and is referred to as the *start global vertical* of the activity, while the other vertical track is maintained for FS and FF lag type connections and is referred to as the *finish global vertical* of the activity. The term *global vertical* refers to either start global vertical or finish global vertical.

NOTE: The use of the term "global" is attributed to the fact that in any connection from an activity to its successor, the global vertical of the activity corresponds to the *only segment* that travels from activity to successor.

Figure 8.11 illustrates the two global verticals of activity **A**.

Figure 8.11 Global Verticals



Activity **A** has four successors: activities **B**, **C**, **D**, and **E**. The lag type of the relationship between **A** and **B** is nonstandard, namely 'Start-to-Start', as is that between **A** and **D**. The other two lag types are standard. The start and finish global verticals of activity **A** are represented by the two dotted lines. The vertical segments of the SS lag type connections from **A** to **B** and from **A** to **D** that are placed along the start global vertical of **A** are labeled PQ and RS, respectively. The vertical segments of the FS lag type connections from **A** to **C** and from **A** to **E** that are placed along the finish global vertical of **A** are labeled TU and UV, respectively.

For a given connection from activity to successor, the vertical segment that is placed on the activity global vertical is connected to the appropriate end of the logic bar by the horizontal segment described in item **a)** above. The minimum length of this horizontal segment is specified with the MINOFFGV= option in the CHART statement. Further, the length of this segment is affected by the MININTGV= option in the CHART statement, which is the minimum interdistance of any two global verticals. In Figure 8.11, the horizontal segments QW and RX connect the vertical segments PQ and RS, respectively, to the logic bar and the horizontal segment YU connects both vertical segments TU and UV to the logic bar.

Local Verticals

Each activity has seven horizontal tracks associated with it, strategically positioned on either end of the logic bar, above the first bar of the activity, and below the last bar of the activity. These tracks are used for the *placement* of the horizontal segments described in items **c)** and **d)**, respectively.

Figure 8.12 illustrates the positions of the horizontal tracks for an activity in a Gantt chart with four schedule bars. Three of the horizontal tracks, namely **`track 1`**, **`track 4`**, and **`track 7`**, service the start of the logic bar and are connected to one another by a vertical track referred to as the *Start Local Vertical*. Similarly, the horizontal tracks **`track 2`**, **`track 3`**, **`track 5`**, and **`track 6`** service the finish of the bar and are interconnected by a vertical track referred to as the *Finish Local Vertical*. The local verticals are used for *placement* of the vertical segment described in item **d)** above.

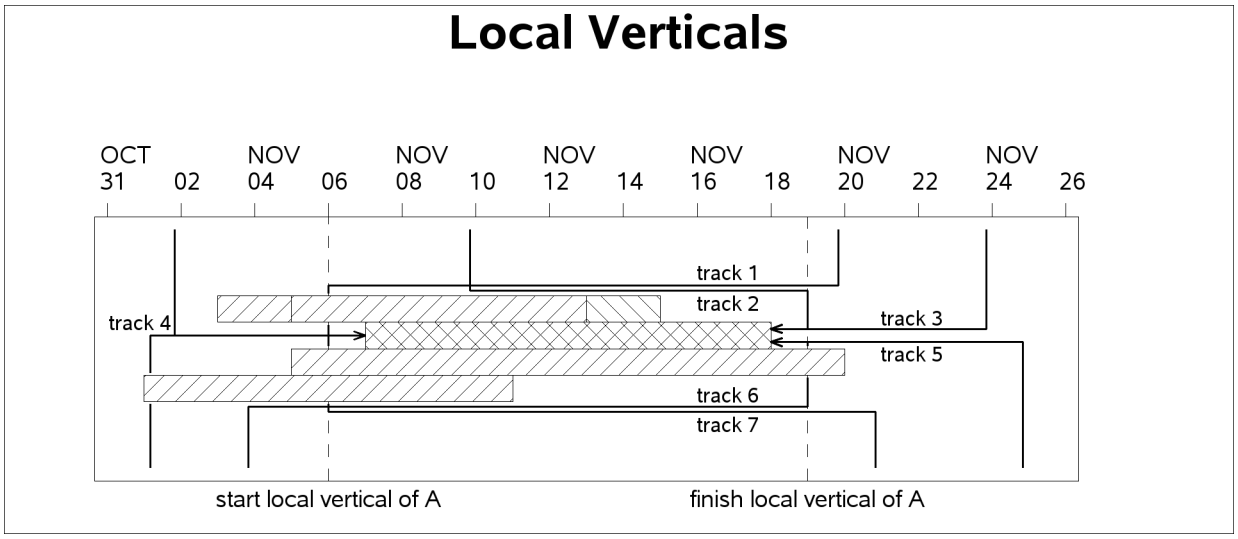
NOTE: The use of the term "local" is attributed to the fact that the local vertical is used to connect horizontal tracks associated with the *same* activity.

Notice that **`track 1`** and **`track 7`** terminate upon their intersection with the start local vertical and that **`track 2`** and **`track 6`** terminate upon their intersection with the finish local vertical.

The minimum distance of a local vertical from its respective bar end is specified with the MINOFFLV= option in the CHART statement. The maximum displacement of the local vertical from this point is specified using the MAXDISLV= option in the CHART statement. The MAXDISLV= option is used to offset the local vertical in order to prevent overlap with any global verticals.

Arrowheads are drawn by default on the horizontal tracks corresponding to the logic bar, namely **`track 3`**, **`track 4`**, and **`track 5`**, upon entering the bar and on continuing pages. The NOARROWHEAD option is used to suppress the display of arrowheads.

Figure 8.12 Local Verticals



Routing the Connection

The routing of the precedence connection from an activity to its successor is dependent on two factors, namely

- the horizontal displacement of the appropriate global vertical of the activity relative to the appropriate local vertical of the successor
- the vertical position on the task axis of the activity relative to the successor

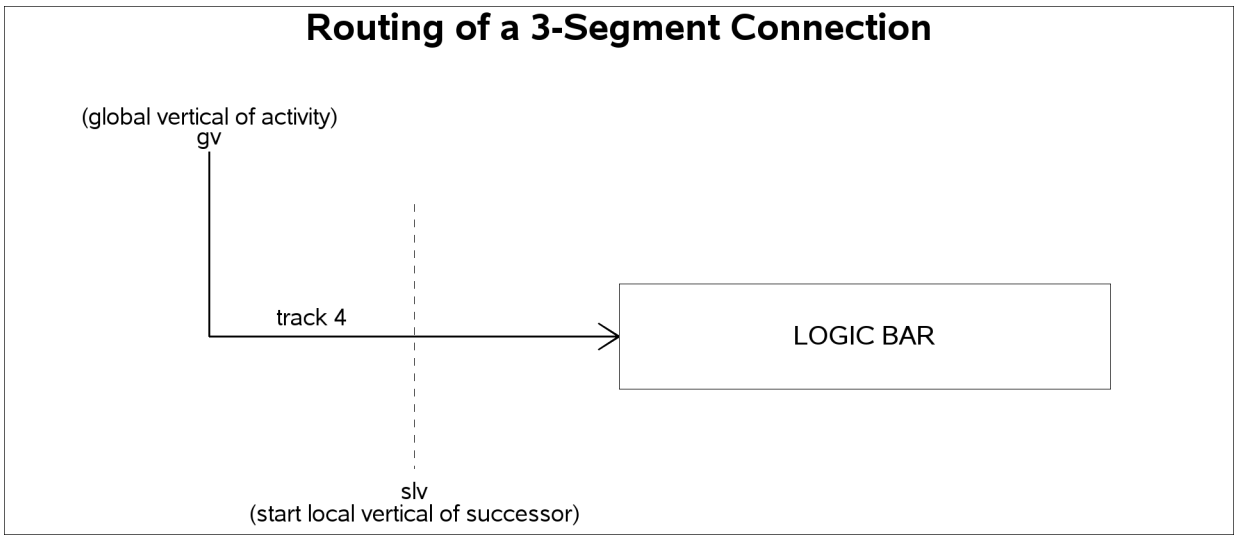
The routing of a SS or FS type precedence connection from activity to successor is described below. A similar discussion holds for the routing of a SF or FF type precedence connection.

Suppose the activity lies above the successor. Let the start local vertical of the successor be denoted by *slv*, and let the appropriate global vertical of the activity be denoted by *gv*.

Case 1:

If *gv* lies to the left of *slv*, then the connection is routed vertically down along *gv* onto **`track 4`** of the successor, on which it is routed horizontally to enter the bar. The resulting 3-segment connection is shown in Figure 8.13.

Figure 8.13 3-Segment Connection

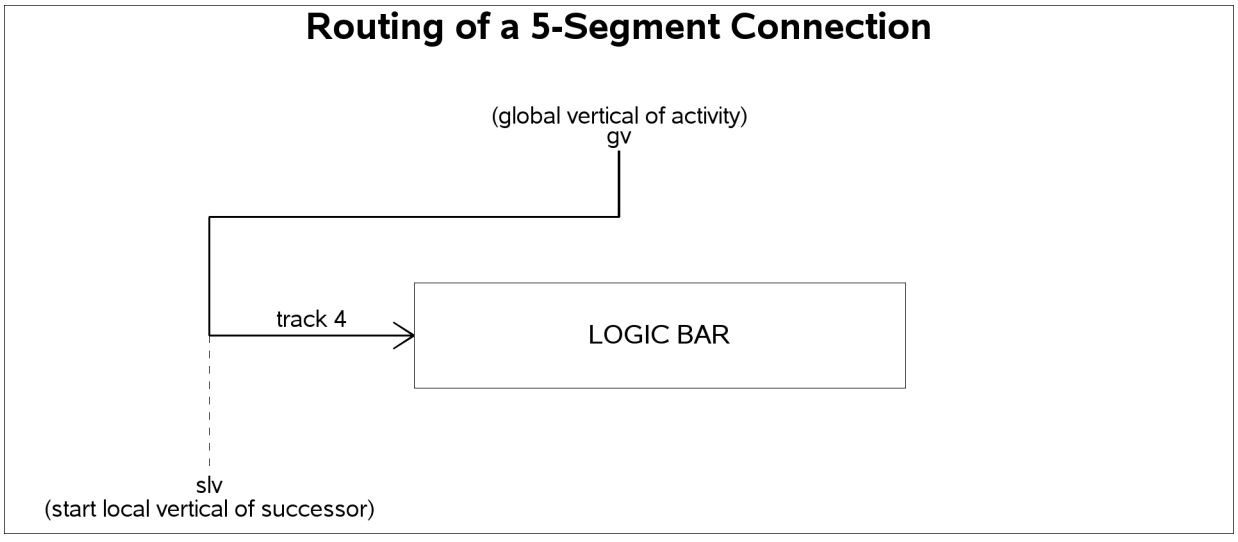


An example of this type of routing is illustrated by the connection between activities **`A`** and **`C`** in Figure 8.10.

Case 2:

If *gv* lies to the right of *slv*, then the connection is routed vertically down along *gv* onto **`track 1`** of the successor, horizontally to the left to meet *slv*, vertically down along *slv* onto **`track 4`** of the successor and horizontally to the right to enter the bar. The resulting 5-segment connection is shown in Figure 8.14.

Figure 8.14 5-Segment Connection



This type of routing is illustrated by the connection between activities **`B`** and **`C`** in Figure 8.10.

An identical description applies when the activity lies below the successor, with the only difference being that **`track 7`** is used in place of **`track 1`** (see Figure 8.12).

Automatic Text Annotation

The automatic text annotation feature is designed specifically for labeling Gantt charts independently of the SAS/GRAPH Annotate facility. This facility enables you to display label strings with a minimum of effort and data entry while providing the capability for more complex chart labeling situations. Some of the properties that characterize this feature are

- the ability to tag labels. This enables you to define 1-1, 1-many, many-1, and many-many relationships.
- the ability to link label coordinates and label strings to variables in the Schedule data set. This enables the Label data set to remain unchanged even if the Schedule data set changes, such as when monitoring a project.
- the ability to automatically format or convert numeric variable values that have been specified for label text strings
- the ability to automatically split strings embedded with blanks to make the pieces as equal in length as possible, with the provision to override this behavior by specifying a split character
- the ability to mix data and percentage coordinates
- the ability to clip labels running off the frame of the Gantt chart

All relevant information is contained in a SAS data set specified using the LABDATA= data set option in the PROC GANTT statement. This data set is also referred to as the Label data set in the context of this documentation. The Label data set is required to contain certain variables in order to determine the label string and the positional information related to the string. At the very least, it requires three variables, one to determine the string to be displayed, one to determine the horizontal position, and one to determine the vertical position. The procedure terminates if it cannot find the required variables.

Table 8.8 Special Symbol Table

Determining the . . .	requires the following variables
Label text string	_LVAR and/or _LABEL
Horizontal placement position	_XVAR and/or _X
Vertical placement position	LABVAR= and/or _Y

The LABVAR variable refers to the variable specified with the LABVAR= option in the CHART statement. It is the LABVAR variable that links the Schedule and Label data sets together. As far as possible, the procedure attempts to use the _X, _Y, and _LABEL variables in the Label data set. However, a link established using the LABVAR variable makes the Schedule data set a secondary source of information for determining positional and text string information for linked observations. The exact meaning of the preceding variables is explained later in this chapter.

Note that, other than the preceding requirements, there are no further restrictions on the Label data set. In fact, the Schedule data set can also be specified as the Label data set as long as the required variables are present. There are several optional variables in the Label data set. These variables enable you to specify offsets in

both horizontal and vertical directions from the given coordinate position; adjust graphical attributes such as baseline angles, character rotations, colors, fonts, and heights; control justification of strings; control placement behavior at pagebreaks; and specify coordinate reference systems for the horizontal and vertical values.

Label Data Set

You specify the Label data set using the LABDATA= option in the PROC GANTT statement. This initiates the labeling of the Gantt chart. The Label data set contains the information that provides the means of determining the label strings and their placement positions. As far as possible, the procedure attempts to use the _X, _Y, and _LABEL variables in the Label data set to extract the horizontal position, the vertical position, and the text string, respectively. The Schedule data set acts as a secondary source of information for all Label data set observations that are linked to it. The priority mechanism is described in the section "Determining the Vertical Position" on page 557.

Determining the Vertical Position

You can specify the vertical position for a label string in one of two ways, either directly by using the _Y variable in the Label data set or indirectly by associating the label with an activity or activities. In the latter case, the vertical position is determined by the relative position of the activity on the activity axis of the Gantt chart.

Directly using _Y

The procedure determines the vertical position using the _Y variable. You specify the coordinate system for the value of _Y with the optional _YSYS variable. A value of DATA or DATAVAL for the _YSYS variable indicates that the unit of measurement is data values. This is also the default coordinate system for _Y. A value of PCT or PCTVAL indicates that the unit of measurement is percentage of the procedure output area. When the coordinate system for _Y is based on data values, the values that _Y can take are restricted to positive real numbers with the exception of -1, which is a special value indicating that the label be displayed for every activity. In effect, this is a more concise way of linking a label to every activity.

Indirectly using LABVAR=

If the _Y variable does not exist or its value is missing, the procedure uses the value of the LABVAR variable to determine the vertical position of the label. If the LABVAR= option is specified and the value of the LABVAR variable is nonmissing, the observation is displayed for every activity that provides a matching value for the LABVAR variable. It is quite possible that there are no activities that provide a match, in which case the Label data set observation is ignored. Likewise, the Label data set observation is ignored if the value of the LABVAR variable is missing,

When the vertical position is based on an integer value for _Y or linkage using the LABVAR variable, the default position for the baseline of the string is the top of the first schedule bar corresponding to the activity (unless offsets _XOFFSET or _YOFFSET are used).

Determining the Horizontal Position

The procedure attempts to determine the horizontal position using the _X variable. You specify the coordinate system for the value of _X with the optional _XSYS variable. A value of DATA or DATAVAL for the _XSYS variable indicates that the unit of measurement is data values. This is also the default coordinate system for

_X. A value of PCT or PCTVAL indicates that the unit of measurement is percentage of the procedure output area.

If the _X variable does not exist or its value is missing, the procedure ignores the Label data set observation if the observation is not linked to an activity in the Schedule data set. However, if the label is linked to an activity (either by the LABVAR variable or a value of -1 for _Y, as described previously), the procedure extracts the horizontal position using the _XVAR variable in the Label data set. The _XVAR variable values are names of numeric variables in the Schedule data set. If the _XVAR value is not missing, the horizontal position is the value of the specified variable in the Schedule data set corresponding to the activity. If no such variable exists in the Schedule data set or its value is missing, no label is displayed for this particular (activity, label) link. As with the _X variable, the _XSYS variable names the unit of measurement for the associated Schedule data set variable.

Coordinate Systems

Coordinates can be specified in data values and percentages. It is important to note a significant difference between these two systems when using multiple pages. A data coordinate value is a point along either the time or activity axis, and it can be related to a page number and to a position on that page in the relevant direction. A percentage value, on the other hand, cannot be related to a particular page and, as such, is treated as applicable to every single page. It is possible to mix data and percentage coordinates. That is, the horizontal position can be in data values and the vertical position can be in percentage values, and vice versa. By mixing coordinate systems, you can get as flexible as you want in labeling Gantt charts.

- If both coordinates are in data values, the label is displayed at a specific coordinate on a specific page.
- If the horizontal coordinate is a percentage, the label is displayed at this horizontal position for every page that corresponds to the vertical position. Likewise, if the vertical position is a percentage, the label is displayed at this vertical position for every page that corresponds to the horizontal position. For example, you can display certain headings at the top of the Gantt chart or at the bottom of the Gantt chart by using a data value for the vertical position and a percentage value for the horizontal position.
- If the horizontal and vertical coordinates are both percentages, the label is displayed on every page at the specified coordinate. This feature can be used to display text that appears on every page, much like titles and footnotes, for example.

Determining the Label String

The technique for determining the label string is similar to that of determining the horizontal position.

As far as possible, the procedure attempts to use the _LABEL variable. If the _LABEL variable does not exist or its value is missing, the procedure ignores the label data observation if the observation is not linked to an activity in the Schedule data set. However, if the label is linked to an activity (either by the LABVAR variable or a value of -1 for _Y, as described previously), the procedure extracts the text string from the Schedule data set using the _LVAR variable. The _LVAR variable values are names of variables in the Schedule data set. If the _LVAR value is not missing, the text string is the value of the specified variable in the Schedule data set corresponding to the activity. If no such variable exists in the Schedule data set or if the value is missing, no label is displayed for this particular (activity, label) link.

Note that the _LABEL variable and the Schedule data set variables named by _LVAR are not restricted to be of character type. These variables can be character or numeric, formatted or unformatted. The strings are displayed using the following rules:

- If the variable is of character type, the label is the character string corresponding to the given activity.
- If the variable is of numeric type and formatted, the label is the formatted string.
- If the variable is of numeric type and unformatted, the label is the number displayed as a string with an integer part of up to LABMAXINT= digits and a maximum of MAXDEC= decimal positions. The LABMAXINT= and MAXDEC= options are specified in the PROC GANTT statement and their default values are 16 and 2, respectively.

Optional Information

In addition to specifying the horizontal and vertical coordinates as described previously, you can also specify a relative offset from these values using the _XOFFSET and _YOFFSET variables. These are optional variables and their default values are both 0. The unit of measurement for the _XOFFSET variable is in MININTERVAL units, and the direction of increase is from left to right. The unit of measurement for the _YOFFSET variable is in barheights, and the direction of increase is from top to bottom. When labels are split, the offset variables pertain only to the first piece of the label. The positions of the remaining split pieces are determined from the positioning of the first piece. The adjusted coordinate after taking the offsets into account is what is used for the placement of the string and is known as the referenced coordinate.

You can control the color and font of the label strings using the _CLABEL and _FLABEL variables, respectively. The values for the _CLABEL variable are any valid SAS/GRAPH color names. If the _CLABEL variable does not exist or its value is missing, the value of the CTEXT= option in the CHART statement is used. The values for the _FLABEL variable are any valid SAS/GRAPH font names. If the _FLABEL variable does not exist or its value is missing, the value of the FONT= option in the CHART statement is used.

You can control the height of the label strings with the _HLABEL variable. The units of measurement are in barheights. If the _HLABEL variable does not exist or its value is missing, the default value of 1 is used.

You can specify the angle of the character baseline with respect to the horizontal in degrees using the _ALABEL variable. If the _ALABEL variable does not exist or its value is missing, the default value of 0 is used. You can specify the rotation angle of each character in the string in degrees with the _RLABEL variable. If the _RLABEL variable does not exist or its value is missing, the default value of 0 is used.

You can control the alignment of the string with the _JLABEL variable. Strings can be displayed left-justified, right-justified, or centered at the specified coordinate. By default, all strings are displayed left-justified. The valid values are L or LEFT for left justification, R or RIGHT for right justification, and C or CENTER for centered justification.

The _PAGEBRK variable gives you displaying control when the referenced coordinate of a label coincides with a pagebreak tickmark and the horizontal coordinate is measured in data values. You can specify on which of the two pages you would like the label to be displayed. The default always displays the label on the first page associated with the common tickmark except when the tickmark is the very first tickmark on the Gantt chart. Valid values are 0 (default), 1 (use first page), or 2 (use second page).

Variables in the LABELDATA= data set

The following table lists all the variables associated with the Label data set and their interpretations by the GANTT procedure. The table also lists for each variable its type, the possible values it can assume, and its default value.

Table 8.9 Label Data Set Variables

Name	Type	Description	Allowed Values	Defaults
_Y	N	y position		
_X	N	x position		
_LABEL	C/N	Label string		
_XVAR	C	Name of numeric SAS var in DATA= ds for x position		
_LVAR	C	Name of SAS var in DATA= ds for label string		
_XSYS	C	Coordinate system for _X, _XVAR	DATA, DATAVAL, PCT, PCTVAL	DATA
_YSYS	C	Coordinate system for _Y	DATA, DATAVAL, PCT, PCTVAL	DATA
_PAGEBRK	N	Resolve pagebreak referenced display	0, 1, 2	0
_XOFFSET	N	Horizontal offset in minintervals		0
_YOFFSET	N	Vertical offset in bar heights		0
_ALABEL	N	Baseline angle in degrees		0
_CLABEL	C	SAS/GRAPH color name		CTEXT=
_FLABEL	C	SAS/GRAPH font name		FONT=
_HLABEL	N	Height in barheights		1
_JLABEL	C	Justify text	L, LEFT, R, RIGHT, C, CENTER	L
_RLABEL	N	Character rotation in degrees		0
LABVAR=	C/N	Variable linking activities to labels		

Web-Enabled Gantt Charts

The WEB variable enables you to define an HTML reference for each activity. This HTML reference is currently associated with all the schedule bars, milestones, and ID variables that correspond to the activity. The WEB variable is a character variable, and the values need to be of the form “HREF=htmlpage.”

In addition, you can also store the coordinate and link information defined by the WEB= option in a SAS data set by specifying the IMAGEMAP= option in the PROC GANTT statement. By processing this SAS data set using a DATA step, you can generate customized HTML pages for your Gantt chart.

Mode-Specific Differences

All the options that are valid for line-printer, full-screen, and graphics mode Gantt charts are explained in detail in the section "Syntax: GANTT Procedure" on page 496. With few exceptions, the options listed in the section "General Options" on page 503 have the same interpretation in all three modes.

Table 8.10 lists those line-printer options that have a different interpretation for the graphics version of PROC GANTT. Table 8.11 lists options specific for graphics charts and the equivalent line-printer/full-screen option. Table 8.12 lists options specific for line-printer and full-screen charts and the equivalent graphics option.

Table 8.10 Line-Printer Options and Corresponding Graphics Interpretation

Line-Printer Option	Graphics Mode Interpretation
SCALE=*scale*	One column is denoted by (1/*h*)% of the screen width, where HPOS=*h*.
SKIP=*skip*	*skip* number of bar heights are skipped between the bars for two consecutive activities. The value 0 is not valid in the graphics case.

Table 8.11 Graphics Mode Options and Line-Printer/Full-Screen Equivalent

Graphics Option/Statement	Line-Printer/Full-Screen Equivalent
LHCON=*linetype*	HCONCHAR='*character*'
LREF=*linetype*	REFCHAR='*character*'
LTNOW=*linetype*	TNCHAR='*character*'
NOFRAME	FORMCHAR='*string*'
PATTERN statement	JOINCHAR='*string*' and SYMCHAR='*string*'
SYMBOL statement	First character of variable name is plotted (See CHART specifications)
VMILE=*value*	MILECHAR='*character*'
WTNOW=*width*	TNCHAR='*character*'

Table 8.12 Line-Printer/Full-Screen Mode Specific Options

Line-Printer/Full-Screen Option	Graphics Equivalent
FORMCHAR='*string*'	NOFRAME
HCONCHAR='*character*'	LHCON=*linetype*, CHCON=*color*
HOLICHAR='*character*'	PATTERN statement 7
JOINCHAR='*string*'	PATTERN statements 1-6, 8, and 9
MILECHAR='*character*'	VMILE=*value*, FMILE=*font*, HMILE=*height*, CMILE=*color*
REFCHAR='*character*'	LREF=*linetype*, CREF=*color*
SYMCHAR='*string*'	PATTERN statements 1-6, 8, and 9
TNCHAR='*character*'	LTNOW=*linetype*, WTNOW=*width*, CTNOW=*color*

Displayed Output

The GANTT procedure produces one or more pages of displayed values and a plot of the schedule. If the SUMMARY option is specified, the chart is preceded by a detailed description of the symbols used. A legend is displayed at the foot of the chart on each page unless suppressed by the NOLEGEND option. The main body of the output consists of columns of the ID values and the Gantt chart of the schedule.

For each activity in the project, PROC GANTT displays the values of the ID variables in the ID columns and plots any combination of the following schedules: the predicted schedule as specified by the early and late start and finish times, the actual schedule as specified by the actual start and finish times, the resource-constrained schedule as specified by the resource-constrained start and finish times, and the baseline schedule as specified by the baseline start and finish times. The procedure looks for default variable names for each of these times (E_START for early start, E_FINISH for early finish, S_START for resource-constrained start times, and so on), or you can explicitly specify the names of the appropriate variables using the ES=, EF=, LS=, . . . options.

By specifying the COMBINE option in the CHART statement, you can request PROC GANTT to represent early, late, and actual schedule information on a single bar rather than use two separate bars (one for the early and late schedules and the other for the actual schedule.) PROC GANTT automatically draws a timenow line when the COMBINE option is specified with the property that all times to the left of the line represent the actual schedule times (that is, times that have already taken place) and all times to the right of the line represent the predicted early/late schedule times (times that have not yet taken place.)

Normally, each observation in the Schedule data set is assumed to denote a new activity, and a new set of ID values are displayed and the schedules corresponding to this activity are plotted on the chart. There are two exceptions to this rule:

- If the ID values for two or more consecutive observations are identical, only the first such observation is used.
- If there is a variable named SEGMT_NO in the Schedule data set, PROC GANTT assumes that the data set contains observations for segments of activities that were split during resource-constrained scheduling. In accordance with the conventions used by PROC CPM, only observations with a missing value for SEGMT_NO are assumed to denote a new activity. Further, the data are assumed to be sorted by SEGMT_NO for each activity. For each activity, PROC GANTT plots the schedules corresponding to the ES, EF, LS, LF, AS, and AF variables on the basis of the first observation for this activity, namely the observation with a missing value for the SEGMT_NO variable. This observation is also the one used for displaying values for the ID variables for this activity. If the activity is not split, this same observation is also the one used to plot the resource-constrained schedule as well as the baseline schedule. However, if the activity is split, then all the observations for this activity with integer values for the variable SEGMT_NO are used to plot the resource-constrained schedule as disjoint segments on the line used for plotting the S_START and S_FINISH times. Furthermore, PROC GANTT plots the baseline schedule corresponding to the BS and BF variables based on the last such observation, namely the observation with the largest value for the SEGMT_NO variable.

In addition to the schedules that are plotted, the Gantt chart also displays any variables specified in the CHART statement. Holidays, weekends, and breaks within a day are marked as appropriate. For details on how to specify holidays, weekends, and breaks within a day, see the section "Multiple Calendars and

Holidays" on page 534. You can also represent zero duration activities with milestone symbols, draw a timenow line to reflect the current time of the project, draw horizontal connect lines, draw vertical reference lines, and group the activities by zones on the Gantt chart. It is important to note that all times are plotted at the start of the appropriate time period. Thus, if the chart starts on June 1, 2004, in column 15 of the page and the value of E_START is '2JUN04,' MININTERVAL=DAY, and SCALE=5, then the early start time is plotted in column 20.

Each activity is identified by a job number (unless the NOJOBNUM option is used), which appears as the first column of activity text. The next column of activity text identifies the values of the ZONE= variable, if specified. This column can be suppressed by specifying the NOZONECOL option in the CHART statement. Next to appear are the ID variables in the order in which they are specified in the CHART statement. If the time axis of the chart is very wide, causing it to be divided across more than one page, the ID variables, by default, do not appear on continuation pages. You need to specify the IDPAGES option to produce the ID variable columns on every page. By default, if the ID variables occupy too much space, leaving no room for the chart to be started on the first page, they are omitted and a warning message is printed to the log. You can override this behavior by using the MAXIDS option. Column headings for the ZONE and ID variables consist of either variable labels (if they are present and if space permits) or variable names. To suppress variable labels in column headings, use the NOLABEL system option. If a ZONE or ID variable is formatted, the value is displayed using that format. If the CRITFLAG option is specified, a flag is displayed to the right of the ID values that indicates how critical the activity is. This flag is also repeated on continuation pages if the time axis occupies more than one page. The body of the chart starts to the right of this flag.

By default, the GANTT procedure is invoked in graphics mode. In graphics mode, you can fit the Gantt chart entirely on one page by specifying the COMPRESS option in the CHART statement. The HPAGES= and VPAGES= options take this one step further by enabling you to control the number of pages that you want the Gantt chart to be compressed into. The PCOMPRESS option behaves much like the COMPRESS option except that all compression is performed in a proportional manner, that is, by maintaining the aspect ratio of the Gantt chart.

PROC GANTT can display the precedence relationships (including nonstandard types) between activities on the Gantt chart by means of directed links between activities. Each link is drawn so as to convey the type of precedence relationship it represents. See the section "Specifying the Logic Options" on page 549 for a detailed description on how this can be done.

In addition, graphics mode provides you with the easy-to-use automatic text annotation facility to generate labels on the Gantt chart. You can link labels and their coordinates to variables in the schedule data set and also have complete control over all attributes such as font, color, angle, rotation, and so forth. You also have the additional capability of annotating text and graphics independently on the Gantt chart by using the SAS/GRAPH Annotate facility.

The GANTT procedure offers you a wide variety of options in addition to text, bar, symbol, and line formatting controls to customize your Gantt chart. These features enable you to create a wide variety of charts such as Logic Gantt charts, zoned Gantt charts, multiproject Gantt charts, Web-enabled Gantt charts, and multiprocess Gantt charts, to name but a few.

Macro Variable _ORGANTT

The GANTT procedure defines a macro variable named _ORGANTT, which is set at procedure termination. This variable contains a character string that indicates the status of the procedure and also provides chart

specific information with respect to each Gantt chart produced by invocation of the GANTT procedure. This includes charts resulting from multiple CHART statements and BY groups.

The format of the _ORGANTT string for a GANTT procedure invocation with *n* CHART statements is as follows:

STATUS= REASON= CHART1 chart1info # . . . CHART*n* chart*n*info #

where the value of STATUS= is either SUCCESSFUL or ERROR_EXIT, and the value of REASON= is one of the following:

- BADDATA_ERROR
- MEMORY_ERROR
- IO_ERROR
- SEMANTIC_ERROR
- SYNTAX_ERROR
- GANTT_BUG
- UNKNOWN_ERROR

The notation chart*i*info is a string of the form

SCALE= INCREMENT= SKIP= HPAGES= VPAGES= SEGNAME=

if there are no BY groups, and it is a string of the form

BY1 by1info: . . . BY*m* by*m*info:

where by*j*info is a string of the form

SCALE= INCREMENT= SKIP= HPAGES= VPAGES= SEGNAME=

if there are *m* BY groups. In other words, the macro contains an informational substring for every chart produced, using the symbol "#" as a CHART statement delimiter and the symbol ":" as a BY statement delimiter within CHART statements.

The chart specific information given in _ORGANTT is described below along with the identifying keyword preceding it. It should be noted that these values refer to those actually used in producing the chart and are not necessarily the same as those specified in the invocation of the procedure.

- SCALE= The value of scale
- INCREMENT= The value of increment
- SKIP= The value of skip
- HPAGES= The number of horizontal pages
- VPAGES= The number of vertical pages
- SEGNAME= The name of the first chart segment in graphics mode

NOTE: Some of the information might be redundant or predictable in certain display modes. For example, the value of SEGNAME= is empty in line-printer and full-screen modes. The values of HPAGES= and VPAGES= are equal to 1 in full-screen mode.

This information can be used when PROC GANTT is one step in a larger program that needs to determine whether the procedure terminated successfully or not. Because _ORGANTT is a standard SAS macro variable, it can be used in the ways that all macro variables can be used.

Computer Resource Requirements

There is no inherent limit on the size of the project that the GANTT procedure can accommodate. The number of activities in the Gantt chart is restricted only by the amount of memory available. Other memory-dependent factors are the type of Gantt chart required and the desired display mode.

Naturally, there needs to be a sufficient amount of core memory available in order to invoke and initialize the SAS System as well as to meet the memory requirements of the specific mode in which you invoke the procedure. For example, more memory is required when you use high-resolution graphics than when you use line-printer mode because the graphics sublibrary has to be loaded. The procedure attempts to store all the data in core memory. However, if there is insufficient core memory available for the entire project, the GANTT procedure resorts to using utility data sets and swaps between core memory and utility data sets as necessary.

The data storage requirement for the GANTT procedure is proportional to the number of activities in the project, and it depends on the number of schedule variables, the number of ID variables, and whether the Logic and Labeling options have been specified or not.

ODS Style Templates

ODS style templates, or *styles*, control the overall look of your output. An ODS style template consists of a set of *style elements*. A style element is a collection of *style attributes* that apply to a particular feature or aspect of the output. You can specify a value for each attribute in a style. See Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*), for a thorough discussion of ODS Graphics.

To create your own style or to modify a style for use with ODS Graphics, you need to understand the relationships between style elements and graph features. This information is provided in the ODS Graphics documentation at **`http://support.sas.com/documentation/onlinedoc/base/`**. You can create and modify style templates with the TEMPLATE procedure. For more information, see the section "TEMPLATE Procedure: Creating a Style Template" in the *SAS Output Delivery System: User's Guide*. Kuhfeld (2010) also offers detailed information and examples.

PROC GANTT Style Template

A predefined ODS style template named GANTT is available for the GANTT procedure. You can use the template to maintain a consistent appearance in all graphical output produced by the procedure.

To change the current style, specify the STYLE= option in an ODS destination statement. The specified style is applied to all output for that destination until you change or close the destination or start a new SAS session. For example, the following statement specifies that ODS should apply the GANTT style template to all HTML output:

```
ods html style=gantt;
```

To disable the use of graphical styles, specify the SAS system option NOGSTYLE.

The parent style template for the GANTT style is the DEFAULT style. Table 8.13 lists the style elements (in bold) and corresponding attributes specified in the GANTT style. The table also indicates which (if any) PROC GANTT options or graphics options (in a GOPTIONS statement) can be used to override the value of a style attribute.

Table 8.13 Style Elements and Attributes in the GANTT Style

Element/Attributes	Description	GANTT Option	GOPTION
GraphColors	Colors of various graph features	PATTERN=	CPATTERN=, COLORS=
gdata1	Duration of a noncritical activity	PATTERN=	CPATTERN=, COLORS=
gdata2	Slack time for a noncritical activity	PATTERN=	CPATTERN=, COLORS=
gdata3	Duration of a critical activity	PATTERN=	CPATTERN=, COLORS=
gdata4	Slack time for a supercritical activity	PATTERN=	CPATTERN=, COLORS=
gdata5	Duration of a supercritical activity	PATTERN=	CPATTERN=, COLORS=
gdata6	Actual duration of an activity	PATTERN=	CPATTERN=, COLORS=
gdata7	Break due to a holiday	PATTERN=	CPATTERN=, COLORS=
gdata8	Resource-constrained duration of an activity	PATTERN=	CPATTERN=, COLORS=
gdata9	Baseline duration of an activity	PATTERN=	CPATTERN=, COLORS=
gaxis	Axis	CAXIS=	COLORS=
ggrid	Horizontal connecting lines, zone lines	CHCON=, CZONE=	COLORS=
gdata	Default		COLORS=
gcdata	Precedence connections	CPREC=	COLORS=
greferencelines	Reference and timenow lines	CREF=, CTNOW=	COLORS=
gtextt	Title text		CTITLE=
gtext	Text		CTEXT=
glabel	Labels		COLORS=

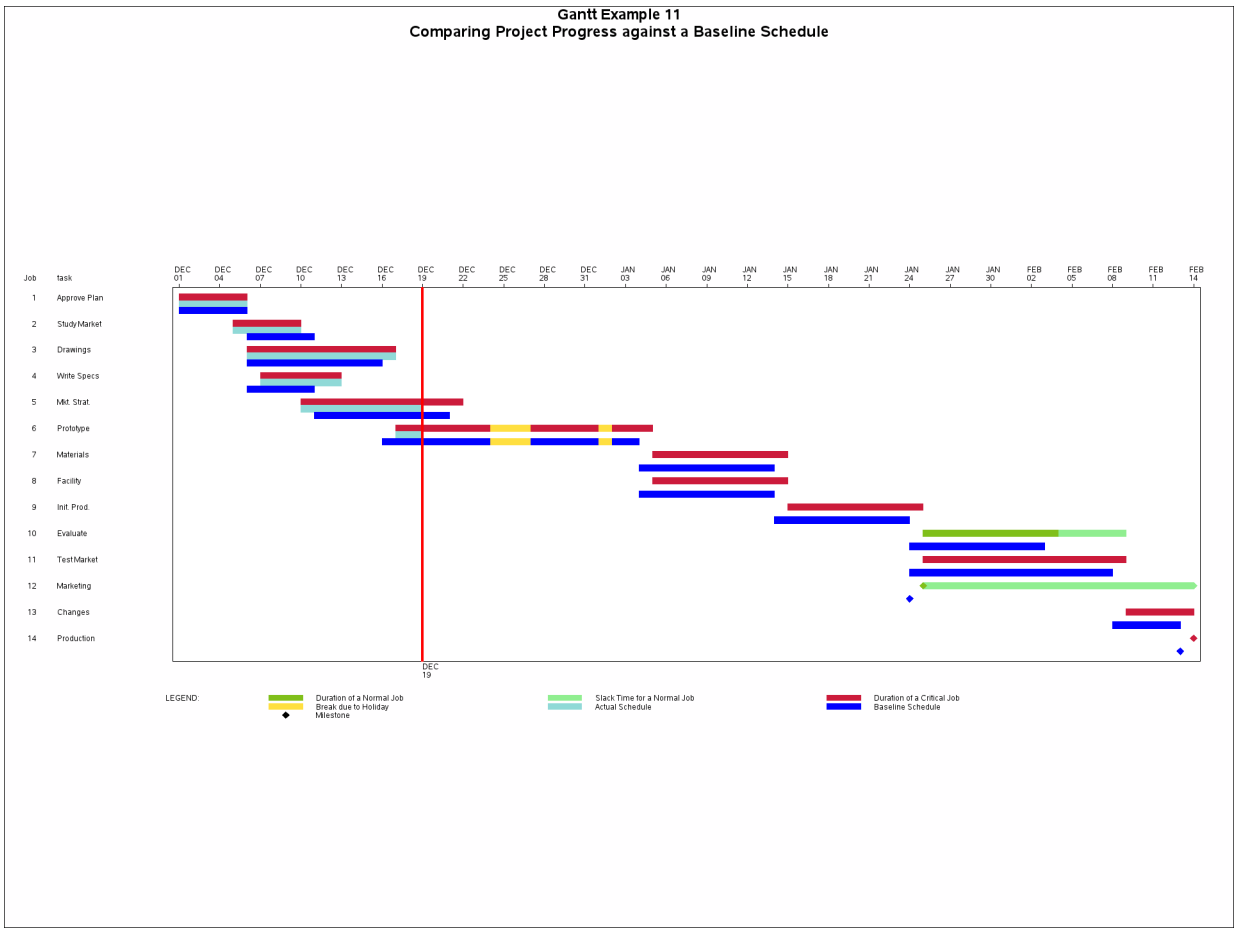
Table 8.13 (continued)

Element/Attributes	Description	GANTT Option	GOPTION
GraphFonts	Fonts for various graph features		
GraphDataFont	Default		FONT=
GraphLabelFont	Labels		FONT=
GraphTitleFont	Title text		FTITLE=
GraphAxisLines	Attributes related to graph axes		
Color	GraphColors('gaxis')	CAXIS=	COLORS=
GraphGridLines	Attributes related to horizontal connecting lines and zone lines		
Color	GraphColors('ggrid')	CHCON=, CZONE=	COLORS=
GraphConnectLine	Attributes related to precedence connections		
Color	GraphColors('gcdata')	CPREC=	COLORS=
GraphReference	Attributes related to reference and timenow lines		
Color	GraphColors('greferencelines')	CREF=, CTNOW=	COLORS=
GraphDataText	Attributes related to general text		
Color	GraphColors('gtext')	CTEXT=	COLORS=
Font	GraphFonts('GraphDataFont')	FONT=	FONT=
GraphTitleText	Attributes related to title text		
Color	GraphColors('gtextt')		CTITLE=
Font	GraphFonts('GraphTitleFont')		FTITLE=
GraphTitle1Text	Attributes related to the first title text		
Color	GraphColors('gtextt')		CTITLE=
Font	GraphFonts('GraphTitleFont')		FTITLE=
GraphLabelText	Attributes related to label text		
Color	GraphColors('glabel')	_CLABEL variable in the LABDATA= data set	COLORS=
Font	GraphFonts('GraphLabelFont')	_FLABEL variable in the LABDATA= data set	FONT=
GraphDataDefault	Default values for the attributes specified in Table 8.14		
Color	GraphColors('gdata')		COLORS=

Attributes that you do not override retain the values specified in the style template.

Figure 8.15 demonstrates features of the GANTT graphical style. The GANTT chart in the figure is the first output from Example 8.11.

Figure 8.15 GANTT Style Template: Example



Default Values

If the SAS system option GSTYLE is in effect (this is the default), then the default values of certain PROC GANTT options can depend on the current ODS style template. Table 8.14 lists these PROC GANTT options and lists the order in which PROC GANTT searches for each option's default value. The order assumes that the GSTYLE system option is in effect; if that is not the case, then the steps that refer to ODS style templates are ignored. Names with arguments indicate style elements and attributes of the current ODS style template. For example, "GraphAxisLines('Color')" refers to the Color attribute of the GraphAxisLines element.

Table 8.14 PROC GANTT Options: Search Orders for Default Values

Option	Search Order for Default Color
CAXIS=	1. The Color attribute of the GraphAxisLines element of the current ODS style template 2. The Color attribute of the GraphDataDefault element of the current ODS style template 3. The first color in the COLORS= list in the GOPTIONS statement

Table 8.14 (continued)

Option	Search Order for Default Color
CFRAME=	1. No color filling the axis area (if the GSTYLE system option is not in effect) 2. The Color attribute of the GraphWalls element of the current ODS style template 3. No color filling the axis area
CHCON=	1. The Color attribute of the GraphGridLines element of the current ODS style template 2. The Color attribute of the GraphDataDefault element of the current ODS style template 3. The first color in the COLORS= list in the GOPTIONS statement
CPREC=	1. The Color attribute of the GraphConnectLine element of the current ODS style template 2. The Color attribute of the GraphDataDefault element of the current ODS style template 3. The first color in the COLORS= list in the GOPTIONS statement
CREF=	1. The Color attribute of the GraphReference element of the current ODS style template 2. The Color attribute of the GraphDataDefault element of the current ODS style template 3. The first color in the COLORS= list in the GOPTIONS statement
CTEXT=	1. The value specified for the CTEXT= option in the GOPTIONS statement 2. The Color attribute of the GraphDataText element of the current ODS style template 3. The Color attribute of the GraphDataDefault element of the current ODS style template 4. The first color in the COLORS= list in the GOPTIONS statement
CTNOW=	1. The Color attribute of the GraphReference element of the current ODS style template 2. The Color attribute of the GraphDataDefault element of the current ODS style template 3. The first color in the COLORS= list in the GOPTIONS statement
CZONE=	1. The Color attribute of the GraphGridLines element of the current ODS style template 2. The Color attribute of the GraphDataDefault element of the current ODS style template 3. The first color in the COLORS= list in the GOPTIONS statement
FONT=	1. The value specified for the FTEXT= option in the GOPTIONS statement 2. The Font attribute of the GraphDataText element of the current ODS style template 3. The default hardware font for the graphics output device

Examples: GANTT Procedure

This section contains examples that illustrate several of the options and statements available with PROC GANTT in the different display modes. Example 8.1 and Example 8.2 illustrate the GANTT procedure in line-printer mode, and Example 8.3 through Example 8.27 illustrate the GANTT procedure in graphics mode.

Line-Printer Examples

Example 8.1 shows how to obtain a basic line-printer Gantt chart using the default options. Example 8.2 demonstrates how to use various options to customize the Gantt chart for the same project.

Example 8.1: Printing a Gantt Chart

This example shows how to use the GANTT procedure to obtain a basic line-printer Gantt chart using the default options. The following data describe the precedence relationships among the tasks involved in the construction of a typical floor in a multistory building. The first step saves the precedence relationships in a SAS data set. The variable ACTIVITY names each task, the variable DUR specifies the time it takes to complete the task in days, and the variables SUCCESS1 to SUCCESS4 specify tasks that are immediate successors to the task identified by the ACTIVITY variable.

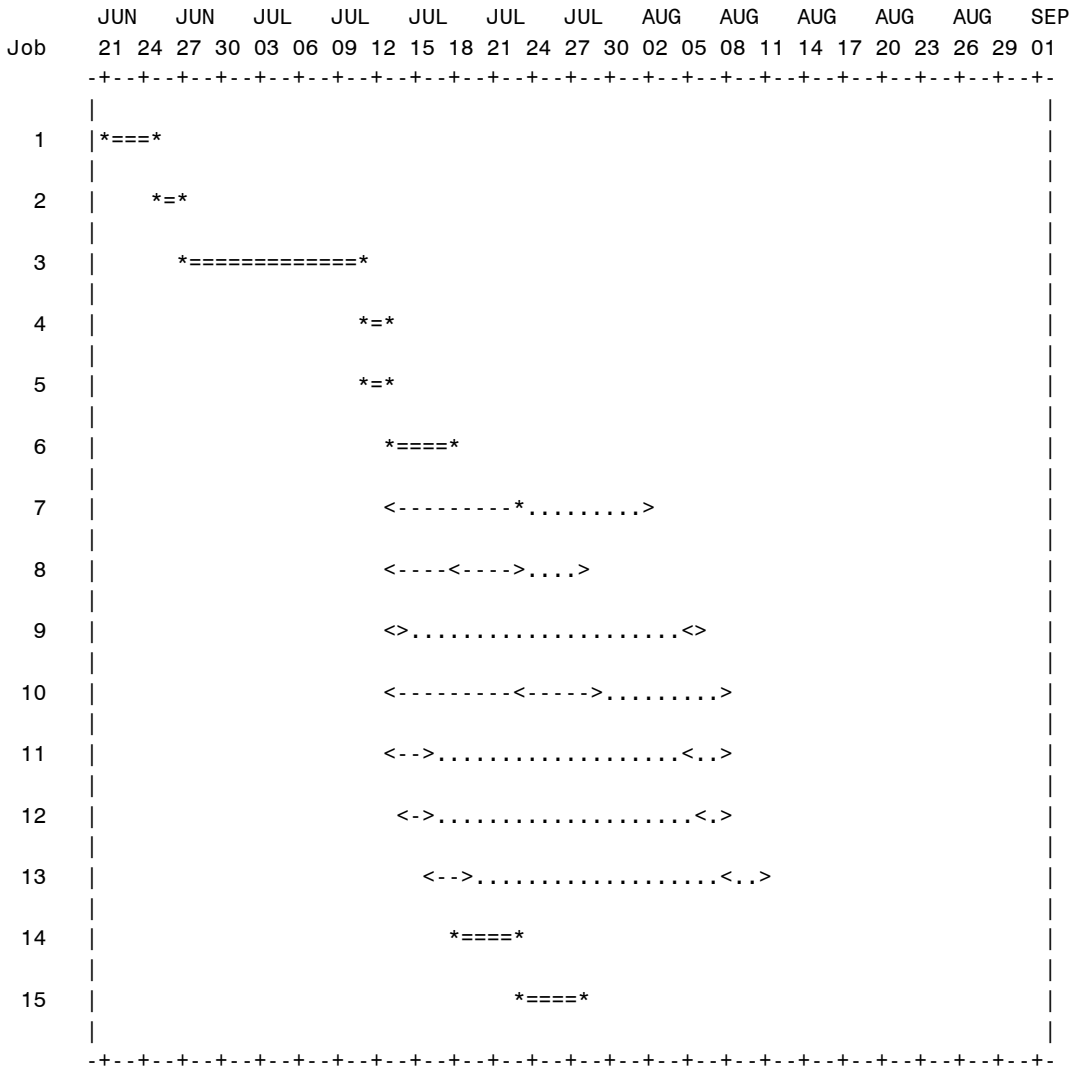
PROC CPM determines the shortest schedule for the project that finishes before September 1, 2004. The solution schedule, saved in a SAS data set, is next sorted by the early start time before invoking the GANTT procedure to plot the schedule. Since the DATA= option is not specified, PROC GANTT uses the sorted data set to produce the schedule since it is the most recently created data set. The Gantt chart in Output 8.1.1 is plotted on two pages because there are too many observations (29) to fit on one page. Note that the observations are split into two groups containing 15 and 14 observations, respectively, so that the chart size on each page is approximately equal. The time axis is labeled from June 21, 2004, to September 1, 2004, since these are the minimum and maximum dates in the Schedule data set. A legend is displayed at the bottom of the chart on each page.

```
title 'Gantt Example 1';
title2 'Printing a Gantt Chart';

data;
   format activity $20. success1 $20. success2 $20. success3 $20.
                        success4 $20.;
   input activity dur success1-success4;
   datalines;
form                 4 pour . . .
pour                 2 core . . .
core                14 strip spray_fireproof insulate_walls .
strip                2 plumbing curtain_wall risers doors
strip                2 electrical_walls balance_elevator . .
curtain_wall         5 glaze_sash . . .
glaze_sash           5 spray_fireproof insulate_walls . .
spray_fireproof      5 ceil_ducts_fixture . . .
```

```
ceil_ducts_fixture   5 test . . .
plumbing            10 test . . .
test                 3 insulate_mechanical . . .
insulate_mechanical  3 lath . . .
insulate_walls       5 lath . . .
risers              10 ceil_ducts_fixture . . .
doors                1 port_masonry . . .
port_masonry         2 lath finish_masonry . .
electrical_walls    16 lath . . .
balance_elevator     3 finish_masonry . . .
finish_masonry       3 plaster marble_work . .
lath                 3 plaster marble_work . .
plaster              5 floor_finish tiling acoustic_tiles .
marble_work          3 acoustic_tiles . . .
acoustic_tiles       5 paint finish_mechanical . .
tiling               3 paint finish_mechanical . .
floor_finish         5 paint finish_mechanical . .
paint                5 finish_paint . . .
finish_mechanical    5 finish_paint . . .
finish_paint         2 caulking_cleanup . . .
caulking_cleanup     4 finished . . .
;

* invoke cpm to find the optimal schedule;

proc cpm finishbefore date='1sep04'd;
   activity activity;
   duration dur;
   successors success1-success4;
run;

* sort the schedule by the early start date;

proc sort;
   by e_start;
run;

* invoke proc gantt to print the schedule;

proc gantt lineprinter;
run;
```

Output 8.1.1 Printing a Gantt Chart

Gantt Example 1
Printing a Gantt Chart

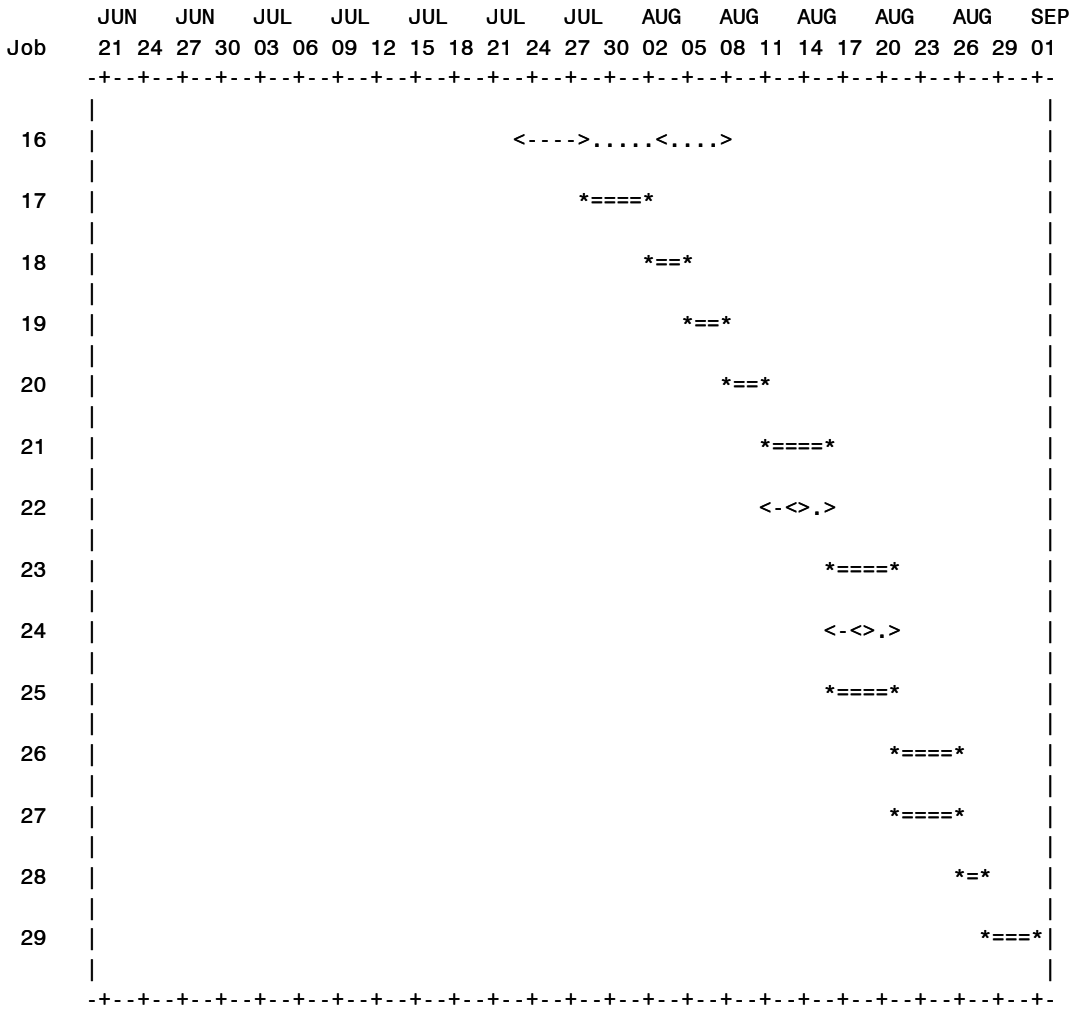


LEGEND

Symbol	Explanation
<--->	Duration of a Normal Job
>....>	Slack Time for a Normal Job
====	Duration of a Critical Job

Output 8.1.1 *continued*

Gantt Example 1
Printing a Gantt Chart



LEGEND

Symbol	Explanation
<---->	Duration of a Normal Job
>....>	Slack Time for a Normal Job
====	Duration of a Critical Job

Example 8.2: Customizing the Gantt Chart

This example shows how to control the format of the Gantt chart using CHART statement options. The Schedule data set used by PROC GANTT is the same as that used in Example 8.1. Output 8.2.1 is on three pages; the first page contains a detailed description of the various symbols used by the procedure to plot the schedule. This description is produced by using the SUMMARY option. The next two pages contain the Gantt chart. The LINEPRINTER option invokes the procedure in line-printer mode. The FILL option causes the first page to be filled as completely as possible before the second page is started. Thus, the first page of the chart contains 20 activities while the second page contains only 8 activities.

The MININTERVAL=WEEK specification defines the units of time for axis labeling. The SCALE=5 specification causes five columns of the chart to be used to display one week. The SKIP=2 specification causes two lines to be skipped between observations. The NOLEGEND option suppresses displaying of the legend, while the NOJOBNUM option causes job numbers to be omitted. The CRITFLAG option is used to produce the flag to the left of the main chart indicating if an activity is critical. Specifying BETWEEN=2 sets the number of columns between consecutive ID columns equal to 2. The REF= option produces the reference lines shown on the chart on the specified dates. The ID statement is used to display the activity names to the left of the chart. The ID statement also causes the activity 'strip' to appear only once in the chart. Thus, there are only 28 activities in this chart instead of 29, as in Example 8.1.

```
title 'Gantt Example 2';
title2 'Customizing the Gantt Chart';

proc gantt lineprinter;
   chart / summary
           fill
           mininterval=week scale=5
           skip=2
           nolegend
           nojobnum critflag between=2
           ref='10jun04'd to '30aug04'd by 15;
   id activity;
   run;
```

Output 8.2.1 Customizing the Gantt Chart

Gantt Example 2
Customizing the Gantt Chart

Summary

Symbols used for different times on the schedule

Variable	Symbol	Variable	Symbol
E_START	<	L_START	<
E_FINISH	>	L_FINISH	>

Miscellaneous Symbols

Symbol	Explanation
|	Reference Line
*	Overprint character when start or finish times coincide

Symbols used for joining start and/or finish times

Symbol	Explanation
-	Duration of non-critical job
.	Slack time for non-critical job
=	Duration of critical job
-	Slack time(neg.) for supercritical job
*	Duration of supercritical job

Output 8.2.1 *continued*

Gantt Example 2
Customizing the Gantt Chart

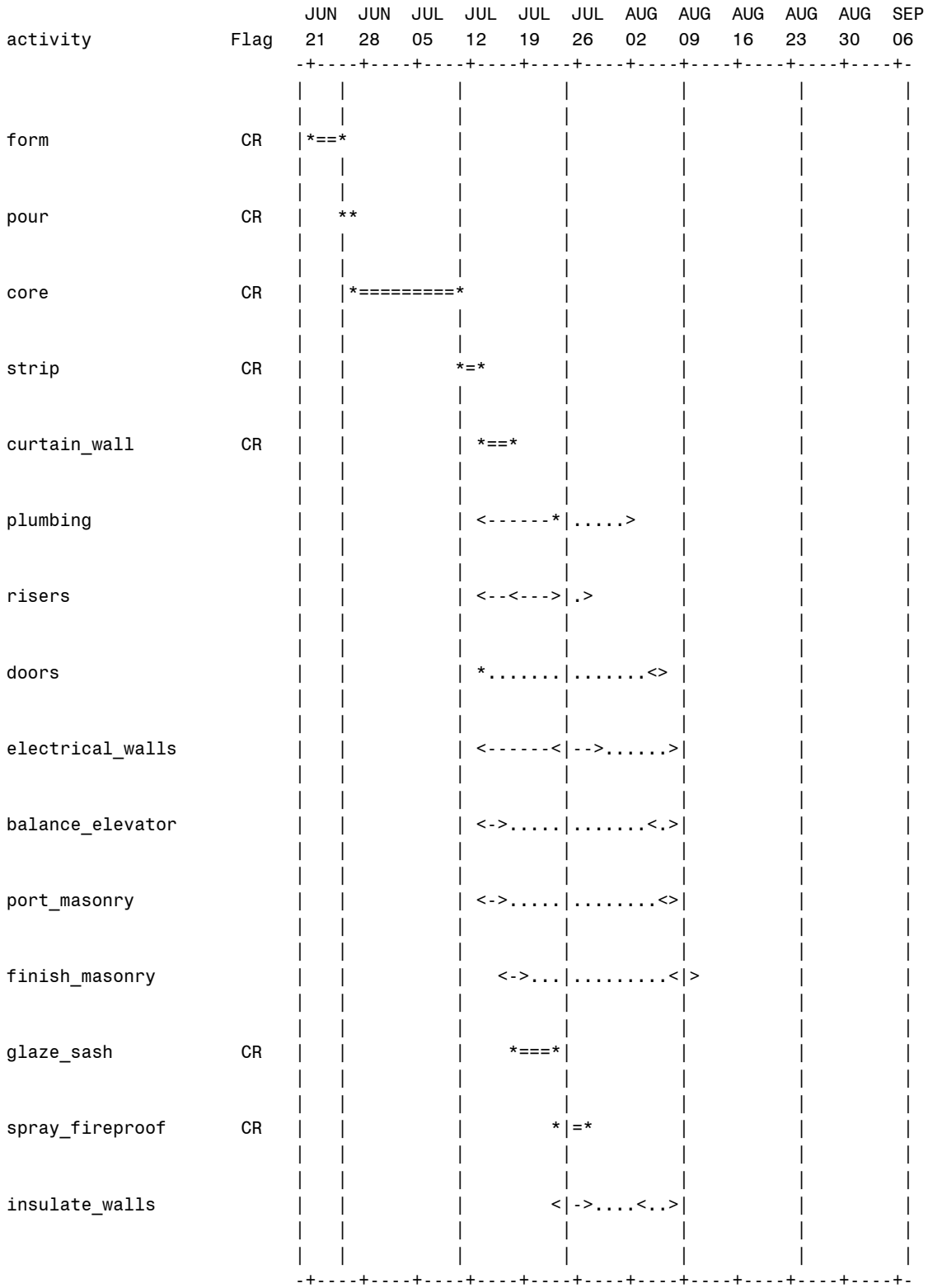
Summary (Contd.)

Some examples of typical strings

String	Description
<--->...<...>	Duration followed by slack time: early finish before late start
<---<--->...>	Duration followed by slack time: early finish after late start
<---*...>	Duration followed by slack time: early finish equals late start
===	Duration of job on critical path
<--->---<***>	Duration preceded by negative slack time for a supercritical job: late finish before early start
<---<***>***>	Duration preceded by negative slack time for a supercritical job: late finish after early start
<---****>	Duration preceded by negative slack time for a supercritical job: late finish equals early start

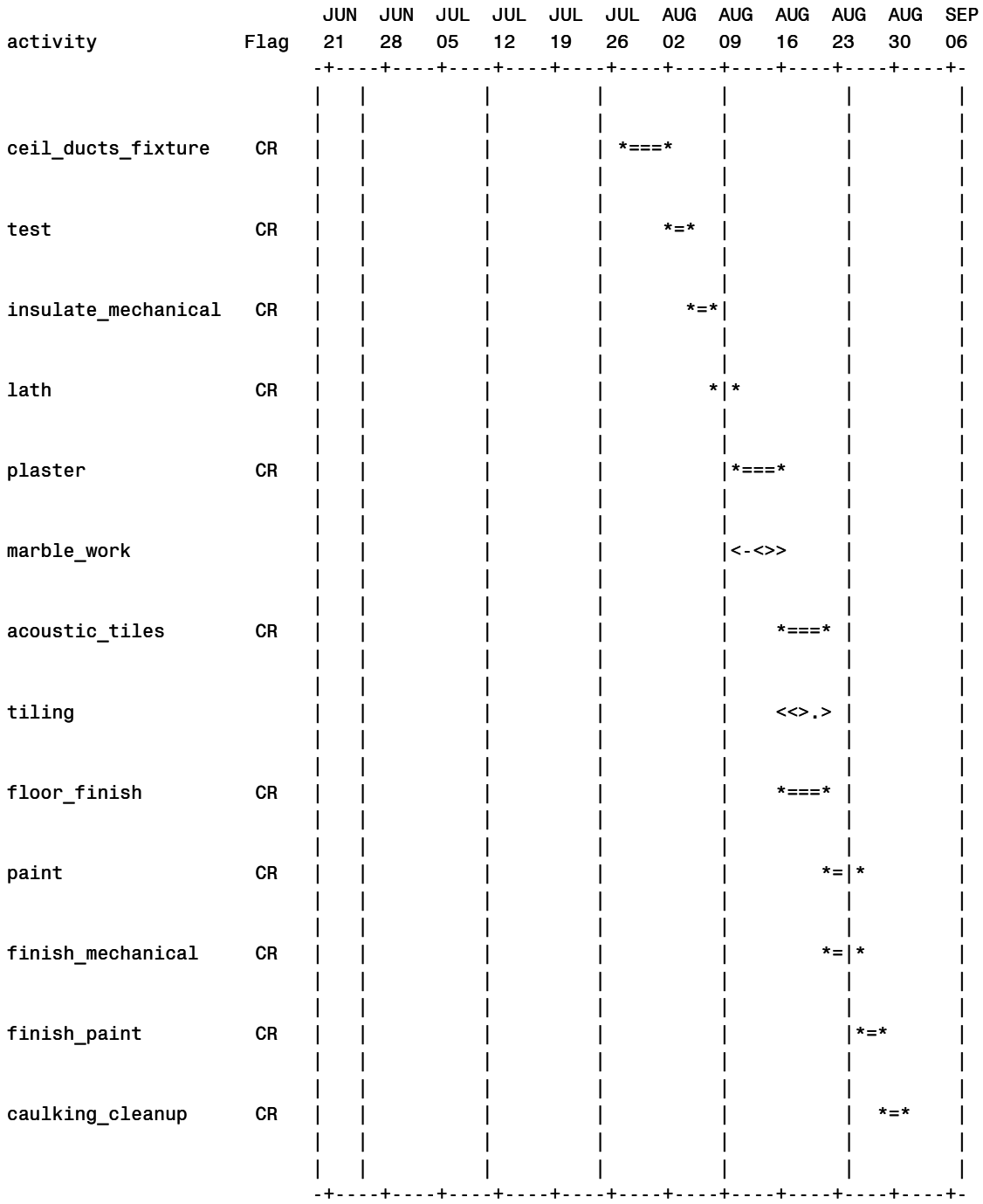
Output 8.2.1 *continued*

Gantt Example 2
Customizing the Gantt Chart



Output 8.2.1 *continued*

Gantt Example 2
Customizing the Gantt Chart



Graphics Examples

The following examples illustrate the use of graphics options and the use of PATTERN and SYMBOL statements to produce high resolution graphics quality Gantt charts. In Example 8.3, an extra input data set containing the holiday information is used to mark the holidays used in computing the schedule by PROC CPM. Example 8.4 illustrates the use of the CHART statement to specify milestones and additional variables to be plotted on the chart. Example 8.5 illustrates the use of the COMPRESS option to fit an entire Gantt chart on one page. Example 8.6 illustrates the use of the MININTERVAL= and SCALE= options to control the width of the chart; this example also shows how the chart is divided and continued on the succeeding page when the time axis extends beyond one page. In Example 8.7, the MINDATE= and MAXDATE= options are used to permit viewing of only a portion of the schedule in greater detail. Example 8.8 uses the HOLIDUR= option in conjunction with the INTERVAL= option to mark holidays of varying lengths on the Gantt chart. Example 8.9 illustrates the use of the CALENDAR and WORKDAY data sets to mark holiday information from different calendars on the chart.

In Example 8.10, the actual schedule for each activity is plotted on a separate line in addition to the early and late schedules. Example 8.11 illustrates tracking a project and comparing its progress against a baseline schedule. In Example 8.12, the COMBINE option is used to concatenate the early, late, and actual schedules of a project in progress to produce a single concise schedule that retains all of the vital information of the former schedules. Example 8.13 shows the resource-constrained schedule containing split segments of activities. The ability to bypass the project scheduler, PROC CPM, and directly specify the schedule information to PROC GANTT is demonstrated in Example 8.14. Example 8.15 illustrates the use of the BY statement to obtain Gantt charts for different projects in a multiproject environment. In Example 8.16, the GANTT procedure is used after some data manipulation steps to produce Gantt charts for individuals, each working on different subsets of activities in the project.

In Example 8.17, the HEIGHT= and HTOFF= options are used to modify the text height in relation to the height of the activity bars. The next three examples show you how to invoke the different logic options in order to draw a Logic Gantt chart that displays the precedence relationships between activities. Example 8.18 illustrates use of the ACTIVITY= and SUCCESSOR= options to specify the precedence information in AON format and the LEVEL= option to specify the bar type for the connections. In Example 8.19, the routing control options MAXDISLV=, MAXOFFGV=, MAXOFFLV=, and MININTGV= are used in connection with a project that is specified in AOA format using the TAIL= and HEAD= options in the CHART statement. Example 8.20 demonstrates the specification of nonstandard lag types using the LAG= option in the CHART statement. This example also illustrates use of the PRECDATA= option in the PROC GANTT statement. In Example 8.21, the ANNOTATE= option is used to add graphics and text on a Gantt chart. Example 8.22 illustrates the Automatic Text Annotation facility to label the Gantt chart independently of the SAS/GRAPH Annotate facility. In Example 8.23 a PATTERN variable and a Label data set are used to generate Gantt charts for multiprojects. A very useful chart in project management and multiprocess environments is the multisegment Gantt chart. Example 8.24 illustrates the use of the SEGMT_NO variable and the PATTERN variable to produce a versatile multisegment Gantt chart. In Example 8.25 the ZONE= option is used to produce a zoned Gantt chart. Example 8.26 shows you how to produce a "Web-enabled" Gantt chart that you can use to drill-down your project. Finally, Example 8.27 uses the CHARTWIDTH= option to produce Gantt charts that are consistent in appearance.

In all the examples presented, the early and late schedules are specified in the data set by means of the variables E_START, E_FINISH, L_START, and L_FINISH; hence, the ES=, EF=, LS=, and LF= options are not needed in the CHART statement. Unless otherwise specified, the pattern statements used in the examples are as follows:

```
pattern1 c=green v=s;    /* duration of a non-critical activity  */
pattern2 c=green v=e;    /* slack time for a noncrit. activity   */
pattern3 c=red v=s;      /* duration of a critical activity      */
pattern4 c=magenta v=e;  /* slack time for a supercrit. activity */
pattern5 c=magenta v=s;  /* duration of a supercrit. activity    */
pattern6 c=cyan v=s;     /* actual duration of an activity       */
pattern7 c=black v=e;    /* break due to a holiday               */
pattern8 c=blue v=s;     /* resource schedule of activity        */
pattern9 c=brown v=s;    /* baseline schedule of activity        */
```

Example 8.3: Marking Holidays

This example uses the widget manufacturing project introduced in Chapter 4, "The CPM Procedure." The data sets used in this example are the same as those used in Example 4.8 to illustrate holiday processing in PROC CPM. The WIDGET data set describes the project in AON format. The variable TASK identifies the activity and the variables SUCC1, SUCC2, and SUCC3 identify the successors to TASK. The variable DAYS defines the duration of an activity. Another data set, HOLIDAYS, defines the holidays that need to be taken into account when scheduling the project. Although the HOLIDAYS data set contains three variables HOLIDAY, HOLIFIN, and HOLIDUR, the HOLIDUR variable is not used in this example. Thus, the Christmas holiday starts on December 24, 2003, and finishes on December 26, 2003. PROC CPM schedules the project to start on December 1, 2003, and saves the schedule in a data set named SAVEH. This data set is shown in Output 8.3.1.

Next, the GANTT procedure is invoked with the specification of HOLIDATA= HOLIDAYS in the PROC GANTT statement and the HOLIDAY= and HOLIEND= options in the CHART statement, causing the Christmas and New Year holidays to be marked on the chart. The resulting Gantt chart is shown in Output 8.3.2. Note that the procedure marks the duration of the holiday with the pattern corresponding to the seventh PATTERN statement. (See the section "Graphics Examples" on page 579 for a list of the pattern statements used in the examples.)

```
options ps=60 ls=80;

title h=2 'Gantt Example 3';
title2 'Marking Holidays';

/* Activity-on-Node representation of the project */
data widget;
   format task $12. succ1-succ3 $12. ;
   input task & days succ1 & succ2 & succ3 & ;
   datalines;
Approve Plan   5  Drawings      Study Market  Write Specs
Drawings      10  Prototype     .             .
Study Market   5  Mkt. Strat.   .             .
Write Specs    5  Prototype     .             .
Prototype     15  Materials     Facility      .
Mkt. Strat.   10  Test Market   Marketing     .
Materials     10  Init. Prod.   .             .
Facility      10  Init. Prod.   .             .
Init. Prod.   10  Test Market   Marketing     Evaluate
Evaluate      10  Changes       .             .
Test Market   15  Changes       .             .
Changes        5  Production    .             .
Production     0  .             .             .
Marketing      0  .             .             .
;

data holidays;
   format holiday holifin date7.;
   input holiday & date7. holifin & date7. holidur;
   datalines;
24dec03  26dec03  4
01jan04  .        .
;

* schedule the project subject to holidays;
proc cpm data=widget holidata=holidays
         out=saveh date='1dec03'd ;
   activity task;
   succ     succ1 succ2 succ3;
   duration days;
   holiday  holiday / holifin=(holifin);
   run;

* sort the schedule by the early start date ;
proc sort;
   by e_start;
   run;

* print the schedule;
proc print data=saveh;
   var task days e_start e_finish l_start l_finish
       t_float f_float;
   run;
```

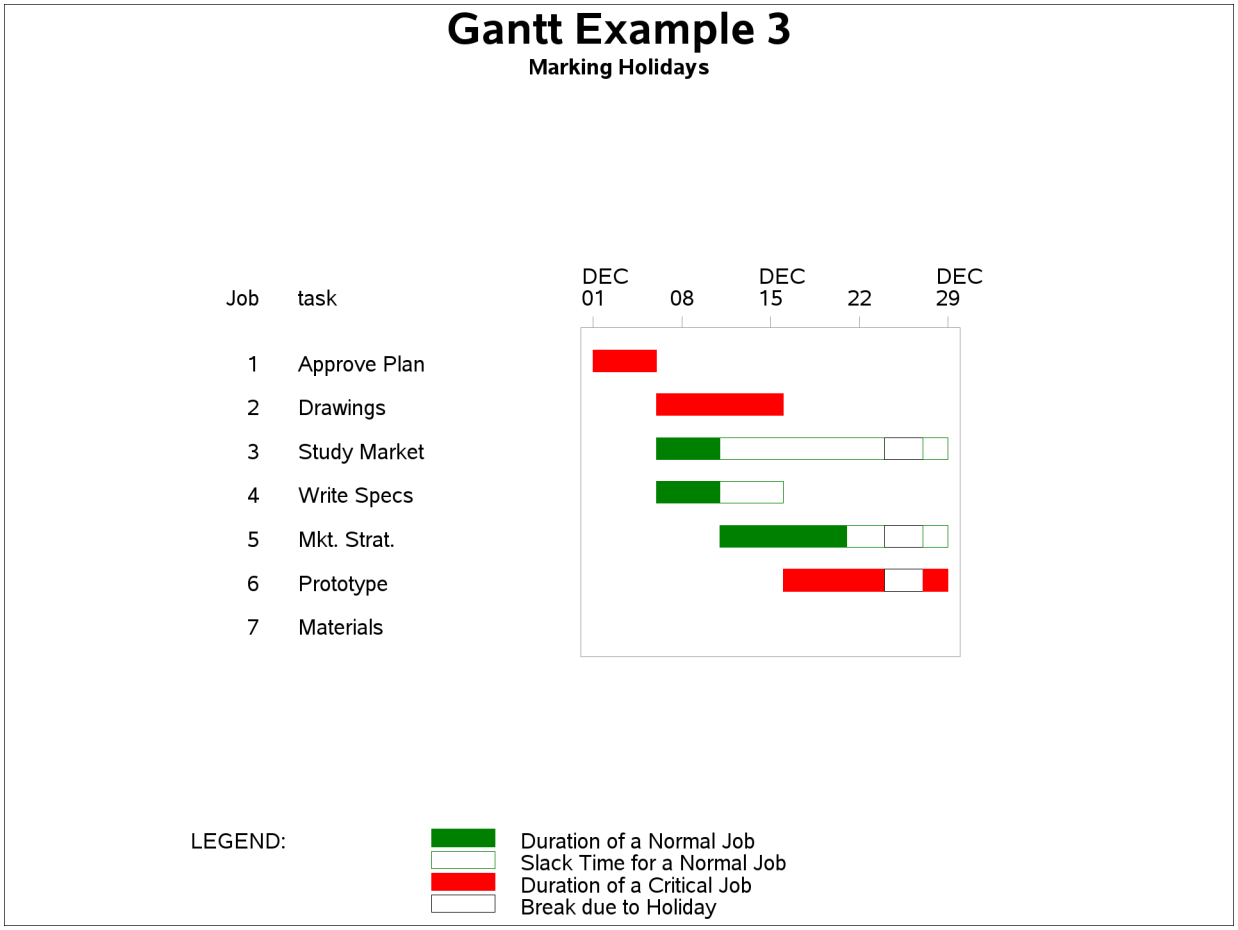
Output 8.3.1 Schedule Data Set SAVEH

Gantt Example 3
Marking Holidays

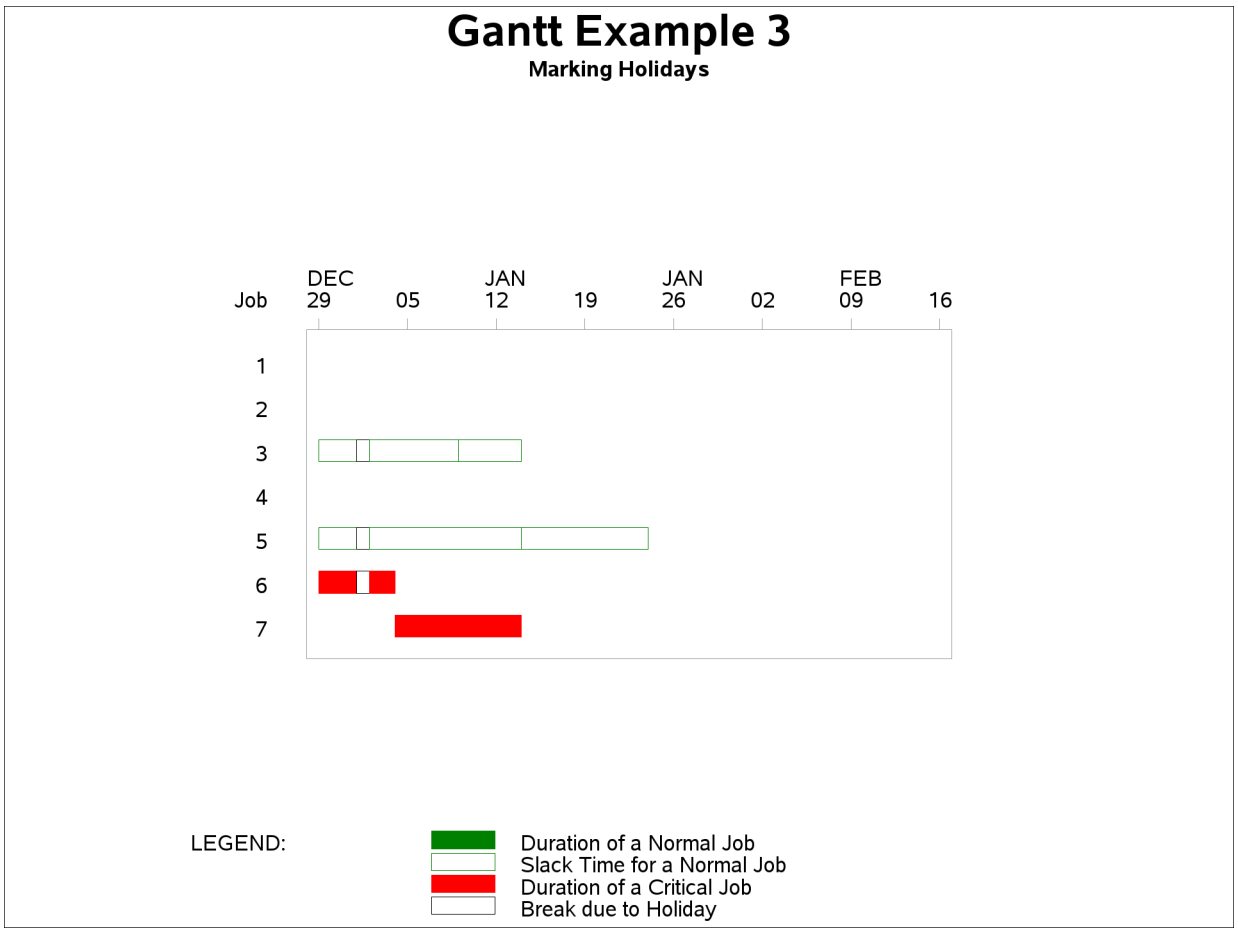
Obs	task	days	E_START	E_FINISH	L_START	L_FINISH	T_FLOAT	F_FLOAT
1	Approve Plan	5	01DEC03	05DEC03	01DEC03	05DEC03	0	0
2	Drawings	10	06DEC03	15DEC03	06DEC03	15DEC03	0	0
3	Study Market	5	06DEC03	10DEC03	09JAN04	13JAN04	30	0
4	Write Specs	5	06DEC03	10DEC03	11DEC03	15DEC03	5	5
5	Mkt. Strat.	10	11DEC03	20DEC03	14JAN04	23JAN04	30	30
6	Prototype	15	16DEC03	03JAN04	16DEC03	03JAN04	0	0
7	Materials	10	04JAN04	13JAN04	04JAN04	13JAN04	0	0
8	Facility	10	04JAN04	13JAN04	04JAN04	13JAN04	0	0
9	Init. Prod.	10	14JAN04	23JAN04	14JAN04	23JAN04	0	0
10	Evaluate	10	24JAN04	02FEB04	29JAN04	07FEB04	5	5
11	Test Market	15	24JAN04	07FEB04	24JAN04	07FEB04	0	0
12	Marketing	0	24JAN04	24JAN04	13FEB04	13FEB04	20	20
13	Changes	5	08FEB04	12FEB04	08FEB04	12FEB04	0	0
14	Production	0	13FEB04	13FEB04	13FEB04	13FEB04	0	0

```
* plot the schedule;
proc gantt holidata=holidays data=saveh;
   chart / holiday=(holiday) holiend=(holifin);
   id task;
   run;
```

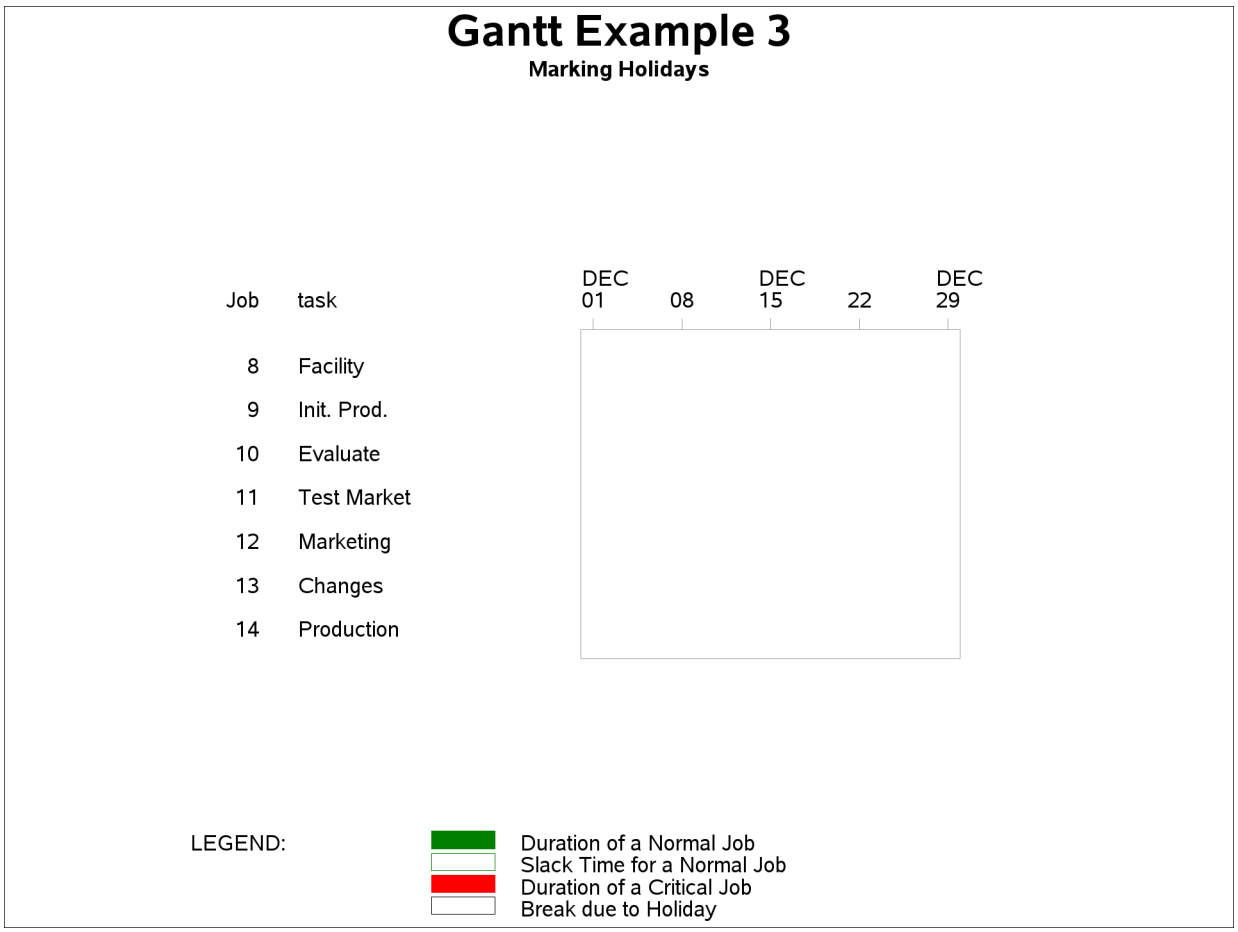
Output 8.3.2 Marking Holidays on the Gantt Chart



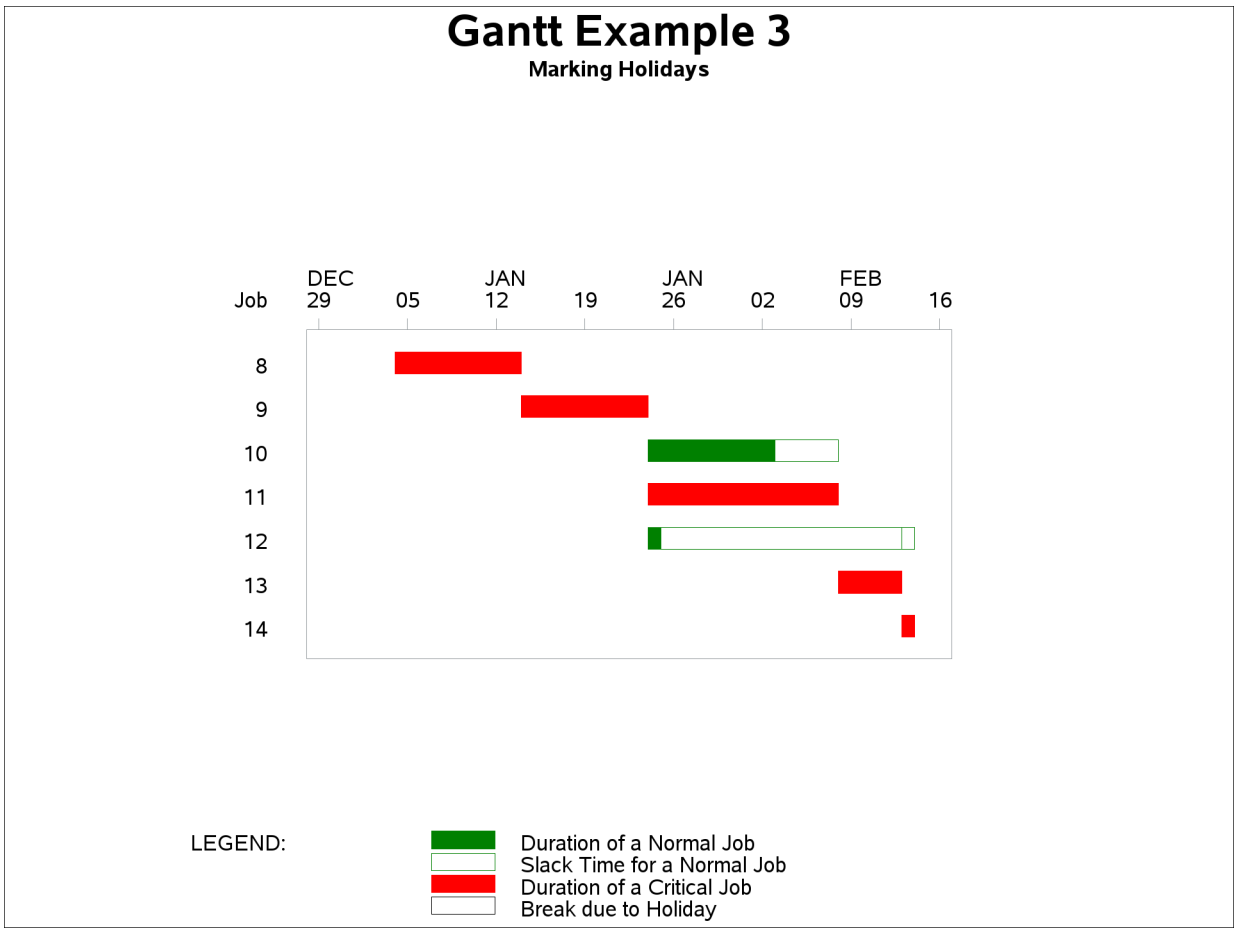
Output 8.3.2 *continued*



Output 8.3.2 *continued*



Output 8.3.2 *continued*



Example 8.4: Marking Milestones and Special Dates

The widget manufacturing project described in Example 8.3 has two activities with zero duration, namely 'Production' and 'Marketing.' By default, PROC GANTT pads finish times by a padding unit, thus these two activities are represented on the Gantt chart as having a duration equal to one day (see the section "Specifying the PADDING= Option" on page 532 for further information on padding). In other words, based on start and finish times alone, PROC GANTT cannot distinguish between activities that are one day or zero days long; it needs knowledge of the activity duration variable, which is specified using the DUR= option in the CHART statement, in order to represent zero duration activities by a milestone symbol.

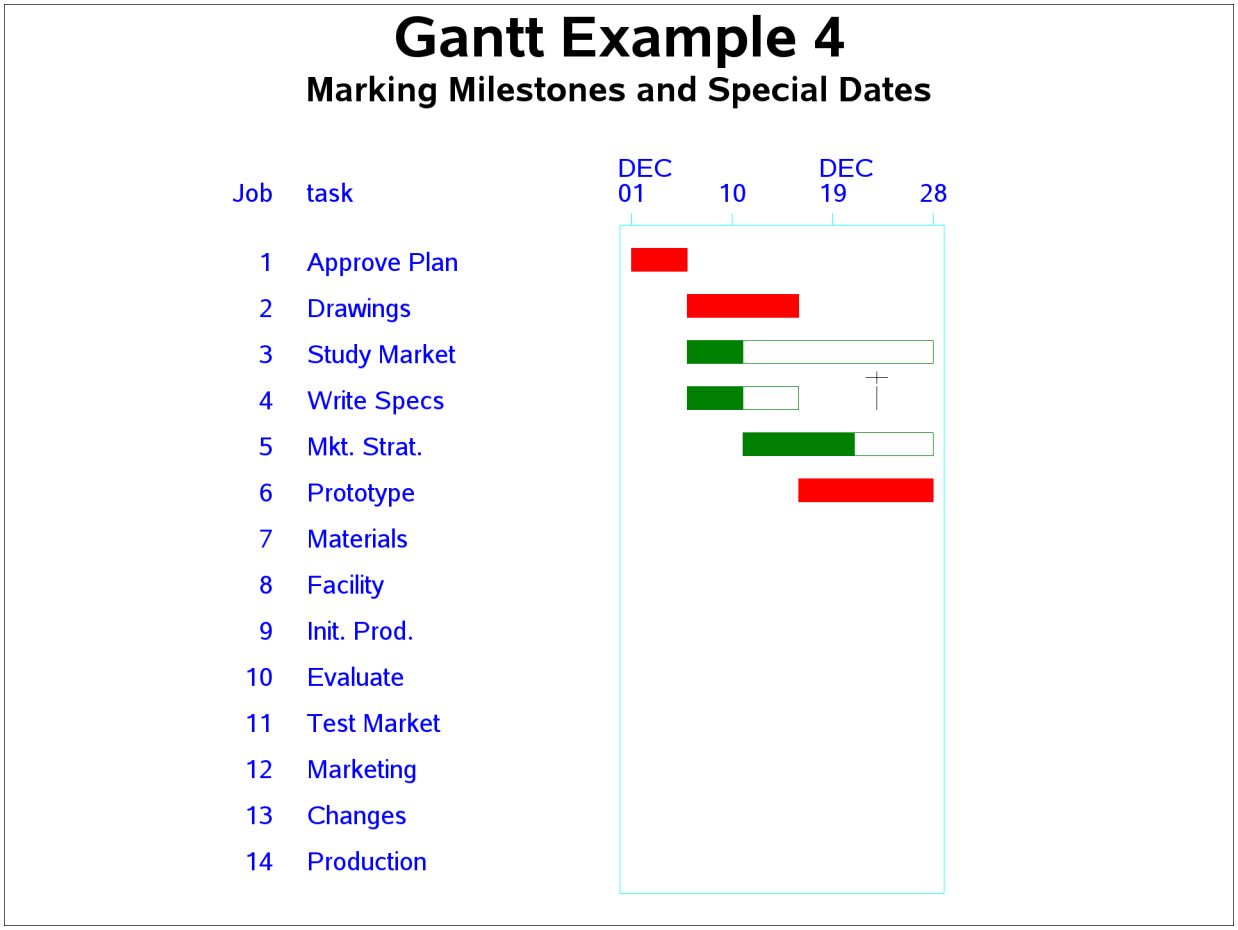
Now, suppose that the Engineering department would like to finish writing up the specifications before Christmas and have the prototype ready by mid-January. In addition, the Marketing department would like to develop a marketing concept by the year's end. The data set, TARGET, contains the target dates for these activities. This data set is merged with the WIDGET data set to produce the WIDGETT data set. The WIDGETT data set is then input to the CPM procedure, which is invoked with an ID statement to ensure that the variable TARGET is passed to the Schedule data set. After sorting the Schedule data set by the early start time, PROC GANTT is used to produce a Gantt chart of the resulting schedule. The Gantt chart is shown in Output 8.4.1.

Before invoking PROC GANTT, you specify the required symbol using a SYMBOL statement. Specifying the variable TARGET in the CHART statement causes target dates to be marked on the chart with the symbol specified in the SYMBOL statement, a PLUS symbol in black. Specifying the DUR= option in the CHART statement causes zero duration schedules to be represented on the chart by the default milestone symbol, a filled diamond. To use a different milestone symbol, use the FMILE= and VMILE= options in the CHART statement. The duration and slack time of the activities are indicated by the use of the appropriate fill patterns as explained in the legend.

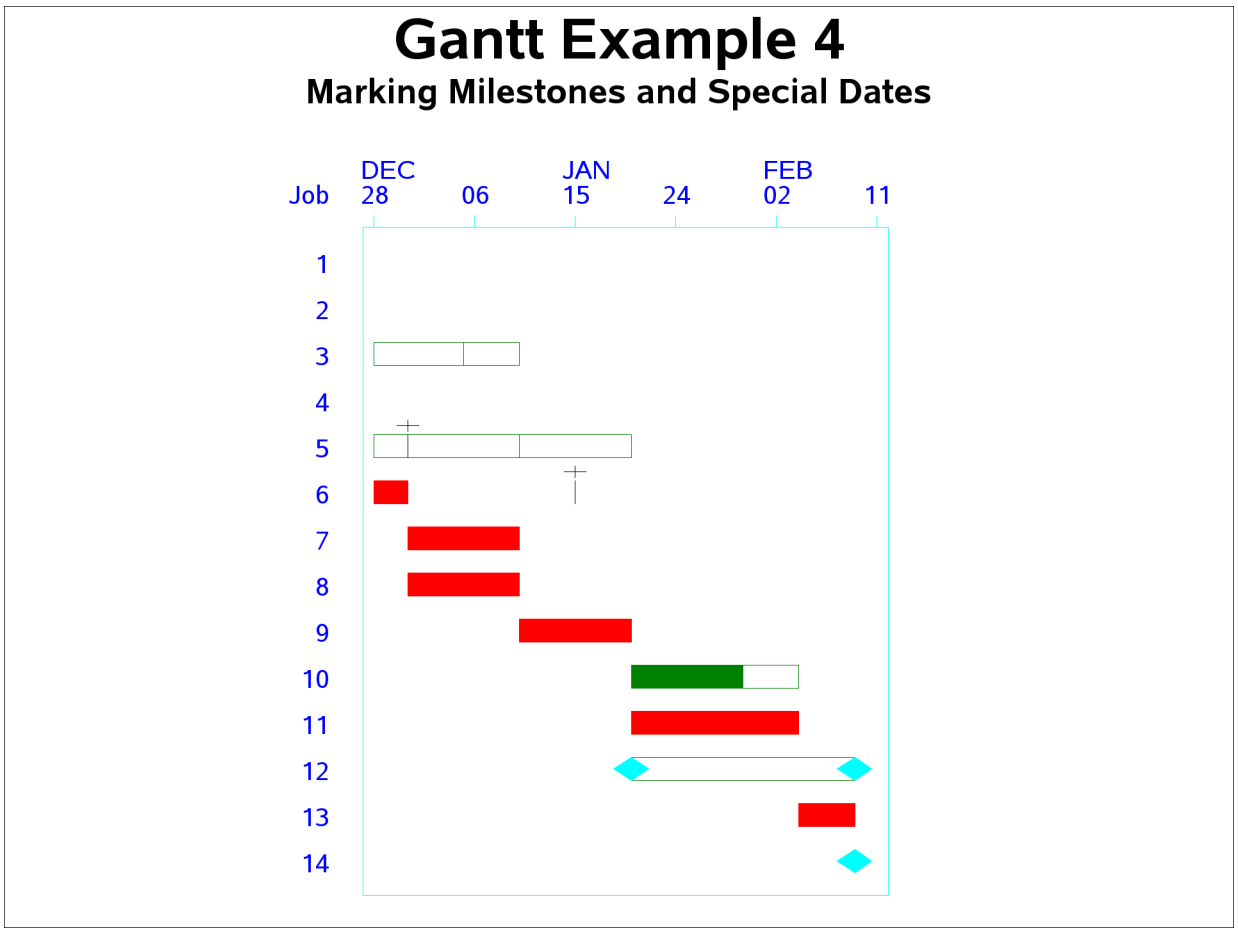
Colors for the milestone, axis, and text are specified using the options CMILE=, CAXIS=, and CTEXT=, respectively.

```
options ps=60 ls=100;

title h=2.5 'Gantt Example 4';
title2 h=1.5 'Marking Milestones and Special Dates';

proc cpm data=widgett date='1dec03'd;
   activity task;
   successor succ1-succ3;
   duration days;
   id target;
   run;

* sort the schedule by the early start date ;
proc sort;
   by e_start;
   run;

goptions htext=1.1 hpos=110 vpos=40;

* set up required pattern and symbol statements;
pattern1 c=green v=s;
pattern2 c=green v=e;
pattern3 c=red   v=s;
symbol   c=black v=plus;

* plot the schedule;
proc gantt;
   chart target / dur=days cmile=cyan
                  ctext=blue caxis=cyan;
   id task;
   run;
```

Output 8.4.1 Marking Milestones and Special Dates in Graphics Mode



Output 8.4.1 *continued*

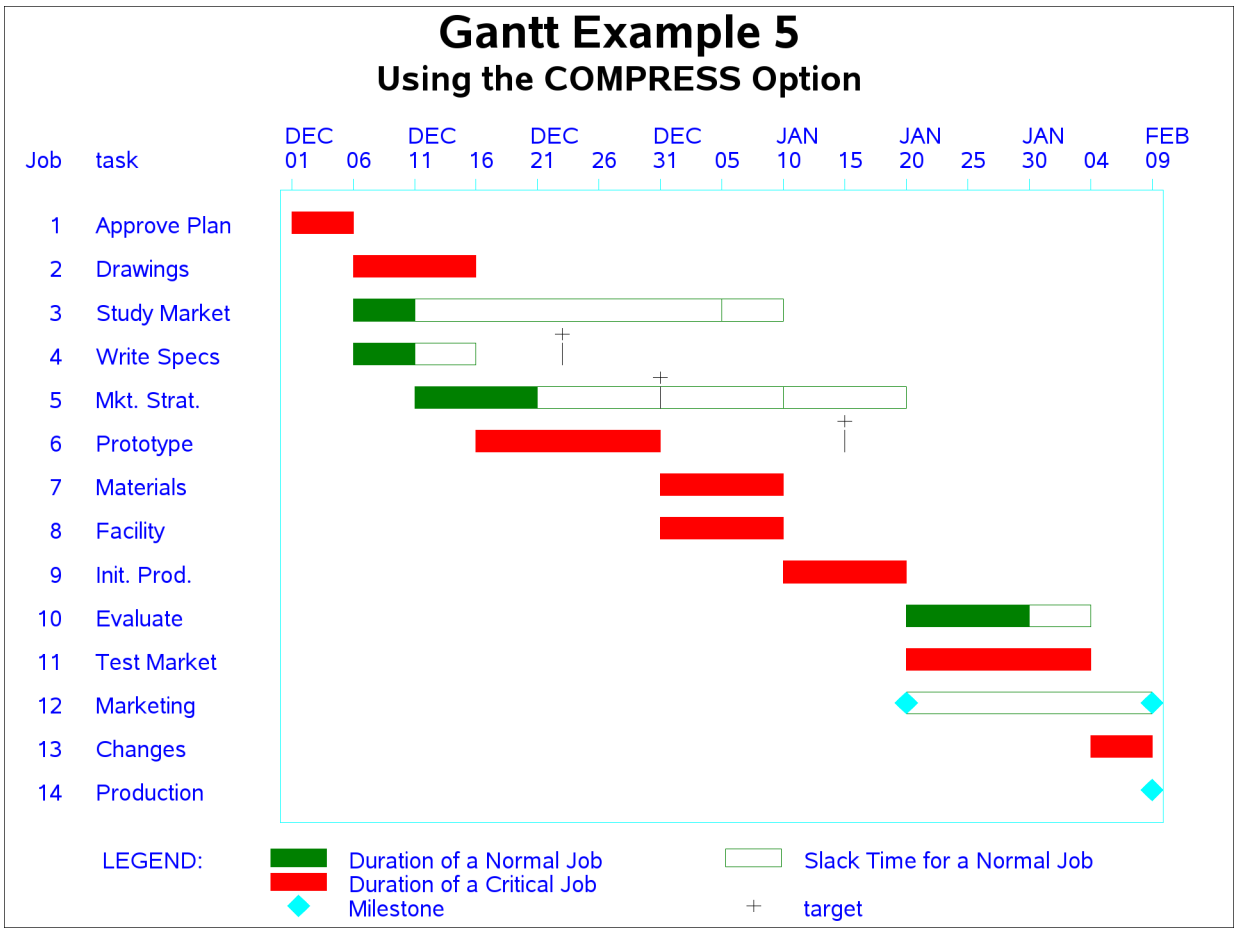


Example 8.5: Using the COMPRESS Option

In the previous example, PROC GANTT produced two pages of output since the chart would not fit on a single page. One way to ensure that the entire chart fits on a single page in graphics mode is to adjust the values of HPOS and VPOS accordingly. An easier way that is independent of the values of HPOS and VPOS is to specify the COMPRESS option in the CHART statement. Output 8.5.1 shows the result of adding the COMPRESS option to the CHART statement in Example 8.4. The PCOMPRESS option would have a similar effect but would also maintain the aspect ratio. Some other options that can be used to control the number of pages generated are the HPAGES= and VPAGES= options.

```
title h=2 'Gantt Example 5';
title2 h=1.5 'Using the COMPRESS Option';

* plot the schedule on one page;
proc gantt;
   chart target / dur=days cmile=cyan
                  ctext=blue caxis=cyan
                  compress;
   id task;
   run;
```

Output 8.5.1 Using the COMPRESS Option



Example 8.6: Using the MININTERVAL= and SCALE= Options

The data sets used for this example are the same as those used to illustrate PROC CPM in Example 4.2. The data set WIDGAOA defines the project using the AOA specification. The data set DETAILS specifies the abbreviated and detailed names for each of the activities in addition to the name of the department that is responsible for that activity. Notice that a dummy activity has been added to the project in order to maintain the precedence relationships established by the WIDGET data set of the previous two examples that define the same project in AON format. The two data sets WIDGAOA and DETAILS are merged to form the WIDGETA data set that is input as the Activity data set to PROC CPM. The data set SAVE produced by PROC CPM and sorted by E_START is shown in Output 8.6.1.

Because MININTERVAL=WEEK and SCALE=10, PROC GANTT uses (1000/*h*)% of the screen width to denote one week, where *h* is the value of HPOS. Note that this choice also causes the chart to become too wide to fit on one page. Thus, PROC GANTT splits the chart into two pages. The first page contains the ID variable as well as the job number while the second page contains only the job number. The chart is split so that the displayed area on each page is approximately equal.

The milestone color is changed to green using the CMILE= option. The resulting Gantt chart is shown in Output 8.6.2.

```
options ps=60 ls=80;

title h=2 'Gantt Example 6';
title2 h=1.5 'Using the MININTERVAL= and SCALE= Options';

data widgaoa;
   format task $12.;
   input task & days tail head;
   datalines;
Approve Plan   5   1   2
Drawings      10   2   3
Study Market   5   2   4
Write Specs    5   2   3
Prototype     15   3   5
Mkt. Strat.   10   4   6
Materials     10   5   7
Facility      10   5   7
Init. Prod.   10   7   8
Evaluate      10   8   9
Test Market   15   6   9
Changes        5   9  10
Production     0  10  11
Marketing      0   6  12
Dummy          0   8   6
;

data details;
   format task $12. dept $13. descrpt $30.;
   input task & dept & descrpt & ;
   label dept = "Department"
         descrpt = "Activity Description";
   datalines;
Approve Plan  Planning       Finalize and Approve Plan
Drawings      Engineering    Prepare Drawings
Study Market  Marketing      Analyze Potential Markets
Write Specs   Engineering    Write Specifications
Prototype     Engineering    Build Prototype
Mkt. Strat.   Marketing      Develop Marketing Concept
Materials     Manufacturing  Procure Raw Materials
Facility      Manufacturing  Prepare Manufacturing Facility
Init. Prod.   Manufacturing  Initial Production Run
Evaluate      Testing        Evaluate Product In-House
Test Market   Testing        Mail Product to Sample Market
Changes       Engineering    Engineering Changes
Production    Manufacturing  Begin Full Scale Production
Marketing     Marketing      Begin Full Scale Marketing
Dummy         .              Production Milestone
;

data widgeta;
   merge widgaoa details;
   run;
```

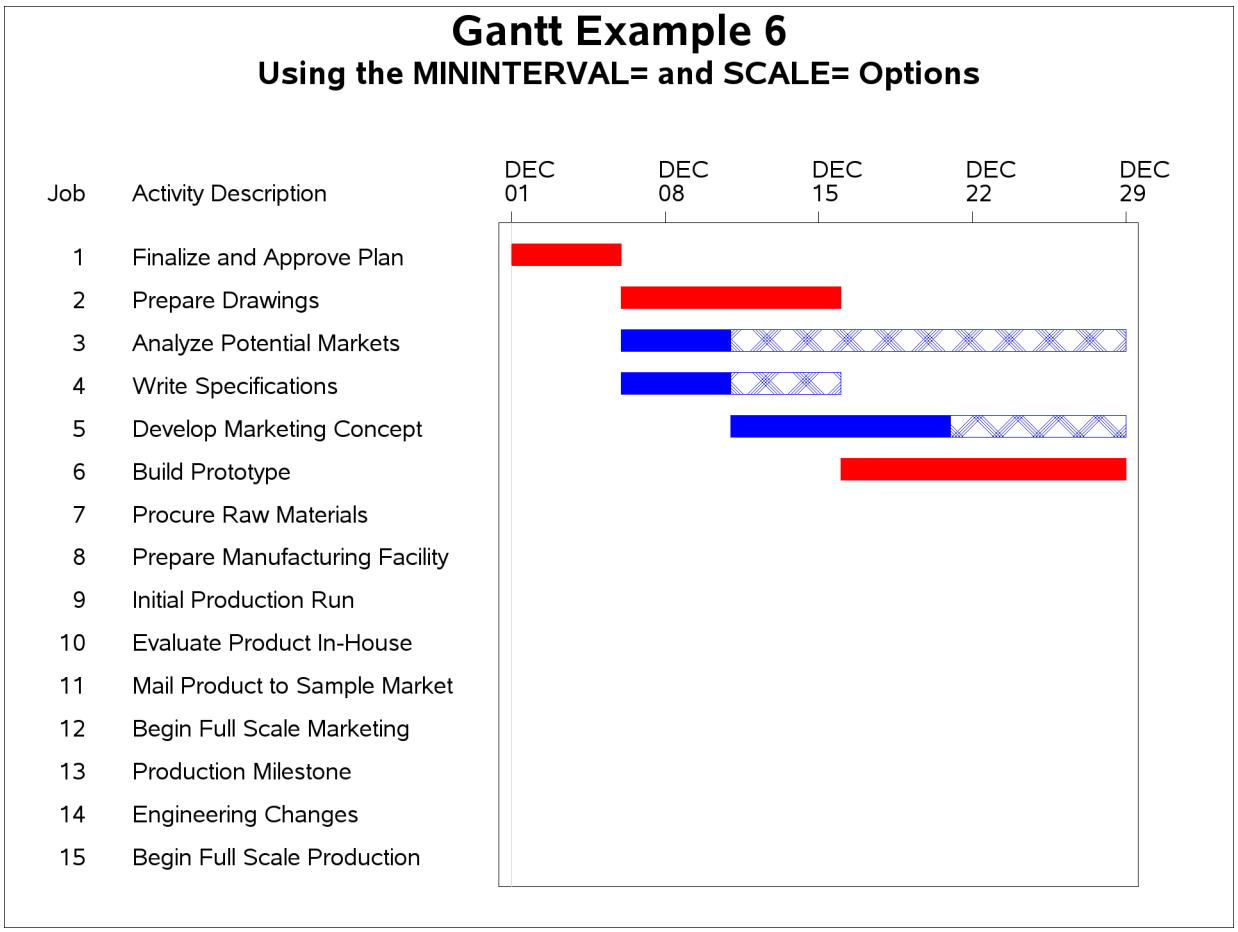
```
* schedule the project;
proc cpm data=widgeta date='1dec03'd out=save;
   tailnode tail;
   headnode head;
   duration days;
   id task dept descrpt;
   run;

* sort the schedule by the early start date ;
proc sort;
   by e_start;
   run;

goptions vpos=43 hpos=80;

* plot the schedule;
proc gantt graphics;
   chart / mininterval=week scale=10 dur=days
           cmile=green nolegend caxis=black
           ref='1dec03'd to '1feb04'd by month;
   id descrpt;
   run;
```

Output 8.6.1 Schedule Data Set SAVE

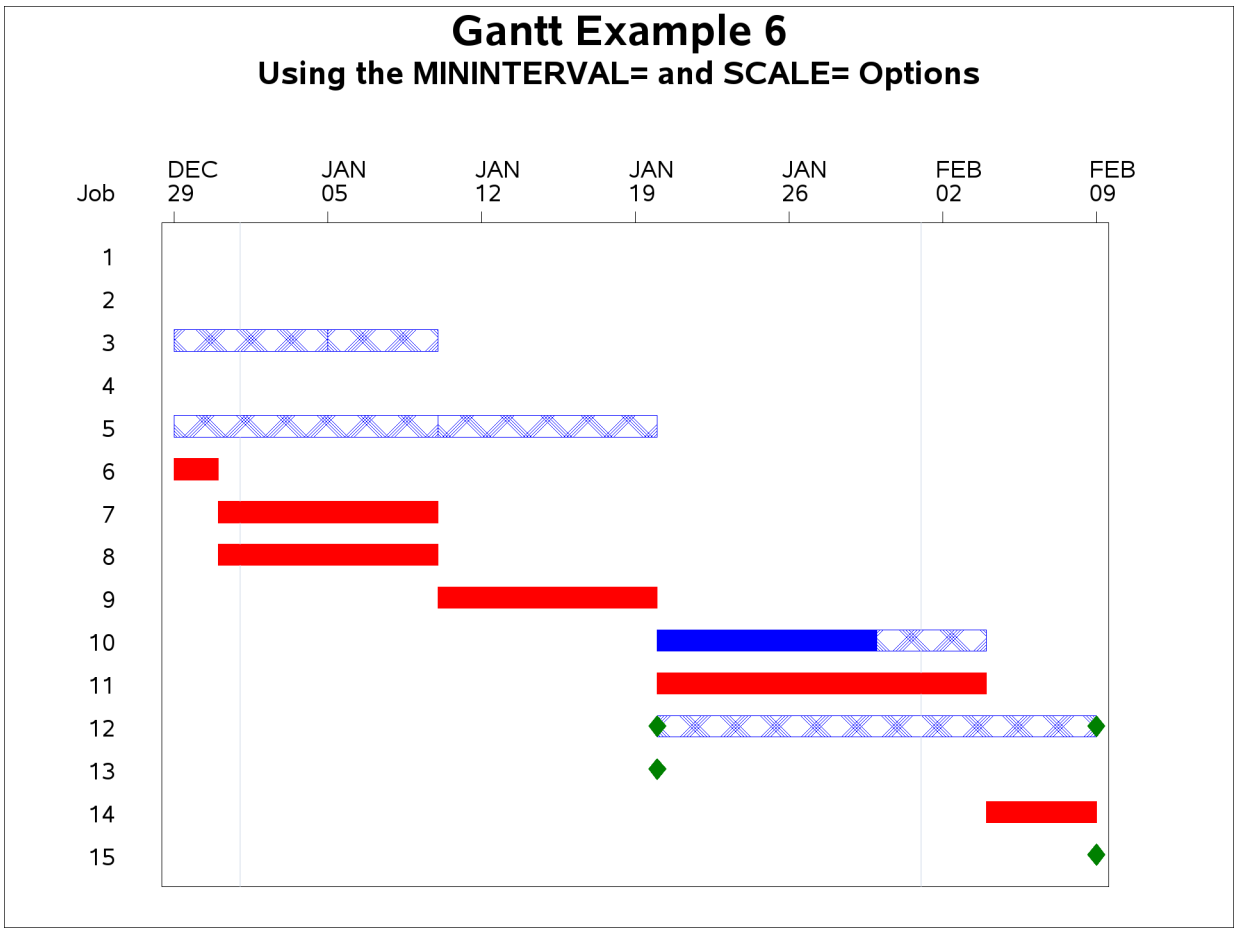
Gantt Example 6
Using the MININTERVAL= and SCALE= Options

descrpt	dept	E_START	E_FINISH	L_START	L_FINISH	T_FLOAT	F_FLOAT
Finalize and Approve Plan	Planning	01DEC03	05DEC03	01DEC03	05DEC03	0	0
Prepare Drawings	Engineering	06DEC03	15DEC03	06DEC03	15DEC03	0	0
Analyze Potential Markets	Marketing	06DEC03	10DEC03	05JAN04	09JAN04	30	0
Write Specifications	Engineering	06DEC03	10DEC03	11DEC03	15DEC03	5	5
Develop Marketing Concept	Marketing	11DEC03	20DEC03	10JAN04	19JAN04	30	30
Build Prototype	Engineering	16DEC03	30DEC03	16DEC03	30DEC03	0	0
Procure Raw Materials	Manufacturing	31DEC03	09JAN04	31DEC03	09JAN04	0	0
Prepare Manufacturing Facility	Manufacturing	31DEC03	09JAN04	31DEC03	09JAN04	0	0
Initial Production Run	Manufacturing	10JAN04	19JAN04	10JAN04	19JAN04	0	0
Evaluate Product In-House	Testing	20JAN04	29JAN04	25JAN04	03FEB04	5	5
Mail Product to Sample Market	Testing	20JAN04	03FEB04	20JAN04	03FEB04	0	0
Begin Full Scale Marketing	Marketing	20JAN04	20JAN04	09FEB04	09FEB04	20	20
Production Milestone		20JAN04	20JAN04	20JAN04	20JAN04	0	0
Engineering Changes	Engineering	04FEB04	08FEB04	04FEB04	08FEB04	0	0
Begin Full Scale Production	Manufacturing	09FEB04	09FEB04	09FEB04	09FEB04	0	0

Output 8.6.2 Using the MININTERVAL= and SCALE= Options in Graphics Mode



Output 8.6.2 *continued*

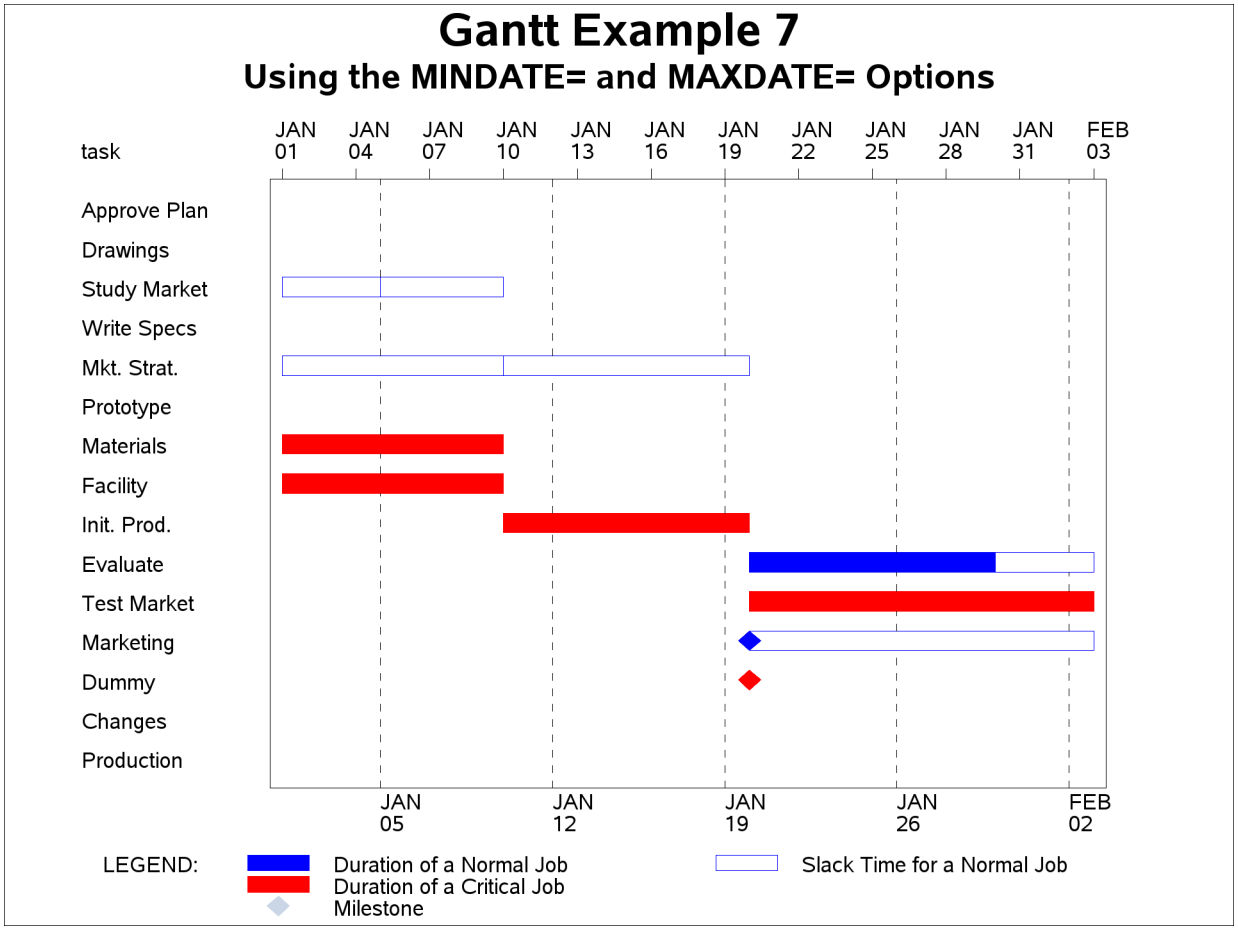


Example 8.7: Using the MINDATE= and MAXDATE= Options

In this example, the SAVE data set from Example 8.6 is used to display the schedule of the project over a limited time period. The start date and end date are specified by the MINDATE= and MAXDATE= options, respectively, in the CHART statement. As in Example 8.5, the COMPRESS option is used to ensure that the region of the Gantt chart lying between January 1, 2004, and February 2, 2004, fits on a single page. The specification REF='5JAN04'D TO '2FEB04'D BY WEEK causes PROC GANTT to draw reference lines at the start of every week. Further, the reference lines are labeled using the REFLABEL option. The CREF= and LREF= options are specified in the CHART statement to indicate the color and line style, respectively, of the reference lines. The resulting Gantt chart is shown in Output 8.7.1.

```
title h=2 'Gantt Example 7';
title2 h=1.5 'Using the MINDATE= and MAXDATE= Options';

goptions vpos=40 hpos=100;

* plot the schedule;
proc gantt graphics data=save;
   chart / mindate='1jan04'd maxdate='2feb04'd
           ref='5jan04'd to '2feb04'd by week
           reflabel cref=black lref=2 caxis=black
           compress dur=days nojobnum;
   id task;
   run;
```

Output 8.7.1 Using the MINDATE= and MAXDATE= Options in Graphics Mode

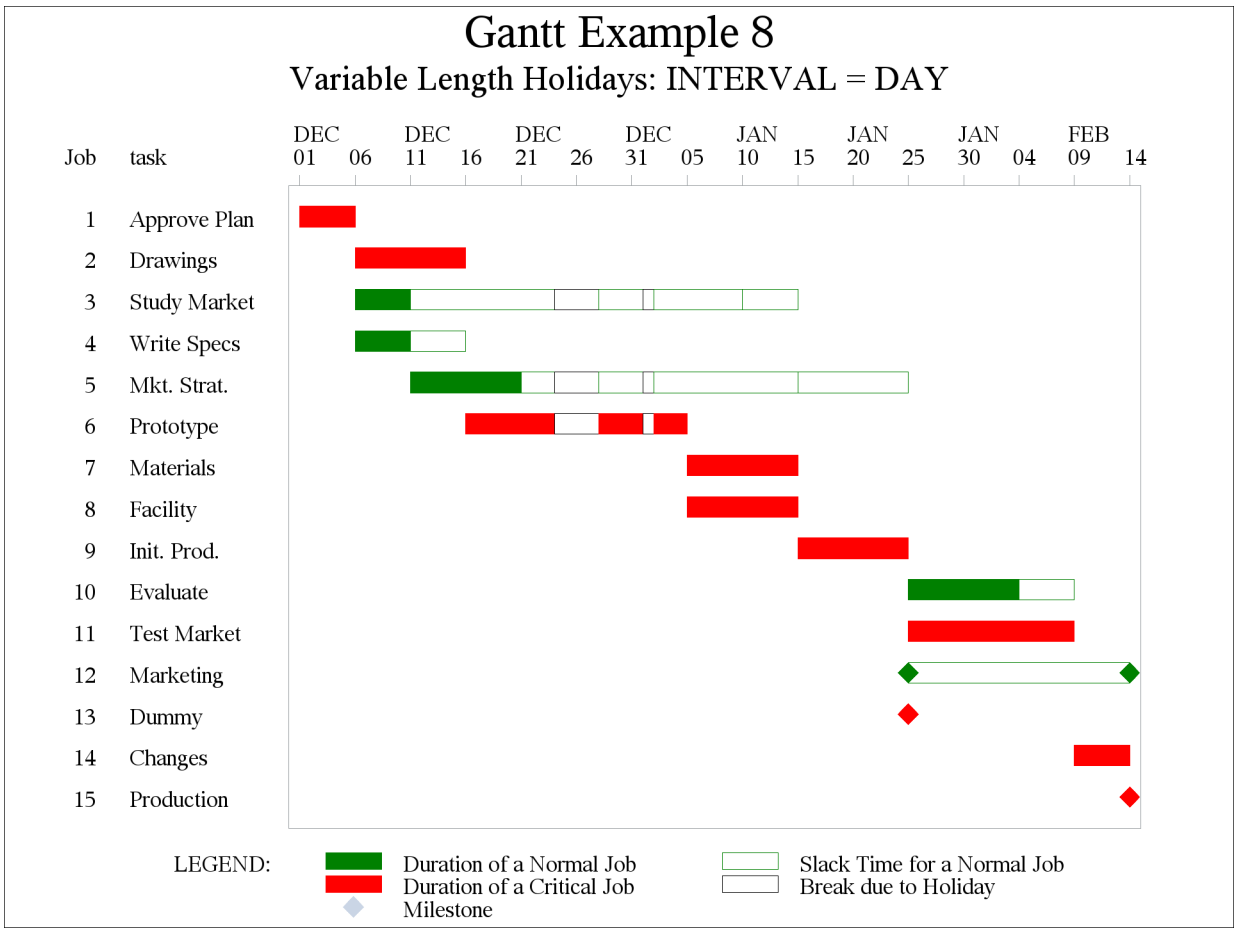


Example 8.8: Variable-Length Holidays

This example shows how you can mark vacation periods that last longer than one day on the Gantt chart. This can be done by using the HOLIDUR= option in the CHART statement. Recall that holiday duration is assumed to be in *interval* units where *interval* is the value specified for the INTERVAL= option. The project data for this example are the same as the data used in the previous example. Suppose that in your scheduling plans you want to assign work on all days of the week, allowing for a Christmas vacation of four days starting from December 24, 2003, and a day off on January 1, 2004 for the New Year. The data set HOLIDAYS contains the holiday information for the project. First, the project is scheduled with INTERVAL=DAY so that the holidays are on December 24, 25, 26, and 27, 2003, and on January 1, 2004. PROC GANTT is invoked with INTERVAL=DAY to correspond to the invocation of PROC CPM. The desired font is specified by using the FONT= option in the CHART statement and the F= option in the TITLE statement. As an alternative, the desired font can be specified globally by using the FTEXT= option in a GOPTIONS statement. The resulting Gantt chart is shown in Output 8.8.1.

```
data holidays;
   format holiday holifin date7.;
   input holiday & date7. holifin & date7. holidur;
   datalines;
24dec03  27dec03  4
01jan04  .        .
;

 * schedule the project subject to holidays;
proc cpm data=widgeta holidata=holidays out=sched1
         date='1dec03'd interval=day;
   tailnode tail;
   headnode head;
   duration days;
   id task dept descrpt;
   holiday  holiday / holidur=(holidur);
   run;

 * sort the schedule by the early start date ;
proc sort;
   by e_start;
   run;

 * plot the schedule;
title h=2 f='Thorndale AMT' 'Gantt Example 8';
title2 h=1.5 f='Thorndale AMT' 'Variable Length Holidays: INTERVAL = DAY';
proc gantt holidata=holidays data=sched1 ;
   chart / holiday=(holiday) holidur=(holidur) font='Thorndale AMT'
           dur=days interval=day pcompress;
   id task;
   run;
```

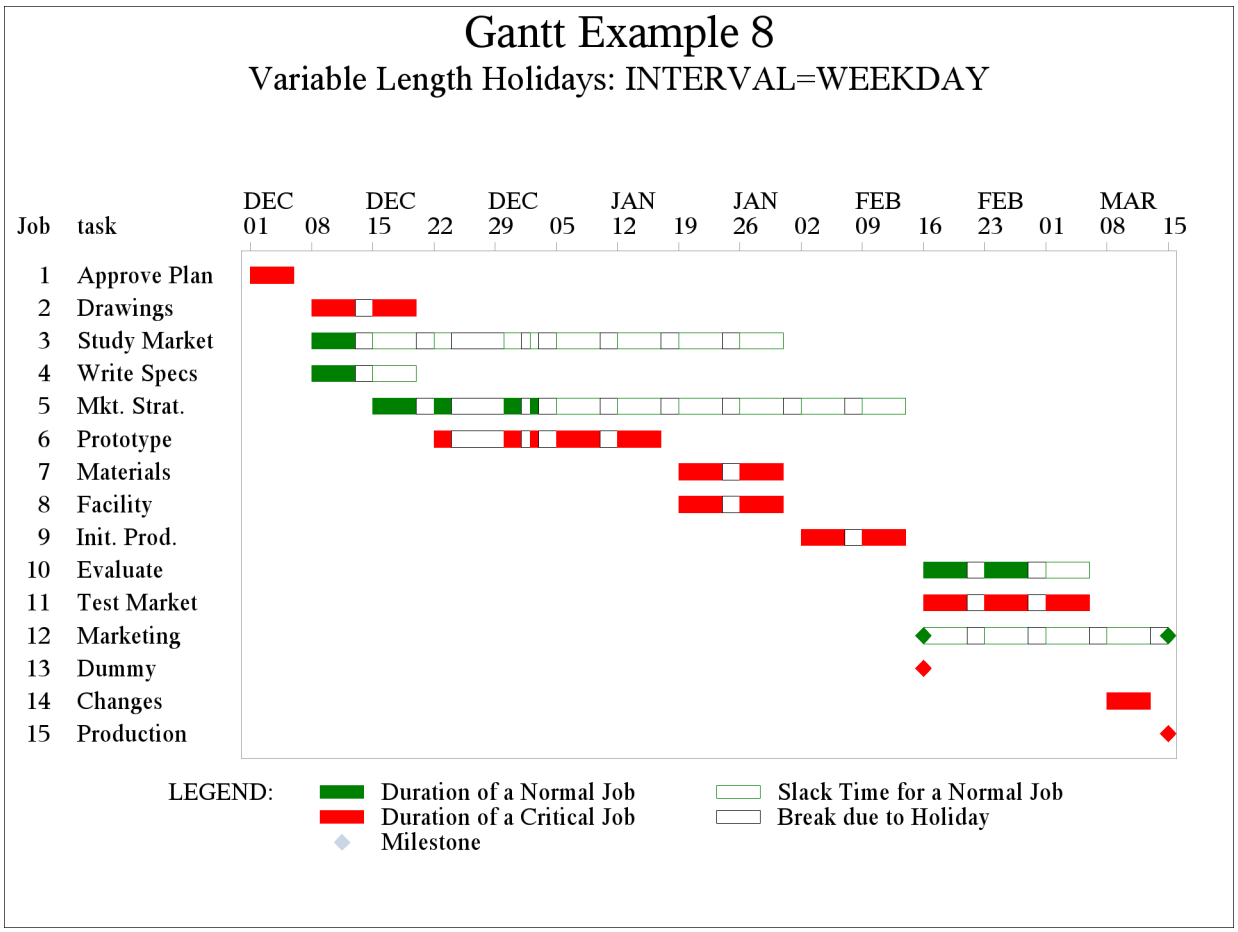
Output 8.8.1 Variable Length Holidays: INTERVAL=DAY



Next, consider the same project and Holiday data set, but invoke PROC CPM with INTERVAL=WEEKDAY. Then, the value '4' specified for the variable HOLIDUR is interpreted as 4 weekdays. The holidays are on December 24, 25, 26, and 29, 2003, and on January 1, 2004, because December 27 and 28 (Saturday and Sunday) are non-working days. The same steps are used as previously, except that INTERVAL is set to WEEKDAY instead of DAY in both PROC CPM and PROC GANTT. Suppose that the resulting data set is saved as SCHED2. The following invocation of PROC GANTT produces Output 8.8.2. Note that the use of INTERVAL=WEEKDAY causes weekends to be also marked on the chart.

```
title2 h=1.5 f='Thorndale AMT' 'Variable Length Holidays: INTERVAL=WEEKDAY';

proc gantt holidata=holidays data=sched2;
   chart / holiday=(holiday) holidur=(holidur)
           font='Thorndale AMT'
           height=1.4
           interval=weekday
           dur=days
           pcompress;
   id task;
   run;
```

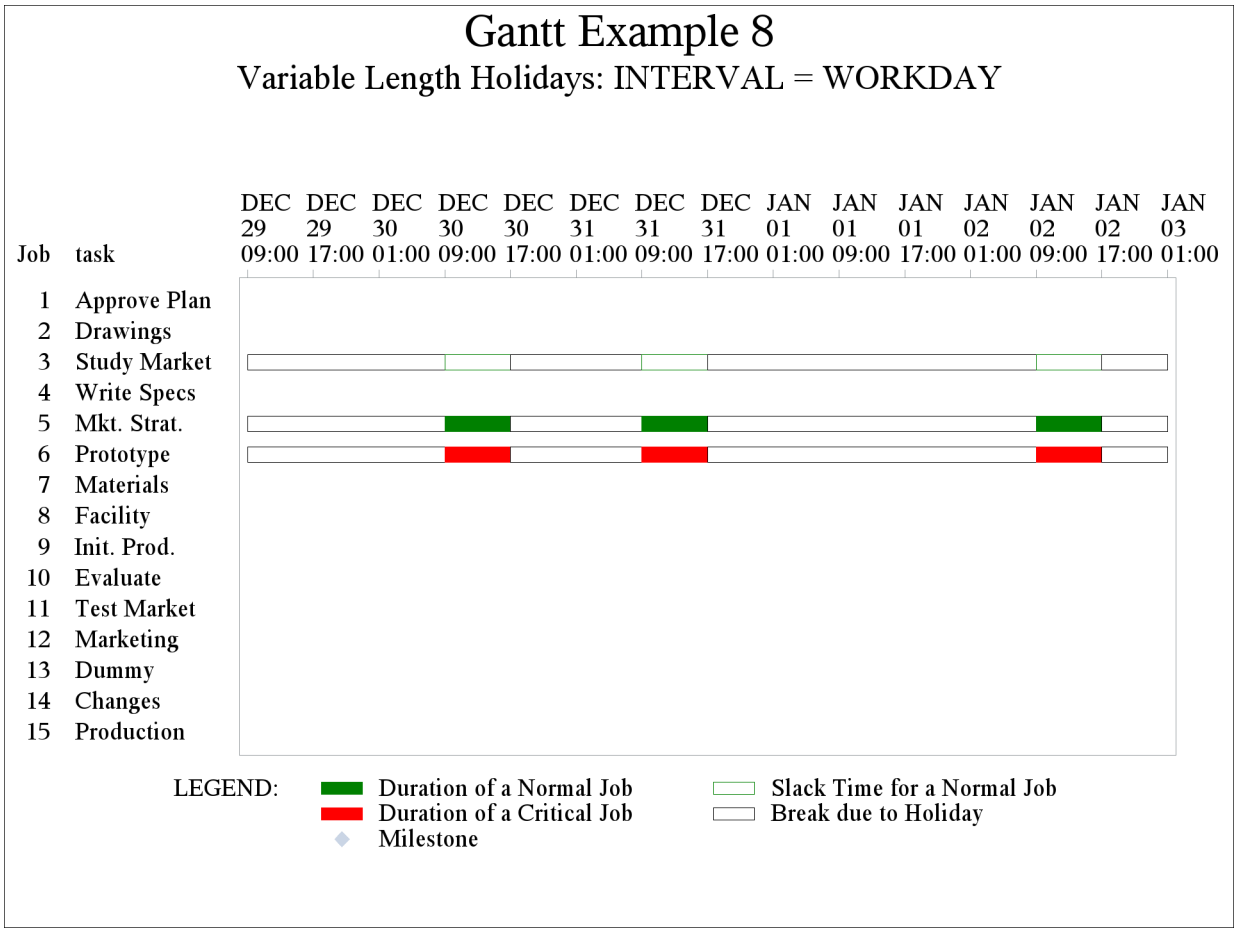
Output 8.8.2 Variable Length Holidays: INTERVAL=WEEKDAY



Finally, when the INTERVAL= option is specified as WORKDAY, the workday is assumed to be from 9:00 a.m. to 5:00 p.m., and the Christmas holiday period begins at 5:00 p.m. on December 23, 2003, and ends at 9:00 a.m. on December 30, 2004. PROC GANTT is invoked with the MARKBREAK option and MININTERVAL=DTHOUR so that all breaks during a day can be seen. Because the SCALE= option is not specified, each column denotes one hour of the schedule. Since the project duration is several days long, the entire Gantt chart would be spread across many pages. Simply specifying the COMPRESS or PCOMPRESS option will not be of much help since the text would be barely legible owing to the extent of the scaling. Hence, only a portion of the Gantt chart is shown in Output 8.8.3 using the MINDATE= and MAXDATE= options. Note that the Gantt chart is labeled with the date as well as the time values on the time axis.

```
title2 h=1.5 f='Thorndale AMT' 'Variable Length Holidays: INTERVAL = WORKDAY';

proc gantt holidata=holidays data=sched3;
   chart / holiday=(holiday) holidur=(holidur)
           dur=days interval=workday
           font='Thorndale AMT'
           mininterval=dthour markbreak
           mindate='29dec03:09:00:00'dt
           maxdate='03jan04:00:00:00'dt
           pcompress height=1.5;
   id task;
   run;
```

Output 8.8.3 Variable Length Holidays: INTERVAL=WORKDAY



Example 8.9: Multiple Calendars

This example illustrates the use of multiple calendars within a project. The data for this example are the same as the data used in Example 4.10 to illustrate the CPM Procedure. The input data sets to PROC CPM are displayed in Output 8.9.1. The WORKDATA data set defines several shift patterns, which in turn are identified with four different calendars in the CALEDATA data set:

- The ‘DEFAULT’ calendar has five 8-hour workdays (8 a.m. - 4 p.m.) on Monday through Friday and holidays on Saturday and Sunday.
- The ‘OVT_CAL’ calendar defines the “overtime” calendar that is followed by the Engineering department to build the prototype. The ‘OVT_CAL’ calendar has five 10-hour workdays (8 a.m. - 6 p.m.) on Monday through Friday, a 4-hour halfday (8 a.m. - 12 noon) on Saturday and a holiday on Sunday.
- The ‘PROD_CAL’ calendar defines the “production” calendar that is used for full-scale production of the widget. The ‘PROD_CAL’ calendar consists of continuous work from Monday 8 a.m. through Saturday 6 p.m. except for two 2-hour breaks per day from 6 a.m. to 8 a.m. and from 6 p.m. to 8 p.m. Thus, ‘PROD_CAL’ is made up of eleven 8-hour shifts per week; six day shifts and five night shifts.
- The ‘Eng_cal’ calendar defines the calendar followed by the Engineering department for writing the specifications for the prototype. The ‘Eng_cal’ calendar has the same work pattern as the default calendar with an extra holiday period of seven days starting on December 8, 2003.

The HOLIDATA data set defines the appropriate holidays for the different calendars. The project data set WIDGVAC includes a variable named CAL to identify the appropriate calendar for each activity.

Output 8.9.1 Multiple Calendars: Data Sets

Multiple Calendars

Workdays Data Set

Obs	fullday	halfday	ovtday	s1	s2	s3
1	8:00	8:00	8:00	.	8:00	.
2	16:00	12:00	18:00	6:00	18:00	6:00
3	.	.	.	8:00	20:00	8:00
4	.	.	.	18:00	.	18:00
5	.	.	.	20:00	.	.
6	.	.	.	.	.	.

Calendar Data Set

Obs	cal	_sun_	_mon_	_tue_	_wed_	_thu_	_fri_	_sat_
1	DEFAULT	holiday	fullday	fullday	fullday	fullday	fullday	holiday
2	OVT_CAL	holiday	ovtday	ovtday	ovtday	ovtday	ovtday	halfday
3	PROD_CAL	holiday	s2	s1	s1	s1	s1	s3
4	Eng_cal							

Holidays Data Set

Obs	holiday	holifin	holidur	cal
1	08DEC03	.	7	Eng_cal
2	24DEC03	26DEC03	.	
3	01JAN04	01JAN04	.	

Output 8.9.1 *continued*

Project Data Set

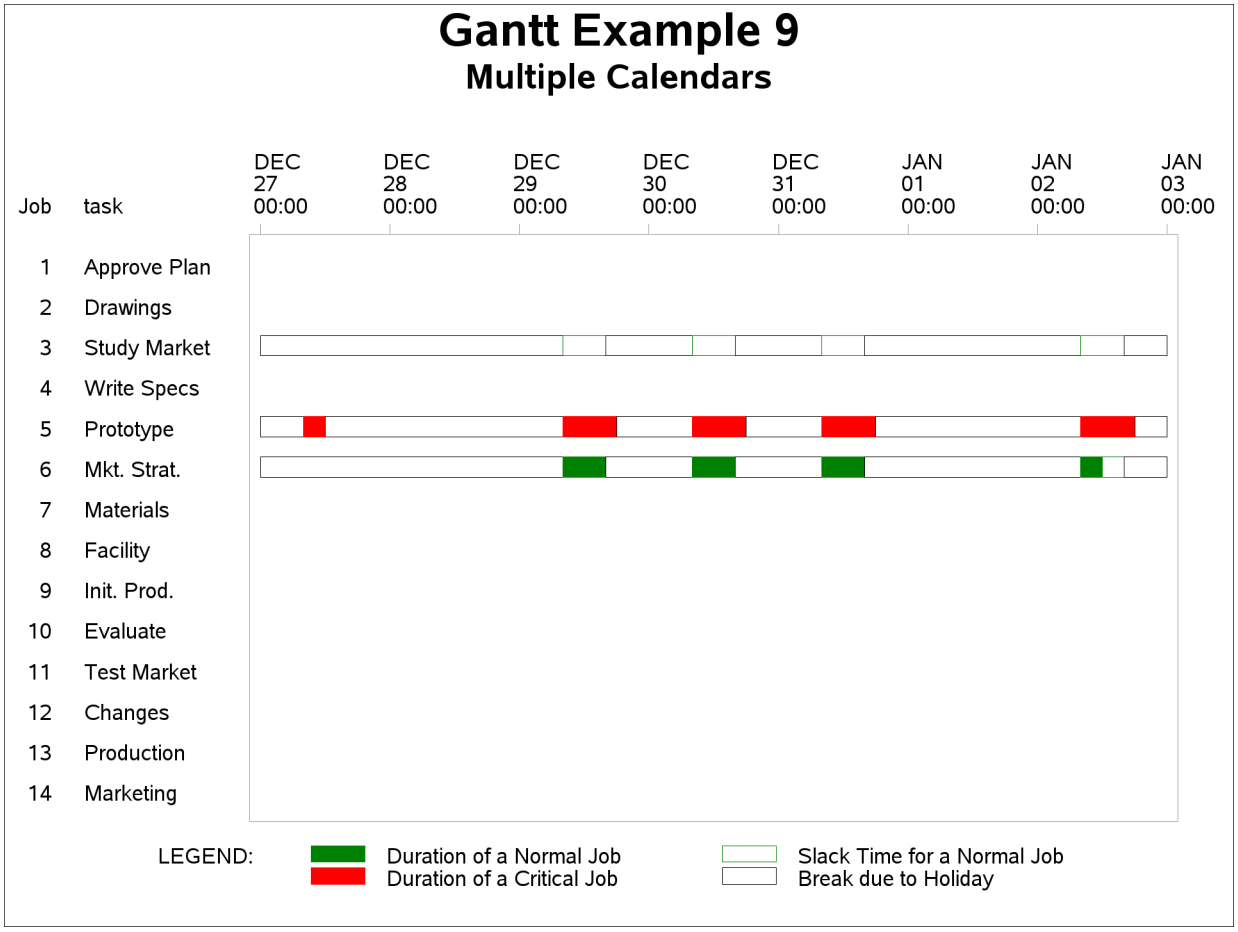
Obs	task	succ1	succ2	succ3	days	cal
1	Approve Plan	Drawings	Study Market	Write Specs	5.5	DEFAULT
2	Drawings	Prototype			10.0	DEFAULT
3	Study Market	Mkt. Strat.			5.0	DEFAULT
4	Write Specs	Prototype			4.5	Eng_cal
5	Prototype	Materials	Facility		15.0	OVT_CAL
6	Mkt. Strat.	Test Market	Marketing		10.0	DEFAULT
7	Materials	Init. Prod.			10.0	DEFAULT
8	Facility	Init. Prod.			10.0	DEFAULT
9	Init. Prod.	Test Market	Marketing	Evaluate	10.0	DEFAULT
10	Evaluate	Changes			10.0	DEFAULT
11	Test Market	Changes			15.0	DEFAULT
12	Changes	Production			5.0	DEFAULT
13	Production				0.0	PROD_CAL
14	Marketing				0.0	DEFAULT

The program used to invoke PROC CPM and PROC GANTT follows. The CALENDAR= and WORKDAY= options are specified in the PROC GANTT statement to identify the CALEDATA and WORKDATA data sets, respectively. The CALID= option in the CHART statement names the variable identifying the calendar that each observation refers to in the WIDGVAC and CALEDATA data sets. Since the value of MININTERVAL= is DTDAY, setting the SCALE= value to 12 ensures that a single column on the Gantt chart represents two hours. This is done in order to be able to detect a two hour difference between schedules. Consequently, the MINDATE= and MAXDATE= options are used to control the output produced by PROC GANTT. The resulting Gantt chart is shown in Output 8.9.2. Notice the 5 column duration for 'Prototype' on December 29, 2003 representing a 10-hour day versus the 4 column duration for 'Mkt. Strat.' for the same day representing 8 hours of work. Although MAXDATE= is set to 8 a.m. on January 2, 2004, the last tick mark is the beginning of January 3, 2004. This is because the specified value of the MAXDATE= option does not correspond to a tick mark (based on the SCALE= and MININTERVAL= options); the value used is the first tick mark appearing after the value of the MAXDATE= option.

```
proc cpm date='01dec03'd interval=workday data=widgvac
         out=schedvac holidata=holidata
         workday=workdata calendar=caledata;
   holiday holiday / holifin=holifin holidur=holidur;
   activity task;
   duration days;
   successor succ1 succ2 succ3;
   calid cal;
   run;

title h=2 'Gantt Example 9';
title2 h=1.5 'Multiple Calendars';

proc gantt data=schedvac holidata=holidata
           workday=workdata calendar=caledata ;
   chart / holiday=(holiday) holiend=(holifin)
           calid=cal
```

```
        markbreak scale=12
        mindate='27dec03:00:00'dt
        maxdate='02jan04:08:00'dt
        pcompress;
id task;
run;
```

Output 8.9.2 Multiple Calendars



Example 8.10: Plotting the Actual Schedule

Suppose that the project is complete and you want to compare the actual progress of the activities with the predicted schedule computed by PROC CPM. The following DATA step stores the actual start and finish times of each activity in a data set named COMPLETE. A data set named WIDGELA is then created that contains both the schedule obtained from PROC CPM (the data set SAVEH from Example 8.3 is used because it does not contain the dummy activity) and the actual schedule. The resulting data set is sorted by early start time.

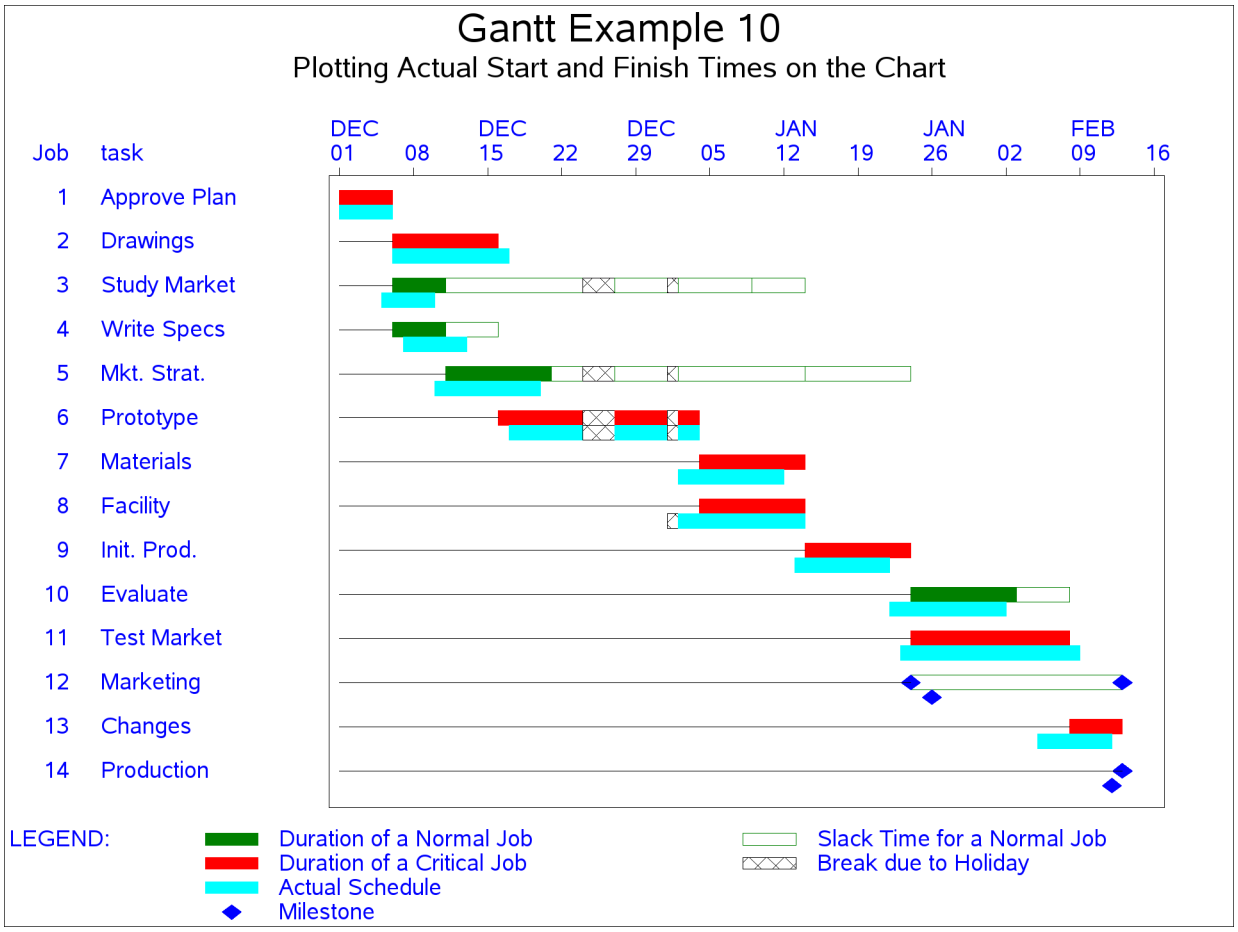
Fill patterns are specified using PATTERN statements, and the COMPRESS option is employed in order to draw the entire Gantt chart on one page. Predicted schedules as well as actual schedules are plotted on separate bars for each activity. The A_START= and A_FINISH= options in the CHART statement are used to specify the variables containing the actual start and finish times for each activity. The actual schedule

is plotted with the fill pattern specified in the sixth PATTERN statement. This example also illustrates the drawing of holidays in graphics mode. PROC GANTT uses the fill pattern specified in the seventh PATTERN statement to represent the holidays defined by the HOLIDATA= data set. The holidays are identified to PROC GANTT by specifying the HOLIDAY= and HOLIFIN= options in the CHART statement.

The HCONNECT option causes a connecting line to be drawn from the left boundary of the chart to the early start time for each activity. The CHCON= option specifies the color for drawing the connect lines. You can use the LHCON= option in the CHART statement to specify a line style other than the default style for the connect lines. The Gantt chart is shown in Output 8.10.1.

```
data complete;
   format activity $12. sdate date7. fdate date7.;
   input activity & sdate & date7. fdate & date7.;
   datalines;
Approve Plan   01dec03  05dec03
Drawings       06dec03  16dec03
Study Market   05dec03  09dec03
Write Specs    07dec03  12dec03
Prototype      17dec03  03jan04
Mkt. Strat.    10dec03  19dec03
Materials      02jan04  11jan04
Facility       01jan04  13jan04
Init. Prod.    13jan04  21jan04
Evaluate       22jan04  01feb04
Test Market    23jan04  08feb04
Changes        05feb04  11feb04
Production     12feb04  12feb04
Marketing      26jan04  26jan04
;

 * merge the computed schedule with the actual schedule;
data widgela;
   merge saveh complete;

* sort the data;
proc sort;
   by e_start;
   run;

* set vpos to 40 and hpos to 100;
goptions vpos=40 hpos=100;

title h=1.75 f='Albany AMT' 'Gantt Example 10';
title2 h=1.25
   f='Albany AMT' 'Plotting Actual Start and Finish Times on the Chart';

* plot the computed and actual schedules using proc gantt;
proc gantt graphics data=widgela holidata=holidays;
   chart / holiday=(holiday) holifin=(holifin)
           a_start=sdate a_finish=fdate
           dur=days font='Albany AMT' chcon=black
           hconnect compress ctext=blue height=1.5
           caxis=black cmile=blue;
   id task;
   run;
```

Output 8.10.1 Plotting the Actual Schedule on the Gantt Chart



Example 8.11: Comparing Progress Against a Baseline Schedule

Suppose that the widget manufacturing project is currently in progress and you want to measure its performance by comparing it with a baseline schedule. For example, the baseline schedule may be the originally planned schedule, a target schedule that you would like to achieve, or an existing schedule that you intend to improve on. The data for this example come from Example 4.13, which was used to illustrate the options available in PROC CPM. Prior to the beginning of the project, the predicted early schedule is saved by PROC CPM as the baseline schedule. Progress information for the project as of December 19, 2003, is saved in the ACTUAL data set. The variables SDATE and FDATE represent the actual start and actual finish times, respectively. The variables PCTC and RDUR represent the percent of work completed and the remaining days of work for each activity, respectively. PROC CPM is then invoked using the baseline and project progress information with TIMENOW set to December 19, 2003. The scheduling is carried out with the AUTOPUPDT option in order to automatically update progress information. The Schedule data set WIDGUPDT produced by PROC CPM is shown in Output 8.11.1. Notice that the development of a marketing strategy (activity 5: 'Mkt. Strategy') and the building of the prototype (activity 6: 'Prototype') have a specified value for A_START and a missing value for A_FINISH, indicating that they are currently in progress at TIMENOW.

PROC GANTT is next invoked with the data set WIDGUPDT. This data set contains the actual schedule variables A_START and A_FINISH and the baseline schedule variables B_START and B_FINISH. The Gantt chart is drawn with three schedule bars per activity. The first bar represents the predicted early/late schedule based on the actual data specified, the second bar represents the actual schedule, and the third bar represents the baseline schedule. The TIMENOW= option is specified in the CHART statement to draw a timenow line on December 19, 2003. Actual schedule bars for 'Mkt. Strategy' and 'Prototype' are drawn up to TIMENOW to indicate that they are currently in progress. You can use the CTNOW=, LTNOW=, and WTNOW= options to change the color, style, and width of the timenow line, respectively. To suppress the timenow label displayed at the bottom of the axis, specify the NOTNLABEL in the CHART statement.

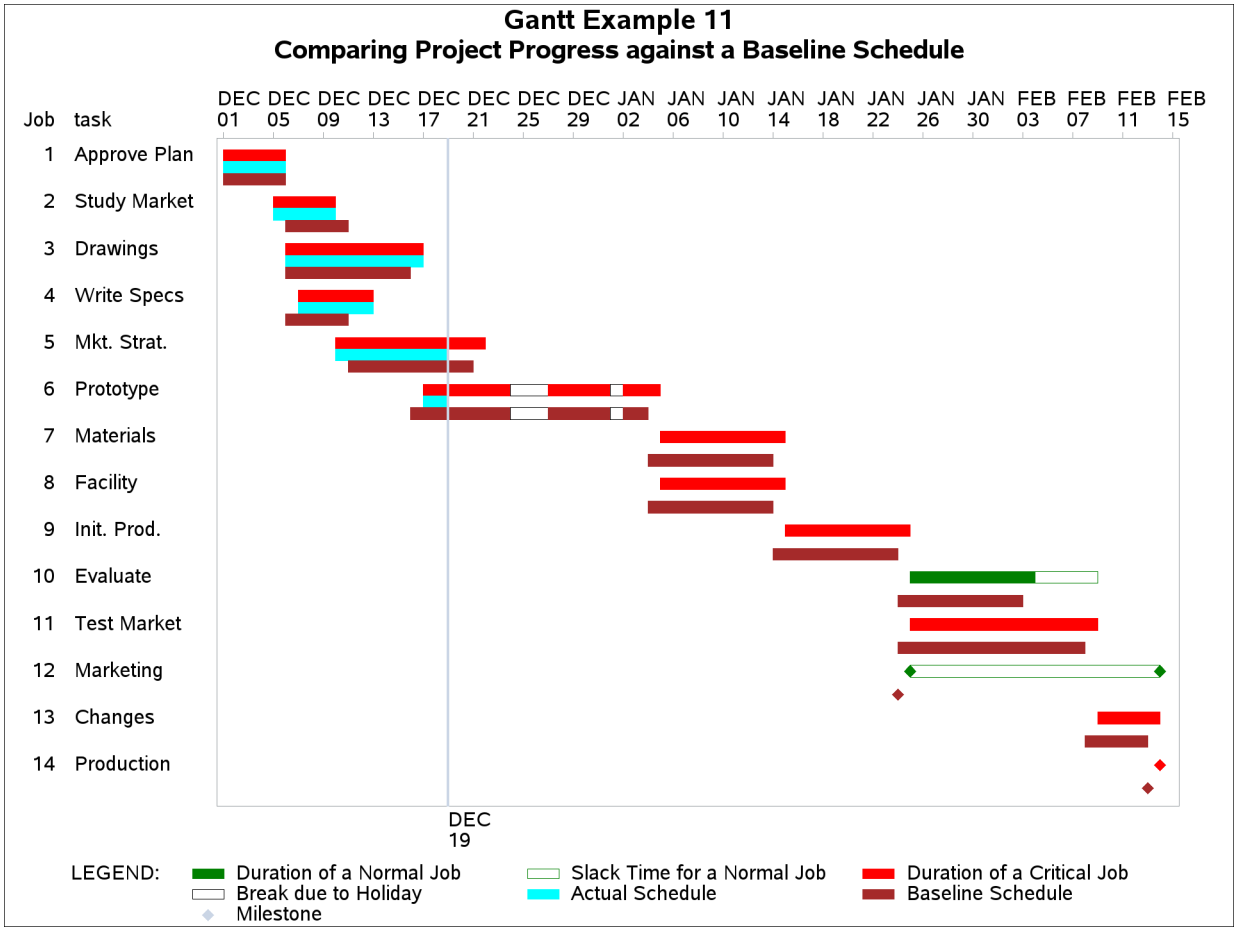
```
title h=1.2 'Gantt Example 11';

* estimate schedule based on actual data;
proc cpm data=widgact holidata=holidays
         out=widgupdt date='1dec03'd;
   activity task;
   succ     succ1 succ2 succ3;
   duration days;
   holiday  holiday / holifin=(holifin);
   baseline / compare=early;
   actual / as=sdate af=fdate timenow='19dec03'd
            remdur=rdur pctcomp=pctc
            autoupdt;
   run;

* sort the data;
proc sort;
   by e_start;
   run;

* print the data;
title2 'Progress Data';

proc print;
   var task e_: l_: a_start a_finish b_: ;
   run;

title2 'Comparing Project Progress against a Baseline Schedule';

* plot the actual and baseline schedules using proc gantt;
proc gantt data=widgupdt holidata=holidays;
   chart / holiday=(holiday) holifin=(holifin)
           timenow='19dec03'd dur=days
           scale=2 height=1.6
           pcompress;
   id task;
   run;
```

Output 8.11.1 Schedule Data Set WIDGUPDT

Gantt Example 11
Progress Data

Obs	task	E_START	E_FINISH	L_START	L_FINISH	A_START	A_FINISH	B_START	B_FINISH
1	Approve Plan	01DEC03	05DEC03	01DEC03	05DEC03	01DEC03	05DEC03	01DEC03	05DEC03
2	Study Market	05DEC03	09DEC03	05DEC03	09DEC03	05DEC03	09DEC03	06DEC03	10DEC03
3	Drawings	06DEC03	16DEC03	06DEC03	16DEC03	06DEC03	16DEC03	06DEC03	15DEC03
4	Write Specs	07DEC03	12DEC03	07DEC03	12DEC03	07DEC03	12DEC03	06DEC03	10DEC03
5	Mkt. Strat.	10DEC03	21DEC03	10DEC03	21DEC03	10DEC03	.	11DEC03	20DEC03
6	Prototype	17DEC03	04JAN04	17DEC03	04JAN04	17DEC03	.	16DEC03	03JAN04
7	Materials	05JAN04	14JAN04	05JAN04	14JAN04	.	.	04JAN04	13JAN04
8	Facility	05JAN04	14JAN04	05JAN04	14JAN04	.	.	04JAN04	13JAN04
9	Init. Prod.	15JAN04	24JAN04	15JAN04	24JAN04	.	.	14JAN04	23JAN04
10	Evaluate	25JAN04	03FEB04	30JAN04	08FEB04	.	.	24JAN04	02FEB04
11	Test Market	25JAN04	08FEB04	25JAN04	08FEB04	.	.	24JAN04	07FEB04
12	Marketing	25JAN04	25JAN04	14FEB04	14FEB04	.	.	24JAN04	24JAN04
13	Changes	09FEB04	13FEB04	09FEB04	13FEB04	.	.	08FEB04	12FEB04
14	Production	14FEB04	14FEB04	14FEB04	14FEB04	.	.	13FEB04	13FEB04

Output 8.11.2 Comparing Project Progress Against a Baseline Schedule



Example 8.12: Using the COMBINE Option

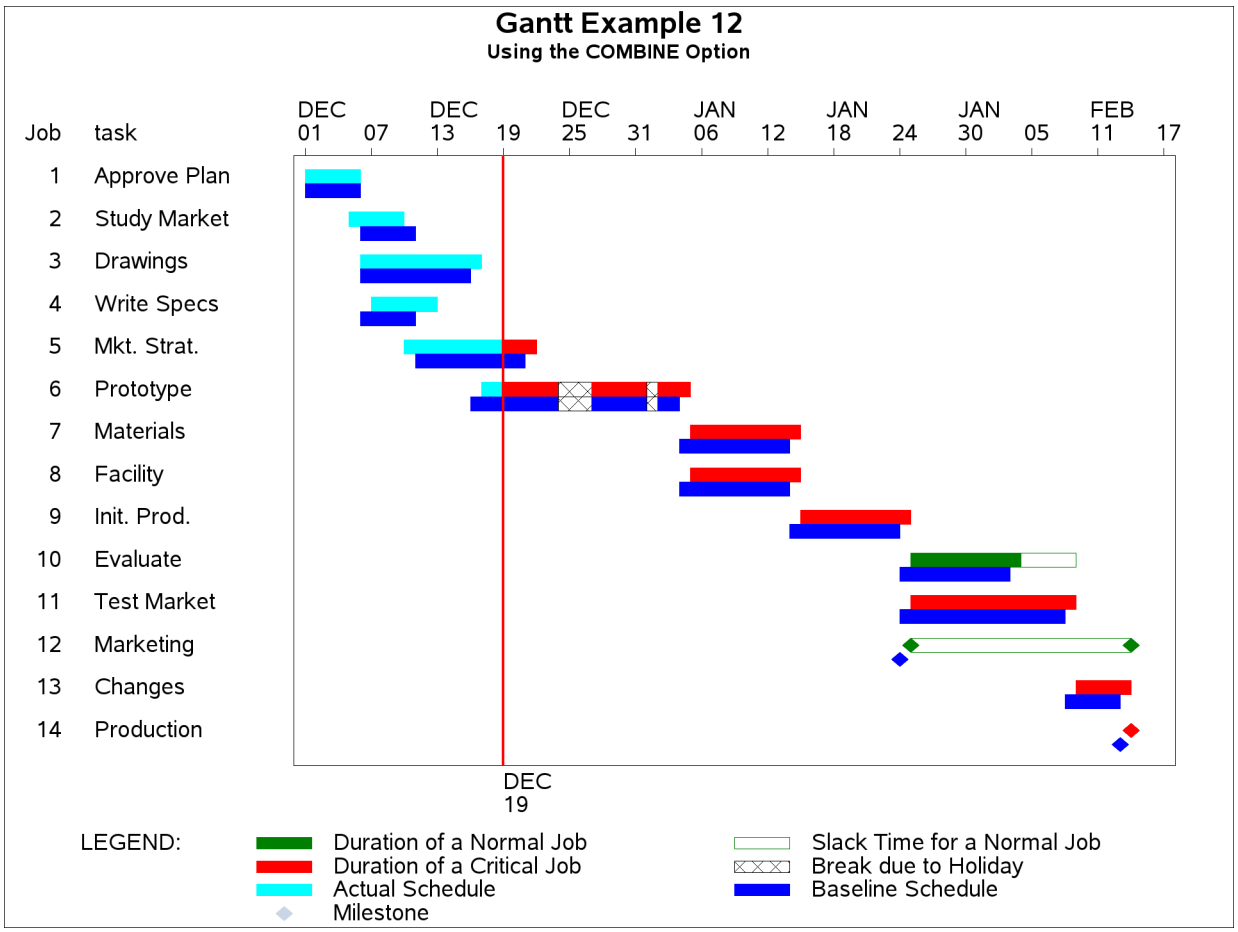
When you monitor a project in progress, as in the previous example, it is evident that there are no actual dates beyond TIMENOW and that PROC CPM sets the early times to the corresponding actual times for activities that are completed or in progress (see Output 8.11.1). For example, activities 1 through 4 have their early schedule equal to the actual schedule. Activities 5 and 6 have their early start equal to the actual start; however the actual finish for these two activities is missing since they are in progress at TIMENOW. Finally, activities 7 through 14 have no actual information.

The COMBINE option in PROC GANTT exploits the fact that the early times are made consistent with the actual times to strip away a lot of the redundancy and produce a more compact Gantt chart while retaining all of the essential schedule information. Specifying the COMBINE option in the CHART statement of the previous example produces the Gantt chart in Output 8.12.1. Instead of using two separate bars to draw the early/late schedule and the actual schedule, the COMBINE option causes PROC GANTT to use one bar to represent all three schedules and draws a timenow line. The actual schedule is shown to the left of TIMENOW and the early/late schedule is shown to the right of TIMENOW. Thus, for activities 1 through 4, the actual schedule is drawn on the first bar to the left of the timenow line. Activities 5 and 6 are in progress at TIMENOW, which is indicated by the actual start positioned to the left of TIMENOW and the predicted early/late schedule, based on the progress made up to TIMENOW, drawn to the right of TIMENOW. Activities 7 through 14 have not yet started, and this is reflected in their predicted early/late schedules drawn to the right of TIMENOW.

The COMBINE option draws a timenow line by default, and if the TIMENOW= option is not specified, the procedure computes the value of TIMENOW based on the schedule data as explained in the “Syntax” section. In this example, specifying the COMBINE option without the TIMENOW= option causes a timenow line to be drawn on December 18, 2003, since this is the first day following the largest actual value. The CTNOW= option is used to specify the color of the timenow line. You can change the line style and line width of the timenow line by specifying the LTNOW= and WTNOW= options, respectively, in the CHART statement.

```
title h=1.6 'Gantt Example 12';

title2 'Using the COMBINE Option';

* set vpos to 50 and hpos to 100;
goptions vpos=50 hpos=100;

* plot the combined and baseline schedules using proc gantt;
proc gantt graphics data=widgupdt holidata=holidays;
   chart / holiday=(holiday) holifin=(holifin)
           compress ctnow=red caxis=black
           height=1.5
           timenow='19dec03'd
           dur=days
           combine;
   id task;
   run;
```

Output 8.12.1 Using the COMBINE Option in Graphics Mode



Example 8.13: Plotting the Resource-Constrained Schedule

This example illustrates plotting the resource-constrained schedules on a Gantt chart. The schedule used is the one produced in Example 4.19 using the CPM procedure. The output data set from PROC CPM is displayed in Output 8.13.2. Notice that the activities 'Drawings' and 'Mkt. Strat.' have been split to produce a shorter project duration than if they had not been split.

PROC GANTT is invoked with all default options and an ID statement. The early/late schedule is drawn on the first bar, and the resource-constrained schedule is drawn on the second bar of each activity. The observations corresponding to the split segments of each activity have been combined to produce the plot of the resource-constrained schedule for that activity. Thus, even though the Schedule data set input to PROC GANTT contains 18 observations, the Gantt chart shows each of the 14 activities only once.

```
title h=1.75 'Gantt Example 13';
title2 h=1.25 'Resource Constrained Schedule';

* set vpos to 50 and hpos to 100;
goptions vpos=50 hpos=100;
```

```
* plot the resource-constrained schedule using proc gantt;
proc gantt data=spltschd holidata=holdata;
   chart / holiday=(hol) dur=days
           height=1.6 pcompress;
   id task;
   run;
```

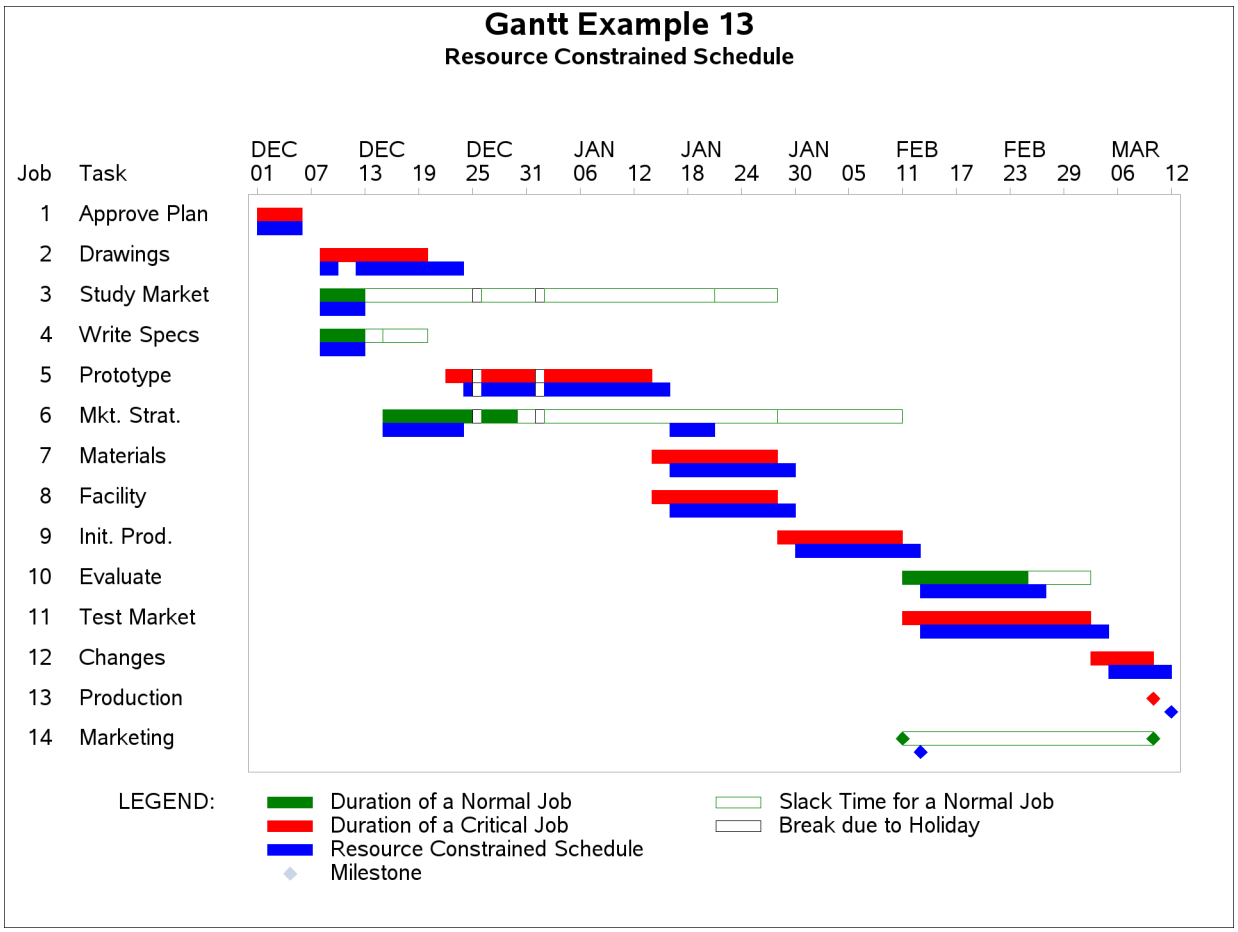
Output 8.13.1 Schedule Data Set SPLTSCHD

Gantt Example 13
Project Schedule: Splitting Allowed

Obs	Task	succ	SEGMT_NO	days	prodman	hrdware
1	Approve Plan	Drawings	.	5	1	.
2	Drawings	Prototype	.	10	.	1
3	Drawings	Prototype	1	2	.	1
4	Drawings	Prototype	2	8	.	1
5	Study Market	Mkt. Strat.	.	5	.	.
6	Write Specs	Prototype	.	5	.	.
7	Prototype	Materials	.	15	1	.
8	Mkt. Strat.	Test Market	.	10	1	.
9	Mkt. Strat.	Test Market	1	7	1	.
10	Mkt. Strat.	Test Market	2	3	1	.
11	Materials	Init. Prod.	.	10	.	.
12	Facility	Init. Prod.	.	10	.	.
13	Init. Prod.	Test Market	.	10	1	.
14	Evaluate	Changes	.	10	1	.
15	Test Market	Changes	.	15	.	.
16	Changes	Production	.	5	.	.
17	Production		.	0	1	.
18	Marketing		.	0	.	.

Obs	S_START	S_FINISH	E_START	E_FINISH	L_START	L_FINISH
1	01DEC03	05DEC03	01DEC03	05DEC03	01DEC03	05DEC03
2	08DEC03	23DEC03	08DEC03	19DEC03	08DEC03	19DEC03
3	08DEC03	09DEC03	08DEC03	19DEC03	08DEC03	19DEC03
4	12DEC03	23DEC03	08DEC03	19DEC03	08DEC03	19DEC03
5	08DEC03	12DEC03	08DEC03	12DEC03	21JAN04	27JAN04
6	08DEC03	12DEC03	08DEC03	12DEC03	15DEC03	19DEC03
7	24DEC03	15JAN04	22DEC03	13JAN04	22DEC03	13JAN04
8	15DEC03	20JAN04	15DEC03	29DEC03	28JAN04	10FEB04
9	15DEC03	23DEC03	15DEC03	29DEC03	28JAN04	10FEB04
10	16JAN04	20JAN04	15DEC03	29DEC03	28JAN04	10FEB04
11	16JAN04	29JAN04	14JAN04	27JAN04	14JAN04	27JAN04
12	16JAN04	29JAN04	14JAN04	27JAN04	14JAN04	27JAN04
13	30JAN04	12FEB04	28JAN04	10FEB04	28JAN04	10FEB04
14	13FEB04	26FEB04	11FEB04	24FEB04	18FEB04	02MAR04
15	13FEB04	04MAR04	11FEB04	02MAR04	11FEB04	02MAR04
16	05MAR04	11MAR04	03MAR04	09MAR04	03MAR04	09MAR04
17	12MAR04	12MAR04	10MAR04	10MAR04	10MAR04	10MAR04
18	13FEB04	13FEB04	11FEB04	11FEB04	10MAR04	10MAR04

Output 8.13.2 Plotting the Resource-Constrained Schedule



Example 8.14: Specifying the Schedule Data Directly

Although each of the examples shown so far uses PROC CPM to produce the Schedule data set for PROC GANTT, this is by no means a requirement of the GANTT procedure. While the CPM procedure is a convenient means for producing different types of schedules, you can create your own schedule and draw a Gantt chart of the schedule without any intervention from PROC CPM. This is done by storing the schedule information in a SAS data set and specifying the data set name using the DATA= option in the PROC GANTT statement. It is also not necessary for the variables in the data set to have specific names, although giving the variables certain names can eliminate the need to explicitly identify them in the CHART statement.

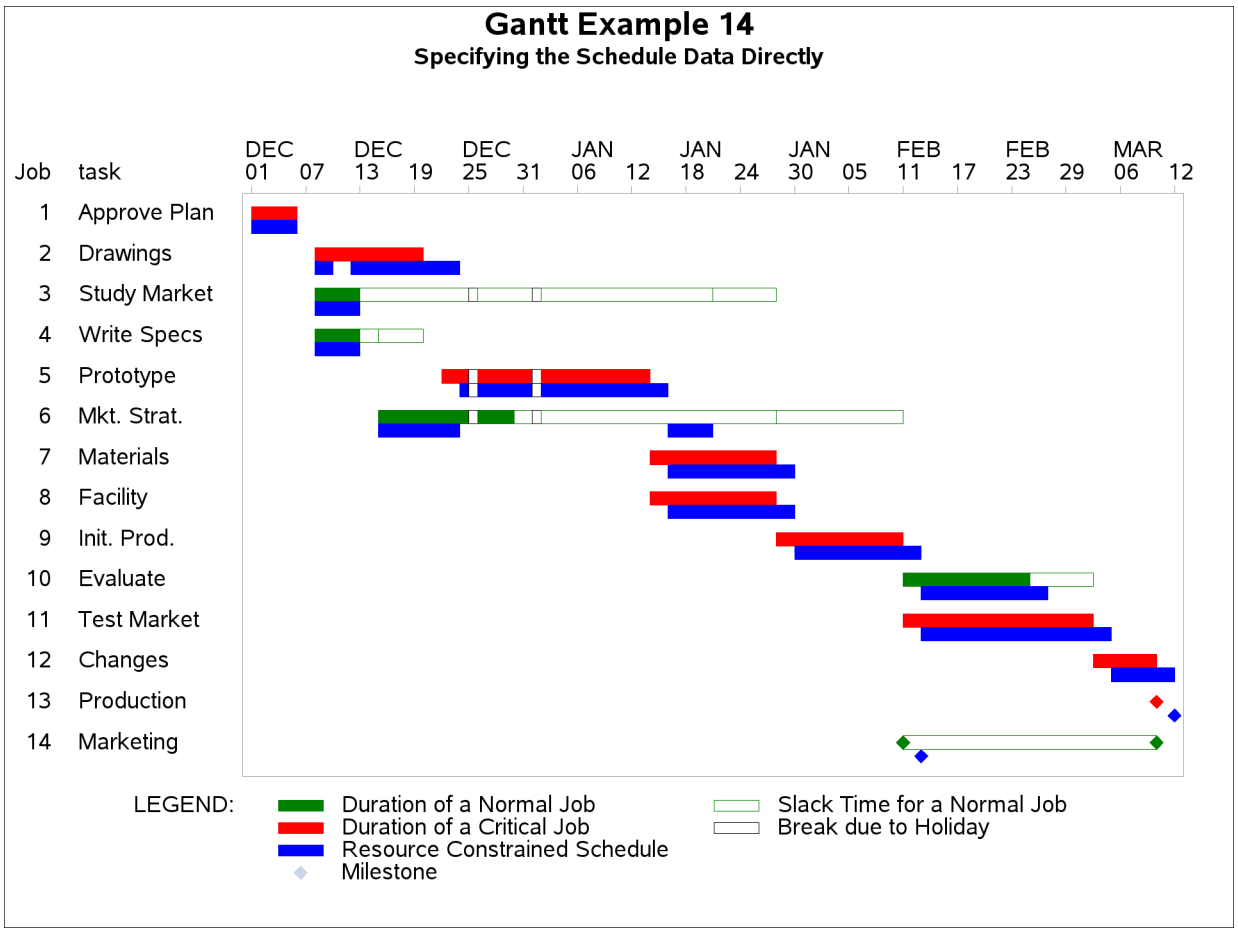
An example of the direct type of input can be seen in Example 8.10 which illustrates plotting of the actual schedule. In Example 8.10, PROC CPM was used to compute the predicted early/late schedule, which was then stored in the SAVEH data set. However, information about the actual schedule, which was provided in the COMPLETE data set, was not used by PROC CPM. Instead, this information was merged with the SAVEH data set to form WIDGELA, the Schedule data set for PROC GANTT. The variables representing the actual start and finish were identified to PROC GANTT using the A_START= and A_FINISH= options, respectively, in the CHART statement. The identification of the variables would not have been necessary if the start and finish variable names were A_START and A_FINISH, respectively.

The following example draws a Gantt chart of the early, late, and resource-constrained schedules for the widget manufacturing project. The schedule information is held in the WIDGDIR data set. The WIDGDIR data set contains the variables TASK, SEGMT_NO, DUR, RS, RF, E_START, E_FINISH, SDATE, and FDATE. The variable TASK identifies the activity. E_START and E_FINISH are recognized as the default names of the early start and early finish variables, respectively. The variables SDATE and FDATE define the late start and late finish times, respectively. Since these are not the default names for the late schedule variables, they need to be identified as such by specifying the LS= and LF= options (or the L_START= and L_FINISH= options) in the CHART statement. The variables RS and RF represent the resource-constrained start and finish times, respectively. As with the late schedule, these variables need to be identified to PROC GANTT by specifying the SS= and SF= options (or the S_START= and S_FINISH= options) in the CHART statement. Further, the SEGMT_NO variable identifies the segment number of the resource constrained schedule that an observation corresponds to since these are activities that start and stop multiple times before completion. The ZDUR variable is identified as a zero duration indicator by specifying the DUR= option in the CHART statement. Since ZDUR is zero for 'Production' and 'Marketing,' these activities are represented by milestones on the chart. Notice that although all the other activities have a value of '1' for the ZDUR variable, any nonzero value will produce the same result. This is due to the fact that PROC GANTT only uses this variable as an *indicator* of whether the activity has zero duration or not, in contrast to the interpretation of the DURATION variable in PROC CPM.

```
options ps=60 ls=100;

title h=1.75 'Gantt Example 14';

 /* Activity-on-Node representation of the project */
data widgdir;
   format task $12. rs rf e_start e_finish sdate fdate date7.;
   input task & segmt_no zdur rs & date7. rf & date7.
         e_start & date7. e_finish & date7.
         sdate & date7.  fdate & date7.;
   datalines;
Approve Plan  . 1 01DEC03  05DEC03  01DEC03  05DEC03  01DEC03  05DEC03
Drawings      . 1 08DEC03  23DEC03  08DEC03  19DEC03  08DEC03  19DEC03
Drawings      1 1 08DEC03  09DEC03  08DEC03  19DEC03  08DEC03  19DEC03
Drawings      2 1 12DEC03  23DEC03  08DEC03  19DEC03  08DEC03  19DEC03
Study Market  . 1 08DEC03  12DEC03  08DEC03  12DEC03  21JAN04  27JAN04
Write Specs   . 1 08DEC03  12DEC03  08DEC03  12DEC03  15DEC03  19DEC03
Prototype     . 1 24DEC03  15JAN04  22DEC03  13JAN04  22DEC03  13JAN04
Mkt. Strat.   . 1 15DEC03  20JAN04  15DEC03  29DEC03  28JAN04  10FEB04
Mkt. Strat.   1 1 15DEC03  23DEC03  15DEC03  29DEC03  28JAN04  10FEB04
Mkt. Strat.   2 1 16JAN04  20JAN04  15DEC03  29DEC03  28JAN04  10FEB04
Materials     . 1 16JAN04  29JAN04  14JAN04  27JAN04  14JAN04  27JAN04
Facility      . 1 16JAN04  29JAN04  14JAN04  27JAN04  14JAN04  27JAN04
Init. Prod.   . 1 30JAN04  12FEB04  28JAN04  10FEB04  28JAN04  10FEB04
Evaluate      . 1 13FEB04  26FEB04  11FEB04  24FEB04  18FEB04  02MAR04
Test Market   . 1 13FEB04  04MAR04  11FEB04  02MAR04  11FEB04  02MAR04
Changes       . 1 05MAR04  11MAR04  03MAR04  09MAR04  03MAR04  09MAR04
Production    . 0 12MAR04  12MAR04  10MAR04  10MAR04  10MAR04  10MAR04
Marketing     . 0 13FEB04  13FEB04  11FEB04  11FEB04  10MAR04  10MAR04
;
```

```
data holdata;
   format hol date7.;
   input hol & date7.;
   datalines;
25dec03
01jan04
;

/* set up required pattern statements */
pattern1 c=green v=s;    /* duration of a non-critical activity  */
pattern2 c=green v=e;    /* slack time for a noncrit. activity   */
pattern3 c=red v=s;      /* duration of a critical activity      */
pattern4 c=magenta v=e;  /* slack time for a supercrit. activity */
pattern5 c=magenta v=s;  /* duration of a supercrit. activity    */
pattern6 c=cyan v=s;     /* actual duration of an activity       */
pattern7 c=black v=e;    /* break due to a holiday               */
pattern8 c=blue v=s;     /* resource schedule of activity        */
pattern9 c=brown v=s;    /* baseline schedule of activity        */

title2 h=1.25 'Specifying the Schedule Data Directly';

proc gantt data=widgdir holidata=holdata;
   chart / holiday=(hol) dur=zdur
           ss=rs sf=rf ls=sdate lf=fdate
           height=1.5 pcompress;
   id task;
   run;
```

Output 8.14.1 Specifying the Schedule Data Directly

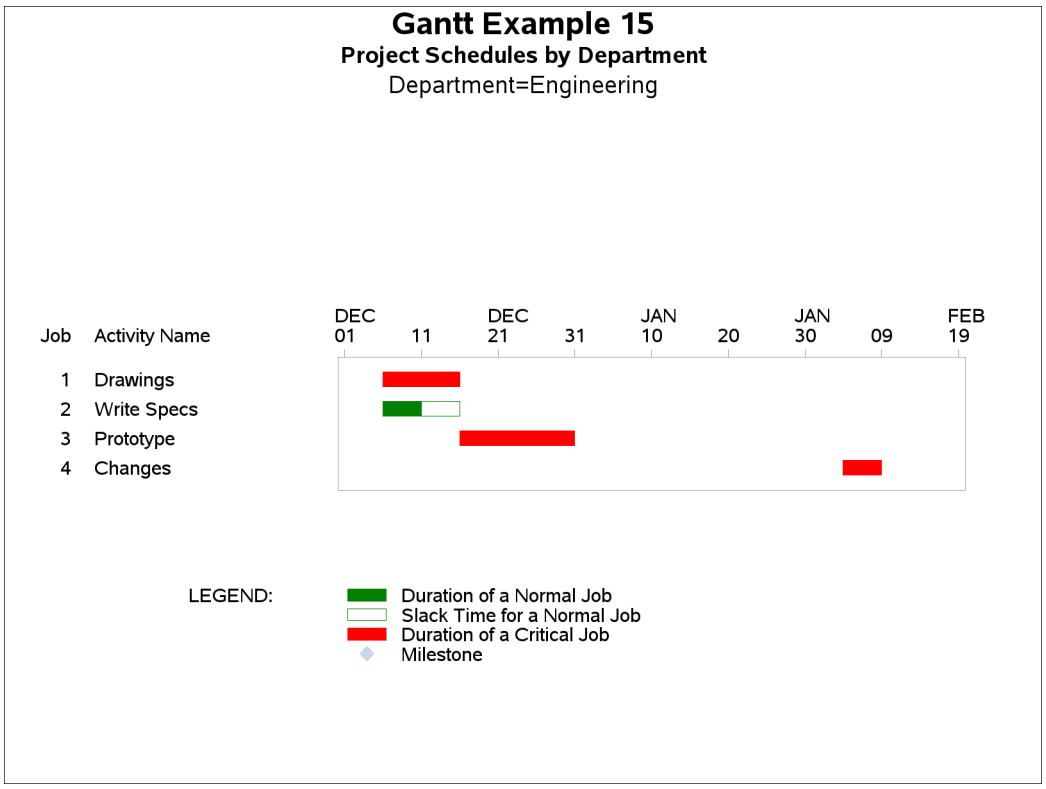


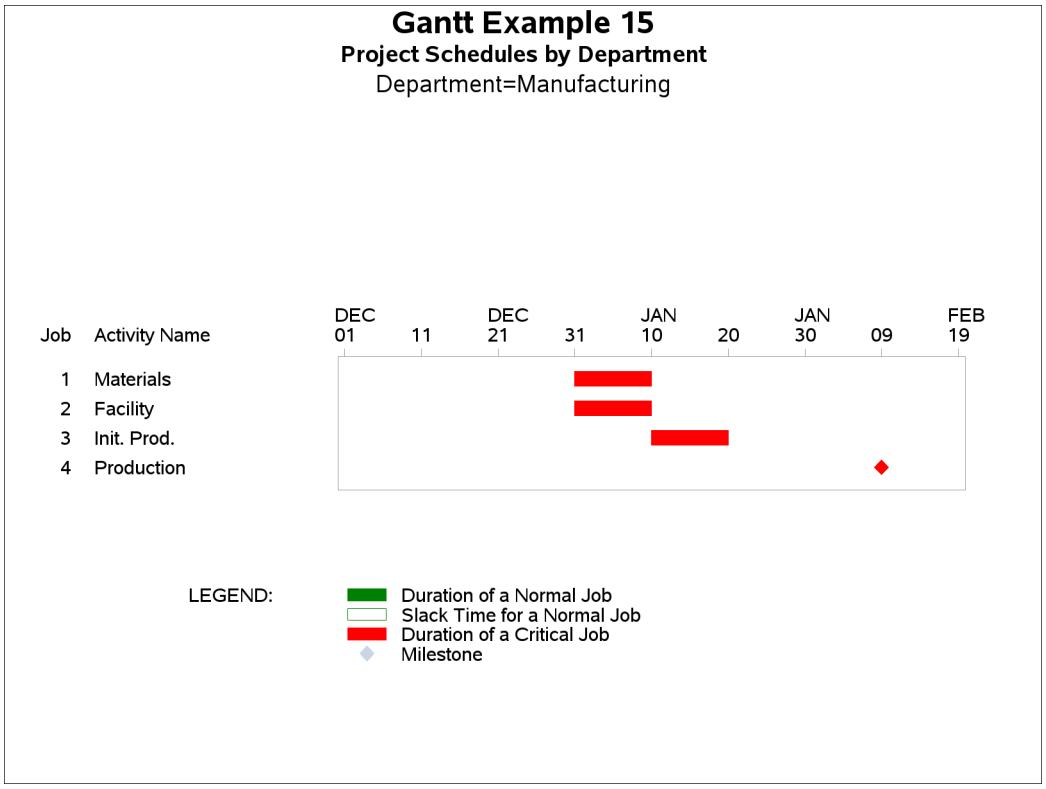
Example 8.15: BY Processing

Every activity in the widget manufacturing project is carried out by one of five departments: Planning, Engineering, Marketing, Manufacturing, and Testing. The DETAILS data set in Example 8.6 identifies the department responsible for each activity. Thus, the project can be thought of as made up of five smaller subprojects, a subproject being the work carried out by a department. A foreseeable need of the project manager and every department is a separate Gantt chart for each subproject. This example uses the WIDGETN data set from Example 4.1, which is formed by merging the WIDGET data set with the DETAILS data set. After scheduling the master project using PROC CPM with DEPT as an ID variable, the Schedule data set is sorted by department name and early start time. The GANTT procedure is then invoked with the variable DEPT specified in the BY statement to obtain individual Gantt charts for each subproject. The Gantt charts for the five different subprojects are shown in Output 8.15.1. The MINDATE= and MAXDATE= options have been specified to ensure a consistent date range across projects. Notice that the TITLE2 statement uses the text substitution option #BYVAR*n*, which substitutes the name of the *n*th BY variable. The BY-LINE that appears below the titles identifies the current values of the BY variables. You can suppress this using the NOBYLINE option in an OPTION statement or the HBY option in a GOPTIONS statement. The SPLIT= option is specified to prevent the TASK variable label from being split on the embedded blank.

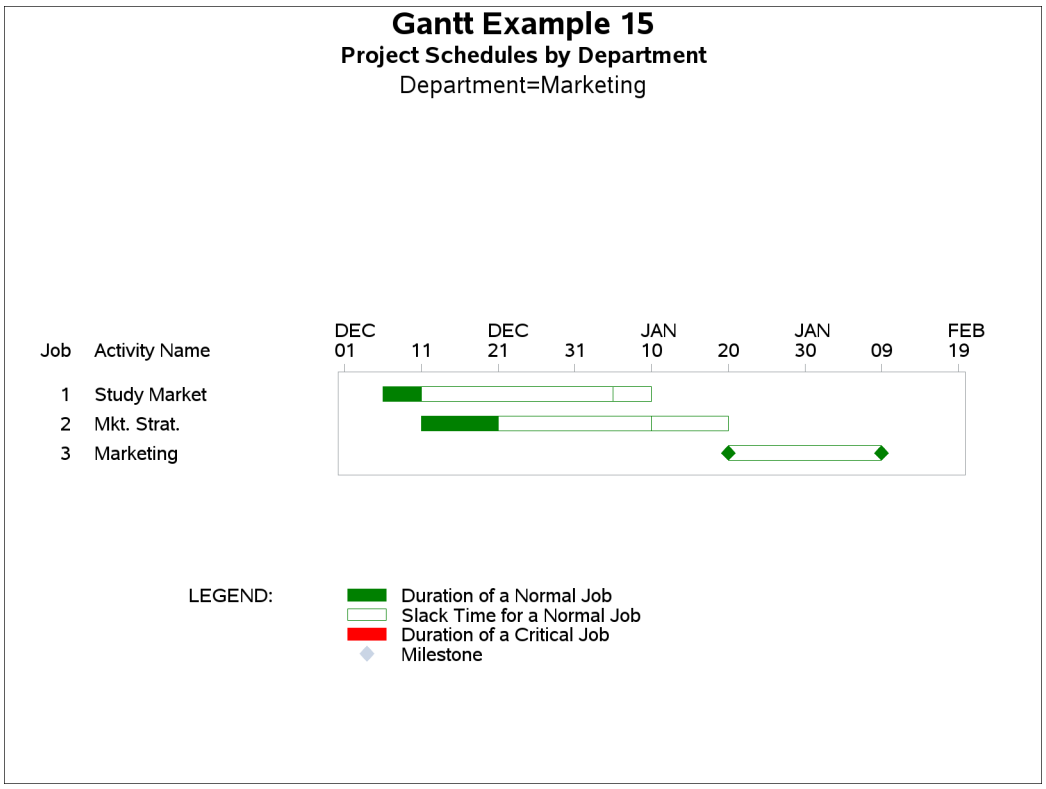
```
data widgetn;
   label task = "Activity Name";
   merge widget details;
   run;

proc cpm date='01dec03'd data=widgetn;
   activity task;
   duration days;
   successor succ1 succ2 succ3;
   id dept;
   run;

proc sort;
   by dept e_start;
   run;


proc gantt split='/';
   chart / pcompress scale=1 dur=days height=1.2
           mindate='01dec03'd maxdate='11feb04'd;
   by dept;
   id task;
   run;
```

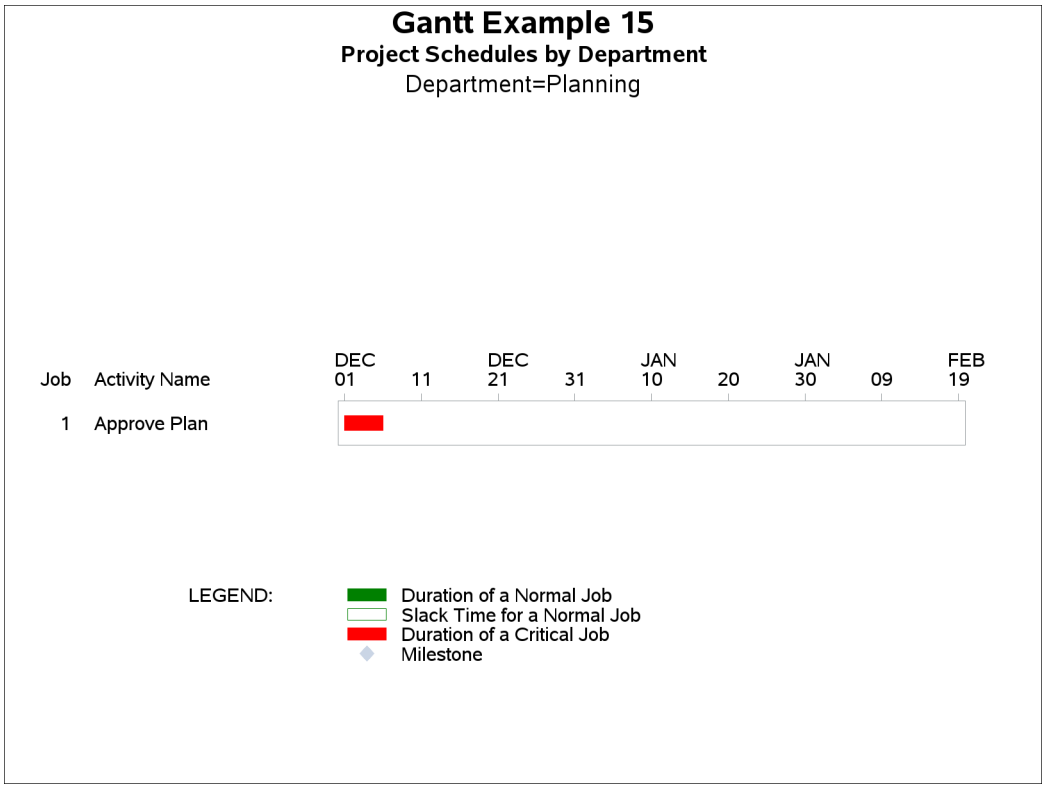
Output 8.15.1 Using BY Processing for Separate Gantt Charts



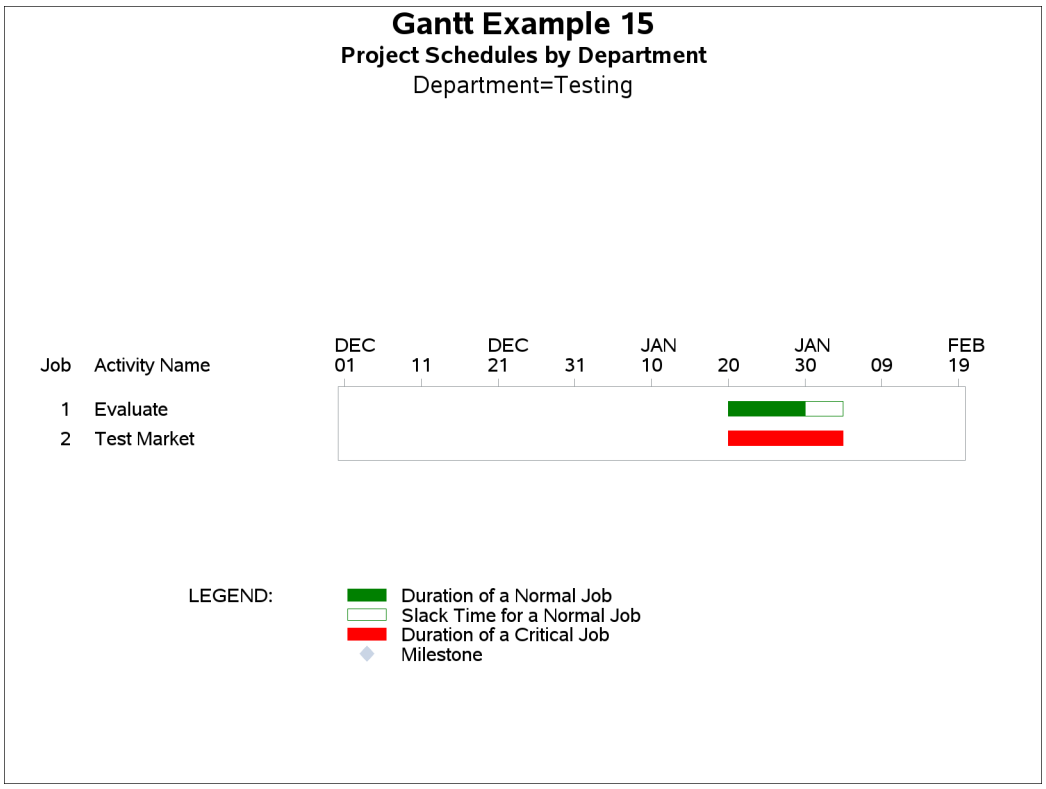


Output 8.15.1 *continued*





Output 8.15.1 *continued*



Example 8.16: Gantt Charts by Persons

Now suppose that you want to obtain individual Gantt charts for two people (Thomas and William) working on the widget manufacturing project. The data set WIDGBYGP, displayed in Output 8.16.1, contains two new variables, THOMAS and WILLIAM. Each variable has a value '1' for activities in which the person is involved and a missing value otherwise. Thus, a value of '1' for the variable THOMAS in observation number 2 indicates that Thomas is working on the activity 'Drawings.'

PROC CPM is used to schedule the project to start on December 1, 2003. A data set named PERSONS is created containing one observation per activity per person working on that activity and a new variable named PERSON containing the name of the person to which the observation pertains. For example, this new data set contains two observations for the activity 'Write Specs,' one with PERSON='Thomas' and the other with PERSON='William,' and no observation for the activity 'Approve Plan.' This data set is sorted by PERSON and E_START, and displayed in Output 8.16.2. PROC GANTT is next invoked with a BY statement to obtain individual charts for each person. The resulting Gantt charts are shown in Output 8.16.3. The BY-LINE is suppressed by specifying the NOBYLINE option in an OPTIONS statement and the name of the person corresponding to the chart is displayed in the subtitle by using the #BYVAL substitution in the TITLE2 statement.

Output 8.16.1 Data Set WIDGBYGP

Data widgbyp

Obs	task	days	tail	head	thomas	william
1	Approve Plan	5	1	2	.	.
2	Drawings	10	2	3	1	.
3	Study Market	5	2	4	.	.
4	Write Specs	5	2	3	1	1
5	Prototype	15	3	5	1	1
6	Mkt. Strat.	10	4	6	.	.
7	Materials	10	5	7	.	1
8	Facility	10	5	7	.	1
9	Init. Prod.	10	7	8	1	.
10	Evaluate	10	8	9	1	1
11	Test Market	15	6	9	.	.
12	Changes	5	9	10	1	.
13	Production	0	10	11	.	1
14	Marketing	0	6	12	.	.
15	Dummy	0	8	6	.	.

```
title h=1.75 'Gantt Example 16';

proc cpm data=widgbyp date='1dec03'd;
   tailnode tail;
   duration days;
   headnode head;
   id task thomas william;
run;

data persons;
   set _last_;
   if william^=. then do;
      person='William';
      output;
      end;
   if thomas^=. then do;
      person='Thomas';
      output;
      end;
   drop thomas william;
run;

proc sort data=persons;
   by person e_start;
run;

title2 'Data PERSONS';
proc print data=persons;
   run;
```

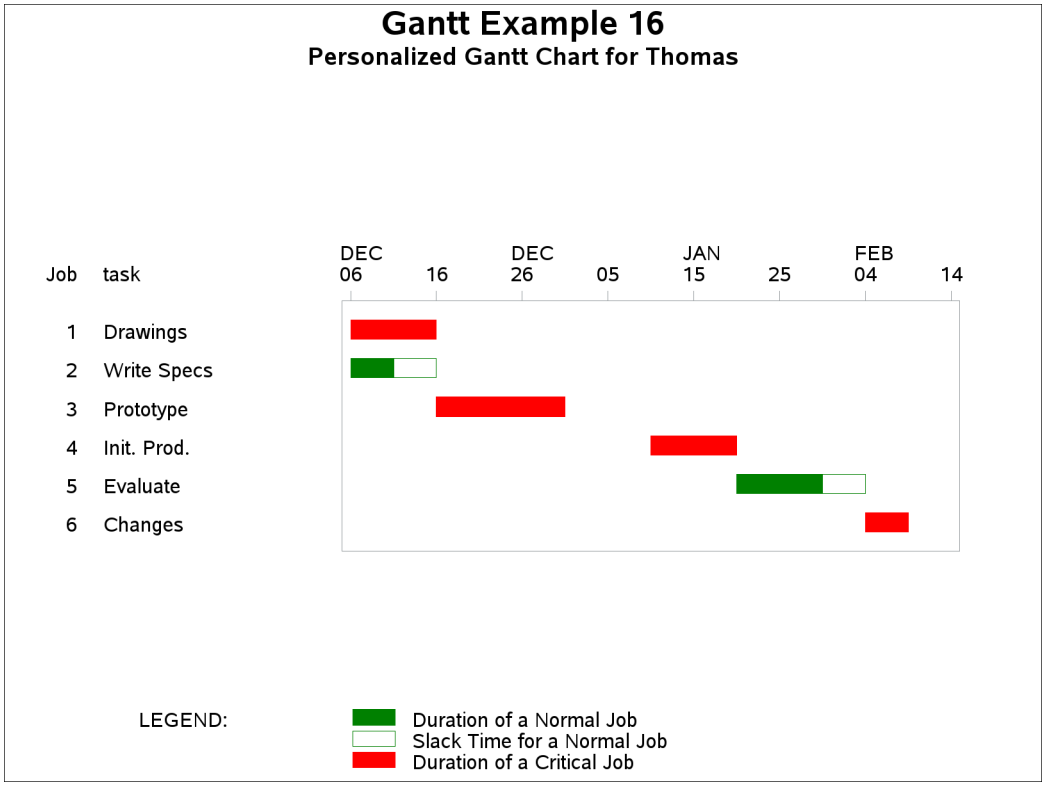
```
/* suppress byline */
options nobyline;

goptions hpos=120 vpos=40 htext=1.1;

title2 h=1.25 'Personalized Gantt Chart for #BYVAL(person)';
proc gantt data=persons;
   chart / pcompress;
   by person;
   id task;
   run;
```

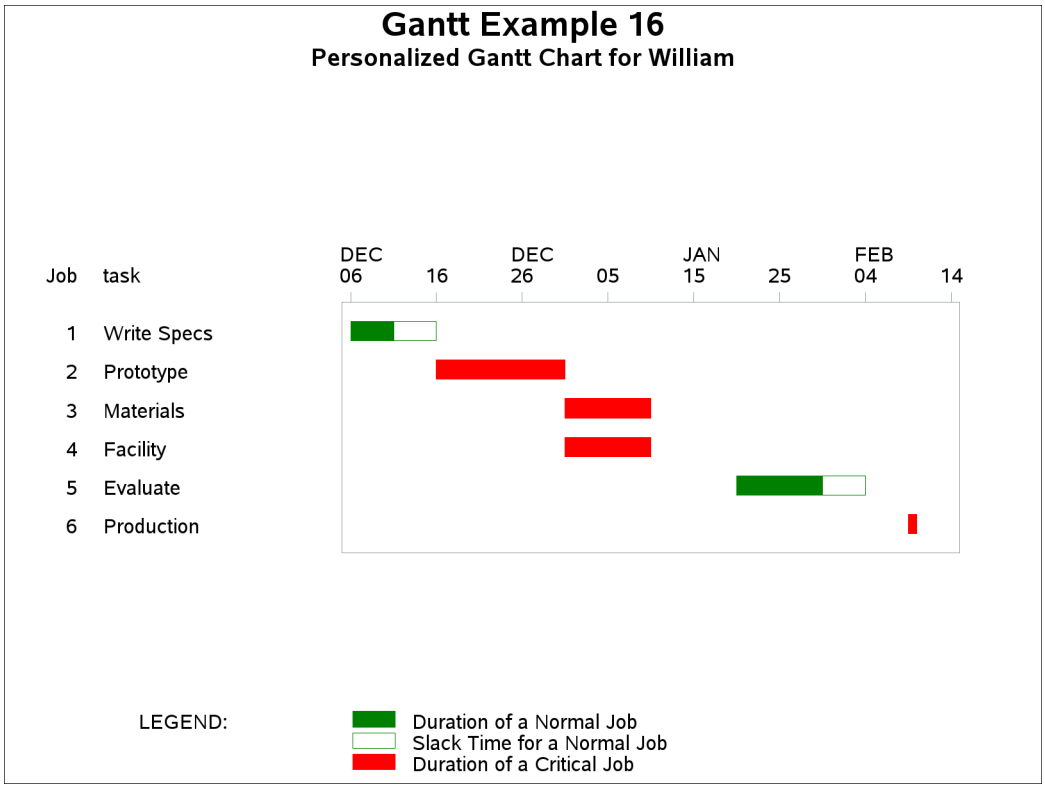
Output 8.16.2 Data Set PERSONS

Gantt Example 16
Data PERSONS

Obs	tail	head	days	task	E_START	E_FINISH	L_START	L_FINISH	T_FLOAT	F_FLOAT	person
1	2	3	10	Drawings	06DEC03	15DEC03	06DEC03	15DEC03	0	0	Thomas
2	2	3	5	Write Specs	06DEC03	10DEC03	11DEC03	15DEC03	5	5	Thomas
3	3	5	15	Prototype	16DEC03	30DEC03	16DEC03	30DEC03	0	0	Thomas
4	7	8	10	Init. Prod.	10JAN04	19JAN04	10JAN04	19JAN04	0	0	Thomas
5	8	9	10	Evaluate	20JAN04	29JAN04	25JAN04	03FEB04	5	5	Thomas
6	9	10	5	Changes	04FEB04	08FEB04	04FEB04	08FEB04	0	0	Thomas
7	2	3	5	Write Specs	06DEC03	10DEC03	11DEC03	15DEC03	5	5	William
8	3	5	15	Prototype	16DEC03	30DEC03	16DEC03	30DEC03	0	0	William
9	5	7	10	Materials	31DEC03	09JAN04	31DEC03	09JAN04	0	0	William
10	5	7	10	Facility	31DEC03	09JAN04	31DEC03	09JAN04	0	0	William
11	8	9	10	Evaluate	20JAN04	29JAN04	25JAN04	03FEB04	5	5	William
12	10	11	0	Production	09FEB04	09FEB04	09FEB04	09FEB04	0	0	William

Output 8.16.3 Gantt Charts by Person



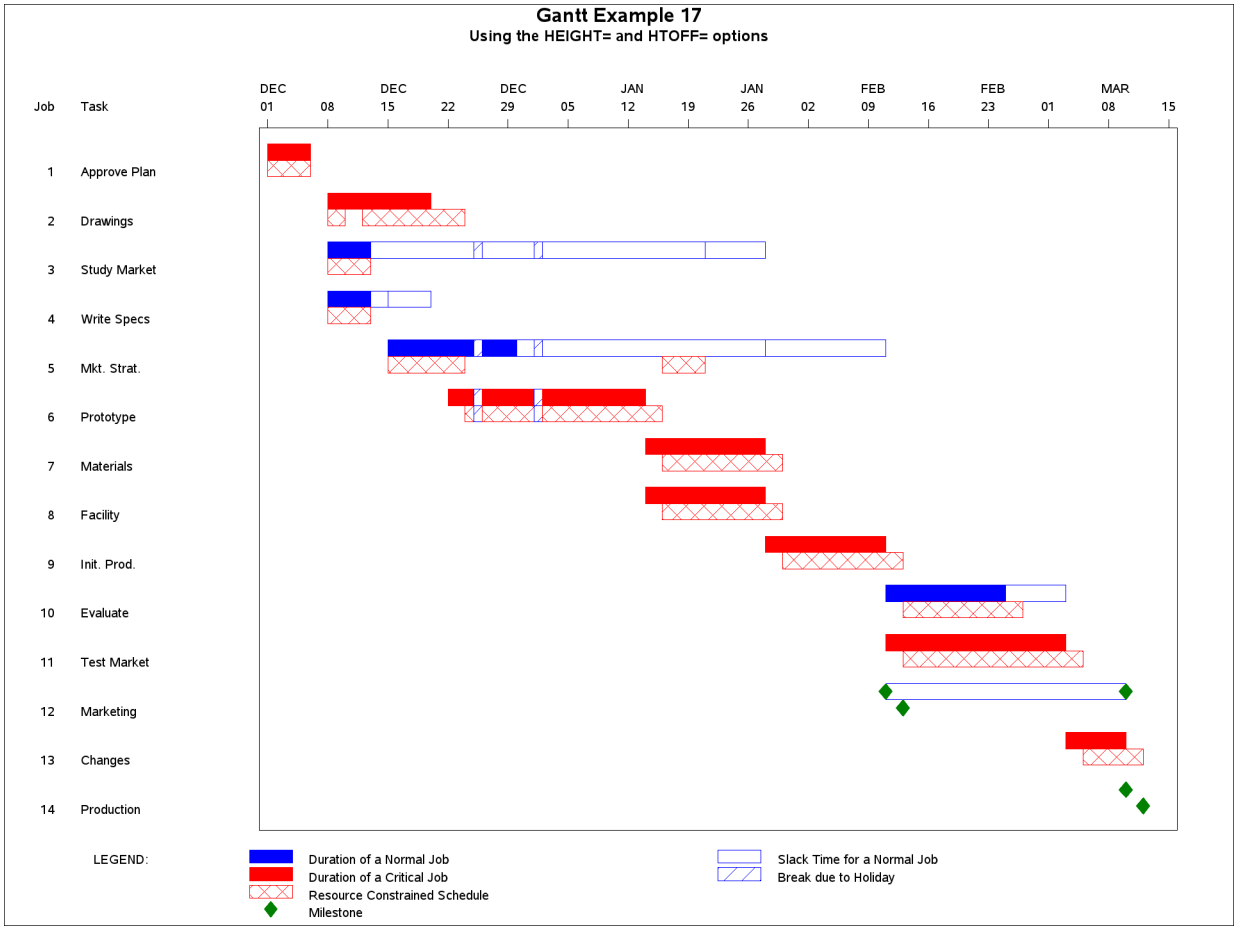


Example 8.17: Using the HEIGHT= and HTOFF= Options

The following example illustrates two options that control the height and positioning of all text produced by PROC GANTT. The data used for this example come from Example 8.13, which illustrates plotting of the resource-constrained schedule. PATTERN statements are specified in order to identify the fill patterns for the different schedule types and holidays. The resource-constrained schedule is drawn using the fill pattern from the eighth PATTERN statement. The HEIGHT= option is set to 2, indicating that the height of all text produced by PROC GANTT be equal to the height of two activity bars. This text includes activity text, legend text, and axis labeling text. The HTOFF= option is also set to 2, which drops the font baseline of the activity text by the height of one schedule bar causing the font baseline to be positioned at the bottom of the resource-constrained schedule bar. The resulting Gantt chart is displayed in Output 8.17.1.

```
title 'Gantt Example 17';

* set up required pattern statements;

pattern1 c=blue v=s;   /* duration of a non-critical activity   */
pattern2 c=blue v=e;   /* slack time for a noncrit. activity    */
pattern3 c=red v=s;    /* duration of a critical activity       */
pattern4 c=red v=e;    /* slack time for a supercrit. activity  */
pattern5 c=red v=r2;   /* duration of a supercrit. activity     */
pattern6 c=cyan v=s;   /* actual duration of an activity        */
pattern7 c=blue v=r1;  /* break due to a holiday                */
pattern8 c=red v=x1;   /* resource schedule of activity         */
pattern9 c=blue v=s;   /* baseline schedule of activity         */

* set vpos to 50 and hpos to 100;
goptions vpos=50 hpos=100;

title2 'Using the HEIGHT= and HTOFF= options';

* draw Gantt chart using height and htoff equal to 2;
proc gantt graphics data=spltschd holidata=holdata;
   chart / holiday=(hol) dur=days compress cmile=green caxis=black
           height=2 htoff=2;
   id task;
   run;
```

Output 8.17.1 Using the HEIGHT= and HTOFF= options

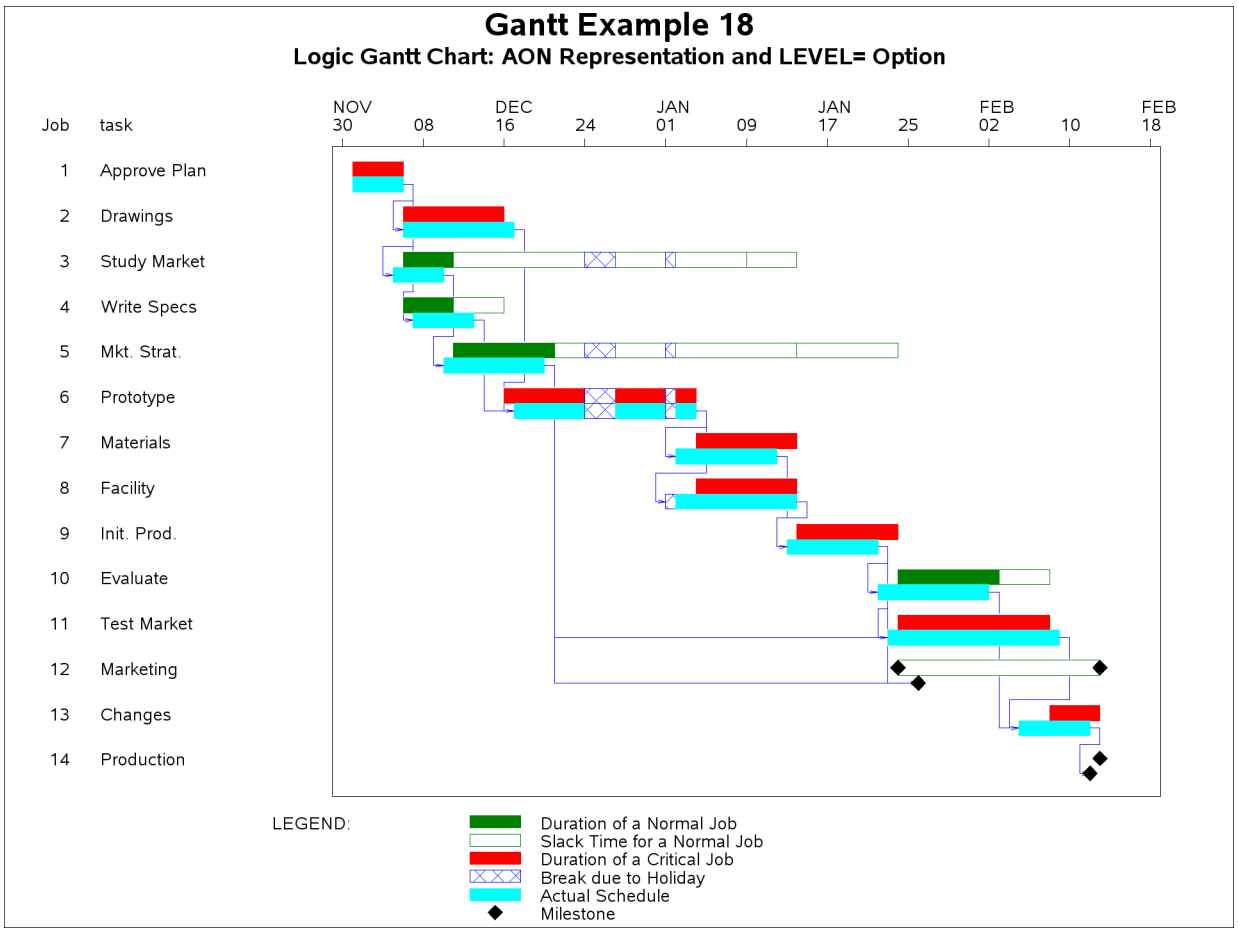


Example 8.18: Drawing a Logic Gantt Chart Using AON Representation

This example uses the data of Example 8.10, which illustrates the drawing of the actual schedule. The ACTIVITY= and SUCCESSOR= options are specified in the CHART statement to define the precedence relationships using the AON format to PROC GANTT. Since no LAG= option is specified, the lag type of each connection is assumed to be Finish-to-Start (FS). In this case, the precedence defining variables exist in the WIDGELA data set; however, this is not a requirement. The precedence defining variables can belong to a different data set as long as the ACTIVITY variable is common to both data sets and the PRECDATA= option, identifying the Precedence data set, is specified in the PROC GANTT statement. Setting the LEVEL= option to 2 causes the actual schedule bar to be used as the logic bar; that is, PROC GANTT draws the precedence connections with respect to the actual schedule. By default, the precedence connections are drawn with respect to the first bar. The color of the precedence connections is specified with the CPREC= option in the CHART statement. You can change the line style and line width of the precedence connections by specifying the LPREC= and WPREC= options in the CHART statement. The resulting Gantt chart is shown in Output 8.18.1.

```
title h=1.75 'Gantt Example 18';

title2 h=1.25 'Logic Gantt Chart: AON Representation and LEVEL= Option';

* sort the data;
proc sort;
   by e_start;
   run;

* set graphics options;
goptions vpos=50 hpos=100 htext=1.2;

* draw the logic Gantt chart;
proc gantt graphics data=widgela holidata=holidays;
   chart / holiday=(holiday) holifin=(holifin)
           a_start=sdate a_finish=fdate dur=days
           compress
           cmile=black
           activity=task successor=(succ1-succ3)
           caxis=black
           level=2
           cprec=blue;
   id task;
   run;
```

Output 8.18.1 Drawing a Logic Gantt Chart Using AON Representation



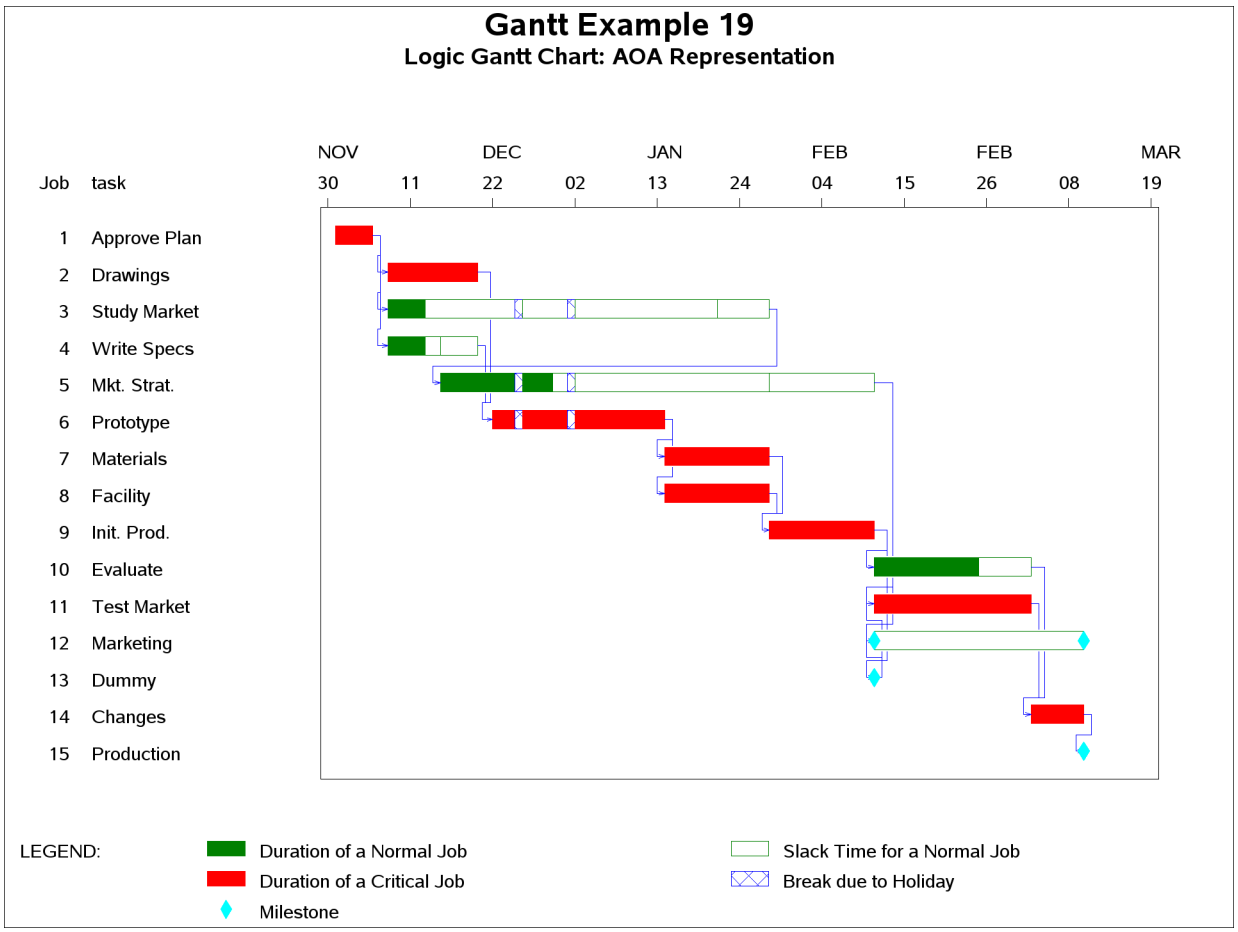
Example 8.19: Specifying the Logic Control Options

This example illustrates four options that control the routing of a precedence connection from an activity to its successor on the logic Gantt chart. The example also illustrates the drawing of a Logic Gantt chart using the Activity-on-Arc format.

The Activity data set for PROC CPM is the WIDGETA data set from Example 4.2, which defines the widget manufacturing project in AOA format. The project is scheduled subject to weekends, and the holidays are defined in the HOLDATA data set. The resulting schedule is stored in the output data set SAVEHP. The GANTT procedure is next invoked to produce a Logic Gantt chart by specifying the HEAD= and TAIL= options in the CHART statement. The resulting Logic Gantt chart is shown in Output 8.19.1.

```
title h=1.75 'Gantt Example 19';

data holdata;
   format hol date7.;
   input hol & date7.;
   datalines;
25dec03
01jan04
;

* schedule the project subject to holidays and weekends;
proc cpm data=widgeta holidata=holdata out=savehp
         date='1dec03'd interval=weekday;
   tailnode tail;
   headnode head;
   duration days;
   holiday  hol;
   id task dept descrpt;
   run;

* sort the schedule by the early start date;
proc sort;
   by e_start;
   run;

* set background to white and text to black;
goptions cback=white ctext=black;

* set additional graphics options;
goptions vpos=50 hpos=100 htext=1.1;

 * plot the logic Gantt chart using AOA representation;
title2 h=1.25 'Logic Gantt Chart: AOA Representation';

proc gantt graphics data=savehp holidata=holdata;
   chart / compress cprec=blue caxis=black cmile=cyan
           increment=7 height=1.5
           dur=days holiday=(hol)
           head=head tail=tail;
   id task;
   run;
```

Output 8.19.1 Logic Gantt Chart: AOA Representation

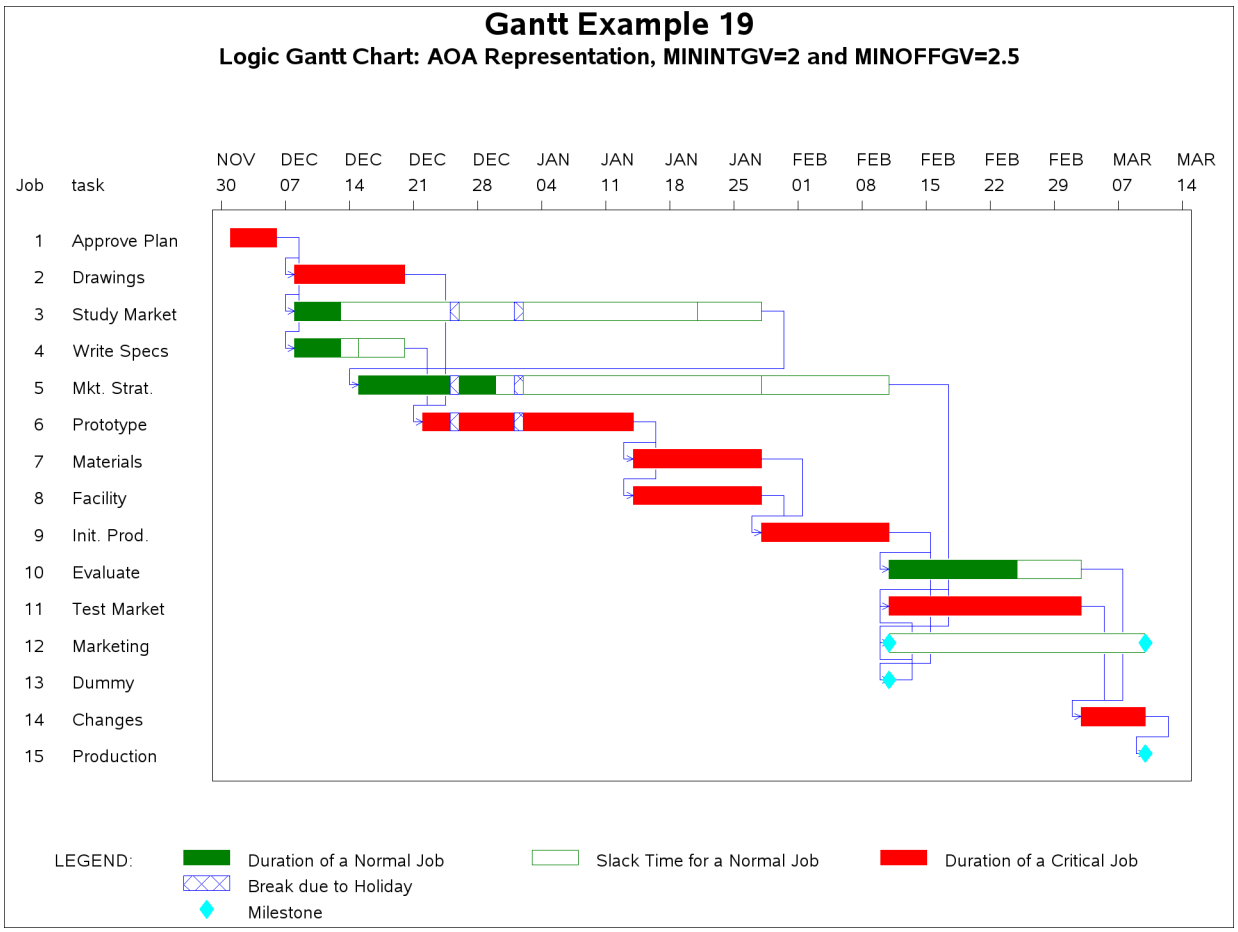


The next invocation of PROC GANTT illustrates the effect of the MININTGV= and MINOFFGV= options, which control placement of the global verticals. The concept of global verticals is explained in the section "Specifying the Logic Options" on page 549. The data sets from the previous invocation of the GANTT procedure remain unchanged. The minimum distance of a global vertical from the end of the bar it is associated with is increased from its default of 1 cell to 2.5 cells by specifying MINOFFGV=2.5. Likewise, the minimum distance between any two global verticals is increased from its default of .75 cells to 2 cells by specifying MININTGV=2.0. The effects of these changes are visible in the resulting Logic Gantt chart shown in Output 8.19.2.

```
goptions htext=1.4;

* illustrate the minintgv and minoffgv options;
title2 h=1.25
   'Logic Gantt Chart: AOA Representation, MININTGV=2 and MINOFFGV=2.5';

proc gantt graphics data=savehp holidata=holdata;
   chart / dur=days holiday=(hol) compress increment=7
           cprec=blue caxis=black cmile=cyan
           head=head tail=tail
           minintgv=2.0 minoffgv=2.5;
   id task;
   run;
```

Notice that now there is greater distance between vertical segments (corresponding to global verticals), and the horizontal segments leaving bars are longer.

Output 8.19.2 Specifying the MININTGV= and MINOFFGV= Options

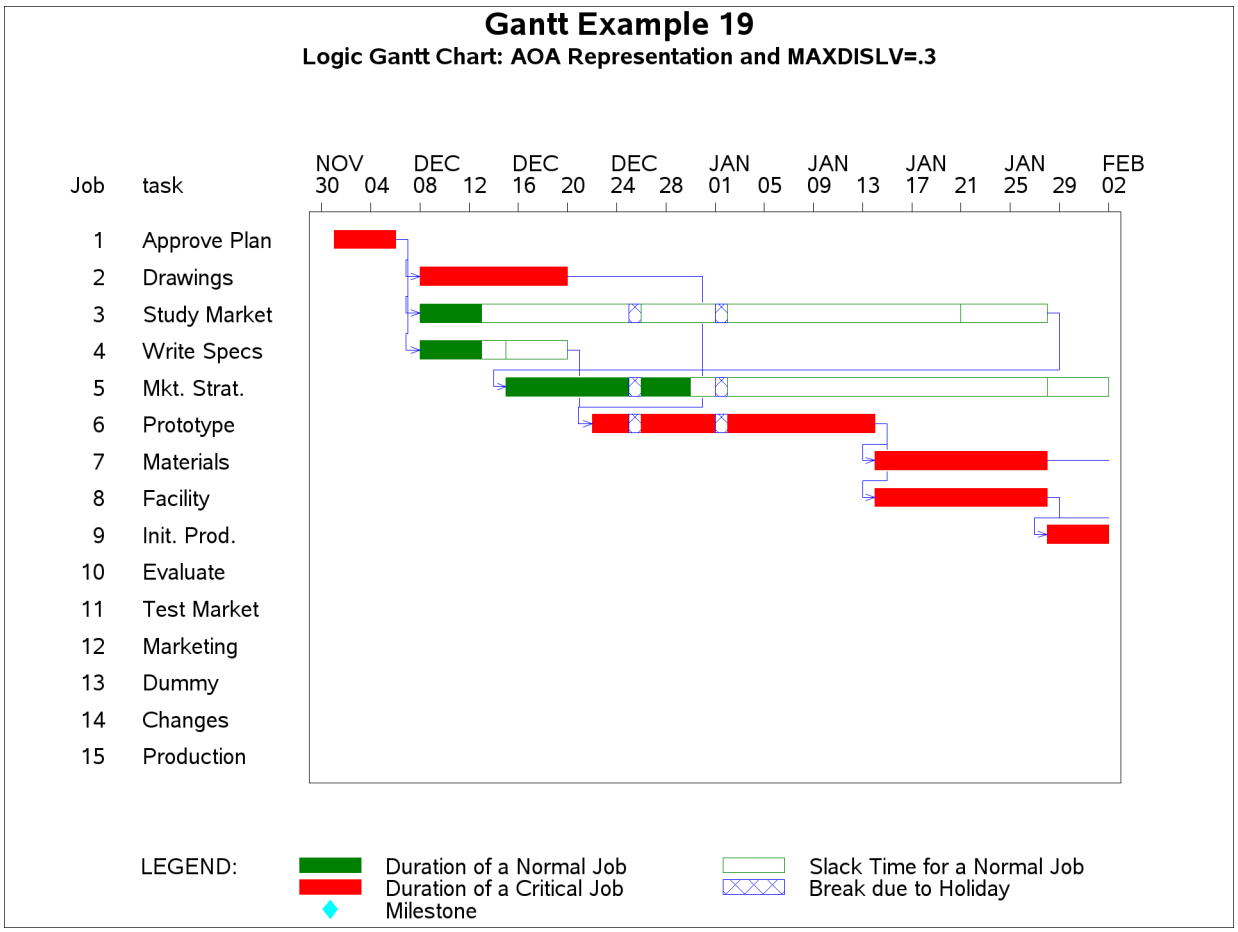


The MAXDATE= option is specified in the remaining Gantt calls in this example in order to focus on the schedule bars of the first few activities in the chart. The next two outputs illustrate the use of the MAXDISLV= option in the CHART statement. The MAXDISLV= option is used as a safeguard to limit the feasible region made available to PROC GANTT for placement of local verticals. The value specified dictates the maximum allowable displacement of the local vertical from its ideal position, that is, at a distance of MINOFFLV= from the end of the bar with which it is associated. However, this ideal position may tend to be positioned too close to a global vertical or even coincide with one. Depending on the cell width, this can result in visual misinterpretation of the Logic Gantt chart. In order to avoid this scenario, you should specify a reasonable value for the MAXDISLV= option to permit a certain amount of freedom for local vertical placement so as to distinguish between local and global verticals. Typically, use of this option is desirable when the value of the MININTGV= option, the minimum distance between global verticals, is relatively much greater than the value of the MAXDISLV= option.

To illustrate, consider the following Gantt call with a large MININTGV= value (10) and a relatively smaller MAXDISLV= value (0.3). Thus, for every local vertical, PROC GANTT has a very small interval that is less than a third of a cell wide in which to place that local vertical regardless of whether a global vertical runs through that interval or not. The result of this constraint is illustrated in the chart shown in Output 8.19.3. The local vertical for 'Drawings' is positioned as far as possible from the global vertical of 'Approve Plan,' but the value of the MAXDISLV= option restricts it from being positioned any further. Visually it is not pleasing, and it is difficult to distinguish the local and global verticals. A similar situation is evident with the local vertical of 'Prototype' and the global vertical of 'Write Specs.'

```
goptions htext=1.2;

* illustrate the maxdislv option;
title2 h=1.25 'Logic Gantt Chart: AOA Representation and MAXDISLV=.3';

proc gantt graphics data=savehp holidata=holdata;
   chart / compress cprec=blue caxis=black cmile=cyan
           dur=days holiday=(hol)
           head=head tail=tail
           maxdislv=.3 minintgv=10
           maxdate='01feb04'd;
   id task;
   run;
```

Output 8.19.3 Specifying the MAXDISLV= Option (I)

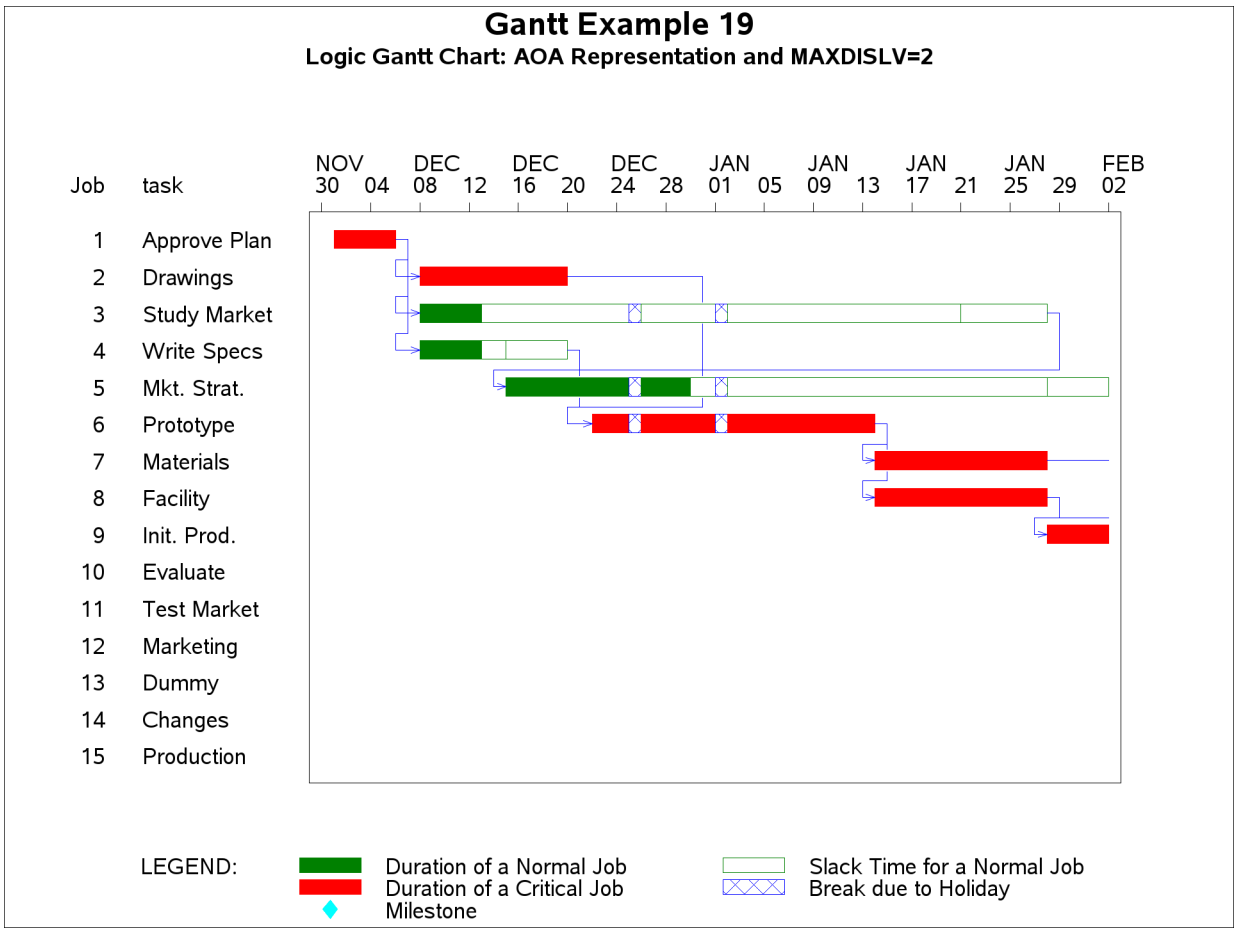


By reducing the value of MAXDISLV= even further, you can produce a chart that gives the appearance of a local vertical overlapping with a global vertical owing to resolution limitations of the display device. Theoretically, by design, this will never be the case. Recall that the value of the MAXDISLV= option is strictly positive and is at least one-tenth of a cell width.

The solution to this problem is to increase the value of the MAXDISLV= option so that the local vertical can be displaced further away from any adjacent global verticals. In the next invocation of PROC GANTT, the value of the MAXDISLV= option is increased to 2, resulting in a Logic Gantt chart in which the local verticals are staggered further away from nearby global verticals. This Gantt chart is displayed in Output 8.19.4.

```
title2 h=1.25 'Logic Gantt Chart: AOA Representation and MAXDISLV=2';

proc gantt graphics data=savehp holidata=holdata;
   chart / compress cprec=blue caxis=black cmile=cyan
           dur=days holiday=(hol)
           head=head tail=tail
           maxdislv=2 minintgv=10
           maxdate='01feb04'd;
   id task;
   run;
```

Output 8.19.4 Specifying the MAXDISLV= Option (II)

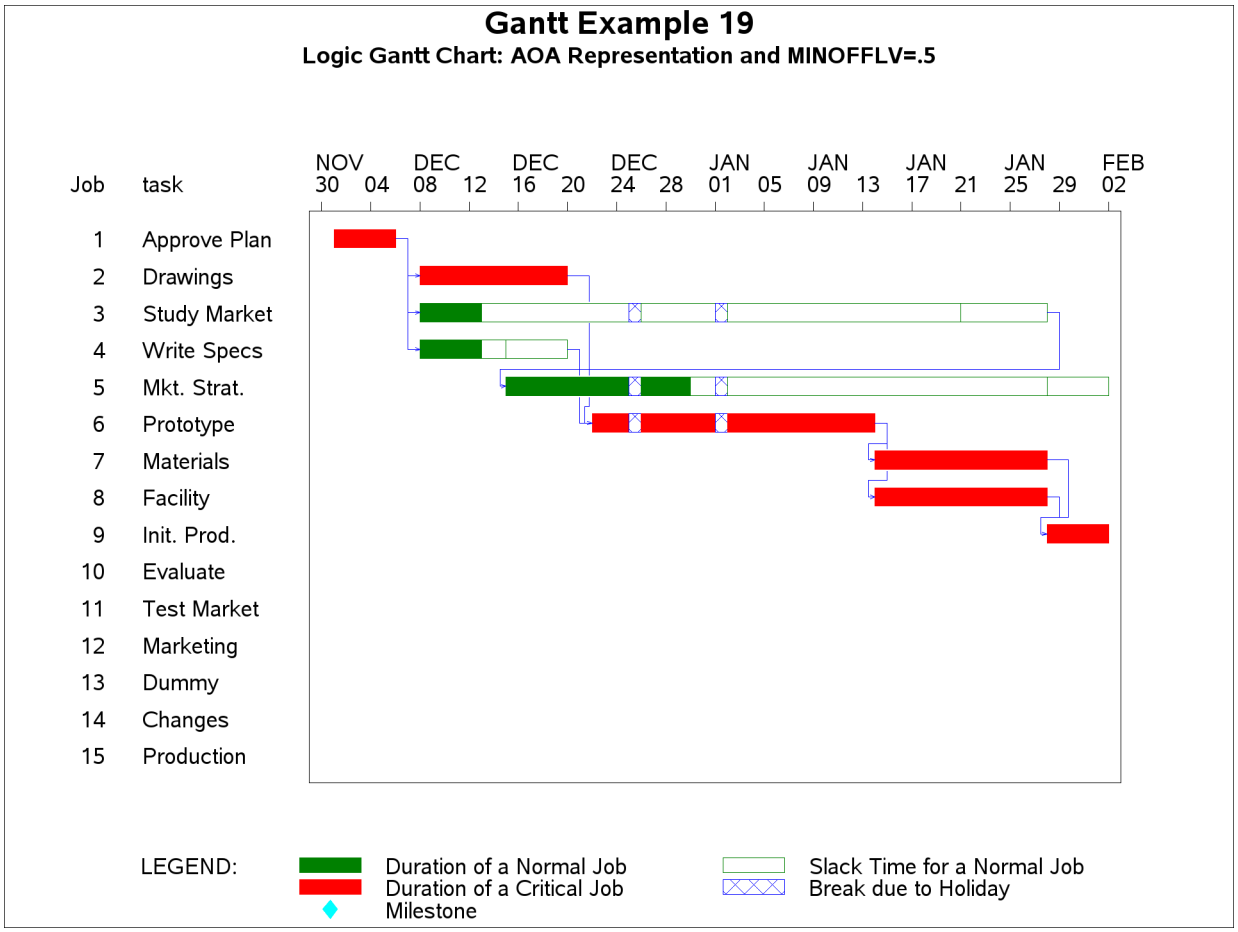


The final Gantt chart in this example illustrates the use of the MINOFFLV= option in the CHART statement. This option specifies the minimum distance of a local vertical from the end of the bar with which it is associated. Although the position corresponding to the MINOFFLV= option is the position of choice for placement of the local vertical, the actual placement can differ from this position owing to the presence of nearby global verticals, as illustrated by Output 8.19.3 and Output 8.19.4. The maximum amount of displacement is determined by the value of the MAXDISLV= option.

In all of the preceding charts in this example, the connection from the activity, 'Approve Plan,' to each of its three successors, 'Drawings', 'Study Market', and 'Write Specs', is a 5-segment connection similar to the type illustrated in Figure 8.14. This is caused by backtracking of the activity's global vertical to the successor's local vertical as described in the section "Controlling the Layout" on page 552. To transform this connection into a 3-segment connection as shown in Figure 8.13, you need to position the local vertical to the right of the global vertical. The following invocation of PROC GANTT achieves this by specifying MINOFFLV=0.5, and the resulting Gantt chart is shown in Output 8.19.5. Notice that this option affects the positioning of *all* local verticals on the chart in contrast to the MAXDISLV= option, which affects only those local verticals that are close to global verticals.

```
* illustrate the minofflv option;
title2 h=1.25
   'Logic Gantt Chart: AOA Representation and MINOFFLV=.5';

proc gantt graphics data=savehp holidata=holdata;
   chart / compress cprec=blue caxis=black cmile=cyan
           dur=days holiday=(hol)
           head=head tail=tail
           minofflv=.5
           maxdate='01feb04'd;
   id task;
   run;
```

Output 8.19.5 Specifying the MINOFFLV= Option



Example 8.20: Nonstandard Precedence Relationships

This example demonstrates the use of nonstandard precedence relationships and specification of the PRECDATA= option in the PROC GANTT statement.

The project and nonstandard precedence relationships are defined by the WIDGLAG2 data set, which is a modification of the WIDGLAG data set that was used in Example 4.11 to illustrate the CPM procedure. The activity and successor variables are represented by the TASK and SUCC variables, respectively, and the lag type of the relationship is defined by the LAGDUR variable. The LAGDUR variable defines the lag type in *keyword_duration_calendar* format for the purpose of passing the information to PROC CPM. Although PROC GANTT accepts this format for a lag variable, it does not use the *duration* and *calendar* values when drawing the connection since the schedule is already computed at this time (presumably by PROC CPM).

As in the WIDGLAG data set, the WIDGLAG2 data set specifies a Start-to-Start lag of nine days between the activity 'Prototype' and its successors, 'Materials' and 'Facility,' and a Finish-to-Start lag of two days between 'Facility' and 'Init. Prod.'. In addition, changes to the widget design are permitted to be made no earlier than six days after in-house evaluation of the product has begun. Furthermore, the Engineering department has to ensure that there will be at least three days available for any changes that need to be carried out after the test market results have come in. These constraints are incorporated in the WIDGLAG data set by setting the value of the LAGDUR variable equal to 'ss_6' for the relationship between 'Evaluate' and 'Changes' and equal to 'ff_3' for the relationship between 'Test Market' and 'Changes.'

The project is scheduled using PROC CPM subject to weekends and the holidays defined in the HOLIDAYS data set. Specifying the COLLAPSE option in the PROC CPM statement ensures that there is one observation per activity. The WIDGLAGH data set is created by deleting the successor variable from the Schedule data set produced by PROC CPM.

Since there is no precedence information contained in the WIDGLAGH data set, specifying DATA=WIDGLAGH in the PROC GANTT statement without the PRECDATA= option produces a nonprecedence Gantt chart. You can produce a Logic Gantt chart by specifying the precedence information using the PRECDATA= option in the PROC GANTT statement as long as the activity variable is common to both the schedule and Precedence data sets.

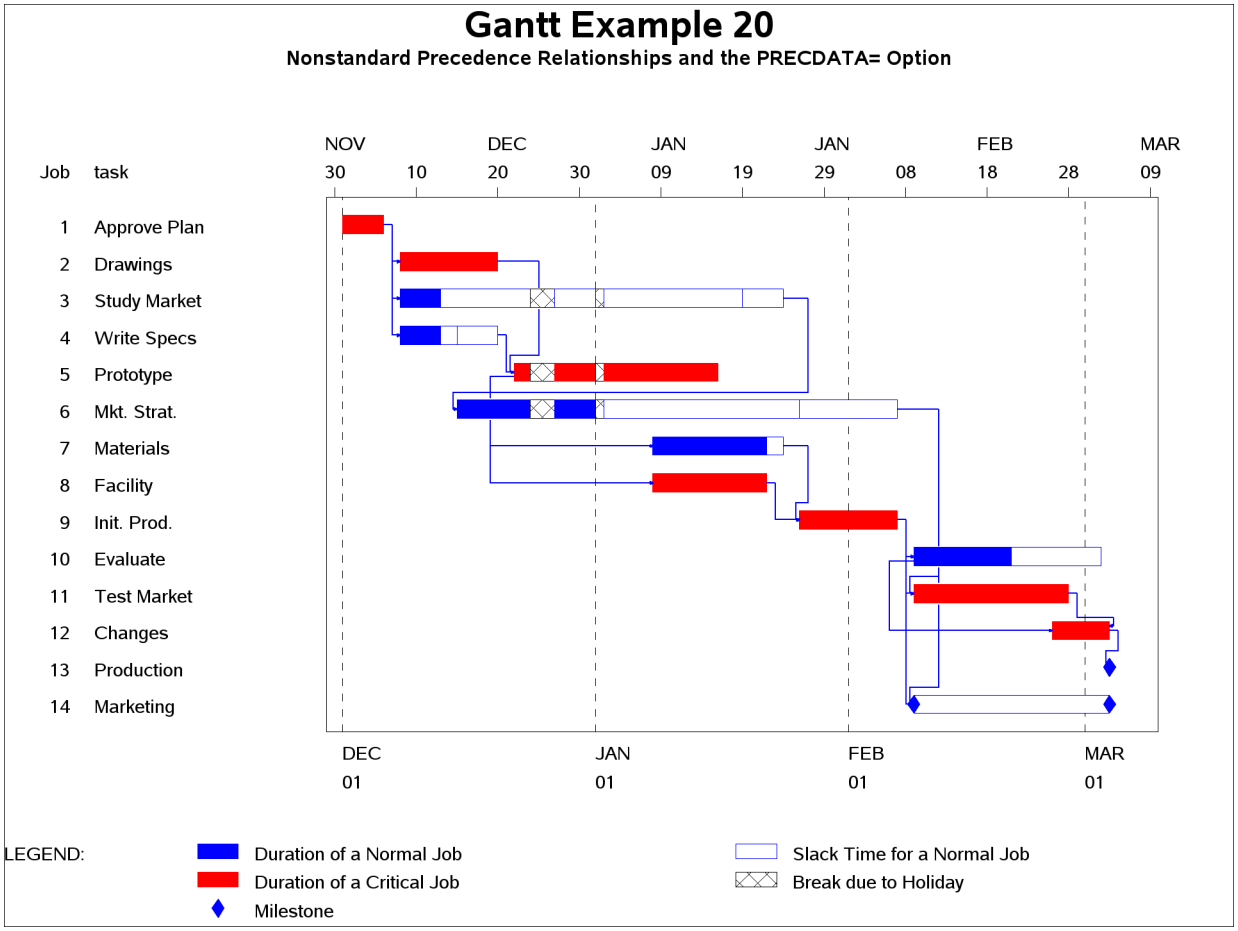
The Gantt chart shown in Output 8.20.1 is produced by specifying PRECDATA= WIDGLAG2. The lag type of the precedence connections is indicated to PROC GANTT using the LAG= option in the CHART statement. The width of the precedence connections is set to 2 with the WPREC= option, and the color of the connections is set to blue using the CPREC= option. The MININTGV= and MINOFFLV= options are specified in the CHART statement in an attempt to minimize the number of 5-segment connections. A reference line with a line style of 2 is drawn at the beginning of every month by using the REF= and LREF= options in the CHART statement.

```
options ps=60 ls=100;

title h=2 'Gantt Example 20';

 /* Activity-on-Node representation of the project with lags */
data widglag2;
   format task $12. succ $12. lagdur $4. ;
   input task & days succ & lagdur $ ;
   datalines;
Approve Plan   5  Drawings      .
```

```
Approve Plan   5  Study Market  .
Approve Plan   5  Write Specs   .
Drawings      10  Prototype     .
Study Market   5  Mkt. Strat.   .
Write Specs    5  Prototype     .
Prototype     15  Materials     ss_9
Prototype     15  Facility      ss_9
Mkt. Strat.   10  Test Market   .
Mkt. Strat.   10  Marketing     .
Materials     10  Init. Prod.   .
Facility      10  Init. Prod.   fs_2
Init. Prod.   10  Test Market   .
Init. Prod.   10  Marketing     .
Init. Prod.   10  Evaluate      .
Evaluate      10  Changes       ss_6
Test Market   15  Changes       ff_3
Changes        5  Production    .
Production     0  .             .
Marketing      0  .             .
;

data holidays;
   format holiday holifin date7.;
   input holiday & date7. holifin & date7. holidur;
   datalines;
24dec03  26dec03  4
01jan04  .        .
;

proc cpm data=widglag2 holidata=holidays date='1dec03'd
         interval=weekday collapse;
   activity task;
   succ     succ / lag = (lagdur);
   duration days;
   holiday  holiday / holifin=(holifin);
   run;

data widglagh;
   set _last_;
   drop succ;
   run;

* set up required pattern statements;
pattern1 c=blue  v=s;  /* duration of a noncrit. activity  */
pattern2 c=blue  v=e;  /* slack time for a noncrit. act.   */
pattern3 c=red   v=s;  /* duration of a critical activity  */
pattern4 c=red   v=e;  /* slack time for a supercrit. act. */
pattern5 c=red   v=r2; /* duration of a supercrit. act.    */
pattern6 c=cyan  v=s;  /* actual duration of an activity   */
pattern7 c=black v=x1; /* break due to a holiday           */

* set graphics options;
goptions vpos=50 hpos=100 htext=1.025;
```

```
title2 c=black h=1.5
   'Nonstandard Precedence Relationships and the PRECDATA= Option';

proc gantt graphics data=widglagh precdata=widglag2
                    holidata=holidays;
   chart / compress dur=days height=1.5
           holiday=(holiday) holifin=(holifin)
           cmile=blue cprec=blue wprec=2
           ref='01dec03'd to '01mar04'd by month
           cref=black lref=2 reflabel caxis=black
           act=task succ=(succ) lag=(lagdur)
           minintgv=2 minofflv=.5;
   id task;
   run;
```

Output 8.20.1 Nonstandard Precedence Relationships



Example 8.21: Using the SAS/GRAPH ANNOTATE= Option

This example illustrates the use of the ANNOTATE= option to add graphics and text to the body of the Gantt chart. The intent of the first invocation of PROC GANTT is to display the resource requirements of each activity on the Gantt chart, while that of the second invocation is to plot the resource usage bar chart for the replenishable resource engineers and the resource availability curve for the consumable resource cost.

The data for this example come from Example 4.15, in which the widget manufacturing project is scheduled using PROC CPM subject to resource constraints. The project network is defined in the WIDGRES data set using AOA format. The number of engineers needed per day per activity is a replenishable resource and is identified by the ENGINEER variable in the WIDGRES data set. The cost incurred per day per activity is a consumable resource and is identified by the ENGCOST variable in the WIDGRES data set. The WIDGRIN data set specifies the resource availabilities for the project. The schedule produced by PROC CPM using the default choice of LST as a heuristic is shown in Output 8.21.1. The following programs assume that the schedule is stored in the WIDGSCH2 data set and that the resource usage is stored in the WIDGROU2 data set.

The Annotate macros are used in this example to simplify the process of creating Annotate observations. The ANNOMAC macro is first used to compile the Annotate macros and make them available for use. The Annotate data set ANNO1 is then created using the Annotate macros. The DCLANNO macro declares all Annotate variables except the TEXT variable, and the SYSTEM macro defines the Annotate reference system. The coordinate system defined here uses *date* for the horizontal scale and *job number* for the vertical scale. The text to be displayed contains the number of engineers required per day and the total cost over the duration of the activity. The LABEL macro is used to annotate the necessary text on the Gantt chart using the BRUSH font.

The GANTT procedure is invoked with the ANNOTATE=ANNO1 specification in the PROC GANTT statement. The resulting Gantt chart is shown in Output 8.21.2. It is important to note that the job number will be used for the vertical scale even if NOJOBNUM is specified in the CHART statement.

Output 8.21.1 Resource Constrained Schedule: Rule = LST

Resource Constrained Schedule: Rule = LST

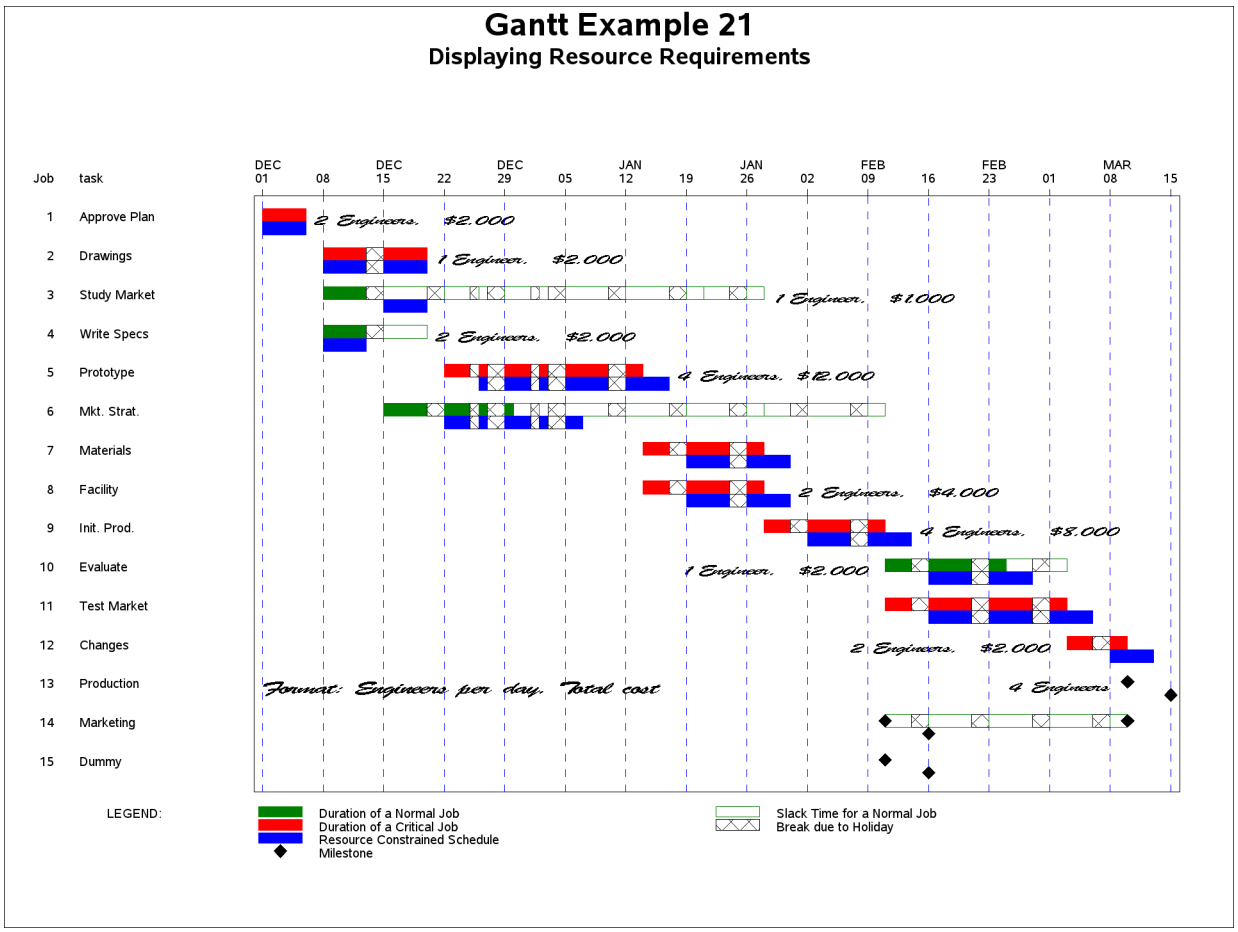
Obs	tail	head	days	task	engineer	engcost	S_START	S_FINISH	E_START	E_FINISH
1	1	2	5	Approve Plan	2	400	01DEC03	05DEC03	01DEC03	05DEC03
2	2	3	10	Drawings	1	200	08DEC03	19DEC03	08DEC03	19DEC03
3	2	4	5	Study Market	1	200	15DEC03	19DEC03	08DEC03	12DEC03
4	2	3	5	Write Specs	2	400	08DEC03	12DEC03	08DEC03	12DEC03
5	3	5	15	Prototype	4	800	26DEC03	16JAN04	22DEC03	13JAN04
6	4	6	10	Mkt. Strat.	.	.	22DEC03	06JAN04	15DEC03	29DEC03
7	5	7	10	Materials	.	.	19JAN04	30JAN04	14JAN04	27JAN04
8	5	7	10	Facility	2	400	19JAN04	30JAN04	14JAN04	27JAN04
9	7	8	10	Init. Prod.	4	800	02FEB04	13FEB04	28JAN04	10FEB04
10	8	9	10	Evaluate	1	200	16FEB04	27FEB04	11FEB04	24FEB04
11	6	9	15	Test Market	.	.	16FEB04	05MAR04	11FEB04	02MAR04
12	9	10	5	Changes	2	400	08MAR04	12MAR04	03MAR04	09MAR04
13	10	11	0	Production	4	800	15MAR04	15MAR04	10MAR04	10MAR04
14	6	12	0	Marketing	.	.	16FEB04	16FEB04	11FEB04	11FEB04
15	8	6	0	Dummy	.	.	16FEB04	16FEB04	11FEB04	11FEB04

Obs	L_START	L_FINISH	R_DELAY	DELAY_R	SUPPL_R
1	01DEC03	05DEC03	0		
2	08DEC03	19DEC03	0		
3	21JAN04	27JAN04	5	engineer	
4	15DEC03	19DEC03	0		
5	22DEC03	13JAN04	3	engineer	
6	28JAN04	10FEB04	0		
7	14JAN04	27JAN04	0		
8	14JAN04	27JAN04	0		
9	28JAN04	10FEB04	0		
10	18FEB04	02MAR04	0		
11	11FEB04	02MAR04	0		
12	03MAR04	09MAR04	0		
13	10MAR04	10MAR04	0		
14	10MAR04	10MAR04	0		
15	11FEB04	11FEB04	0		

```
title c=black h=1.75 'Gantt Example 21';
title2 c=black h=1.25 'Displaying Resource Requirements';

* set background to white and text to black;
goptions ctext=black cback=white;

* set graphics options;
goptions vpos=50 hpos=100 htext=1.01;

* begin annotate process;

* compile annotate macros;
%annomac;
```

```
* create annotate data set for first chart;
data anno1;
   %dclanno;       /* set length and type for annotate variables */
   %system(2,2,4); /* define annotate reference system           */
   set widgsch2;
   length lab $20;
   length TEXT $ 37;
   Y1 = _n_;
   lab='    ';

   if _n_=1 then do;
      %label('01dec03'd,13,
             'Format: Engineers per day, Total cost',*,0,0,1.2,brush,6);
      end;

   if engineer ^= . then do;
      /* create a text label */
      lab = put(engineer, 1.) || " Engineer";
      if engineer > 1 then lab = trim(lab) || "s";
      if days > 0 then lab = trim(lab) || ", " ||
                                          put(engcost*days, dollar7.);

      /* position the text label */
      if y1 < 10 then do;
         x1 = max(l_finish, s_finish) + 2;
         %label(x1,y1,lab,black,0,0,1.0,brush, 6);
         end;
      else do;
         x1 = e_start - 2;
         %label(x1,y1,lab,black,0,0,1.0,brush, 4);
         end;
      end;
run;

* annotate the Gantt chart;
proc gantt graphics data=widgsch2 holidata=holdata annotate=anno1;
chart / pcompress holiday=(hol) interval=weekday increment=7
        ref='1dec03'd to '21mar04'd by week
        cref=blue lref=2
        dur=days cmile=black caxis=black;
id task;
run;
```

Output 8.21.2 Using the ANNOTATE= Option



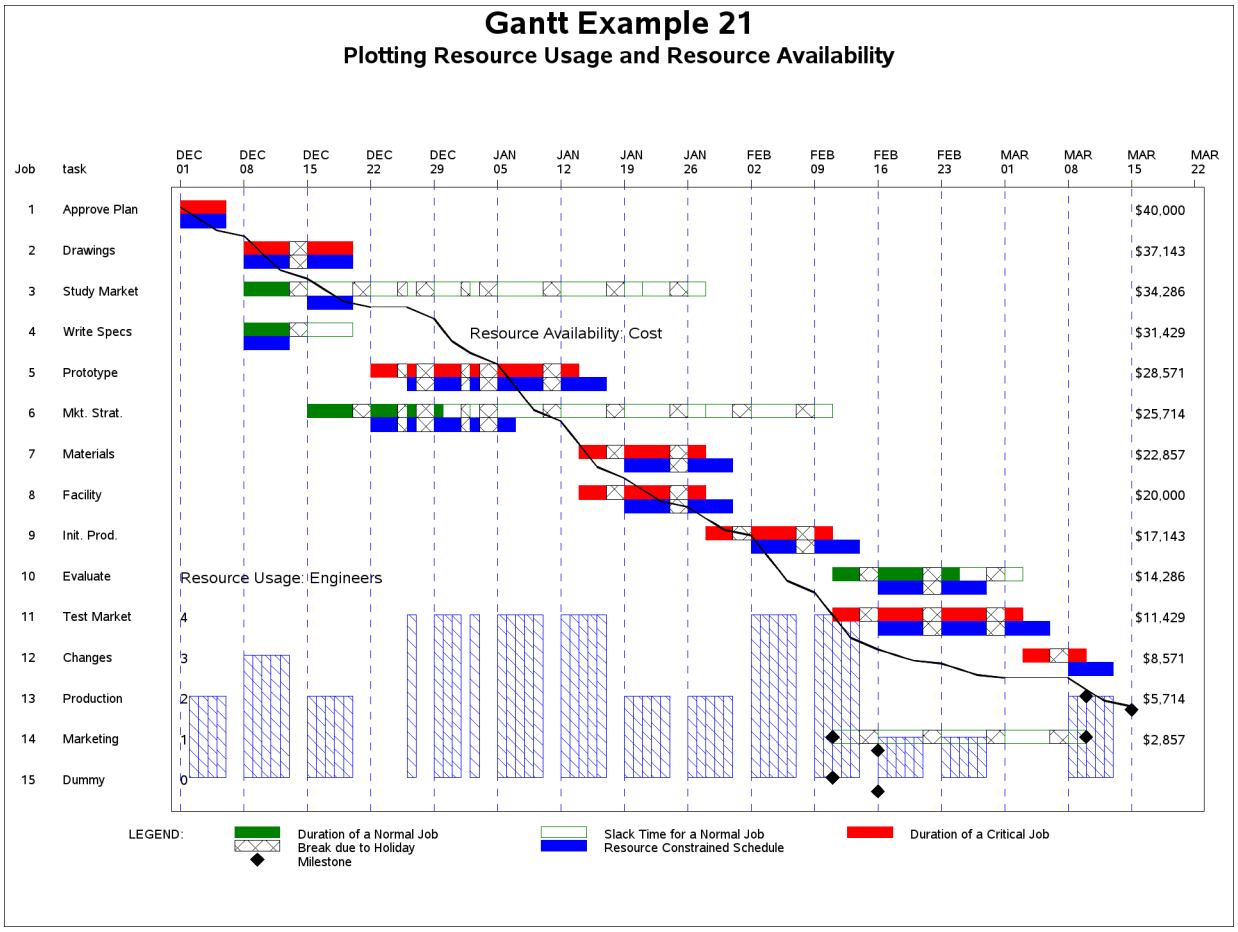
The next illustration of the ANNOTATE= option is to plot the resource usage bar chart for the replenishable resource engineers and the resource availability curve for the consumable resource cost. A DATA step determines the largest value of the cost availability throughout the life of the project in order to scale the costs accordingly. The CSCALE macro variable is required to represent cost availabilities on the Gantt chart. Since there are no further cash inflows after December 1, 2003, and there are 15 jobs represented on the chart, the value of the macro variable CSCALE is $(15 - 1)/40000$.

An Annotate data set, ANNO2, is created in much the same fashion as ANNO1, but it employs some additional macros. The BAR macro is used to draw the resource usage bar chart, and the DRAW and MOVE macros are used to draw the resource availability curve. The PUSH and POP macros are used as necessary to store and retrieve the last used coordinates from the stack, respectively. The resulting Gantt chart is displayed in Output 8.21.3.

```
* calculate scaling factor for cost curve;
data _null_;
   set widgrou2 end=final;
   retain maxcost;
   if aengcost > maxcost then maxcost=aengcost;
   if final then call symput('cscale',14/maxcost);
   run;

* create annotate data set for second chart;
data anno2;
```

```
   %dclanno;       /* set length and type for annotate variables */
   %system(2,2,4); /* define annotate reference system           */
   set widgrou2;
   length lab $16;
   length TEXT $27;
   x1=_time_;
   y1=15-aengcost*symget('cscale');
   y2=15-rengineer;
   lab='    ';

   if _n_=1 then do;
      /* print labels */
      do i = 1 to 14 by 1;
         lab=put( (15-i) / symget('cscale'), dollar7.);
         %label('21mar04'd,i,lab,black,0,0,1.0,,4);
         end;
      do i = 0 to 4 by 1;
         lab=put(i,1. );
         %label('01dec03'd,15-i,lab,black,0,0,1.0,,6);
         end;
      %label('01dec03'd,10,
             'Resource Usage: Engineers',*,0,0,1.2,,6);
      %label('02jan04'd,4,
             'Resource Availability: Cost',*,0,0,1.2,,6);
      %move(x1,y1);
      %push;
      end;
   else do;
      /* draw cost availability curve */
      %pop;
      when='a';
      %draw(x1,y1,black,1,2);
      %push;
      /* draw engineer usage barchart */
      when='b';
      if y2 <= 14 then do;
         %bar(x1,15,x1+1,y2,blue,0,11);
         end;
      end;
   run;

title c=black h=1.75 'Gantt Example 21';
title2 c=black h=1.25
       'Plotting Resource Usage and Resource Availability';

* annotate the Gantt chart;
proc gantt graphics data=widgsch2 holidata=holdata annotate=anno2;
chart / pcompress holiday=(hol) interval=weekday increment=7
        mindate='1dec03'd maxdate='21mar04'd
        ref='1dec03'd to '21mar04'd by week
        cref=blue lref=2
        dur=days cmile=black caxis=black;
id task;
run;
```

Output 8.21.3 Using the ANNOTATE= Option



Example 8.22: Using the Automatic Text Annotation Feature

The following example is a subproject of a larger project involving the maintenance of a pipeline and steam calender (Moder, Phillips, and Davis 1983), and it illustrates the automatic text annotation feature of the GANTT procedure. The SHUTDOWN data set is input as the activity data set to PROC CPM, and the project is scheduled to begin on June 1, 2004. PROC GANTT is used to produce a Gantt chart of the resulting schedule with the data set LABELS specified as a Label data set; the output is shown in Output 8.22.1. The LABVAR= option in the CHART statement specifies the ACT variable as the common linking variable. The LABSPLIT= option is specified in order to prevent the labels from splitting on embedded blanks.

The first observation in the LABELS data set causes the value of the ACT variable to be displayed at the E_START time for every activity in the project. The value of _YOFFSET='-.2' positions the baseline of the displayed text at 0.2 barheights above the top of the first bar for the activity. Similarly the second observation displays the ID variable at the E_START time for each activity with the baseline positioned at 0.8 barheights below the bottom of the first bar for the activity. The heights for both these strings is 1 barheight. The next two observations in the LABELS data set display the symbols corresponding to the values 'N' and 'M' in the ORFONT font, rotated at an angle of 90 degrees, beside the milestones corresponding to the deactivation and activation of the calender, respectively. Observations 5 and 6 indicate the start and finish of the "Maintenance

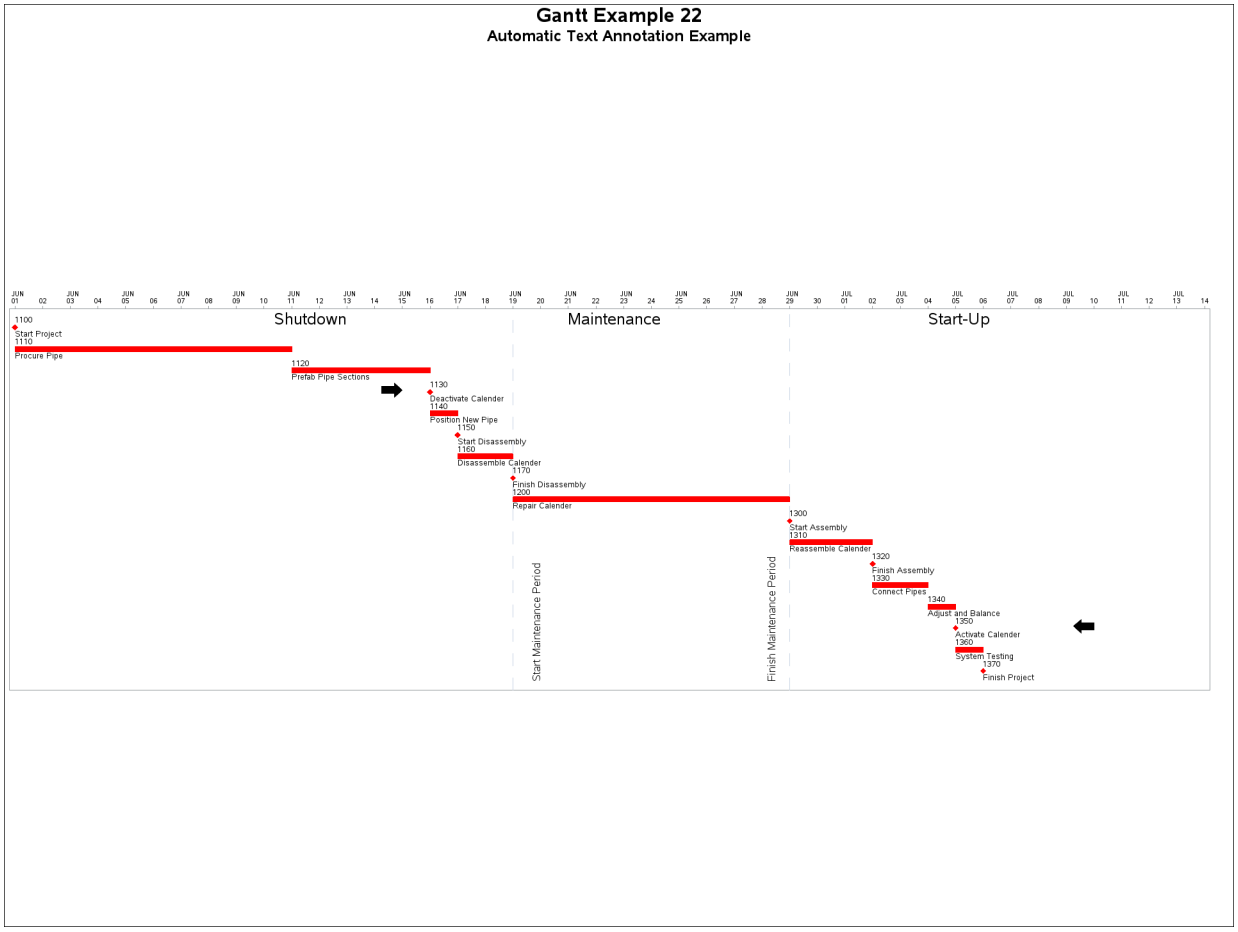
Period" by displaying the indicated strings rotated 90 degrees at the start and finish times of the activity 'Repair Calender.' Finally, the last three observations provide headings for each of the three distinct regions on the chart. The _JLABEL variable is used along with the _XVAR variable to place the strings in the regions defined by the start and finish times of the 'Repair Calender' activity.

It should be noted that since the plot times are linked to variables rather than absolute values, the Label data set need not be changed even if the project is rescheduled. This is a convenient feature when monitoring a project in progress, since the annotation automatically places the labels at the appropriate times.

```
title c=black 'Gantt Example 22';

data shutdown;
  input act succ id & $20. dur;
  datalines;
1100 1110 Start Project         0
1110 1120 Procure Pipe         10
1120 1130 Prefab Pipe Sections  5
1130 1140 Deactivate Calender   0
1140 1150 Position New Pipe     1
1150 1160 Start Disassembly     0
1160 1170 Disassemble Calender  2
1170 1200 Finish Disassembly    0
1200 1300 Repair Calender      10
1300 1310 Start Assembly        0
1310 1320 Reassemble Calender   3
1320 1330 Finish Assembly       0
1330 1340 Connect Pipes         2
1340 1350 Adjust and Balance    1
1350 1360 Activate Calender     0
1360 1370 System Testing        1
1370 .    Finish Project        0
;

proc cpm data=shutdown date='01jun04'd interval=day
  out=sched;
  act act;
  succ succ;
  dur dur;
  id id;
run;

data labels;
  input act _y _xvar $ _lvar $ _yoffset _xoffset _label & $25.
        _alabel _hlabel _jlabel $ _flabel $ _clabel $;
  datalines;
.    -1 e_start  act -.3  0 .                          0 1.5 .   .      .
.    -1 e_start  id  2.3  0 .                          0 1.5 .   .      .
1130  . e_start  .   1.5 -1 N                         90   2 L orfont   .
1350  . e_finish .   1.5  5 M                         90   2 L orfont   .
1200 17 e_start  .   2.5  1 Start Maintenance Period  90   2 .   .      .
1200 17 e_finish .   2.5 .5 Finish Maintenance Period 90   2 .   .      .
1200  1 e_start  .     . -6 Shutdown                   0   3 R   .      .
1200  1 e_start  .     .  2 Maintenance                0   3 L   .      .
1200  1 e_finish .     .  6 Start-Up                   0   3 L   .      .
;
```

```
title2 'Automatic Text Annotation Example';

proc gantt data=sched labdata=labels graphics maxdec=0;
  chart / pcompress nolegend nojobnum dur=dur
          mininterval=day scale=5 skip=3 maxdate='14jul04'd
          labvar=act labsplit='/' ref='19jun04'd '29jun04'd
          height=1.5 lref=20;
run;
```

Output 8.22.1 Using the LABDATA= Option



Example 8.23: Multiproject Gantt Charts

The following example illustrates an application of the PATTERN variable to display summary bars for subprojects. The LAN Selection Project (Bostwick 1986) consists of eight subprojects, two of which represent the beginning and ending of the master project. The data set LANACT defines the structure of the project. The ACT and SUCC variables define the precedence relationships, the PARENT variable defines the parent task, and the DAYS variable contains the duration of the activity.

The project is scheduled using the CPM procedure with a PARENT statement to identify the parent. The schedule data set, SCHED, is created by appending a _PATTERN variable to the output data set generated by CPM. The value of this variable is set to '4,' corresponding to subprojects, and set to missing otherwise. This results in the subproject bars being filled using PATTERN4, namely a solid black pattern. The ACTID variable is indented within the DATA step to reflect the level of each activity in the project hierarchy when used as the ID variable.

A Label data set, LABELS, is created in order to add markers to both ends of the schedule bars that correspond to subprojects. The two observations in the LABELS data set are linked to the SCHED data set with the _PATTERN variable.

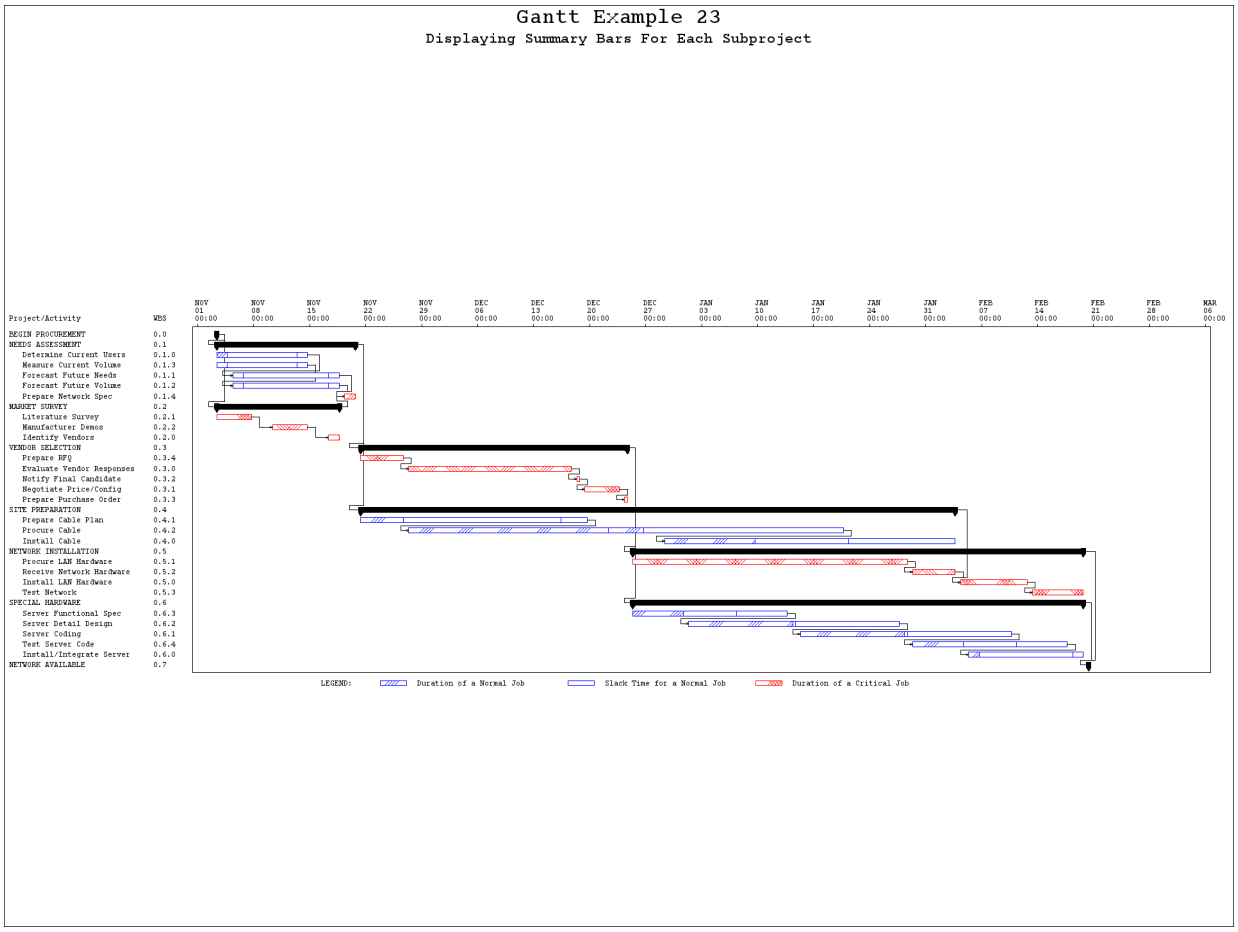
The GANTT procedure is next invoked to produce the Gantt chart in Output 8.23.1. The LABVAR=_PATTERN specification establishes the link between the Schedule and Label data sets. The ACT= and SUCC= options are used to display the precedence relationships between activities.

```
pattern1 c=blue   v=r5;         /* Non-critical duration  */
pattern2 c=blue   v=e;          /* Slack duration         */
pattern3 c=red    v=x5;         /* Critical duration      */
pattern4 c=black  v=s;          /* Project duration       */

data lanact;
   format act $30. succ $30. parent $20.;
   input act & succ & parent & days;
   datalines;
Measure Current Volume     Forecast Future Volume     NEEDS ASSESSMENT       2
Literature Survey          Manufacturer Demos         MARKET SURVEY          5
Determine Current Users    Forecast Future Needs      NEEDS ASSESSMENT       2
Forecast Future Volume     Prepare Network Spec       NEEDS ASSESSMENT       2
Manufacturer Demos         Identify Vendors           MARKET SURVEY          5
Forecast Future Needs      Prepare Network Spec       NEEDS ASSESSMENT       2
Identify Vendors           .                          MARKET SURVEY          2
Prepare Network Spec       .                          NEEDS ASSESSMENT       2
Prepare RFQ                Evaluate Vendor Responses  VENDOR SELECTION       4
Prepare Cable Plan         Procure Cable              SITE PREPARATION       4
Evaluate Vendor Responses  Notify Final Candidate     VENDOR SELECTION      15
Procure Cable              Install Cable              SITE PREPARATION      22
Notify Final Candidate     Negotiate Price/Config     VENDOR SELECTION       1
Install Cable              .                          SITE PREPARATION      10
Negotiate Price/Config     Prepare Purchase Order     VENDOR SELECTION       3
Prepare Purchase Order     .                          VENDOR SELECTION       1
Server Functional Spec     Server Detail Design       SPECIAL HARDWARE       5
Procure LAN Hardware       Receive Network Hardware   NETWORK INSTALLATION  25
Server Detail Design       Server Coding              SPECIAL HARDWARE      10
Receive Network Hardware   Install LAN Hardware       NETWORK INSTALLATION   4
Server Coding              Test Server Code           SPECIAL HARDWARE      10
Install LAN Hardware       Test Network               NETWORK INSTALLATION   7
Test Server Code           Install/Integrate Server   SPECIAL HARDWARE       5
Test Network               .                          NETWORK INSTALLATION   5
Install/Integrate Server   .                          SPECIAL HARDWARE       2
BEGIN PROCUREMENT          NEEDS ASSESSMENT           .                      .
BEGIN PROCUREMENT          MARKET SURVEY              .                      .
NEEDS ASSESSMENT           VENDOR SELECTION           .                      .
NEEDS ASSESSMENT           SITE PREPARATION           .                      .
MARKET SURVEY              Prepare Network Spec       .                      .
```

```
VENDOR SELECTION          NETWORK INSTALLATION      .                    .
VENDOR SELECTION          SPECIAL HARDWARE          .                    .
SITE PREPARATION          Install LAN Hardware      .                    .
NETWORK INSTALLATION      NETWORK AVAILABLE         .                    .
SPECIAL HARDWARE          NETWORK AVAILABLE         .                    .
;

proc sort data=lanact;
   by act;
   run;

proc cpm data=lanact out=lanout
     expand interval=workday date='03nov03'd;
   parent parent / wbs eso;
   activity act;
   duration days;
   successor succ;
   run;

 /* create the schedule data set with a pattern variable */
data sched;
   label wbs_code='WBS';
   label actid='Project/Activity';
   set lanout;
   if proj_lev !0 then do;
   if parent='' then _pattern=4;
      actid=act;
      do i=1 to proj_lev-1;
         actid = "   " || actid;
      end;
      output;
   end;
;

proc sort data=sched;
   by es_asc wbs_code;
   run;

 /* create the label data set */
data labels;
   _pattern=4;
   _flabel='orfont';
   _jlabel='c';
   _yoffset=0.925;
   _label='Z';
   _xvar='e_start ';
   output;
   _xvar='l_finish';
   output;
;

title1 f='Cumberland AMT' h=1.75 'Gantt Example 23';
title2 f='Cumberland AMT' h=1.25 'Displaying Summary Bars For Each Subproject';
```

```
proc gantt graphics data=sched labdata=labels;
  id actid wbs_code;
  chart / pcompress nojobnum ctext=black caxis=black
          mindate='01nov03'd maxdate='29feb04'd increment=7
          labvar=_pattern font='Cumberland AMT' height=1.5
          minoffgv=1.5 minofflv=1.5 cprec=black wprec=1
          scale=1.5 act=act succ=succ;
  run;
```

Output 8.23.1 Using the PATTERN Variable and Labels



Example 8.24: Multisegment Gantt Charts

The following is a simple example that illustrates the generation of multisegmented Gantt charts. The SCHED data set identifies the city, the arrival time, and the departure time for each of four traveling salespeople. In addition, a _PATTERN variable is used to identify the pattern to be used for drawing the bar. The objective is to display the complete schedule for each sales person on a single row. This would require displaying several bars on a single row, each bar corresponding to the time spent in a city. In order to do this, you need first to sort the SCHED data set by Salesperson and Arrival Time and then to add a SEGMT_NO variable that identifies the number of the segment that, in this case, is the order in which the salesperson visits the city. The resulting data set, NEWSCHED, is shown in Output 8.24.1. You next create the LABELS data set in order to identify the names of the cities above the bars; the resulting Gantt chart is shown in Output 8.24.2.

Notice that each bar is drawn using the pattern identified by the _PATTERN variable in the SCHED data set. In the absence of the _PATTERN variable, the pattern associated with the resource-constrained schedule would have been used for all the bars. This is the same mechanism that produced the split segments in Example 8.13 although the SEGMT_NO variable in this case was automatically created by the CPM procedure.

```
data sched;
   format city $12. from to date7. ;
   input person $ city & from & date7. to & date7. _pattern;
   datalines;
Clark   New York     01May04  03May04  10
Clark   Boston       06May04  09May04  11
Clark   Wisconsin    12May04  15May04  12
Clark   Chicago      18May04  24May04  13
Clark   New York     28May04  02Jun04  10
Stevens Charlotte    02May04  04May04  14
Stevens Atlanta      08May04  10May04  15
Stevens Dallas       12May04  15May04  16
Stevens Denver       17May04  20May04  17
Stevens Nashville    27May04  02Jun04  18
Stevens Charlotte    04Jun04  06Jun04  14
Jackson Los Angeles  01May04  08May04  19
Jackson Las Vegas    11May04  18May04  20
Jackson Portland     21May04  23May04  21
Jackson Seattle      25May04  29May04  22
Rogers  Miami        02May04  07May04  23
Rogers  Tampa        11May04  15May04  24
Rogers  New Orleans  18May04  24May04  25
Rogers  Houston      28May04  01Jun04  26
;

 /* Sort data by person, from */
proc sort data=sched;
   by person from;
   run;

/* Add Segmt_no variable */
data newsched;
   set sched;
   retain segmt_no;
   if person ne lag(person) then segmt_no=1;
   else segmt_no = segmt_no + 1;
   output;
   run;

proc print data=newsched;
   title2 'Data NEWSCHED';
   run;

data labels;
   _y=-1;
   _lvar="city";
   _xvar="from";
   _flabel="";
   _hlabel=0.75;
```

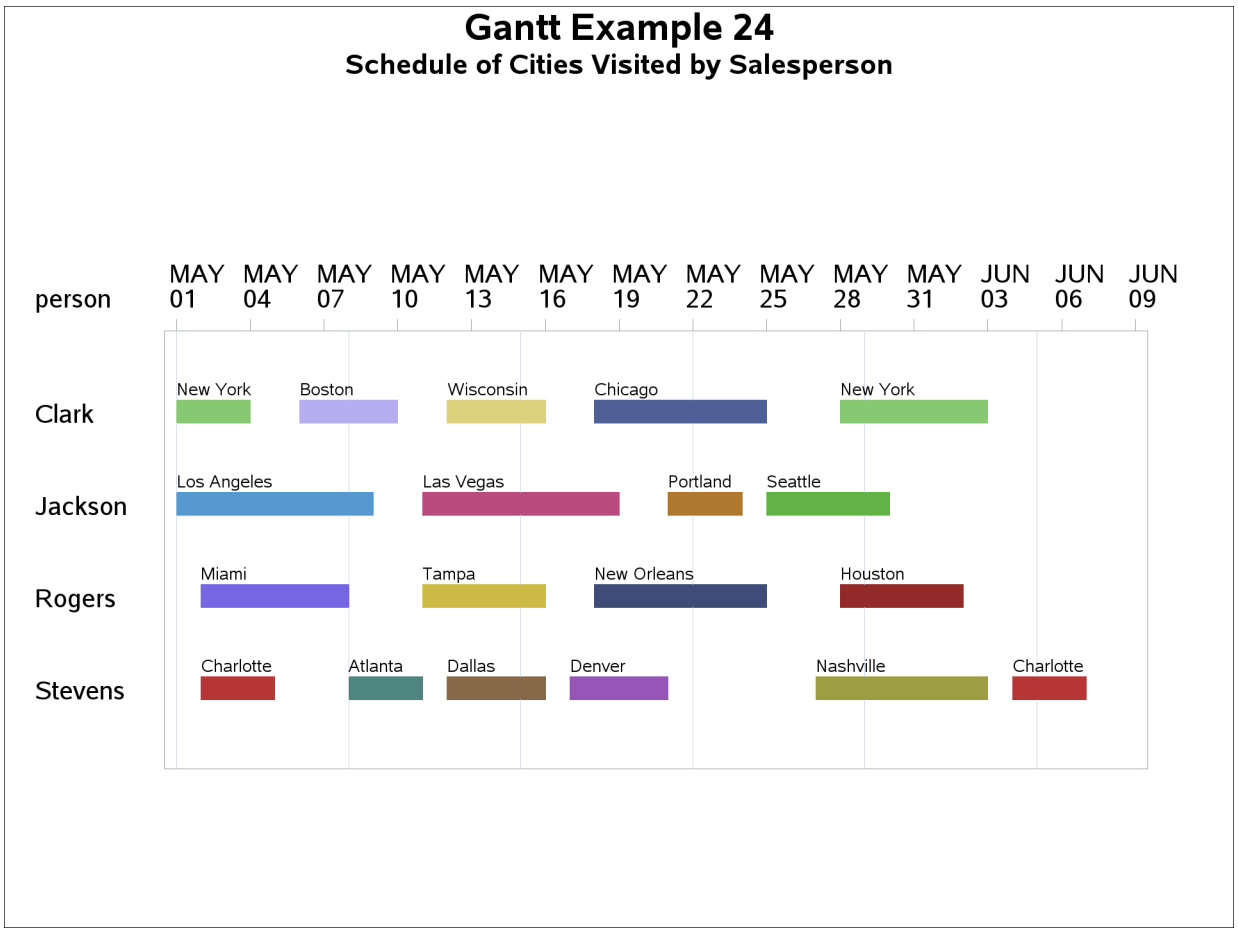
```
   _yoffset = -.2;
   run;

* set up required pattern statements;
pattern1 v=s r=25;

* set graphics options;
proc gantt graphics data=newsched labdata=labels;
id person;
chart / ss=from sf=to compress labsplit='.' scale=2
        nolegend nojobnum skip=3
        ref='01may04'd to '30jun04'd by week;
run;
```

Output 8.24.1 NEWSCHED Data Set

Data NEWSCHED

Obs	city	from	to	person	_pattern	segmt_no
1	New York	01MAY04	03MAY04	Clark	10	1
2	Boston	06MAY04	09MAY04	Clark	11	2
3	Wisconsin	12MAY04	15MAY04	Clark	12	3
4	Chicago	18MAY04	24MAY04	Clark	13	4
5	New York	28MAY04	02JUN04	Clark	10	5
6	Los Angeles	01MAY04	08MAY04	Jackson	19	1
7	Las Vegas	11MAY04	18MAY04	Jackson	20	2
8	Portland	21MAY04	23MAY04	Jackson	21	3
9	Seattle	25MAY04	29MAY04	Jackson	22	4
10	Miami	02MAY04	07MAY04	Rogers	23	1
11	Tampa	11MAY04	15MAY04	Rogers	24	2
12	New Orleans	18MAY04	24MAY04	Rogers	25	3
13	Houston	28MAY04	01JUN04	Rogers	26	4
14	Charlotte	02MAY04	04MAY04	Stevens	14	1
15	Atlanta	08MAY04	10MAY04	Stevens	15	2
16	Dallas	12MAY04	15MAY04	Stevens	16	3
17	Denver	17MAY04	20MAY04	Stevens	17	4
18	Nashville	27MAY04	02JUN04	Stevens	18	5
19	Charlotte	04JUN04	06JUN04	Stevens	14	6

Output 8.24.2 Multisegment Gantt Chart

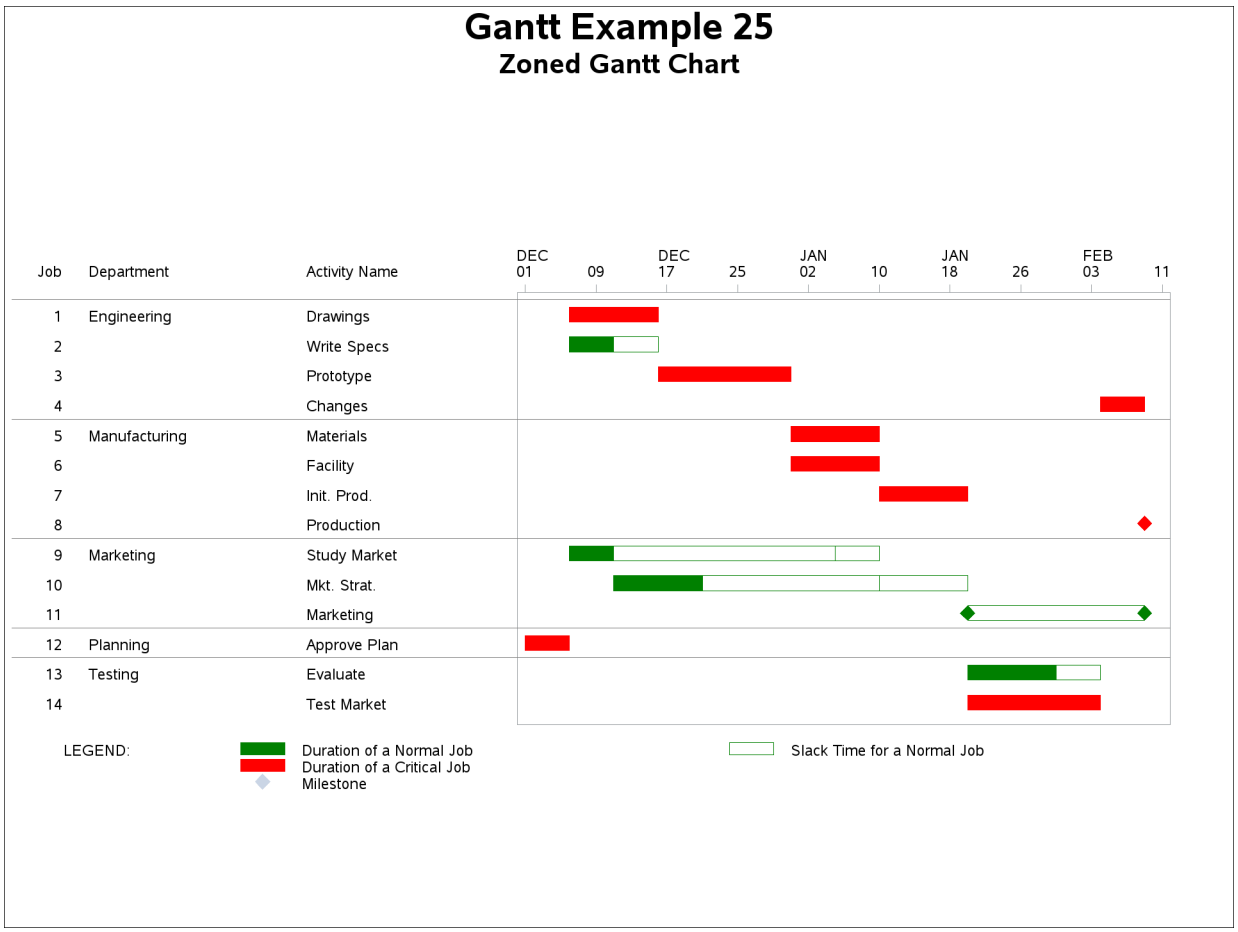


Example 8.25: Zoned Gantt Charts

Example 8.15 illustrated the use of BY processing with the GANTT procedure to present separate Gantt charts for each department. Alternatively, you can use a zoned Gantt chart to display each of the departmental schedules on the same chart with the different department schedules separated by horizontal zone lines running across the chart. The ZONE variable divides the Activity axis into distinct zones. Activities with the same value of the ZONE variable belong to the same zone. This example produces a zoned Gantt chart using the schedule data from Example 8.15. The ZONE=DEPT specification in the CHART statement identifies the DEPT variable as the ZONE variable. The ONEZONEVAL option specifies that the value of the ZONE variable be displayed only when beginning new zones. The resulting Gantt chart is shown in Output 8.25.1. You can customize the color, style and width of the zone line by using the CZONE=, LZONE=, and WZONE= options, respectively. You can also control the span and offset of the zone line by specifying the ZONESPAN= and ZONEOFF= options, respectively, in the CHART statement.

```
proc cpm date='01dec03'd data=widgetn;
   activity task;
   duration days;
   successor succ1 succ2 succ3;
   id dept;
   run;

proc sort;
   by dept e_start;
   run;

proc gantt split='/';
   chart / pcompress scale=1 dur=days
           mindate='01dec03'd maxdate='11feb04'd
           zone=dept onezoneval czone=gray;
   id task;
   run;
```

Output 8.25.1 Gantt Charts Zoned by Department



Example 8.26: Web-Enabled Gantt Charts

This example illustrates the process of "Web-enabling" your Gantt charts. This feature enables you to associate a URL with each activity on a Gantt chart. By using this feature together with SAS/IntrNet software, you can develop some very powerful Project Management applications. SAS/IntrNet software provides you with the capability to perform data set queries and execute SAS applications in real time and view the results in HTML format using a Web browser.

This example takes advantage of the Output Delivery System (ODS) HTML statement to create a very simple "drill-down" Gantt application beginning from a summary Gantt chart of the "top level" projects in Example 8.23. The objective is to display a detailed Gantt chart of the activities in a subproject when you click on the subproject bar.

In order to be able to click on an activity and invoke an action, you need to add variables to the schedule data set that associate a URL with each of the activities that you want linked. The following code adds the WEBVAR and WEBVAR2 variables to the LANOUT data set in Example 8.23 to create the LANWEB data set. The WEBVAR variable uses the ALT= portion to identify information about an activity's schedule that is to be displayed when the mouse hovers over the schedule bar. In addition, it uses the HREF= portion to associate the URL with the linked activity. The WEBVAR2 variable uses only the ALT= portion, so information in the detailed Gantt chart can still be displayed by hovering over the schedule bars.

The LANWEB data set is then sorted by the WBS_CODE variable.

```
data lanweb;
   set lanout;
   length webvar $500;
   length webvar2 $500;

   /* WEBVAR is for the top-level summary chart */
   webvar='alt='|| quote(
     'Activity: '||trim(left(act))||'0D'x||
     '----------------------'||'0D'x||
     'Early Start: '||put(e_start,  datetime.)||'0D'x||
     'Early Finish:'||put(e_finish, datetime.)||'0D'x||
     'Late Start:  '||put(l_start,  datetime.)||'0D'x||
     'Late Finish: '||put(l_finish, datetime.) )||
     ' HREF=#'||trim(wbs_code)  /* link to the anchors */
     ;

   /* WEBVAR2 is for the detailed charts */
   webvar2='alt='|| quote(
     'Activity: '||trim(left(act))||'0D'x||
     '----------------------'||'0D'x||
     'Early Start: '||put(e_start,  datetime.)||'0D'x||
     'Early Finish:'||put(e_finish, datetime.)||'0D'x||
     'Late Start:  '||put(l_start,  datetime.)||'0D'x||
     'Late Finish: '||put(l_finish, datetime.) )
     ;
   run;

proc sort data=lanweb;
   by wbs_code;
   run;
```

Before creating the charts, you need to specify that the GIF driver be used to create graphics output. ODS HTML output always creates a "body" file, which is a single HTML document containing the output from one or more procedures and is specified using the FILE= option in the ODS HTML statement.

```
goptions reset=all device=gif;

ods html file="Gantt_Sum.html";
```

For example, when you click on any of the schedule bars for an activity with WBS_CODE='0.2', you link to an anchor labeled '0.2' in the body file Gantt_Sum.html.

You are now ready to create the summary Gantt chart. You identify the WEBVAR variable to the GANTT procedure using the HTML= option in the CHART statement and invoke the procedure using a WHERE clause to produce a Gantt chart of the top-level activities.

```
/* Create the Summary Gantt Chart with Drill Down Action */
pattern1  c=green  v=s;         /* Non-critical duration */
pattern2  c=green  v=e;         /* Slack duration        */
pattern3  c=red    v=s;         /* Critical duration     */
goptions cback=white htext=1.1;

title1 h=2 'Gantt Example 26';
```

```
title2 h=1.5 'Project Summary Gantt Chart';

proc gantt data=lanweb;
   id act wbs_code;
   where proj_lev=1;
   label act='SUBPROJECT' wbs_code='WBS CODE';
   chart / pcompress nojobnum
           duration=days
           mininterval=week scale=2.5
           mindate='30oct03'd maxdate='29feb04'd
           ref='30oct03:00:00'dt to '01mar04:00:00'dt by dtmonth
           reflabel
           html=webvar
           act=act succ=succ cprec=black;
   run;
```

The graph that is displayed when you click on one of the subprojects is determined by the name of the anchor that has been defined for the subproject. Before creating these graphs, you need to define the anchor name in an ODS HTML statement using the ANCHOR= option to add the anchor to the HTML body file. Since you have to create a chart for each subproject, you can automate this process by using a SAS macro.

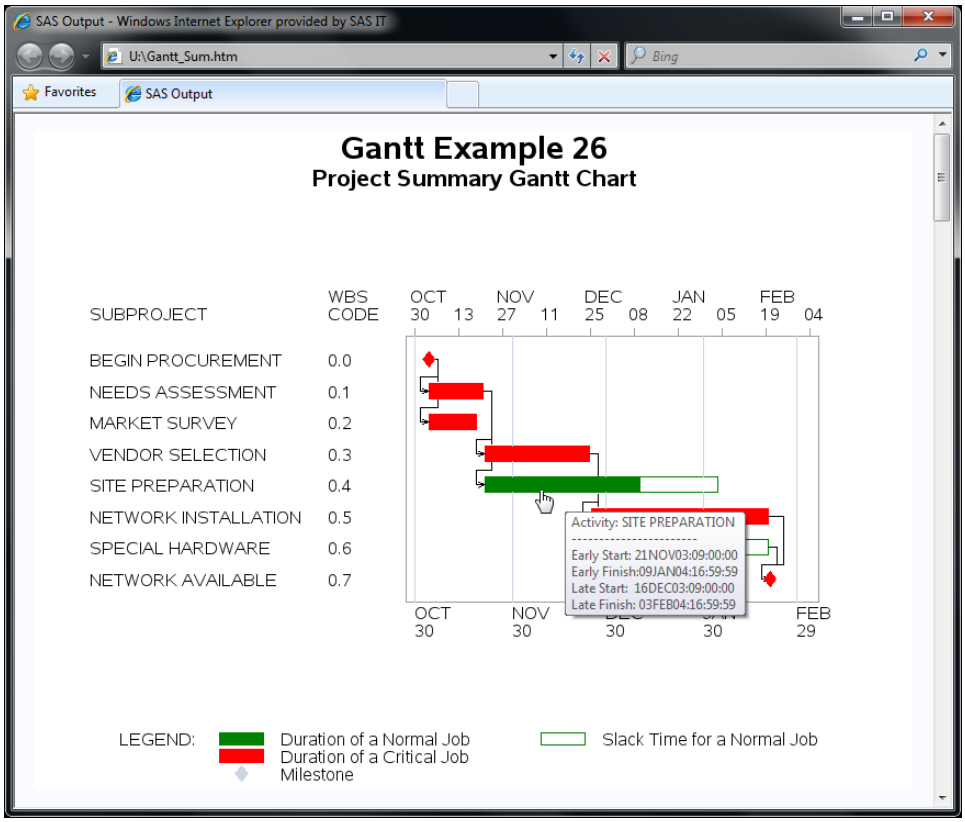
```
/* Define the macro to generate the detail charts */
%macro gandet(wbs);

goptions device=gif;
ods html anchor=&wbs;

title1 h=2 'Gantt Example 26';
title2 h=1.5 "Detail Gantt Chart for WBS="&wbs;

proc gantt data=lanweb;
   id act wbs_code;
   where index(wbs_code,&wbs)=1;
   label act='SUBPROJECT' wbs_code='WBS CODE';
   chart / pcompress nojobnum
           duration=days
           mininterval=week scale=2.5
           mindate='30oct03'd maxdate='29feb04'd
           ref='30oct03:00:00'dt to '01mar04:00:00'dt by dtmonth
           reflabel html=webvar2
           act=act succ=succ cprec=black;
   run;
%mend;

/* Generate each of the detail Gantt Charts */
%gandet('0.1');
%gandet('0.2');
%gandet('0.3');
%gandet('0.4');
%gandet('0.5');
%gandet('0.6');
```

Finally, use the ODS HTML CLOSE statement to close the body file and stop generating HTML output.

```
ods html close;
```

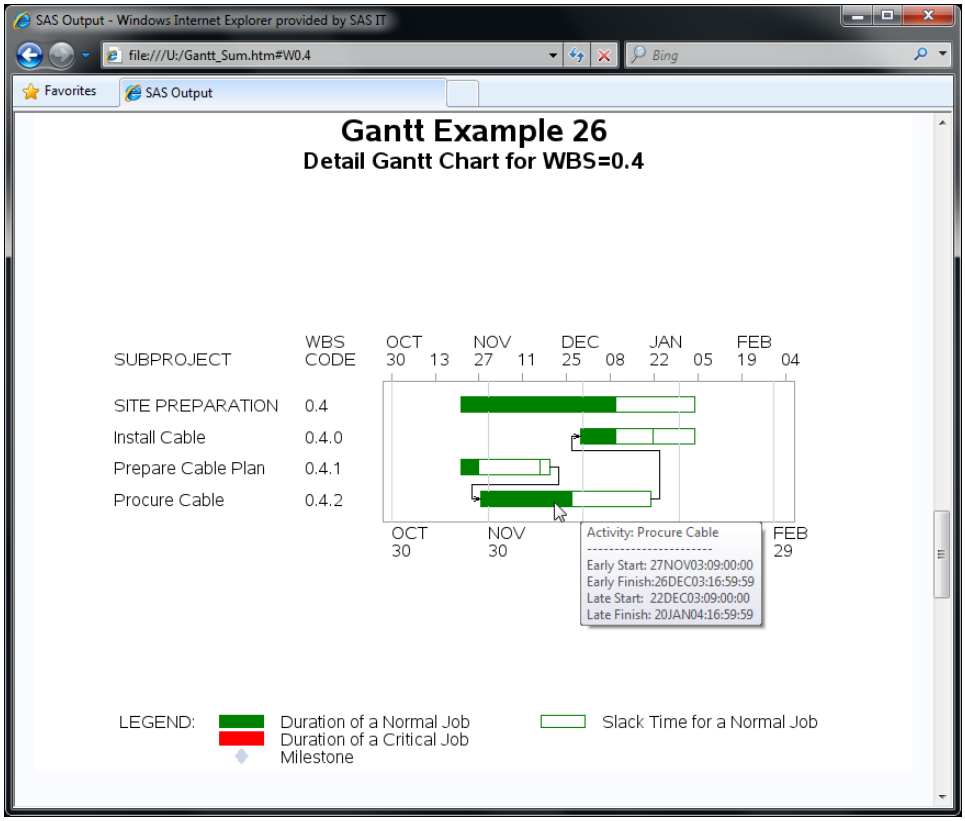
After you have closed the body file, you can display it in a browser window, as shown in Output 8.26.1, to view the output generated by this example.

Output 8.26.1 Summary Gantt Chart



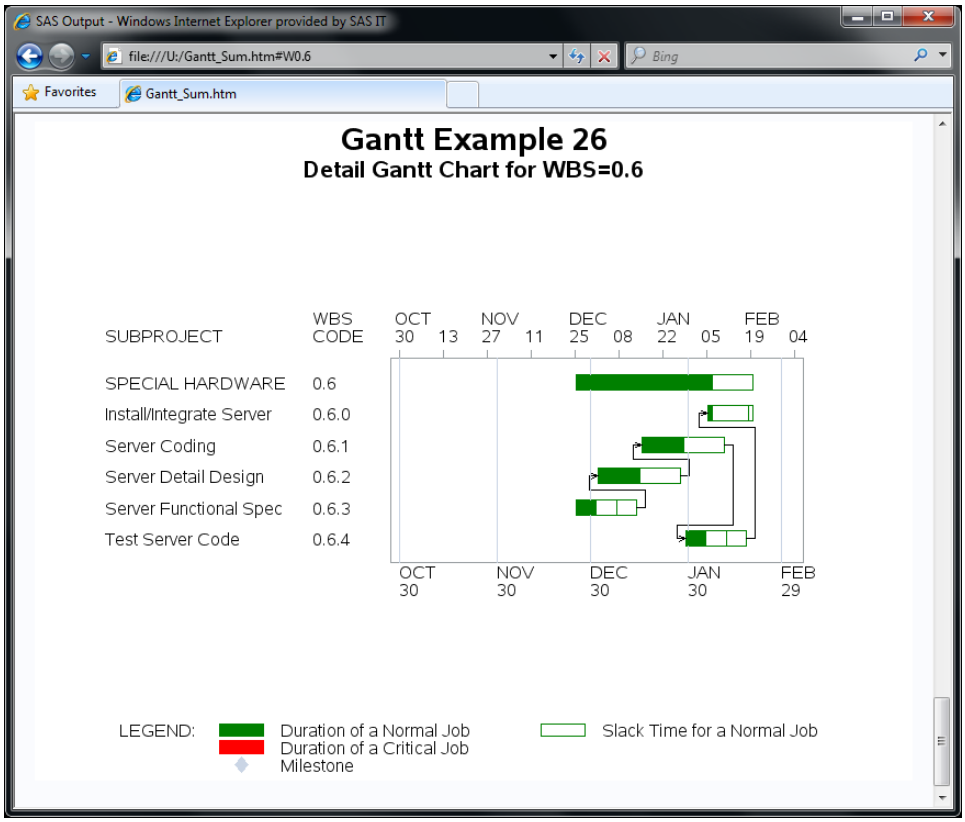
Notice the hand-shaped cursor on the SITE PREPARATION bar, which indicates that this bar is a "hot" link. The alternate text box displays the early and late schedules of the SITE PREPARATION activity. The status bar of the browser also shows that clicking the SITE PREPARATION bar will take you to the location identified by "Gantt_Sum.html#W0.4," which is shown in Output 8.26.2.

Output 8.26.2 Detail Gantt Chart for SITE PREPARATION



Similarly, the detail Gantt chart that is displayed when you click on the SPECIAL HARDWARE summary bar is shown in Output 8.26.3.

Output 8.26.3 Detail Gantt Chart for SPECIAL HARDWARE



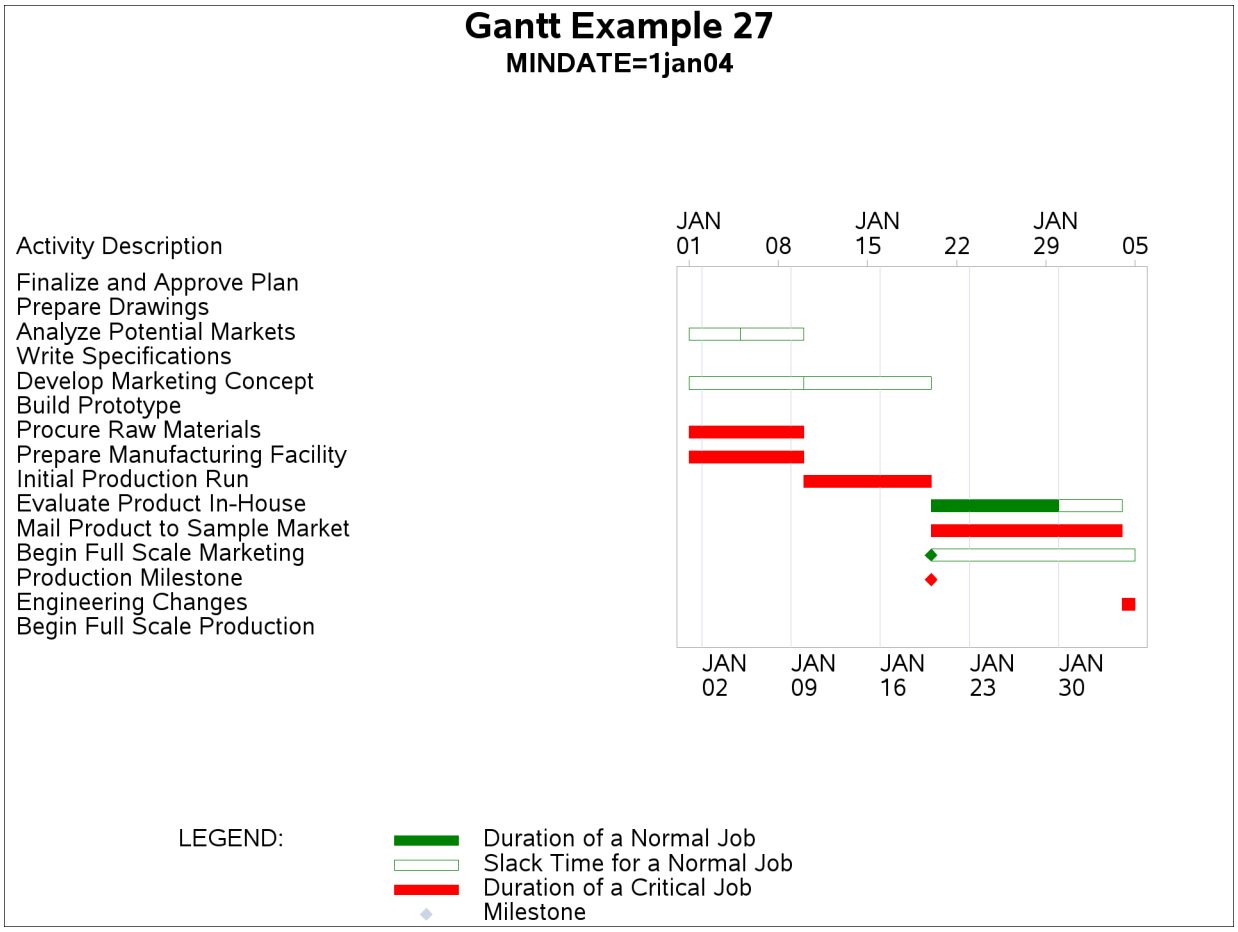
Example 8.27: Using the CHARTWIDTH= Option

This example illustrates the use of the CHARTWIDTH= option to create Gantt charts that are consistent in appearance. The data set used in this example is the SAVE data set created in Example 8.6.

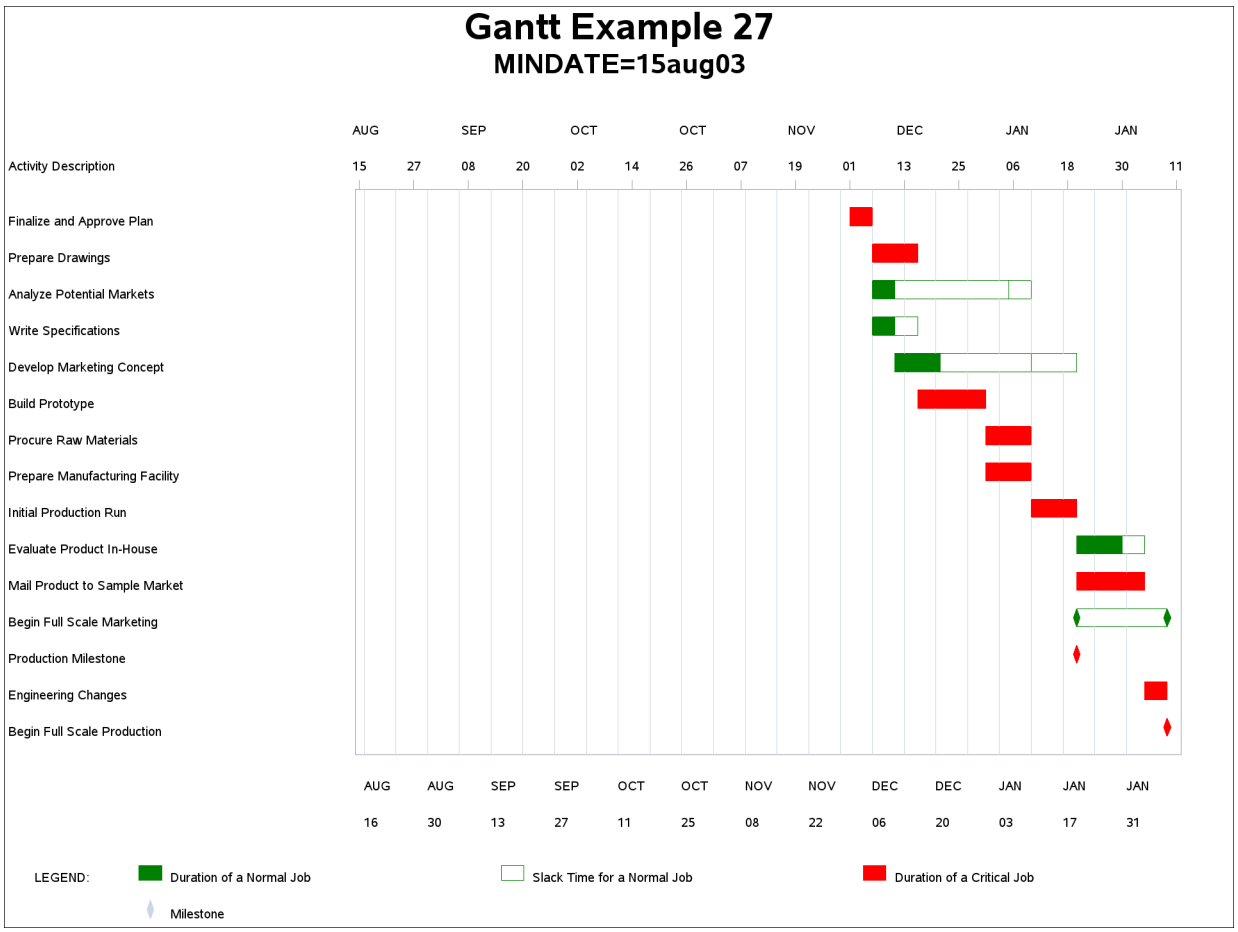
Gantt charts are first produced using different values of the MINDATE= option, and without specifying the CHARTWIDTH= option. Output 8.27.1 shows a Gantt chart using MINDATE='1jan04', and Output 8.27.2 shows a Gantt chart using MINDATE='15aug03'. Notice that the chart in Output 8.27.2 has a much larger chart area than the chart in Output 8.27.1, and the 'Activity Description' column is compressed and rather difficult to read.

```
proc gantt data=save;
   chart / mindate='1jan04'd maxdate='1feb04'd
           dur=days nojobnum compress height=2.0
           ref='2jan04'd to '2feb04'd by week
           reflabel;
   id descrpt;
   run;

proc gantt data=save;
   chart / mindate='15aug03'd maxdate='1feb04'd
           dur=days nojobnum compress height=2.0
           ref='16aug03'd to '2feb04'd by week
           reflabel;
   id descrpt;
   run;
```

Output 8.27.1 Without the CHARTWIDTH= Option (MINDATE=1Jan04)



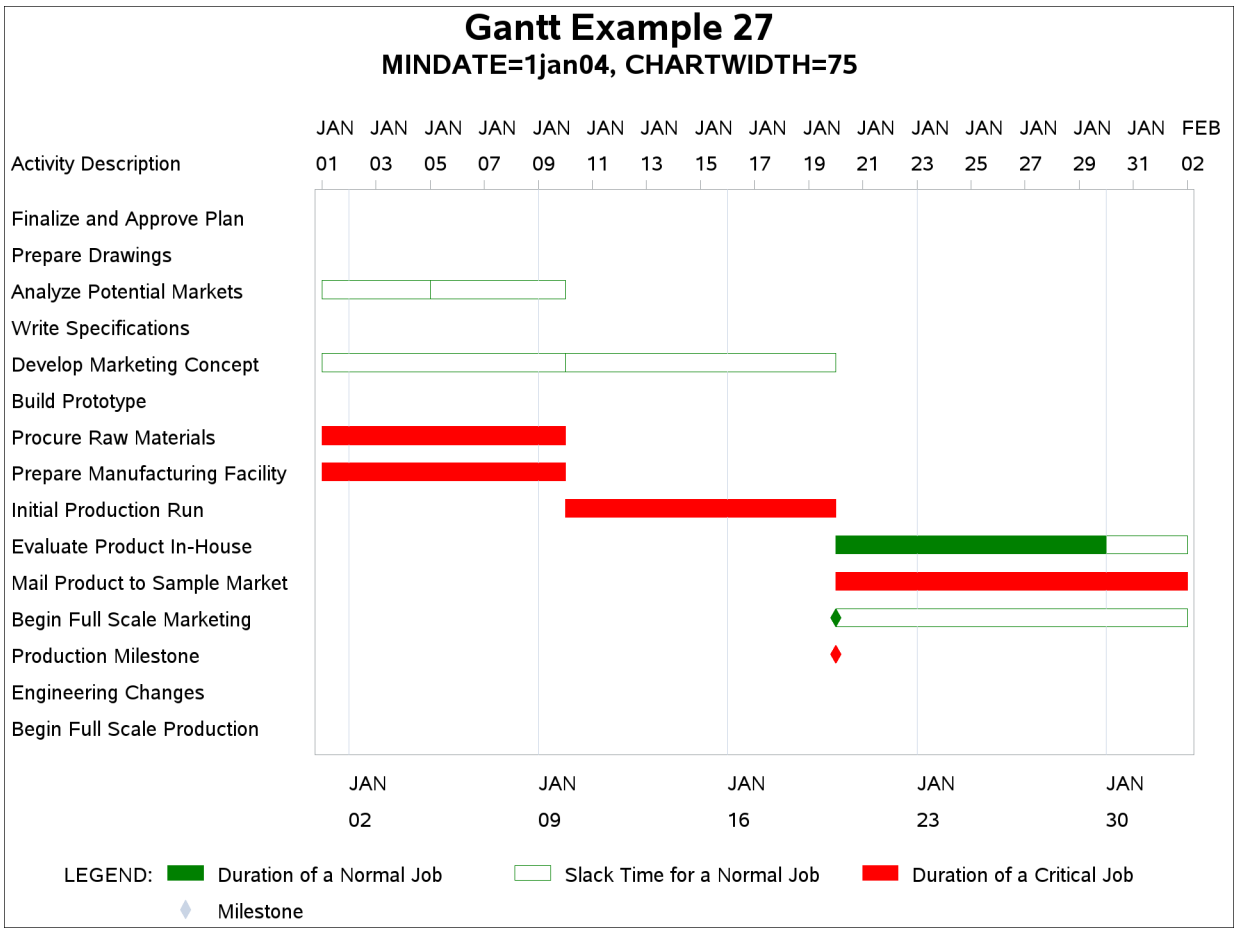
Output 8.27.2 Without the CHARTWIDTH= Option (MINDATE=15Aug03)



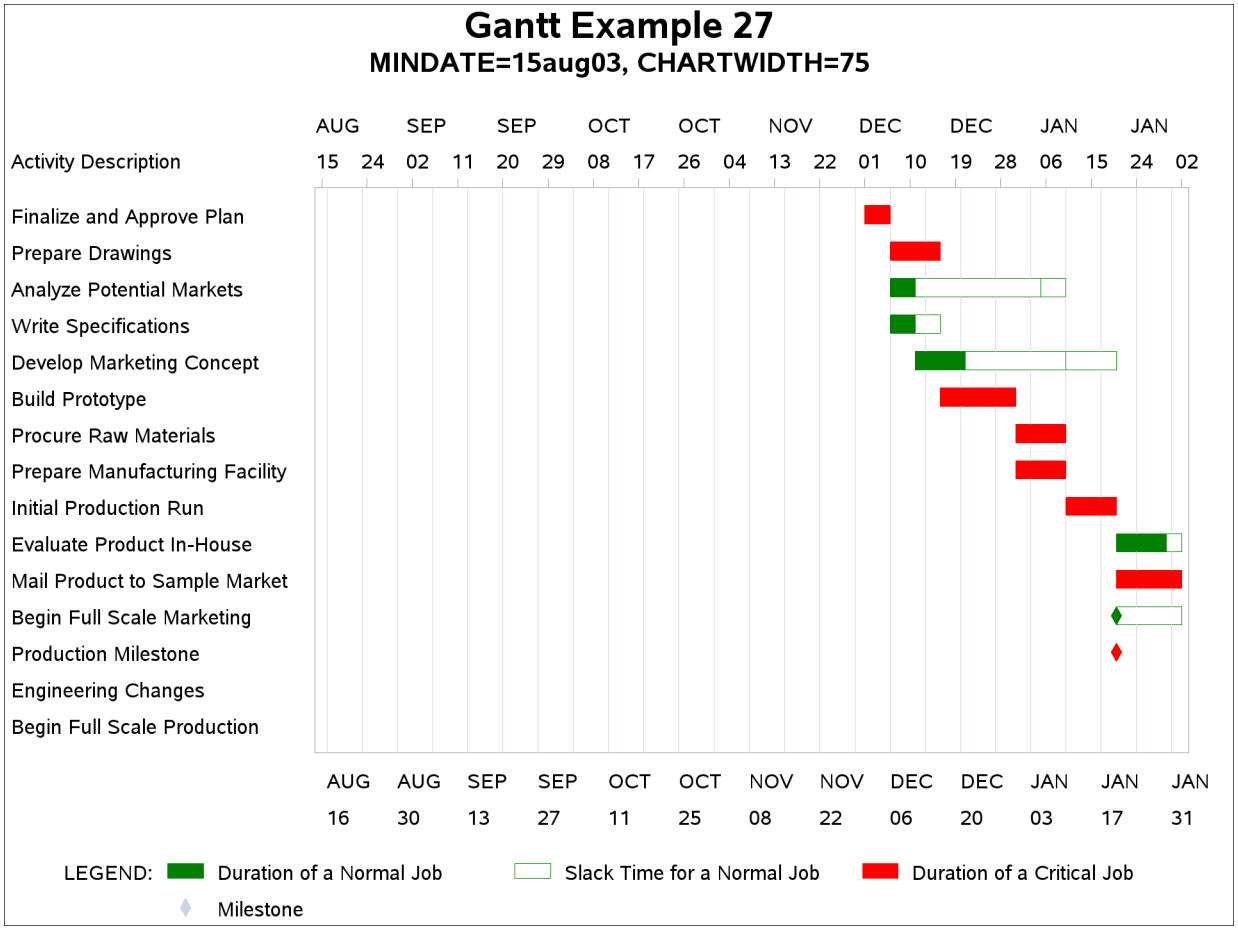
The same charts are now plotted with the CHARTWIDTH= option. The specification CHARTWIDTH=75 indicates that the chart is rescaled so the axis area is 75% of the chart width and the text area is 25% of the chart width. Therefore, specifying CHARTWIDTH=75 for both charts gives the two charts a consistent appearance. The output is shown in Output 8.27.3 and Output 8.27.4.

```
proc gantt data=save;
   chart / mindate='1jan04'd maxdate='1feb04'd
           dur=days nojobnum compress height=2.0
           ref='2jan04'd to '2feb04'd by week
           reflabel chartwidth=75;
   id descrpt;
   run;

proc gantt data=save;
   chart / mindate='15aug03'd maxdate='1feb04'd
           dur=days nojobnum compress height=2.0
           ref='16aug03'd to '2feb04'd by week
           reflabel chartwidth=75;
   id descrpt;
   run;
```

Output 8.27.3 Using the CHARTWIDTH= Option (MINDATE=1Jan04)



Output 8.27.4 Using the CHARTWIDTH= Option (MINDATE=15Aug03)



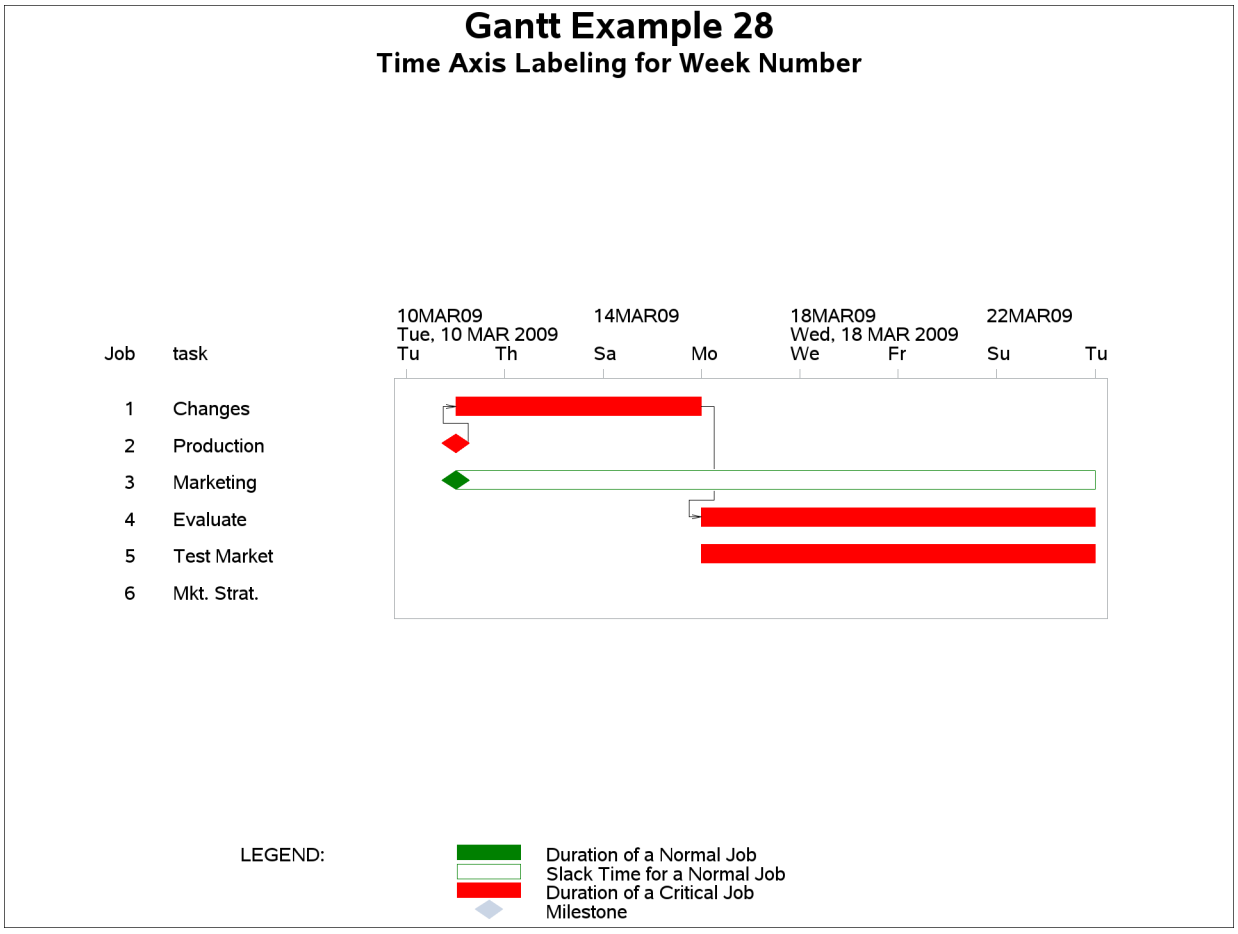
Example 8.28: Using the TIMEAXISFORMAT= Option

The following statements illustrate the use of the TIMEAXISFORMAT= option to specify formats for up to three rows of time-axis labels. The Activity data set for PROC CPM is the WIDGETA data set from Example 4.2, which defines the widget manufacturing project in AOA format.

```
* schedule the project subject to holidays and weekends;
proc cpm data=widgeta out=savehp
         date='11mar09'd;
   successor tail;
   activity head;
   duration days;
   id task dept descrpt;
run;

* sort the schedule by the early start date ;
proc sort;
   by e_start;
run;

* define a date format that includes the day of the week;
proc format;
    picture  dowdate (default=16) low-high = '%a, %d %b %Y'
        (datatype=date fill='0');
run;

* set up pattern statements;
pattern1 c=green v=s;    /* duration of a non-critical activity  */
pattern2 c=green v=e;    /* slack time for a noncrit. activity   */
pattern3 c=red v=s;      /* duration of a critical activity      */
pattern4 c=magenta v=e;  /* slack time for a supercrit. activity */
pattern5 c=magenta v=s;  /* duration of a supercrit. activity    */
pattern6 c=cyan v=s;     /* actual duration of an activity       */
pattern7 c=black v=e;    /* break due to a holiday               */
pattern8 c=blue v=s;     /* resource schedule of activity        */
pattern9 c=brown v=s;    /* baseline schedule of activity        */

* set graphics options;

* plot the logic Gantt chart using AOA representation;
proc gantt data=savehp (obs=6);
   chart / compress
           activity=head
           successor=tail
           mininterval=day
           increment=1
           dur=days
           maxdate='24MAR09'd
           taformat=(date7., dowdate., downame2.)
           cprec=black
           ;
   id task;
run;
```

The resulting Gantt chart, displayed in Output 8.28.1, contains one time-axis row for each format specified.

Output 8.28.1 Using the TIMEAXISFORMAT= Option



Statement and Option Cross-Reference Tables

The next two tables show which examples in this section use each of the statements and options in the GANTT procedure.

Table 8.15 Options Specified in Examples 4.1–4.14

Option	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A_FINISH=										X				
A_START=										X				
BETWEEN=		X												
CALID=									X					
CALENDAR=									X					
CAXIS=				X	X	X	X			X		X		
CHART var				X	X									
CHCON=										X				
CMILE=				X	X	X				X				
COMBINE												X		
COMPRESS					X		X			X		X		
CREF=							X							
CRITFLAG		X												
CTEXT=				X	X					X				
CTNOW=												X		
DATA=			X				X	X	X	X	X	X	X	X
DUR=				X	X	X	X	X		X	X	X	X	X
FILL		X												
FONT=								X		X				
HCONNECT										X				
HEIGHT=								X		X	X	X	X	X
HOLIDATA=			X					X	X	X	X	X	X	X
HOLIDAY=			X					X	X	X	X	X	X	X
HOLIDUR=								X						
HOLIFIN=			X						X	X	X	X		
ID		X	X	X	X	X	X	X	X	X	X	X	X	X
INTERVAL=								X						
LINEPRINTER	X	X												
L_FINISH=														X
LREF=							X							
L_START=														X
MARKBREAK								X	X					
MAXDATE=							X	X	X					
MINDATE=							X	X	X					
MININTERVAL=		X				X		X						
NOJOBNUM		X					X							
NOLEGEND		X				X								
PCOMPRESS								X	X		X		X	X
REF=		X				X	X							

Table 8.15 (continued)

Option	1	2	3	4	5	6	7	8	9	10	11	12	13	14
REFLABEL							X							
SCALE=		X				X			X		X			
S_FINISH=														X
SKIP=		X												
S_START=														X
SUMMARY		X												
TIMENOW=											X	X		
WORKDATA=									X					

Table 8.16 Options Specified in Examples 4.15–4.28

Option	15	16	17	18	19	20	21	22	23	24	25	26	27	28
A_FINISH=				X										
A_START=				X										
ACTIVITY=				X		X			X			X		X
ANNOTATE=							X							
BY	X	X												
CAXIS=			X	X	X	X	X		X					
CHARTWIDTH=													X	
CMILE=			X	X	X	X	X							
COMPRESS			X	X	X	X				X			X	X
CPREC=				X	X	X			X			X		X
CREF=						X	X							
CTEXT=									X					
CZONE=											X			
DATA=		X	X	X	X	X	X	X	X	X		X	X	X
DUR=	X		X	X	X	X	X	X			X	X	X	X
FONT=									X					
HEAD=					X									
HEIGHT=	X		X		X	X		X	X				X	
HOLIDATA=			X	X	X	X	X							
HOLIDAY=			X	X	X	X	X							
HOLIFIN=				X		X								
HTML=												X		
HTOFF=			X											
ID	X	X	X	X	X	X	X		X	X	X	X	X	X
INCREMENT=					X		X		X					X
INTERVAL=							X							
LABDATA=								X	X	X				
LABSPLIT=								X		X				
LABVAR=								X	X					
LAG=						X								
LEVEL=				X										
LREF=						X	X	X						
MAXDATE=	X				X		X	X	X		X	X	X	X

Table 8.16 (continued)

Option	15	16	17	18	19	20	21	22	23	24	25	26	27	28
MAXDEC=								X						
MAXDISLV=					X									
MINDATE=	X						X		X		X	X	X	
MININTERVAL=								X				X		X
MININTGV=					X	X								
MINOFFGV=					X				X					
MINOFFLV=					X	X			X					
NOJOBNUM								X	X	X		X	X	
NOLEGEND								X		X				
ONEZONEVAL											X			
PCOMPRESS	X	X					X	X	X		X	X		
PRECDATA=						X								
REF=						X	X	X		X		X	X	
REFLABEL						X						X	X	
SCALE=	X							X	X	X	X	X		
S_FINISH=										X				
SKIP=								X		X				
SPLIT=	X										X			
S_START=										X				
SUCCESSOR=				X		X			X			X		X
TAIL=					X									
TAFORMAT=														X
WPREC=						X			X					
ZONE=											X			

References

Bostwick, S. (1986). "Software Helps Solve LAN Selection Puzzle." *Mini-Micro Systems* (February 14): 20–22.

Kuhfeld, W. F. (2010). *Statistical Graphics in SAS: An Introduction to the Graph Template Language and the Statistical Graphics Procedures*. Cary, NC: SAS Institute Inc.

Moder, J. J., Phillips, C. R., and Davis, E. W. (1983). *Project Management with CPM, PERT, and Precedence Diagramming*. New York: Van Nostrand Reinhold.

Subject Index

Syntax Index