# Chapter 9
## The NETDRAW Procedure

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview: NETDRAW Procedure</td>
<td>666</td>
</tr>
<tr>
<td>Getting Started: NETDRAW Procedure</td>
<td>668</td>
</tr>
<tr>
<td>Syntax: NETDRAW Procedure</td>
<td>672</td>
</tr>
<tr>
<td>Functional Summary</td>
<td>672</td>
</tr>
<tr>
<td>PROC NETDRAW Statement</td>
<td>676</td>
</tr>
<tr>
<td>ACTNET Statement</td>
<td>677</td>
</tr>
<tr>
<td>Details: NETDRAW Procedure</td>
<td>689</td>
</tr>
<tr>
<td>Network Input Data Set</td>
<td>689</td>
</tr>
<tr>
<td>Variables in the Network Data Set</td>
<td>691</td>
</tr>
<tr>
<td>Missing Values</td>
<td>692</td>
</tr>
<tr>
<td>Layout of the Network</td>
<td>692</td>
</tr>
<tr>
<td>Format of the Display</td>
<td>694</td>
</tr>
<tr>
<td>Page Format</td>
<td>695</td>
</tr>
<tr>
<td>Layout Data Set</td>
<td>696</td>
</tr>
<tr>
<td>Controlling the Layout</td>
<td>697</td>
</tr>
<tr>
<td>Time-Scaled Network Diagrams</td>
<td>698</td>
</tr>
<tr>
<td>Zoned Network Diagrams</td>
<td>700</td>
</tr>
<tr>
<td>Organizational Charts or Tree Diagrams</td>
<td>700</td>
</tr>
<tr>
<td>Full-Screen Version</td>
<td>701</td>
</tr>
<tr>
<td>Graphics Version</td>
<td>704</td>
</tr>
<tr>
<td>Using the Annotate Facility</td>
<td>705</td>
</tr>
<tr>
<td>Web-Enabled Network Diagrams</td>
<td>706</td>
</tr>
<tr>
<td>Macro Variable _ORNETDR</td>
<td>706</td>
</tr>
<tr>
<td>Computer Resource Requirements</td>
<td>707</td>
</tr>
<tr>
<td>ODS Style Templates</td>
<td>707</td>
</tr>
<tr>
<td>Examples: NETDRAW Procedure</td>
<td>712</td>
</tr>
<tr>
<td>Example 9.1: Line-Printer Network Diagram</td>
<td>712</td>
</tr>
<tr>
<td>Example 9.2: Graphics Version of PROC NETDRAW</td>
<td>717</td>
</tr>
<tr>
<td>Example 9.3: Spanning Multiple Pages</td>
<td>718</td>
</tr>
<tr>
<td>Example 9.4: The COMPRESS and PCOMPRESS Options</td>
<td>722</td>
</tr>
<tr>
<td>Example 9.5: Controlling the Display Format</td>
<td>725</td>
</tr>
<tr>
<td>Example 9.6: Nonstandard Precedence Relationships</td>
<td>730</td>
</tr>
<tr>
<td>Example 9.7: Controlling the Arc-Routing Algorithm</td>
<td>732</td>
</tr>
<tr>
<td>Example 9.8: PATTERN and SHOWSTATUS Options</td>
<td>735</td>
</tr>
<tr>
<td>Example 9.9: Time-Scaled Network Diagram</td>
<td>738</td>
</tr>
<tr>
<td>Example 9.10: Further Time-Scale Options</td>
<td>742</td>
</tr>
</tbody>
</table>
Overview: NETDRAW Procedure

The NETDRAW procedure draws a network diagram of the activities in a project. Boxes (or nodes) are used to represent the activities, and lines (or arcs) are used to show the precedence relationships among the activities. Though the description of the procedure is written using project management terminology, PROC NETDRAW can be used to draw any network such as an organizational chart or a software flow diagram. The only information required by the procedure for drawing such a diagram is the name of each activity in the project (or node in the network) and a list of all its immediate successor activities (or nodes connected to it by arcs). Note that project networks are acyclic. However, the procedure can also be used to draw cyclic networks by specifying explicitly the coordinates for the nodes or by requesting the procedure to break the cycles in an arbitrary fashion.

The ACTNET statement in the NETDRAW procedure is designed to draw activity networks that represent a project in Activity-On-Node (AON) format. All network information is contained in SAS data sets. The input data sets used by PROC NETDRAW and the output data set produced by the procedure are as follows:

- The Network input data set contains the precedence information, namely, the activity-successor information for all the nodes in the network. This data set can be an Activity data set that is used as input to the CPM procedure or a Schedule data set that is produced by the CPM procedure, or it can even be a Layout data set produced by the NETDRAW procedure. The minimum amount of information that is required by PROC NETDRAW is the activity-successor information that can be obtained from any one of the preceding three possible types of data sets. The additional information in the input data set can be used by the procedure to add detail to the nodes in the diagram, and, in the case of the Layout data set, the procedure can use the _X_ and _Y_ variables to lay out the nodes and arcs of the diagram.

- The Annotate input data set contains the graphics and text that are to be annotated on the network diagram. This data set is used by the procedure through the Annotate facility in SAS/GRAPH software.

- The Layout output data set produced by PROC NETDRAW contains all the information about the layout of the network. For each node in the network, the procedure saves the (x, y) coordinates; for each arc between each pair of nodes, the procedure saves the (x, y) coordinates of each turning point of the arc in a separate observation. Using these values, the procedure can draw the network diagram without recomputing node placement and arc routing.
Two issues arise in drawing and displaying a network diagram: the layout of the diagram and the format of the display. The layout of the diagram consists of placing the nodes of the network and routing the arcs of the network in an appropriate manner. The format of the display includes the size of the nodes, the distance between nodes, the color of the nodes and arcs, and the information that is placed within each node. Several options available in the ACTNET statement enable you to control the format of the display and the layout of the diagram; these options and their uses are explained in detail later in this chapter.

Following is a list of some of the key aspects of the procedure:

- The Network input data set specifies the activities (or nodes) in the network and their immediate successors. The amount of information displayed within each node can be controlled by the ID= option and by the use of default variables in the data set.

- The procedure uses the node-successor information to determine the placement of the nodes and the layout of the arcs connecting the nodes.

- By default, PROC NETDRAW uses the topological ordering of the activity network to determine the coordinates of the nodes. In a time-based network diagram, the nodes can be ordered according to any numeric, SAS date, time, or datetime variable (the ALIGN= variable) in the input data set.

- The network does not have to represent a project. You can use PROC NETDRAW to draw any network. If the network has no cycles, then the procedure bases the node placement and arc routing on the precedence relationships. Alternately, you can specify explicitly the node positions or use the ALIGN= variable, and let the procedure determine the arc routing.

- To draw networks with cycles, use the BREAKCYCLE option. Alternately, you can use the ALIGN= option or specify the node positions so that the procedure needs only to determine the arc routing. See Example 9.12 for an illustration of a cyclic network.

- The ZONE= option enables you to divide the network into horizontal bands or zones. This is useful in grouping the activities of the project according to some appropriate classification.

- The TREE option instructs PROC NETDRAW to check if the network is indeed a tree, and, if so, to exploit the tree structure in the node layout. This feature is useful for drawing organizational charts, hierarchical charts, and work break-down structures.

- PROC NETDRAW gives you the option of displaying the network diagram in one of three modes: graphics, line-printer, or full-screen. 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 Options” on page 683 for more information about producing high-resolution quality network diagrams. You can also produce line-printer quality network diagrams by specifying the LINEPRINTER (LP) option in the PROC NETDRAW statement. In addition to sending the output to either a plotter or printer, you can view the network diagram at the terminal in full-screen mode by specifying the FULLSCREEN (FS) option in the PROC NETDRAW statement. See the section “Full-Screen Options” on page 682 for more information about viewing network diagrams in full-screen mode.

- The full-screen version of the procedure enables you to move the nodes around on the screen (subject to maintaining the precedence order of the activities) and thus change the layout of the network diagram.

- The graphics version of the procedure enables you to annotate the network diagram using the Annotate facility in SAS/GRAPH software.
• The positions of the nodes and arcs of the layout determined by PROC NETDRAW are saved in an output data set called the Layout data set. This data set can be used again as input to PROC NETDRAW; using such a data set saves some processing time because the procedure does not need to determine the node and arc placement.

• If necessary, the procedure draws the network across page boundaries. The number of pages that are used depends on the number of print positions that are available in the horizontal and vertical directions.

• In graphics mode, the COMPRESS and PCOMPRESS options enable you to produce the network on one page. You can also control the number of pages used to create the network diagram with the HPAGES= and VPAGES= options.

• In graphics mode, the ROTATE and ROTATETEXT options enable you to produce a top-down tree diagram.

Getting Started: NETDRAW Procedure

The first step in defining a project is to make a list of the activities in the project and determine the precedence constraints that need to be satisfied by these activities. It is useful at this stage to view a graphical representation of the project network. In order to draw the network, you specify the nodes of the network and the precedence relationships among them. Consider the software development project that is described in the “Getting Started” section of Chapter 4, “The CPM Procedure.” The network data are in the SAS data set SOFTWARE, displayed in Figure 9.1.

**Figure 9.1 Software Project**

<table>
<thead>
<tr>
<th>Obs</th>
<th>descript</th>
<th>duration</th>
<th>activity</th>
<th>succesr1</th>
<th>succesr2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial Testing</td>
<td>20</td>
<td>TESTING</td>
<td>RECODE</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Prel. Documentation</td>
<td>15</td>
<td>PRELDOC</td>
<td>DOCEDREV QATEST</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Meet Marketing</td>
<td>1</td>
<td>MEETMKT</td>
<td>RECODE</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Recoding</td>
<td>5</td>
<td>RECODE</td>
<td>DOCEDREV QATEST</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>QA Test Approve</td>
<td>10</td>
<td>QATEST</td>
<td>PROD</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Doc. Edit and Revise</td>
<td>10</td>
<td>DOCEDREV</td>
<td>PROD</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Production</td>
<td>1</td>
<td>PROD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The following code produces the network diagram shown in Figure 9.2:

```
pattern1 v=e c=green;
pattern2 v=e c=red;

title h=3 'Software Project';
proc netdraw graphics data=software;
   actnet / act=activity htext=2
   succ=(successr1 succesr2)
   pcompress separatearcs;
run;
```

**Figure 9.2** Software Project

The procedure determines the placement of the nodes and the routing of the arcs on the basis of the topological ordering of the nodes and attempts to produce a compact diagram. You can control the placement of the nodes by specifying explicitly the node positions. The data set SOFTNET, shown in Figure 9.3, includes the variables _X_ and _Y_, which specify the desired node coordinates. Note that the precedence information is conveyed using a single SUCCESSOR variable unlike the data set SOFTWARE, which contains two SUCCESSOR variables.
**Figure 9.3** Software Project: Specify Node Positions

<table>
<thead>
<tr>
<th>Obs</th>
<th>descrip</th>
<th>duration</th>
<th>activity</th>
<th>succesor</th>
<th><em>x</em></th>
<th><em>y</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial Testing</td>
<td>20</td>
<td>TESTING</td>
<td>RECODE</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Meet Marketing</td>
<td>1</td>
<td>MEETMKT</td>
<td>RECODE</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Prel. Documentation</td>
<td>15</td>
<td>PRELDOC</td>
<td>DOCEDREV</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Prel. Documentation</td>
<td>15</td>
<td>PRELDOC</td>
<td>QATEST</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>Recoding</td>
<td>5</td>
<td>RECODE</td>
<td>DOCEDREV</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Recoding</td>
<td>5</td>
<td>RECODE</td>
<td>QATEST</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>QA Test Approve</td>
<td>10</td>
<td>QATEST</td>
<td>PROD</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Doc. Edit and Revise</td>
<td>10</td>
<td>DOCEDREV</td>
<td>PROD</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Production</td>
<td>1</td>
<td>PROD</td>
<td></td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

The following code produces a network diagram (shown in **Figure 9.4**) with the new node placement:

```plaintext
title h=3 'Software Project';
title2 h=2 'Controlled Layout';
proc netdraw graphics data=softnet;
   actnet / act=activity htext=1.25
      succ=(succesor)
         pcompress;
run;
```

**Figure 9.4** Software Project: Controlled Layout
While the project is in progress, you may want to use the network diagram to show the current status of each activity as well as any other relevant information about each activity. PROC NETDRAW can also be used to produce a time-scaled network diagram using the schedule produced by PROC CPM. The schedule data for the software project described earlier are saved in a data set, INTRO1, which is shown in Figure 9.5.

To produce a time-scaled network diagram, use the TIMESCALE option in the ACTNET statement, as shown in the following program. The MININTERVAL= and the LINEAR options are used to control the time axis on the diagram. The ID=, NOLABEL, and NODEFID options control the amount of information displayed within each node. The resulting diagram is shown in Figure 9.6.

``` Sas
Title h=3 'Software Project';
Title2 h=2 'Time-Scaled Diagram';
Proc netdraw graphics data=intro1;
   Actnet / act=activity succ=(succ:)
      Separatearcs pcompress htext=2
      Timescale linear frame mininterval=week
      Id=activity duration nolabel nodefid;
Run;
```
Several other options are available to control the layout of the nodes, the appearance of the network, and the format of the time axis. For projects that have natural divisions, you can use the ZONE= option to divide the network into horizontal zones or bands. For networks that have an embedded tree structure, you can use the TREE option to draw the network like a tree laid out from left to right, with the root at the left edge of the diagram; in graphics mode, you can obtain a top-down tree with the root at the top of the diagram. For cyclic networks you can use the BREAKCYCLE option to enable the procedure to break cycles. All of these options are discussed in detail in the following sections.

Syntax: NETDRAW Procedure

The following statements are used in PROC NETDRAW:

```
PROC NETDRAW options;
ACTNET / options;
```

Functional Summary

Table 9.1 outlines the options available for the NETDRAW procedure classified by function.
### Table 9.1 Functional Summary

<table>
<thead>
<tr>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Color Options</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifies the color of arcs</td>
<td>ACTNET</td>
<td>CARCS=</td>
</tr>
<tr>
<td>Specifies the color of time axis</td>
<td>ACTNET</td>
<td>CAXIS=</td>
</tr>
<tr>
<td>Specifies the fill color for critical nodes</td>
<td>ACTNET</td>
<td>CCRNODEFILL=</td>
</tr>
<tr>
<td>Specifies the color of critical arcs</td>
<td>ACTNET</td>
<td>CCRITARCS=</td>
</tr>
<tr>
<td>Specifies the outline color of critical nodes</td>
<td>ACTNET</td>
<td>CCRITOUT=</td>
</tr>
<tr>
<td>Specifies the fill color for nodes</td>
<td>ACTNET</td>
<td>CNODEFILL=</td>
</tr>
<tr>
<td>Specifies the outline color of nodes</td>
<td>ACTNET</td>
<td>COUTLINE=</td>
</tr>
<tr>
<td>Specifies the color of reference lines</td>
<td>ACTNET</td>
<td>CREF=</td>
</tr>
<tr>
<td>Specifies the color of reference break lines</td>
<td>ACTNET</td>
<td>CREFBRK=</td>
</tr>
<tr>
<td>Specifies the text color</td>
<td>ACTNET</td>
<td>CTEXT=</td>
</tr>
<tr>
<td><strong>Data Set Specifications</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifies the Annotate data set</td>
<td>ACTNET</td>
<td>ANNOTATE=</td>
</tr>
<tr>
<td>Specifies the Annotate data set</td>
<td>NETDRAW</td>
<td>ANNOTATE=</td>
</tr>
<tr>
<td>Specifies the Activity data set</td>
<td>NETDRAW</td>
<td>DATA=</td>
</tr>
<tr>
<td>Specifies the Network output data set</td>
<td>NETDRAW</td>
<td>OUT=</td>
</tr>
<tr>
<td><strong>Format Control Options</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifies the height of node in character cells</td>
<td>ACTNET</td>
<td>BOXHT=</td>
</tr>
<tr>
<td>Specifies the width of node in character cells</td>
<td>ACTNET</td>
<td>BOXWIDTH=</td>
</tr>
<tr>
<td>Specifies the DURATION variable</td>
<td>ACTNET</td>
<td>DURATION=</td>
</tr>
<tr>
<td>Specifies the ID variables</td>
<td>ACTNET</td>
<td>ID=</td>
</tr>
<tr>
<td>Suppresses default ID variables</td>
<td>ACTNET</td>
<td>NODEFID</td>
</tr>
<tr>
<td>Suppresses ID variable labels</td>
<td>ACTNET</td>
<td>NOLABEL</td>
</tr>
<tr>
<td>Specifies the upper limit on number of pages</td>
<td>ACTNET</td>
<td>PAGES=</td>
</tr>
<tr>
<td>Indicates completed or in-progress activities</td>
<td>ACTNET</td>
<td>SHOWSTATUS</td>
</tr>
<tr>
<td>Specifies the horizontal distance between nodes</td>
<td>ACTNET</td>
<td>XBETWEEN=</td>
</tr>
<tr>
<td>Specifies the vertical distance between nodes</td>
<td>ACTNET</td>
<td>YBETWEEN=</td>
</tr>
<tr>
<td><strong>Full-Screen Options</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifies the reference break line character</td>
<td>ACTNET</td>
<td>BRKCHAR=</td>
</tr>
<tr>
<td>Specifies the characters for node outlines and connections</td>
<td>ACTNET</td>
<td>FORMCHAR=</td>
</tr>
<tr>
<td>Specifies the reference character</td>
<td>ACTNET</td>
<td>REFCHAR=</td>
</tr>
<tr>
<td><strong>Graphics Catalog Options</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifies the description for the catalog entry</td>
<td>ACTNET</td>
<td>DESCRIPTION=</td>
</tr>
<tr>
<td>Specifies the name for the catalog entry</td>
<td>ACTNET</td>
<td>NAME=</td>
</tr>
<tr>
<td>Specifies the name of the graphics catalog</td>
<td>NETDRAW</td>
<td>GOUT=</td>
</tr>
<tr>
<td><strong>Graphics Display Options</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifies the length of arrowhead in character cells</td>
<td>ACTNET</td>
<td>ARROWHEAD=</td>
</tr>
<tr>
<td>Centers each ID variable within node</td>
<td>ACTNET</td>
<td>CENTERID</td>
</tr>
<tr>
<td>Compresses the diagram to a single page</td>
<td>ACTNET</td>
<td>COMPRESS</td>
</tr>
</tbody>
</table>
### Table 9.1 continued

<table>
<thead>
<tr>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifies the text font</td>
<td>ACTNET</td>
<td>FONT=</td>
</tr>
<tr>
<td>Specifies the text height</td>
<td>ACTNET</td>
<td>HEIGHT=</td>
</tr>
<tr>
<td>Specifies the horizontal margin in character cells</td>
<td>ACTNET</td>
<td>HMARGIN=</td>
</tr>
<tr>
<td>Specifies the number of horizontal pages</td>
<td>ACTNET</td>
<td>HPAGES=</td>
</tr>
<tr>
<td>Specifies the reference line style</td>
<td>ACTNET</td>
<td>LREF=</td>
</tr>
<tr>
<td>Specifies the reference break line style</td>
<td>ACTNET</td>
<td>LREFBRK=</td>
</tr>
<tr>
<td>Specifies the width of lines used for critical arcs</td>
<td>ACTNET</td>
<td>LWCRIT=</td>
</tr>
<tr>
<td>Specifies the width of lines</td>
<td>ACTNET</td>
<td>LWIDTH=</td>
</tr>
<tr>
<td>Specifies the width of outline for nodes</td>
<td>ACTNET</td>
<td>LWOUTLINE=</td>
</tr>
<tr>
<td>Suppresses filling of arrowheads</td>
<td>ACTNET</td>
<td>NOARROWFILL</td>
</tr>
<tr>
<td>Suppresses page number</td>
<td>ACTNET</td>
<td>NOPAGENUMBER</td>
</tr>
<tr>
<td>Suppresses vertical centering</td>
<td>ACTNET</td>
<td>NOVCENTER</td>
</tr>
<tr>
<td>Specifies the number of nodes in horizontal direction</td>
<td>ACTNET</td>
<td>NXNODENES=</td>
</tr>
<tr>
<td>Specifies the number of nodes in vertical direction</td>
<td>ACTNET</td>
<td>NYNODENES=</td>
</tr>
<tr>
<td>Displays page number at upper right corner</td>
<td>ACTNET</td>
<td>PAGENUMBER</td>
</tr>
<tr>
<td>Specifies the PATTERN variable</td>
<td>ACTNET</td>
<td>PATTERN=</td>
</tr>
<tr>
<td>Proportionally compresses the diagram</td>
<td>ACTNET</td>
<td>PCOMPRESS</td>
</tr>
<tr>
<td>Draws arcs with rectangular corners</td>
<td>ACTNET</td>
<td>RECTILINEAR</td>
</tr>
<tr>
<td>Reverses the order of the y pages</td>
<td>ACTNET</td>
<td>REVERSEY</td>
</tr>
<tr>
<td>Rotates the network diagram</td>
<td>ACTNET</td>
<td>ROTATE</td>
</tr>
<tr>
<td>Rotates text within node by 90 degrees</td>
<td>ACTNET</td>
<td>ROTATETEXT</td>
</tr>
<tr>
<td>Draws arcs along distinct tracks</td>
<td>ACTNET</td>
<td>SEPARATEARCS</td>
</tr>
<tr>
<td>Specifies the vertical margin in character cells</td>
<td>ACTNET</td>
<td>VMARGIN=</td>
</tr>
<tr>
<td>Specifies the number of vertical pages</td>
<td>ACTNET</td>
<td>VPAGES=</td>
</tr>
</tbody>
</table>

#### Layout Options

<table>
<thead>
<tr>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breaks cycles in cyclic networks</td>
<td>ACTNET</td>
<td>BREAKCYCLE</td>
</tr>
<tr>
<td>Uses dynamic programming algorithm to route arcs</td>
<td>ACTNET</td>
<td>DP</td>
</tr>
<tr>
<td>Specifies the number of horizontal tracks between nodes</td>
<td>ACTNET</td>
<td>HTRACKS=</td>
</tr>
<tr>
<td>Routes arcs along potential node positions</td>
<td>ACTNET</td>
<td>NODETRACK</td>
</tr>
<tr>
<td>Disables use of dynamic programming to route arcs</td>
<td>ACTNET</td>
<td>NONDP</td>
</tr>
<tr>
<td>Blocks track along potential node positions</td>
<td>ACTNET</td>
<td>NONODETRACK</td>
</tr>
<tr>
<td>Restricts scope of arc layout algorithm</td>
<td>ACTNET</td>
<td>RESTRICTSEARCH</td>
</tr>
<tr>
<td>Uses spanning tree layout</td>
<td>ACTNET</td>
<td>SPANNINGTREE</td>
</tr>
<tr>
<td>Draws network as a tree, if possible</td>
<td>ACTNET</td>
<td>TREE</td>
</tr>
<tr>
<td>Specifies the number of vertical tracks between nodes</td>
<td>ACTNET</td>
<td>VTRACKS=</td>
</tr>
</tbody>
</table>

#### Line-Printer Options

<table>
<thead>
<tr>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifies the reference break line character</td>
<td>ACTNET</td>
<td>BRKCHAR=</td>
</tr>
<tr>
<td>Specifies the characters for node outlines and connections</td>
<td>ACTNET</td>
<td>FORMCHAR=</td>
</tr>
<tr>
<td>Specifies the reference character</td>
<td>ACTNET</td>
<td>REFCHAR=</td>
</tr>
</tbody>
</table>

#### Mode Options

<table>
<thead>
<tr>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invokes full-screen version</td>
<td>NETDRAW</td>
<td>FULLSCREEN</td>
</tr>
</tbody>
</table>
### Table 9.1 continued

<table>
<thead>
<tr>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invokes graphics version</td>
<td>NETDRAW</td>
<td>GRAPHICS</td>
</tr>
<tr>
<td>Invokes line-printer version</td>
<td>NETDRAW</td>
<td>LINEPRINTER</td>
</tr>
<tr>
<td>Suppresses display of diagram</td>
<td>NETDRAW</td>
<td>NODISPLAY</td>
</tr>
</tbody>
</table>

#### Network Specifications

- Specifies the ACTIVITY variable: ACTNET ACTIVITY=
- Specifies the LAG variables: ACTNET LAG=
- Specifies the SUCCESSOR variables: ACTNET SUCCESSOR=

#### Time-Scale Options

- Specifies the ALIGN variable: ACTNET ALIGN=
- Draws reference lines at every level: ACTNET AUTOREF
- Draws a border around the network diagram and axes: ACTNET FRAME
- Draws all vertical levels: ACTNET LINEAR
- Specifies the maximum number of empty columns between tick marks: ACTNET MAXNULLCOLUMN=
- Specifies the smallest interval per level: ACTNET MININTERVAL=
- Specifies the number of levels per tick mark: ACTNET NLEVELS PERCOLUMN=
- Suppresses time axis on continuation pages: ACTNET NOREPEATAXIS
- Omits the time axis: ACTNET NOTIMEAXIS
- Stops procedure if align value is missing: ACTNET QUITMISSINGALIGN
- Draws zigzag reference line at breaks: ACTNET REFBREAK
- Shows all breaks in time axis: ACTNET SHOWBREAK
- Draws time-scaled diagram: ACTNET TIMESCALE
- Uses format of variable and not default: ACTNET USEFORMAT

#### Tree Options

- Centers each node with respect to subtree: ACTNET CENTERSUBTREE
- Specifies the order of the children of each node: ACTNET CHILDORDER=
- Separates the sons of a node for symmetry: ACTNET SEPARATESONS
- Uses spanning tree layout: ACTNET SPANNINGTREE
- Draws network as a tree, if possible: ACTNET TREE

#### Web Options

- Specifies the Image map output data set: NETDRAW IMAGEMAP=
- Specifies the HTML variable: ACTNET WEB=

#### Zone Options

- Divides network into connected components: ACTNET AUTOZONE
- Suppresses zone labels: ACTNET NOZONELABEL
- Specifies the ZONE variable: ACTNET ZONE=
- Labels zones: ACTNET ZONELABEL
- Sets missing pattern values using zone: ACTNET ZONEPAT
- Leaves extra space between zones: ACTNET ZONESPACE
**PROC NETDRAW Statement**

```sas
PROC NETDRAW options ;
```

The following options can appear in the PROC NETDRAW statement.

- **ANNOTATE=SAS-data-set**
  specifies the input data set that contains the appropriate annotate variables for the purpose of adding text and graphics to the network diagram. The data set specified must be an Annotate data set. See the section “Using the Annotate Facility” on page 705 for further details about this option.

- **DATA=SAS-data-set**
  names the SAS data set to be used by PROC NETDRAW for producing a network diagram. If DATA= is omitted, the most recently created SAS data set is used. This data set, also referred to as the Network data set, contains the network information (ACTIVITY and SUCCESSOR variables) and any ID variables that are to be displayed within the nodes. For details about this data set, see the section “Network Input Data Set” on page 689.

- **FULLSCREEN**
  indicates that the network be drawn in full-screen mode. This enables you to view the network diagram produced by NETDRAW in different scales; you can also move nodes around the diagram to modify the layout.

- **GOUT=graphics-catalog**
  specifies the name of the graphics catalog used to save the output produced by PROC NETDRAW for later replay. This option is valid only if the GRAPHICS option is specified.

- **GRAPHICS**
  indicates that the network diagram produced be of high-resolution quality. If you specify the GRAPHICS option, but you do not have SAS/GRAPH software licensed at your site, the procedure stops and issues an error message. GRAPHICS is the default mode.

- **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 Network data set. This gives you complete control over the appearance and structure of your HTML pages.

- **LINEPRINTER**
  produces a network diagram of line-printer quality.

- **NODISPLAY**
  requests the procedure not to display any output. The procedure still produces the Layout data set containing the details about the network layout. This option is useful to determine node placement and arc routing for a network that can be used at a later time to display the diagram.
OUT=SAS-data-set

specifies a name for the output data set produced by PROC NETDRAW. This data set, also referred to as the Layout data set, contains the node and arc placement information determined by PROC NETDRAW to draw the network. This data set contains all the information that was specified in the Network data set to define the project; in addition, it contains variables that specify the coordinates for the nodes and arcs of the network diagram. For details about the Layout data set, see the section “Layout Data Set” on page 696.

If the OUT= option is omitted, the procedure creates a data set and names it according to the DATA\n convention.

ACTNET Statement

ACTNET / options ;

The ACTNET statement draws the network diagram. You can specify several options in this statement to control the appearance of the network. All these options are described in the current section 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 (full-screen, graphics, or line-printer) of invocation of the procedure.

General Options

ACTIVITY=variable

specifies the variable in the Network data set that names the nodes in the network. If the data set contains a variable called _FROM_, this specification is ignored; otherwise, this option is required.

ALIGN=variable

specifies the variable in the Network data set containing the time values to be used for positioning each activity. This options triggers the TIMESCALE option that adds a time axis at the top of the network and aligns the nodes of the network according to the values of the ALIGN= variable. The minimum and maximum values of this variable are used to determine the time axis. The format of this variable is used to determine the default value of the MININTERVAL= option, which, in turn, determines the format of the time axis.

AUTOREF

draws reference lines at every tick mark. This option is valid only for time-scaled network diagrams.

AUTOZONE

enables automatic zoning (or dividing) of the network into connected components. This option is equivalent to defining an automatic zone variable that associates a tree number for each node. The tree number refers to a number assigned (by the procedure) to each distinct tree of a spanning tree of the network.

BREAKCYCLE

breaks cycles by reversing the back arcs of the network. The back arcs are determined by constructing an underlying spanning tree of the network. Once cycles are broken, the nodes of the network are laid out using a topological ordering of the new network formed from the original network by ignoring the back arcs. The back arcs are drawn after determining the network layout. Note that only the back arcs go from right to left.
Chapter 9: The NETDRAW Procedure

**BOXHT**=`boxht`
specifies the height of the box (in character cell positions) used for denoting a node. If this option is not specified, the height of the box equals the number of lines required for displaying all of the ID variable values for any of the nodes. See the ROTATETEXT option (under “Graphics Options”) for an exception.

**BOXWIDTH**=`boxwdth`
specifies the width of the box (in character cell positions) used for denoting a node. If this option is not specified, the width of the box equals the maximum number of columns required for displaying all of the ID variable values for any of the nodes. See the ROTATETEXT option (under “Graphics Options”) for an exception.

**CENTERSUBTREE**
positions each node at the center of the subtree that originates from that node instead of placing it at the midpoint of its children (which is the default behavior). Note that the nodes are placed at integral positions along an imaginary grid, so the positioning may not be exactly at the center. This option is valid only in conjunction with the TREE option.

**CHILDORDER**=`order`
orders the children of each node when the network is laid out using either the TREE or the SPANNINGTREE option. The valid values for this option are TOPDOWN and BOTTOMUP for default orientation, and LEFTRIGHT and RIGHTLEFT for rotated networks (drawn with the ROTATETEXT option). The default is TOPDOWN.

**DP**
causes PROC NETDRAW to use a dynamic programming (DP) algorithm to route the arcs. This DP algorithm is memory and CPU-intensive and is not necessary for most applications.

**DURATION**=`variable`
specifies a variable that contains the duration of each activity in the network. This value is used only for displaying the durations of each activity within the node.

**FRAME**
encloses the drawing area with a border. This option is valid only for time-scaled or zoned network diagrams.

**HTRACKS**=`integer`
controls the number of arcs that are drawn horizontally through the space between two adjacent nodes. This option enables you to control the arc-routing algorithm. The default value is based on the maximum number of successors of any node.

**ID**=`(variables)`
specifies the variables in the Network data set that are displayed within each node. In addition to the ID variables, the procedure displays the ACTIVITY variable, the DURATION variable (if the DURATION= option was specified), and any of the following variables in the Network data set: E_START, E_FINISH, L_START, L_FINISH, S_START, S_FINISH, A_START, A_FINISH, T_FLOAT, and F_FLOAT. See Chapter 4, “The CPM Procedure,” for a description of these variables. If you specify the NODEFID option, only the variables listed in the ID= option are displayed.
LAG=variable
LAG=(variables)
specifies the variables in the Network data set that identify 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, or FF, which denote ‘Finish-to-Start’, ‘Start-to-Start’, ‘Start-to-Finish’, and ‘Finish-to-Finish’, respectively. You can also use the keyword_duration_calendar specification used by the CPM procedure, although PROC NETDRAW 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 NETDRAW interprets the lag as a ‘Finish-to-Start’ type.

This option enables the procedure to identify the different types of nonstandard precedence constraints (Start-to-Start, Start-to-Finish, and Finish-to-Finish) on graphics quality network diagrams by drawing the arcs from and to the appropriate edges of the nodes.

LINEAR
plots one column for every $mininterval$ between the minimum and maximum values of the ALIGN= variable. By default, only those columns that contain at least one activity are displayed. This option is valid only for time-scaled network diagrams.

MAXNULLCOLUMN=maxncol
MAXEMPTY=maxncol
MAXZCOL=maxncol
MAXNCOL=maxncol
specifies the maximum number of empty columns between two consecutive nonempty columns. The default value for this option is 0. Note that specifying the LINEAR option is equivalent to specifying the MAXNULLCOLUMN= option to be infinity. This option is valid only for time-scaled network diagrams.

MININTERVAL=mininterval
specifies the smallest interval to be used per column of the network diagram. Thus, if MININTERVAL=DAY, each column is used to represent a day, and all activities that start on the same day are placed in the same column. The valid values for $mininterval$ are SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QTR, and YEAR. The default value of $mininterval$ is determined by the format of the ALIGN= variable. The tick labels are formatted on the basis of $mininterval$; for example, if $mininterval$ is DAY, the dates are marked using the DATE7. format, and if $mininterval$ is HOUR, the labels are formatted as TIME5. and so on. This option is valid only for time-scaled network diagrams.

NLEVELSPERCOLUMN=npercol
NPERCOL=npercol
contracts the time axis by specifying that activities that differ in ALIGN= value by less than npercol units of MININTERVAL can be plotted in the same column. The default value of npercol is 1. This option is valid only for time-scaled network diagrams.
NODEFID
indicates that the procedure need not check for any of the default ID variables in the Network data set; if this option is in effect, only the variables specified in the ID= option are displayed within each node.

NODETRACK
specifies that the arcs can be routed along potential node positions if there is a clear horizontal track to the left of the successor (or _TO_) node. This is the default option. To prevent the use of potential node positions, use the NONODETRACK option.

NOLABEL
suppresses the labels. By default, the procedure uses the first three letters of the variable name to label all the variables that are displayed within each node of the network. The only exception is the variable that is identified by the ACTIVITY= option.

NONDP
uses a simple heuristic to connect the nodes. The default mode of routing is NONDP, unless the HTRACKS= or VTRACKS= option (or both) are specified and set to a number that is less than the maximum number of successors. The NONDP option is faster than the DP option.

NONODETRACK
blocks the horizontal track along potential node positions. This option may lead to more turns in some of the arcs. The default is NODETRACK.

NOREPEATAXIS
displays the time axis only on the top of the chart and not on every page. This option is useful if the different pages are to be glued together to form a complete diagram. This option is valid only for time-scaled network diagrams.

NOTIMEAXIS
suppresses the display of the time axis and its labels. Note that the nodes are still placed according to the time scale, but no axis is drawn. This option is valid only for time-scaled network diagrams.

NOZONELABEL
NOZONEDESCR
omits the zone labeling and the dividing lines. The network is still divided into zones based on the ZONE variable, but there is no demarcation or labeling corresponding to the zones.

PAGES=npages
specifies the maximum number of pages to be used for the network diagram in graphics and line-printer modes. The default value is 100.

QUITMISSINGALIGN
stops processing if the ALIGN= variable has any missing values. By default, the procedure tries to fill in missing values using the topological order of the network. This option is valid only for time-scaled network diagrams.

REFBREAK
shows breaks in the time axis by drawing a zigzag line down the diagram just before the tick mark at the break. This option is valid only for time-scaled network diagrams.
ACTNET Statement

RESTRICTSEARCH

RSEARCH restricts the scope of the arc layout algorithm by restricting the area of search for the arc layout when the DP option is in effect; this is useful in reducing the computational complexity of the dynamic programming algorithm. By default, using the DP algorithm to route the arcs, the \( y \) coordinates of the arcs can range through the entire height of the network. The RESTRICTSEARCH option limits the \( y \) coordinates to the minimum and the maximum of the \( y \) coordinates of the node and its immediate successors.

SEPARATESONS

separates the children (immediate successors) of a given node by adding an extra space in the center whenever it is needed to enable the node to be positioned at integral \((x, y)\) coordinates. For example, if a node has two children, placing the parent node at the midpoint between the two children requires the \( y \) coordinate to be noninteger, which is not allowed in the Layout data set. By default, the procedure positions the node at the same \( y \) level as one of its children. The SEPARATESONS option separates the two children by adding a dummy child in between, thus enabling the parent node to be centered with respect to its children. This option is valid only in conjunction with the TREE option.

SHOWBREAK

shows breaks in the time axis by drawing a jagged break in the time axis line just before the tick mark corresponding to the break. This option is valid only for time-scaled network diagrams.

SHOWSTATUS

uses the variable STATUS (if it exists) in the Network data set to determine if an activity is in-progress or completed. Note that the STATUS variable exists in the Schedule data set produced by PROC CPM when used with an ACTUAL statement. If there is no STATUS variable or if the value is missing, the procedure uses the A_FINISH and A_START values to determine the status of the activity. If the network is drawn in line-printer or full-screen mode, activities in progress are outlined with the letter \( P \) and completed activities are outlined with the letter \( F \); in high-resolution graphics mode, in-progress activities are marked with a diagonal line across the node from the bottom left to the top right corner, while completed activities are marked with two diagonal lines.

SPANNINGTREE

uses a spanning tree to place the nodes in the network. This method typically results in a wider layout than the default. However, for networks that have totally disjoint pieces, this option separates the network into connected components (or disjoint trees). This option is not valid for time-scaled or zoned network diagrams, because the node placement dictated by the spanning tree may not be consistent with the zone or the tickmark corresponding to the node.

SUCCESSOR=(variables)

specifies the variables in the Network data set that name all the immediate successors of the node specified by the ACTIVITY variable. This specification is ignored if the data set contains a variable named \_TO_. At least one SUCCESSOR variable must be specified if the data set does not contain a variable called \_TO_.

TIMESCALE

indicates that the network is to be drawn using a time axis for placing the nodes. This option can be used to align the network according to default variables. If the TIMESCALE option is specified without the ALIGN= option, the procedure looks for default variables in the following order: E_START,
Chapter 9: The NETDRAW Procedure

L_START, S_START, and A_START. The first of these variables that is found is used as the ALIGN= variable.

TREE
TREELAYOUT
requests the procedure to draw the network as a tree if the network is indeed a tree (that is, all the nodes have at most one immediate predecessor). The option is ignored if the network does not have a tree structure.

USEFORMAT
indicates that the explicit format of the ALIGN= variable is to be used instead of the default format based on the MININTERVAL= option. Thus, for example, if the ALIGN variable contains SAS date values, by default, the procedure uses the DATE7. format for the time axis labels irrespective of the format of the ALIGN= variable. The USEFORMAT option specifies that the variable’s format should be used for the labels instead of the default format. This option is valid only for time-scaled network diagrams.

VTRACKS=integer
controls the number of arcs that are drawn vertically through the space between two adjacent nodes. A default value is based on the maximum number of successors of any node.

XBETWEEN=integer
HBETWEEN=integer
specifies the horizontal distance (in character cell positions) between two adjacent nodes. The value for this option must be at least 3; the default value is 5.

YBETWEEN=integer
VBETWEEN=integer
specifies the vertical distance (in character cell positions) between two adjacent nodes. The value for this option must be at least 3; the default value is 5.

ZONE=variable
names the variable in the Network data set used to separate the network diagram into zones.

ZONELABEL
ZONEDESCR
labels the different zones and draws dividing lines between two consecutive zones. This is the default behavior; to omit the labels and the dividing lines, use the NOZONELABEL option.

ZONESPACE
ZONELEVADD
draws the network with an extra row between two consecutive zones.

Full-Screen Options

BRKCHAR=brkchar
specifies the character used for drawing the zigzag break lines down the chart at break points of the time axis. The default value is >. This option is valid only for time-scaled network diagrams.
CARCS=color
specifies the color of the connecting lines (or arcs) between the nodes. The default value of this option is CYAN.

CAXIS=color
specifies the color of the time axis. The default value is WHITE. This option is valid only for time-scaled network diagrams.

CCRITARCS=color
specifies the color of arcs connecting critical activities. The procedure uses the values of the E_FINISH and L_FINISH variables (if they are present) in the Network data set to determine the critical activities. The default value is the value of the CARCS= option.

CREF=color
specifies the color of the reference lines. The default value is WHITE. This option is valid only for time-scaled network diagrams.

CREFBRK=color
specifies the color of the lines drawn to denote breaks in the time axis. The default value is WHITE. This option is valid only for time-scaled network diagrams.

FORMCHAR [index list]=‘string’
specifies the characters used for node outlines and arcs. See the section “Line-Printer Options” on page 689 for a description of this option.

PATTERN=variable
specifies an integer-valued variable in the Network data set that identifies the color number for each node of the network. If the data set contains a variable called _PATTERN, this specification is ignored. All the colors available for the full-screen device are used in order corresponding to the number specified in the PATTERN variable; if the value of the PATTERN variable is more than the number of colors available for the device, the colors are repeated starting once again with the first color. If a PATTERN variable is not specified, the procedure uses the first color for noncritical activities, the second color for critical activities, and the third color for supercritical activities.

REFCHAR=refchar
specifies the reference character used for drawing reference lines. The default value is “|”. This option is valid only for time-scaled network diagrams.

ZONEPAT
indicates that if a PATTERN variable is not specified or is missing and if a ZONE= variable is present, then the node colors are based on the value of the ZONE= variable.

**Graphics Options**

ANNOTATE=SAS-data-set
specifies the input data set that contains the appropriate annotate variables for the purpose of adding text and graphics to the network diagram. The data set specified must be an Annotate data set. See the section “Using the Annotate Facility” on page 705 for further details about this option.
**ARROWHEAD=integer**

specifies the length of the arrowhead in character cell positions. You can specify ARROWHEAD = 0 to suppress arrowheads altogether. The default value is 1.

**CARCS=color**

specifies the color to use for drawing the connecting lines between the nodes. The default color depends on the GOPTIONS statement and the GSTYLE system option; see the section “**ODS Style Templates**” on page 707 for more information.

**CAXIS=color**

specifies the color of the time axis. This option is valid only for time-scaled network diagrams. The default color depends on the GSTYLE system option and the value of the CTEXT= option; see the section “**ODS Style Templates**” on page 707 for more information.

**CCNODEFILL=color**

specifies the fill color for all critical nodes of the network diagram. If you specify this option, the procedure uses a solid fill pattern (with the color specified in this option) for all critical nodes, ignoring any fill pattern specified in the PATTERN statements; the PATTERN statements are used only to obtain the color of the outline for these nodes unless you specify the **CCRITOUT=** option. The default value for this option is the value of the **CNODEFILL=** option, if it is specified; otherwise, the procedure uses the PATTERN statements to determine the fill pattern and color.

**CCRITARCS=color**

specifies the color of arcs connecting critical activities. The procedure uses the values of the E_FINISH and L_FINISH variables (if they are present) in the Network data set to determine the critical activities. The default value of this option is the value of the **CARCS=** option.

**CCRITOUT=color**

specifies the outline color for critical nodes. The default value for this option is the value of the **COUTLINE=** option, if it is specified; otherwise, it is the same as the pattern color for the node.

**CENTERID**

centers the ID values placed within each node. By default, character valued ID variables are left justified and numeric ID variables are right justified within each node. This option centers the ID values within each node.

**CNODEFILL=color**

specifies the fill color for all nodes of the network diagram. If you specify this option, the procedure uses a solid fill pattern with the specified color, ignoring any fill pattern specified in the PATTERN statements; the PATTERN statements are used only to obtain the color of the outline for the nodes, unless you specify the **COUTLINE=** option.

**COMPRESS**

draws the network on one physical page. By default, the procedure draws the network across multiple pages if necessary, using a default scale that allots one character cell position for each letter within the nodes. Sometimes, to get a broad picture of the network and all its connections, you may want to view the entire network on one screen. If the **COMPRESS** option is specified, PROC NETDRAW determines the horizontal and vertical transformations needed so that the network is compressed to fit on one screen.
COUTLINE=color

specifies an outline color for all nodes. By default, the procedure sets the outline color for each node to be the same as the fill pattern for the node. This option is useful when used in conjunction with a solid fill using a light color. Note that if an empty fill pattern is specified, then the COUTLINE= option will cause all nodes to appear the same.

CREF=color

specifies the color of the reference lines. This option is valid only for time-scaled network diagrams. The default color depends on the GSTYLE system option and the value of the CTEXT= option; see the section “ODS Style Templates” on page 707 for more information.

CREFBRK=color

specifies the color of the zigzag break lines. This option is valid only for time-scaled network diagrams. The default color depends on the GSTYLE system option and the value of the CTEXT= option; see the section “ODS Style Templates” on page 707 for more information.

CTEXT=color

 specifies the color of all text on the network diagram including variable names or labels, 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 707 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 in PROC GREPLAY. If the DESCRIPTION= option is omitted, the description field contains a description assigned by PROC NETDRAW.

FILLPAGE

causes the diagram on each page to be magnified (if necessary) to fill up the page.

FONT=font

 specifies the font of the text. The default font depends on the GOPTIONS statement and the GSTYLE system option; see the section “ODS Style Templates” on page 707 for more information.

HEIGHT=h

HTEXT=h

 specifies that the height for all text in PROC NETDRAW (excluding the titles and footnotes) be h times the value of the global HTEXT= option, which is the default text height specified in the GOPTIONS statement of SAS/GRAPH. The value of h must be a positive real number; the default value is 1.0.

HMARGIN=integer

 specifies the width of a horizontal margin (in number of character cell positions) for the network in graphics mode. The default width is 1.

HPAGES=h

NXPAGES=h

 specifies that the network diagram is to be produced using h horizontal pages. However, it may not be possible to use h horizontal pages due to intrinsic constraints on the output.
For example, PROC NETDRAW requires that every horizontal page should contain at least one x level. Thus, the number of horizontal pages can never exceed the number of vertical levels in the network. The exact number of horizontal pages used by the network diagram is given in the _ORNETDR macro variable. See the section “Macro Variable _ORNETDR” on page 706 for further details.

The appearance of the diagram 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 (unless the HPAGES= and VPAGES= options are specified). If neither the COMPRESS nor PCOMPRESS option is specified, the number of vertical pages is computed in order to display as much of the chart as possible in a proportional manner.

LREF=linestyle
specifies the linestyle (1-46) of the reference lines. The default linestyle is 1, a solid line. See Figure 8.5 in Chapter 8, “The GANTT Procedure,” for examples of the various line styles available. This option is valid only for time-scaled network diagrams.

LREFBRK=linestyle
specifies the linestyle (1-46) of the zigzag break lines. The default linestyle is 1, a solid line. See Figure 8.5 in Chapter 8, “The GANTT Procedure,” for examples of the various line styles available. This option is valid only for time-scaled network diagrams.

LWCRIT=integer
specifies the line width for critical arcs and the node outlines for critical activities. If the LWCRIT= option is not specified, the procedure uses the value specified for the LWIDTH= option.

LWIDTH=integer
specifies the line width of the arcs and node outlines. The default line width is 1.

LWOUTLINE=integer
specifies the line width of the node outlines. The default line width for the node outline is equal to LWIDTH for noncritical nodes and LWCRIT for critical nodes.

NAME='string'
specifies a string of up to eight characters that appears in the name field of the catalog entry for the graph. The default name is NETDRAW. If either the name specified or the default name duplicates an existing name in the catalog, then the procedure adds a number to the duplicate name to create a unique name, for example, NETDRAW2.

NOARROWFILL
draws arrowheads that are not filled. By default, the procedure uses filled arrowheads.

NOPAGENUMBER
NONUMBER
suppresses the page numbers that are displayed in the top right corner of each page of a multipage network diagram. Note that the pages are ordered from left to right, bottom to top (unless the REVERSEY option is specified).
NOVCENTER
draws the network diagram just below the titles without centering in the vertical direction.

NXNODES=nx
specifies the number of nodes that should be displayed horizontally across each page of the network diagram. This option determines the value of the HPAGES= option; this computed value of HPAGES overrides the specified value for the HPAGES= options.

NYNODES=ny
specifies the number of nodes that should be displayed vertically across each page of the network diagram. This option determines the value of the VPAGES= option; this computed value of VPAGES overrides the specified value for the VPAGES= options.

PAGENUMBER
PAGENUM
numbers the pages of the network diagram on the top right-hand corner of the page if the diagram exceeds one page. The numbering scheme is from left to right, bottom to top (unless the REVERSEY option is specified).

PATTERN=variable
specifies an integer-valued variable in the Network data set that identifies the pattern for filling each node of the network. If the data set contains a variable called _PATTERN, this specification is ignored. The patterns are assumed to have been specified using PATTERN statements. If a PATTERN variable is not specified, the procedure uses the first PATTERN statement for noncritical activities, the second PATTERN statement for critical activities, and the third PATTERN statement for supercritical activities.

PCOMPRESS
draws the network diagram on one physical page. As with the COMPRESS option, the procedure determines the horizontal and vertical transformation needed so that the network is compressed to fit on one screen. However, in this case, the transformations are such that the network diagram is proportionally compressed. See Example 9.4 for an illustration of this option.

If the HPAGES= and VPAGES= options are used to control the number of pages, each page of the network diagram is drawn while maintaining the original aspect ratio.

RECTILINEAR
draws arcs with rectangular corners. By default, the procedure uses rounded turning points and rounded arc merges in graphics mode.

REVERSEY
reverses the order in which the y pages are drawn. By default, the pages are ordered from bottom to top in the graphics mode. This option orders them from top to bottom.

ROTATE
rotates the network diagram to change the orientation of the network to be from top to bottom instead of from left to right. For example, you can use this option to draw a Bill of Materials diagram that is traditionally drawn from top to bottom with the Final Product drawn at the top of the tree. In addition to rotating the orientation of the network, use the ROTATETEXT option to rotate the text within each node. See Example 9.18 for an illustration of this option.

This option is similar to the global graphics option, ROTATE (GOPTIONS ROTATE). Note that if the global graphics option is used, titles and footnotes also need to be drawn with an angle specification:
A=90. However, some device drivers ignore the global graphics option, ROTATE (for example, the SASGDDMX driver). Use the ROTATE option on the ACTNET statement for such device drivers.

**ROTATETEXT**

**RTEXT**

Rotates the text within the nodes by 90 degrees. This option is useful when used in conjunction with the **ROTATE** option in the ACTNET statement (or the global graphics option ROTATE) to change the orientation of the network to be from top to bottom instead of from left to right. For example, you can use this option to draw an organizational chart that is traditionally drawn from top to bottom with the head of the organization at the top of the chart. If the ROTATETEXT option is specified, then the definitions of the **BOXHT=** and **BOXWIDTH=** options are reversed. See Example 9.18 for an illustration of this option.

**SEPARATEARCS**

Separates the arcs to follow distinct tracks. By default, the procedure draws all segments of the arcs along a central track between the nodes, which may cause several arcs to be drawn on top of one another. If the SEPARATEARCS option is specified, the procedure may increase the values of the **XBETWEEN=** and **YBETWEEN=** options to accommodate the required number of lines between the nodes.

**VMARGIN=integer**

Specifies the width of a vertical margin (in number of character cell positions) for the network. The default width is 1.

**VPAGES=v**

**NYPAGES=v**

Specifies that the network diagram is to be produced using v vertical pages. This, however, may not be possible due to intrinsic constraints on the output. For example, PROC NETDRAW requires that every vertical page should contain at least one y level. Thus, the number of vertical pages can never exceed the number of horizontal levels in the network. The exact number of vertical pages used by the procedure is provided in the _ORNETDR macro variable. See the section “Macro Variable _ORNETDR” on page 706 for further details.

The appearance of the diagram 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 (or the NXNODES= option). If the COMPRESS or PCOMPRESS option is specified (without the HPAGES= or NXNODES= options), 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 Network data set that identifies an HTML page for each activity. The procedure generates an HTML image map using this information for each node in the network diagram.
ZONEPAT
indicates that if a PATTERN= variable is not specified or is missing and if a ZONE= variable is present, then the node patterns are based on the value of the ZONE= variable.

Line-Printer Options

BRKCHAR=brkchar
specifies the character used for drawing the zigzag break lines down the chart at break points of the time axis. The default value is >. This option is valid only for time-scaled network diagrams.

FORMCHAR [index list]='string'
specifies the characters used for node outlines and arcs. The value is a string 20 characters long. The first 11 characters define the 2 bar characters, vertical and horizontal, and the 9 corner characters: upper-left, upper-middle, upper-right, middle-left, middle-middle (cross), middle-right, lower-left, lower-middle, and lower-right. These characters are used to outline each node and connect the arcs. The nineteenth character denotes a right arrow. The default value of the FORMCHAR= option is |----|+|----+=|-/<>*. Any character or hexadecimal string can be substituted to customize the appearance of the diagram. Use an index list to specify which default form character each supplied character replaces, or replace the entire default string by specifying the full character replacement string without an index list. For example, change the four corners of each node and all turning points of the arcs to asterisks by specifying

    FORMCHAR(3 5 7 9 11)= '*****'

Specifying

    formchar=' ' (11 blanks)

produces a network diagram with no outlines for the nodes (as well as no arcs). For further details about the FORMCHAR= option see Chapter 7, “The DTREE Procedure,” and Chapter 8, “The GANTT Procedure.”

REFCHAR=refchar
specifies the reference character used for drawing reference lines. The default value is “|”. This option is valid only for time-scaled network diagrams.

Details: NETDRAWE Procedure

Network Input Data Set

The Network input data set contains the precedence information, namely the activity-successor information for all the nodes in the network. The minimum amount of information that is required by PROC NETDRAW is the activity-successor information for the network. Additional information in the input data set can be used by the procedure to add detail to the nodes in the diagram or control the layout of the network diagram.
Three types of data sets are typically used as the Network data set input to PROC NETDRAW. Which type of data set you use depends on the stage of the project:

- The Activity data set that is input to PROC CPM is the first type. In the initial stages of project definition, it may be useful to get a graphical representation of the project showing all the activity precedence constraints.

- The Schedule data set produced by PROC CPM (as the OUT= data set) is the second type. When a project is in progress, you may want to obtain a network diagram showing all the relevant start and finish dates for the activities in the project, in addition to the precedence constraints. You may also want to draw a time-scaled network diagram, with the activities arranged according to the start or finish times corresponding to any of the different schedules produced by PROC CPM.

- The Layout data set produced by PROC NETDRAW (as the OUT= data set) is the third type. Often, you may want to draw network diagrams of the project every week showing updated information (as the project progresses); if the network logic has not changed, it is not necessary to determine the placement of the nodes and the routing of the arcs every time. You can use the Layout data set produced by PROC NETDRAW that contains the node and arc positions, update the start and finish times of the activities or merge in additional information about each activity, and use the modified data set as the Network data set input to PROC NETDRAW. The new network diagram will have the same layout as the earlier diagram but will contain updated information about the schedule. Such a data set may also be useful if you want to modify the layout of the network by changing the positions of some of the nodes. See the section “Controlling the Layout” on page 697 for details about how the layout information is used by PROC NETDRAW. If the Layout data set is used, it contains the variables _FROM_ and _TO_; hence, it is not necessary to specify the ACTIVITY= and SUCCESSOR= options. See Example 9.13 and Example 9.14 for illustrations of the use of the Layout data set.

The minimum information required by PROC NETDRAW from the Network data set is the variable identifying each node in the network and the variable (or variables) identifying the immediate successors of each node. In addition, the procedure can use other optional variables in the data set to enhance the network diagram. The procedure uses the variables specified in the ID= option to label each node. The procedure also looks for default variable names in the Network data set that are added to the list of ID variables; the default variable names are E_START, E_FINISH, L_START, L_FINISH, S_START, S_FINISH, A_START, A_FINISH, T_FLOAT, and F_FLOAT. The format used for determining the location of these variables within each node is described in the section “Format of the Display” on page 694. See the section “Variables in the Network Data Set” on page 691 for a table of all the variables in the Network data set and their interpretations by PROC NETDRAW.

If the Network data set contains the variables _X_ and _Y_ identifying the x and y coordinates of each node and each turning point of each arc in the network, then this information is used by the procedure to draw the network. Otherwise, the precedence relationships among the activities are used to determine the layout of the network. It is possible to specify only the node positions and let the procedure determine the routing of all the arcs. However, partial information cannot be augmented by the procedure.

**NOTE:** If arc information is provided, the procedure assumes that it is complete and correct and uses it exactly as specified.
Variables in the Network Data Set

The NETDRAW procedure expects all the network information to be contained in the Network input data set named by the DATA= option. The network information is contained in the ACTIVITY and SUCCESSOR variables. In addition, the procedure uses default variable names in the Network data set for specific purposes. For example, the _X_ and _Y_ variables, if they are present in the Network data set, represent the coordinates of the nodes, the _SEQ_ variable indexes the turning points of each arc of the network, and so on.

In addition to the network precedence information, the Network data set may also contain other variables that can be used to change the default layout of the network. For example, the nodes of the network can be aligned in the horizontal direction using the ALIGN= specification, or they can be divided into horizontal bands (or zones) using a ZONE variable.

Table 9.2 lists all of the variables associated with the Network data set and their interpretations by the NETDRAW procedure. Note that all the variables are identified to the procedure in the ACTNET statement. Some of the variables use default names that are recognized by the procedure to denote specific information, as explained previously. The table indicates if the variable is default or needs to be identified in the ACTNET statement.

### Table 9.2  Network Data Set and Associated Variables

<table>
<thead>
<tr>
<th>Statement</th>
<th>Variable Name</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTNET</td>
<td>ACTIVITY</td>
<td>Activity or node name</td>
</tr>
<tr>
<td></td>
<td>ALIGN</td>
<td>Align variable for time-scaled network</td>
</tr>
<tr>
<td></td>
<td>DURATION</td>
<td>Duration of activity</td>
</tr>
<tr>
<td></td>
<td>ID</td>
<td>Additional variables to be displayed</td>
</tr>
<tr>
<td></td>
<td>PATTERN</td>
<td>Pattern number</td>
</tr>
<tr>
<td></td>
<td>SUCCESSOR</td>
<td>Immediate successor</td>
</tr>
<tr>
<td></td>
<td>WEB</td>
<td>HTML page corresponding to activity</td>
</tr>
<tr>
<td></td>
<td>ZONE</td>
<td>Zone variable for dividing network</td>
</tr>
<tr>
<td>Default</td>
<td>A_FINISH</td>
<td>Default ID variable</td>
</tr>
<tr>
<td>Variable</td>
<td>A_START</td>
<td>Default ID variable</td>
</tr>
<tr>
<td>Names</td>
<td>E_FINISH</td>
<td>Default ID variable</td>
</tr>
<tr>
<td></td>
<td>E_START</td>
<td>Default ID variable</td>
</tr>
<tr>
<td></td>
<td>F_FLOAT</td>
<td>Default ID variable</td>
</tr>
<tr>
<td></td>
<td>L_FINISH</td>
<td>Default ID variable</td>
</tr>
<tr>
<td></td>
<td>L_START</td>
<td>Default ID variable</td>
</tr>
<tr>
<td></td>
<td>S_FINISH</td>
<td>Default ID variable</td>
</tr>
<tr>
<td></td>
<td>S_START</td>
<td>Default ID variable</td>
</tr>
<tr>
<td></td>
<td>T_FLOAT</td>
<td>Default ID variable</td>
</tr>
<tr>
<td></td>
<td><em>FROM</em></td>
<td>Supersedes ACTIVITY= specification</td>
</tr>
<tr>
<td></td>
<td>_PATTERN</td>
<td>Supersedes PATTERN= specification</td>
</tr>
<tr>
<td></td>
<td><em>SEQ</em></td>
<td>Index of turning point in arc</td>
</tr>
<tr>
<td></td>
<td><em>TO</em></td>
<td>Supersedes SUCCESSOR= specification</td>
</tr>
<tr>
<td></td>
<td><em>X</em></td>
<td>x coordinate of node or arc turning point</td>
</tr>
<tr>
<td></td>
<td><em>Y</em></td>
<td>y coordinate of node or arc turning point</td>
</tr>
</tbody>
</table>
Missing Values

Missing values are not allowed for the ACTIVITY, _X_, _Y_, and _SEQ_ variables. Missing values for the SUCCESSOR and ID variables are ignored. Missing values are not allowed for the ALIGN= variable if the QUITMISSINGALIGN option is specified; otherwise, the procedure determines suitable values for the ALIGN= variable using the topological ordering of the network nodes.

Layout of the Network

The network layout is determined in two stages. First, the precedence relationships are used to determine the positions of the nodes, which are then used to determine a routing of the arcs. The positions of the nodes and arcs are identified by specifying their $x$ and $y$ coordinates in a grid. Figure 9.7 shows a sample grid and explains some of the conventions followed by PROC NETDRAW in determining the node and arc layout. This notation will be useful in later sections that describe the Layout data set and how you can control the layout of the diagram. The asterisks in the figure represent possible positions for the nodes of the network. The arcs are routed between the possible node positions. For example, node A has coordinates (1, 3) and node B has coordinates (2, 1). The arc connecting them has two turning points and is completely determined by the two pairs of coordinates (1.5, 3) and (1.5, 1); here, $x = 1.5$ implies that the position is midway between the $x$ coordinates 1 and 2.
PROC NETDRAW sets $x = 1$ for all nodes with no predecessors; the $x$ coordinates for the other nodes are determined so that each node is placed to the immediate right of all its predecessors; in other words, no node will appear to the left of any of its predecessors or to the right of any of its successors in the network diagram. The nodes are placed in topological order: a node is placed only after all its predecessors have been placed. Thus, the node-placement algorithm requires that there should be no cycles in the network. The $y$ coordinates of the nodes are determined by the procedure using several heuristics designed to produce a reasonable compact diagram of the network. To draw a network that has cycles, use the BREAKCYCLE option, or you can specify the node coordinates or an ALIGN= variable to circumvent the requirement of a topological ordering of the nodes (see the second part of Example 9.12).

Note that the $x$ and $y$ coordinates fix only a relative positioning of the nodes and arcs. The actual distance between two nodes, the width and height of each node, and so on can be controlled by specifying desired values for the options that control the format of the display, namely, BOXHT=, BOXWIDTH=, and so on. See the section “Format of the Display” on page 694 for details about these options.

By default, the procedure routes the arcs using a simple heuristic that uses, at most, four turning points: the arc leaves the predecessor node from its right edge, turns up or down according to whether the successor is above or below the current node position, then tracks horizontally across to the vertical corridor just before the successor node, and then tracks in a vertical direction to meet the successor node. For example, see the tracking of the arc connecting nodes C and D in Figure 9.7.
Chapter 9: The NETDRAW Procedure

For networks that include some nonstandard precedence constraints, the arcs may be drawn from and to the appropriate edges of the nodes, depending on the type of the constraint.

The default routing of the arcs may lead to an unbalanced diagram with too many arcs in one section and too few in another. The **DP** option in the **ACTNET** statement causes the procedure to use a dynamic programming algorithm to route the arcs. This algorithm tries to route the arcs between the nodes so that not too many arcs pass through any interval between two nodes. The procedure sets the maximum number of arcs that are allowed to be routed along any corridor to be equal to the maximum number of successors for any node. The **HTRACKS=** and **VTRACKS=** options enable you to set these maximum values: **HTRACKS** specifies the maximum number of arcs that are allowed to pass horizontally through any point while **VTRACKS** specifies the same for arcs in the vertical direction. See Example 9.7 for an illustration of the **HTRACKS=** option.

The layout of the network for time-scaled and zoned network diagrams is discussed in the section “Time-Scaled Network Diagrams” on page 698 and the section “Zoned Network Diagrams” on page 700, respectively. The section “Organizational Charts or Tree Diagrams” on page 700 describes the layout of the diagram when the **TREE** option is specified.

---

**Format of the Display**

As explained in the previous section, the layout of the network is determined by the procedure in terms of \( x \) and \( y \) coordinates on a grid as shown in Figure 9.7. The distance between nodes and the width and height of each node is determined by the values of the format control options: **XBETWEEN=**, **YBETWEEN=**, **BOXHT=**, and **BOXWIDTH=**. Note that if the **ROTATETEXT** option is specified (in graphics mode), then the definitions of the **BOXHT=** and **BOXWIDTH=** options are reversed.

The amount of information that is displayed within each node is determined by the variables specified by the **ID=** option, the number of default variables found in the Network data set, and whether the **NOLABEL** and **NODEFID** options are specified. The values of the variables specified by the **ID=** option are placed within each node on separate lines. If the **NOLABEL** option is in effect, only the values of the variables are written; otherwise, each value is preceded by the name of the ID variable truncated to three characters. Recall from the section “Syntax: NETDRAW Procedure” on page 672 that, in addition to the variables specified using the **ID=** option, the procedure also displays additional variables. These variables are displayed below the variables explicitly specified by the **ID=** option, in pre-determined relative positions within each node (see Table 9.3.)

---

**Table 9.3** Display Format for the Variables within Each Node

<table>
<thead>
<tr>
<th>ID1</th>
<th>[.]</th>
<th>[.]</th>
<th>IDn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity variable</td>
<td>Duration variable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E_START</td>
<td>E_FINISH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L_START</td>
<td>L_FINISH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_START</td>
<td>S_FINISH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_START</td>
<td>A_FINISH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_FLOAT</td>
<td>F_FLOAT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NOTE: If a node is identified as a successor (through a SUCCESSOR variable) and is never identified with the ACTIVITY variable, the ID values for this node are never defined in any observation; hence, this node will have missing values for all the ID variables.

If the SHOWSTATUS option is specified and the Network data set contains progress information (in either the STATUS variable or the A_START and A_FINISH variables), the procedure appropriately marks each node referring to activities that are completed or in progress. See Example 9.8 for an illustration of the SHOWSTATUS option.

The features just described pertain to all three modes of the procedure. In addition, there are options to control the format of the display that are specific to the mode of invocation of the procedure. For graphics quality network diagrams, you can choose the color and pattern used for each node separately by specifying a different pattern number for the PATTERN variable, identified in the ACTNET statement (for details, see the section “Graphics Version” on page 704). For line-printer or full-screen network diagrams, the FORMCHAR= option enables you to specify special boxing characters that enhance the display; for full-screen network diagrams, you can also choose the color of the nodes using the PATTERN= option.

By default, all arcs are drawn along the center track between two consecutive nodes. The SEPARATEARCS option, which is available in the graphics version, separates arcs in the same corridor by drawing them along separate tracks, thus preventing them from being drawn on top of each other.

If the network fits on one page, it is centered on the page; in the graphics mode, you can use the NOVCENTER option to prevent centering in the vertical direction so that the network is drawn immediately below the title. If the network cannot fit on one page, it is split onto different pages appropriately. See the section “Page Format” on page 695 for a description of how the pages are split.

---

Figure 9.8  Page Layout

As explained in the section “Format of the Display” on page 694, if the network fits on one page, it is centered on the page (unless the NOVCENTER option is specified); otherwise, it is split onto different pages appropriately, and each page is drawn starting at the bottom left corner. If the network is drawn on multiple pages, the procedure numbers each page of the diagram on the top right corner of the page. The pages are numbered starting with the bottom left corner of the entire picture. Thus, if the network diagram is broken...
into three horizontal and three vertical levels and you want to paste all the pieces together to form one picture, they should be arranged as shown in Figure 9.8.

The number of pages of graphical output produced by the NETDRAW procedure depends on several options such as the NXNODES=, NYNODES=, HPAGES=, VPAGES=, COMPRESS, PCOMPRESS, HEIGHT=, and the ID= options. The value of the HTEXT= option and the number of variables specified in the ID= options determines the size of each node in the network diagram, which in turn affects the number of horizontal and vertical pages needed to draw the entire network. The number of pages is also affected by the global specification of the HPOS=, VPOS=, HSIZE=, and VSIZE= graphics options.

The COMPRESS and PCOMPRESS options force the entire network diagram to be drawn on a single page. You can explicitly control the number of horizontal and vertical pages using the HPAGES= and VPAGES= options. The NXNODES= and NYNODES= options enable you to specify the number of nodes in the horizontal and vertical directions, respectively, on each page of the network diagram.

For examples of these options and how they affect the network diagram output, see Example 9.5.

### Layout Data Set

The Layout data set produced by PROC NETDRAW contains all the information needed to redraw the network diagram for the given network data. In other words, the Layout data set contains the precedence information, the ID variables that are used in the current invocation of the procedure, and variables that contain the coordinate information for all the nodes and arcs in the network.

The precedence information used by the procedure is defined by two new variables named _FROM_ and _TO_, which replicate the ACTIVITY and SUCCESSOR variables from the Network data set. Note that the Layout data set has only one _TO_ variable even if the Network data set has multiple SUCCESSOR variables; if a given observation in the Network data set defines multiple successors for a given activity, the Layout data set defines a new observation for each of the successors. In fact, for each (node, successor) pair, a sequence of observations, defining the turning points of the arc, is saved in the Layout data set; the number of observations corresponding to each pair is equal to one plus the number of turns in the arc connecting the node to its successor. Suppose that a node ‘C’ has two successors, ‘D’ and ‘E,’ and the arcs connecting ‘C’ and ‘D’ and ‘C’ and ‘E’ are routed as shown in Figure 9.7. Then, Table 9.4 illustrates the format of the observations corresponding to the two (_FROM_, _TO_) pairs of nodes, (‘C’, ‘D’) and (‘C’, ‘E’).

<table>
<thead>
<tr>
<th><em>FROM</em></th>
<th><em>TO</em></th>
<th><em>X</em></th>
<th><em>Y</em></th>
<th><em>SEQ</em></th>
<th><em>PATTERN</em></th>
<th>ID variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>D</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>3.5</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>3.5</td>
<td>2.5</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>5.5</td>
<td>2.5</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>5.5</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9.4 Sample Observations in the Layout Data Set
For every (node, successor) pair, the first observation (\_SEQ\_ = ‘0’) gives the coordinates of the predecessor node; the succeeding observations contain the coordinates of the turning points of the arc connecting the predecessor node to the successor. The data set also contains a variable called \_PATTERN\_, which contains the pattern number that is used for coloring the node identified by the \_FROM\_ variable. The value of this variable is missing for observations with \_SEQ\_ > 0.

Controlling the Layout

As explained in the section “Layout of the Network” on page 692, the procedure uses the precedence constraints between the activities to draw a reasonable diagram of the network. A very desirable feature in any procedure of this nature is the ability to change the default layout. PROC NETDRAW provides two ways of modifying the network diagram:

- using the full-screen interface
- using the Network data set

The full-screen method is useful for manipulating the layout of small networks, especially networks that fit on a handful of screens. You can use the full-screen mode to examine the default layout of the network and move the nodes to desired locations using the MOVE command from the command line or by using the appropriate function key. When a node is moved, the procedure reroutes all the arcs that connect to or from the node; other arcs are unchanged. For details about the MOVE command, see the section “Full-Screen Version” on page 701.

You can use the Network data set to modify or specify completely the layout of the network. This method is useful if you want to draw the network using information about the network layout that has been saved from an earlier invocation of the procedure. Sometimes you may want to specify only the positions of the node and let the procedure determine the routing of the arcs. The procedure looks for three default variables in the data set: \_X\_, \_Y\_, and \_SEQ\_. The \_X\_ and \_Y\_ variables are assumed to denote the x and y coordinates of the nodes and all the turning points of the arcs connecting the nodes. The variable \_SEQ\_ is assumed to denote the order of the turning points. This interpretation is consistent with the values assigned to the \_X\_, \_Y\_, and \_SEQ\_ variables in the Layout data set produced by PROC NETDRAW. If there is no variable called \_SEQ\_ in the data set, the procedure assumes that only the node positions are specified and uses the specified coordinates to place the nodes and determines the routing of the arcs corresponding to these positions. If there is a variable called \_SEQ\_, the procedure requires that the turning points for each arc be specified in the proper order, with the variable \_SEQ\_ containing numbers sequentially starting with 1 and continuing onward. The procedure then draws the arcs exactly as specified, without checking for consistency or interpolating or extrapolating turning points that may be missing.

The ALIGN= variable provides another means of controlling the node layout (see the section “Time-Scaled Network Diagrams” on page 698). This variable can be used to specify the x coordinates for the different nodes of the network; the procedure then determines the y coordinates. Note that time-scaled network diagrams (without an ALIGN= specification) are equivalent to network diagrams drawn with the ALIGN= variable being set to the E_START variable.

You can also control the placement of the nodes using the ZONE=option (see the section “Zoned Network Diagrams” on page 700). The procedure uses the values of the ZONE variable to divide the network into...
horizontal zones. Thus, you can control the horizontal placement of the nodes using the ALIGN= option and the vertical placement of the nodes using the ZONE= option.

For networks that have a tree structure, the TREE option draws the network as a tree, thus providing another layout option (see the section “Organizational Charts or Tree Diagrams” on page 700). The procedure draws the tree from left to right, with the root at the left edge of the diagram. Thus, the children of each node are drawn to the right of the node. In the graphics mode of invocation, you can use the ROTATETEXT option in conjunction with the ROTATE option in the ACTNET statement (or the global graphics option ROTATE) to obtain a top-down tree diagram.

Time-Scaled Network Diagrams

By default, PROC NETDRAW uses the topological ordering of the activity network to determine the x coordinates of the nodes. As a project progresses, you may want to display the activities arranged according to their time of occurrence. Using the TIMESCALE option, you can draw the network with a time axis at the top and the nodes aligned according to their early start times, by default. You can use the ALIGN= option to specify any of the other start or finish times in the Network data set. In fact, PROC NETDRAW enables you to align the nodes according to any numeric variable in the data set.

If the TIMESCALE option is specified without any ALIGN= specification, the procedure chooses one of the following variables as the ALIGN= variable: E_START, L_START, S_START, or A_START, in that order. The first of these variables that is found is used to align the nodes. The minimum and maximum values of the ALIGN= variable are used to determine the time axis. The format of this variable is used to determine the default value for the MININTERVAL= option. The value of the MININTERVAL= option (or the default value) is used to determine the format of the time axis. You can override the format based on mininterval by specifying the desired format for the ALIGN= variable (using the FORMAT statement to indicate a standard SAS format or a special user-defined format) and the USEFORMAT option in the ACTNET statement. Table 9.5 lists the valid values of mininterval corresponding to the type of the ALIGN= variable and the default format corresponding to each value of mininterval. For each value in the first column, the first value of mininterval listed is the default value of the MININTERVAL= option corresponding to that type of the ALIGN= variable.

Several options are available in PROC NETDRAW to control the spacing of the nodes and the scaling of a time-scaled network diagram:

- The MININTERVAL= option enables you to scale the network diagram: one tick mark is associated with one unit of mininterval. Thus, if mininterval is DAY, each column is used to represent one day and all activities that start on the same day are placed in the same column. By default, the procedure omits any column (tick mark) that does not contain any node.

- The LINEAR option enables you to print a tick mark corresponding to every day (or the unit of mininterval). Note that, for a project that has few activities spread over a large period of time, the LINEAR option can lead to a network diagram that is very wide.

- The MAXNULLCOLUMN= option specifies the maximum number of empty columns that is allowed between two consecutive nonempty columns. The LINEAR option is equivalent to specifying maxncol = infinity, while the default time-scaled network diagram is drawn with maxncol = 0.
The NLEVELSPERCOLUMN= option enables you to contract the network diagram by combining a few columns. For example, if mininterval is DAY and nlevelspercol is 7, each column contains activities that start within seven days of each other; note that the same effect can be achieved by setting mininterval to be WEEK.

Table 9.5 MININTERVAL Values and Axis Format

<table>
<thead>
<tr>
<th>ALIGN Variable Type</th>
<th>MININTERVAL</th>
<th>Axis Label Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>MININTERVAL</td>
<td>numeric format</td>
</tr>
<tr>
<td>SAS time</td>
<td>HOUR</td>
<td>HHMM5.</td>
</tr>
<tr>
<td></td>
<td>MINUTE</td>
<td>HHMM5.</td>
</tr>
<tr>
<td></td>
<td>SECOND</td>
<td>TIME8.</td>
</tr>
<tr>
<td>SAS date</td>
<td>DAY</td>
<td>DATE7.</td>
</tr>
<tr>
<td></td>
<td>WEEKDAY</td>
<td>DATE7.</td>
</tr>
<tr>
<td></td>
<td>WEEK</td>
<td>DATE7.</td>
</tr>
<tr>
<td></td>
<td>MONTH</td>
<td>MONYY5.</td>
</tr>
<tr>
<td></td>
<td>QTR</td>
<td>MONYY5.</td>
</tr>
<tr>
<td></td>
<td>YEAR</td>
<td>MONYY5.</td>
</tr>
<tr>
<td>SAS datetime</td>
<td>DTDAY</td>
<td>DATE7.</td>
</tr>
<tr>
<td></td>
<td>WORKDAY</td>
<td>DATE7.</td>
</tr>
<tr>
<td></td>
<td>DTWRKDAY</td>
<td>DATE7.</td>
</tr>
<tr>
<td></td>
<td>DTSECOND</td>
<td>DATETIME16.</td>
</tr>
<tr>
<td></td>
<td>DTMINUTE</td>
<td>DATETIME16.</td>
</tr>
<tr>
<td></td>
<td>DTHOUR</td>
<td>DATETIME13.</td>
</tr>
<tr>
<td></td>
<td>DTWEEK</td>
<td>DATE7.</td>
</tr>
<tr>
<td></td>
<td>DTMONTH</td>
<td>MONYY5.</td>
</tr>
<tr>
<td></td>
<td>DTQTR</td>
<td>MONYY5.</td>
</tr>
<tr>
<td></td>
<td>DTYEAR</td>
<td>MONYY5.</td>
</tr>
</tbody>
</table>

The node-placement algorithm described in the section “Layout of the Network” on page 692 is modified slightly for time-scaled network diagrams. The x coordinate of each node is determined by the value of the ALIGN= variable. The scaling options just described are used to determine the tick mark corresponding to the node. The y coordinate is determined as before. Once the node placement is completed, the arc routing algorithm is the same as described earlier.

NOTE: Since the node placement for time-scaled networks is determined by the ALIGN= variable, it is possible that some of the arcs between the nodes may have to be routed from right to left instead of from left to right; in other words, there may be some backward arcs. Note also that, if the ALIGN= variable is used to determine the x coordinates of the nodes, the procedure can also draw networks that contain cycles (see the second part of Example 9.12).

Several other options are available to control the appearance of time-scaled network diagrams: AUTOREF, BRKCHAR=, CAXIS=, CREF=, CREFBRK=, FRAME, LREF=, LREFBRK=, NOREPEATAXIS, NOTIMEAXIS, REFBREAK, REFCHAR=, and SHOWBREAK. These options are described in the section “Syntax: NETDRAW Procedure” on page 672.
Zoned Network Diagrams

Most projects have at least one natural classification of the different activities in the project: department, type of work involved, location of the activity, and so on. The ZONE= option enables you to divide the network diagram into horizontal bands or zones corresponding to this classification. The procedure uses the following rules to place the nodes in a zoned network diagram:

- The values of the ZONE variable are used to define as many zones as there are distinct values of this variable.
- Each node of the network is drawn within its corresponding zone.
- The number of rows within each zone is determined by the maximum number of nodes in any given column that correspond to that zone.
- The values of the ZONE variable do not need to be sorted in any particular order, nor do they need to be grouped by distinct values.
- The zones are ordered according to the order of appearance of the different values of the ZONE variable in the Network data set. This enables you to choose any order for the zone values.
- For arcs that connect two nodes within the same zone, the arc lies entirely within the zone; in other words, all the turning points of the arc have $y$ coordinates that are between the minimum and maximum $y$ coordinates for the zone.
- Each zone is labeled by the value of the ZONE variable unless the NOZONELABEL option is specified.
- Each zone is separated from the next by a horizontal line drawn across the width of the network unless the NOZONELABEL option is specified.
- In the graphics and full-screen modes of invocation of the procedure, you can use the ZONEPATTERN option to color the nodes in each zone differently using different pattern statements. In the graphics mode, the first zone uses the first PATTERN statement, the second zone uses the second PATTERN statement, and so on; in full-screen mode, the colors available for the device are repeated in cyclic order. Note that the values of the PATTERN variable (or the default _PATTERN variable, if it exists in the Network data set) override the node patterns dictated by the ZONEPATTERN option.

Organizational Charts or Tree Diagrams

The NETDRAW procedure automatically draws any acyclic network; it does not have to be a representation of a project. You can also use the procedure to draw a general directed graph that has cycles, if node location is specified or if the BREAKCYCLE option is specified. The procedure attempts to draw the network in a compact fashion, which may not always produce the expected result. Trees form one such class of directed graphs that have an inherent natural layout that may not be produced by the default layout of PROC NETDRAW. The TREE option in the ACTNET statement exploits the tree structure of the network by laying the nodes out in the form of a tree.
A directed graph is said to be a tree if it has a root and there is a unique directed path from the root to every node in the tree. An equivalent characterization of a tree is that the root node has no predecessors and every other node has exactly one predecessor (Even 1979). Typical examples of trees that arise in project management are organizational charts or work breakdown structures. If the TREE option is specified, the NETDRAW procedure checks if the network has a tree structure and draws the network with the root at the left edge of the diagram and the children of each node appearing to the right of the node. In other words, the tree is drawn from left to right.

The NETDRAW procedure enables you to specify multiple trees in the same Network data set; each tree is drawn separately in the same diagram with all the roots appearing at the left edge of the diagram. Thus, you can use the TREE option as long as every node in the network has at most one predecessor. If you specify the TREE option and some node has multiple predecessors, the TREE option is ignored and the procedure uses the default node-layout algorithm.

There are several features that control the appearance of the tree:

- The children of each node are placed in the order of occurrence in the Network data set. The \((x, y)\) coordinates of each node are required to be integers. The procedure attempts to place each node at the center of all its children, subject to the requirement that the coordinates must be integers. This requirement may cause some of the nodes to be positioned slightly off-center. See Example 9.15.

- The SEPARATESONS option separates the children of a node, if necessary, to enable the parent node to be exactly centered with respect to its children. See the second part of Example 9.15.

- The CENTERSUBTREE option can be used to center each node with respect to the entire subtree originating from the node instead of centering it with respect to its children.

- In graphics mode, you can change the orientation of the network to be from top to bottom instead of from left to right. To do so, use the ROTATETEXT option in the ACTNET statement to rotate the text within the nodes and the ROTATE option in the ACTNET statement (or the ROTATE global graphics option) to rotate the entire diagram by 90 degrees. See Example 9.18 for an illustration of this feature.

**Full-Screen Version**

You can invoke PROC NETDRAW in full-screen mode by specifying FS (or FULLSCREEN) in the PROC NETDRAW statement. The statement specifications are the same as for the line-printer mode. The full-screen mode offers you a convenient way to browse the network diagram of the project and change the layout of the network by moving the nodes of the network to desired locations. However, you cannot move a node to any position that violates the precedence constraints that must be satisfied by the node. In other words, you cannot move a node to the left of any of its predecessors or to the right of any of its successors. For time-scaled network diagrams, you cannot move a node out of the column corresponding to the value of the ALIGN= variable. For zoned network diagrams you cannot move a node out of its zone.

The format control options are treated in the same way as for the line-printer version, with some minor changes. It is assumed that the main purpose of invoking the procedure is to gain a general picture of the layout of the entire network and to modify it to some extent. In an effort to display as much of the network as possible, the initial display on the screen is drawn with only one row and three columns for each node. In other words, the BOXHT=, BOXWIDTH=, XBETWEEN=, and YBETWEEN= options are ignored by the
procedure in drawing the initial display. However, the full-screen commands supported by PROC NETDRAW enable you to change the scale of the diagram. You can display as much or as little information within each node by invoking the SCALE ROW or the SCALE COL command or both. The SCALE MAX command causes the procedure to display the diagram using the values specified in the ACTNET statement or the dimensions that would be required to display all the ID information, whichever is larger. The SCALE RESET command returns the scaling to the initial values used for display.

The nodes of the network are color coded on the basis of the PATTERN variable. If there is no PATTERN variable, then the nodes are color coded depending on whether the activities are normal, critical, or supercritical. The nodes are drawn in reverse video. By default, the nodes are drawn without an outline; however, there is an OUTLINE command that lets you toggle back and forth between an outlined or non-outlined node. Using an outline for the node is useful if you want to obtain a printout of the screen display using SPRINT; it helps mark the boundary of each node clearly.

**Commands**

Table 9.6 lists the commands that can be invoked from the command line in the full-screen version of PROC NETDRAW. These commands are explained in greater detail in this section.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKWARD</td>
<td>Scrolls toward the top of the network by the VSCROLL amount. BACKWARD MAX scrolls to the top of the network. You can specify the vertical scroll amount for the current command as BACKWARD PAGE</td>
</tr>
<tr>
<td>BOTTOM</td>
<td>Scrolls to the bottom of the network.</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Ends the current invocation of the procedure.</td>
</tr>
<tr>
<td>CLEAR</td>
<td>Clears any outstanding move commands.</td>
</tr>
</tbody>
</table>
GEND
ends the current invocation of the procedure after drawing the network in graphics mode with the compress option.

END
ends the current invocation of the procedure.

FORWARD
scrolls toward the bottom of the network by the VSCROLL amount. FORWARD MAX scrolls to the bottom of the network. You can also specify the vertical scroll amount for the current command as FORWARD PAGE | HALF | n.

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

HOME
moves the cursor to the command line.

HSCROLL
sets the amount that information 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 horizontal levels. HSCROLL PAGE sets the scroll amount to be the number of horizontal levels that fit on one screen; HSCROLL HALF is half that amount; HSCROLL n sets the horizontal scroll amount to n levels.

KEYS
displays current function key settings for the NETDRAW procedure.

LEFT
scrolls toward the left boundary of the network by the HSCROLL amount. LEFT MAX scrolls to the left boundary. You can specify the horizontal scroll amount for the current command as LEFT PAGE | HALF | n.

MOVE
specifies a node to be moved or a place to move a node to. You can specify these in any order. Thus, you can first position the cursor on the node that you want to move, issue the MOVE command, and then position the cursor at a target position and issue the MOVE command again. If the target position is valid, the node is moved. You can also first specify the target position and then indicate the node that is to be moved.

NOTE: For a standard network, a node cannot be moved to any position that violates the topological ordering of the nodes in the network. For time-scaled network diagrams, you cannot move a node to a level corresponding to a different tick mark. For zoned network diagrams, you cannot move a node out of its zone.

OUTLINE
causes an outline to be drawn around each node in the network. This is useful if you want to print a copy of the screen by using the SPRINT command. The OUTLINE command works like an on/off switch: you can turn it off by entering the command again.
RIGHT
scrolls toward the right boundary of the network by the HSCROLL amount. RIGHT MAX scrolls to the right boundary. You can also specify the horizontal scroll amount for the current command as RIGHT PAGE | HALF | n.

SCALE
controls the scaling of the nodes and the space between nodes. The format of this command is SCALE MAX | MIN | RESET | ROW MAX | COL MAX | ROW MIN | COL MIN | ROW n | COL n | +n | -n. The number n denotes the number of character positions. SCALE MIN displays as many nodes on the screen as can fit. SCALE MAX enables as many rows and columns per node as is required to display all the information that pertains to it. SCALE ROW MAX displays the maximum number of rows per node. SCALE COL MAX displays the maximum number of columns per node. SCALE ROW n sets the number of rows per node to n. SCALE ROW +n increases the number of rows per node by n. SCALE COL n sets the number of columns per node to n. SCALE COL +n increases the number of columns per node by n. SCALE RESET sets the values to be the same as for the initial display. Note that none of these values can be greater than the dimensions of the screen.

TOP
scrolls to the top of the network.

VSCROLL
sets the amount by which information 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 vertical levels. VSCROLL PAGE sets the scroll amount to be the number of vertical levels that fit on one screen; VSCROLL HALF is half that amount; VSCROLL n sets the vertical scroll amount to n levels.

Full-Screen Global Commands

Most of the global commands used in SAS/FSP software are also valid with PROC NETDRAW. Some of the commands used for printing screens are described in the section “Global Commands” on page 537 in Chapter 8, “The GANTT Procedure.”

Graphics Version

Several options are available in the ACTNET statement to enhance the appearance of the network diagram in graphics mode. These are described in the section “Graphics Options” on page 683. The format control options BOXWIDTH=, BOXHT=, XBETWEEN=, and YBETWEEN= are also valid in this mode and can be used to control the width and height of each node and the distance between the nodes. These parameters are specified in terms of number of character cell positions. The number of positions available on one page depends on the graphics device that is used; thus, if a plotter is used with large paper, more of the network will be drawn on a single page. Further, you can control the number of character cell positions on a page by changing the values of the global graphics options (HPOS= and VPOS=). Note that the NETDRAW procedure is not supported with the ActiveX or Java series of devices on the OPTIONS statement.

You can also control the number of nodes on a given page by specifying the NXNODES= and NYNODES= options. The HPAGES= and VPAGES= options control the number of pages in the horizontal and vertical
Using the Annotate Facility

The Annotate facility enables you to enhance graphics output produced by PROC NETDRAW. To use this facility, you must create an Annotate data set, which contains a set of graphics commands that can be superimposed on the network diagram. This data set has a specific format and must contain key variables as described in SAS/GRAPH Software: Reference. The chapter entitled “The Annotate Data Set” lists the variables that are required in this data set and explains the coordinate systems used by the Annotate facility. The present section explains the use of data coordinates specifically with reference to the NETDRAW procedure.

When annotating a graph produced by any of the graphics procedures, it is 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 node of a network diagram with additional text, you can position the text accurately if you use data coordinates instead of screen coordinates. With respect to PROC NETDRAW, the Annotate facility uses the _X_ and _Y_ values in the Layout data set as the basis for the data coordinate system. To use this feature, you can invoke PROC NETDRAW (with the NODISPLAY option, if necessary) for the given network to produce the Layout data set that contains the _X_ and _Y_ coordinates for each node of the network. This data set can then be used to create the required Annotate data set containing the graphics commands positioning the primitives appropriately on the diagram using the data coordinates. See Example 9.16 and Example 9.17 for illustrations of this feature.
**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 network diagram that runs from one page of the diagram to another.

---

**Web-Enabled Network Diagrams**

The **WEB** variable enables you to define a HTML reference for each activity. This HTML reference is associated with the node corresponding 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 NETDRAW statement**. By processing this SAS data set using a **DATA step**, you can generate customized HTML pages for your network diagram.

---

**Macro Variable _ORNETDR**

The **NETDRAW** procedure defines a macro variable named **_ORNETDR**. This variable contains a character string that indicates the status of the procedure. It is set at procedure termination. The form of the **_ORNETDR** character string is **STATUS= REASON=**, where **STATUS=** is either **SUCCESSFUL** or **ERROR_EXIT** and **REASON=** (if **PROC NETDRAW** terminated unsuccessfully) can be one of the following:

- CYCLE
- BADDATA_ERROR
- MEMORY_ERROR
- IO_ERROR
- SEMANTIC_ERROR
- SYNTAX_ERROR
- NETDRAW_BUG
- UNKNOWN_ERROR

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

In addition to providing the “**STATUS= REASON=**” string that indicates the status of the procedure, the macro variable **_ORNETDR** also provides some information about the network diagram produced by the current invocation of **PROC NETDRAW**.

The information given in **_ORNETDR** is described in the following list, along with the keyword that identifies it. These values refer to those actually used in producing the network diagram and are not necessarily the same as those specified in the invocation of the procedure.
Computer Resource Requirements

There is no inherent limit on the size of the network that you can draw with the NETDRAW procedure. Naturally, a sufficient amount of core memory must be available in order to invoke and initialize the SAS system. Furthermore, the amount of memory that is required depends on the mode of invocation of the procedure. The procedure attempts to store all the data in core memory. However, if the problem is too large to fit in core memory, the procedure resorts to using utility data sets and swaps between core memory and utility data sets as necessary.

The storage requirement for the data area that the procedure requires is proportional to the number of nodes and arcs in the network. You can further increase the memory that is required by specifying the DP option in the ACTNET statement. Recall that the DP option requests the use of a dynamic programming algorithm to route the arcs between the nodes, and such algorithms tend to grow exponentially with the size of the problem being solved.

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 NETDRAW Style Template

A predefined ODS style template named NETDRAW is available for the NETDRAW 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 NETDRAW style template to all HTML output:
To disable the use of graphical styles, specify the SAS system option NOGSTYLE.

The parent style template for the NETDRAW style is the DEFAULT style. Table 9.7 lists the style elements (in bold) and corresponding attributes specified in the NETDRAW style. The table also indicates which, if any, PROC NETDRAW options or graphics options (in a GOPTIONS statement) can be used to override the value of a style attribute.

<table>
<thead>
<tr>
<th>Element/Attributes</th>
<th>Description</th>
<th>NETDRAW Option</th>
<th>GOPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GraphColors</strong></td>
<td>Colors of various graph features</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata1</td>
<td>Noncritical nodes or nodes in the first zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata2</td>
<td>Critical nodes or nodes in the second zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata3</td>
<td>Nodes in the third zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata4</td>
<td>Nodes in the fourth zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata5</td>
<td>Nodes in the fifth zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata6</td>
<td>Nodes in the sixth zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata7</td>
<td>Nodes in the seventh zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata8</td>
<td>Nodes in the eighth zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata9</td>
<td>Nodes in the ninth zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata10</td>
<td>Nodes in the tenth zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata11</td>
<td>Nodes in the eleventh zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gdata12</td>
<td>Nodes in the twelfth zone</td>
<td>PATTERN=, ZONEPAT</td>
<td>CPATTERN=, COLORS=</td>
</tr>
<tr>
<td>gaxis</td>
<td>Borderlines</td>
<td>COLORS=</td>
<td></td>
</tr>
<tr>
<td>greferencelines</td>
<td>Horizontal and vertical reference lines</td>
<td>COLORS=</td>
<td></td>
</tr>
<tr>
<td>gtext</td>
<td>Text</td>
<td>CTEXT=</td>
<td></td>
</tr>
<tr>
<td>gtexttt</td>
<td>Title</td>
<td>CTITLE=</td>
<td></td>
</tr>
<tr>
<td>gcdata</td>
<td>Arcs</td>
<td>COLORS=</td>
<td></td>
</tr>
<tr>
<td><strong>GraphFonts</strong></td>
<td>Fonts for various graph features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphDataFont</td>
<td>Default</td>
<td>FTEXT=</td>
<td></td>
</tr>
<tr>
<td>GraphLabelFont</td>
<td>Annotation text</td>
<td>FTEXT=</td>
<td></td>
</tr>
<tr>
<td>GraphTitleFont</td>
<td>Title text</td>
<td>FTITLE=</td>
<td></td>
</tr>
<tr>
<td><strong>GraphAxisLines</strong></td>
<td>Attributes related to graph axes</td>
<td>CAXIS=</td>
<td>COLORS=</td>
</tr>
<tr>
<td>Color</td>
<td>GraphColors(‘gaxis’)</td>
<td>COLORS=</td>
<td></td>
</tr>
<tr>
<td><strong>GraphConnectLine</strong></td>
<td>Attributes related to arcs</td>
<td>CARCS=, CCRITARCS=</td>
<td>COLORS=</td>
</tr>
<tr>
<td>Color</td>
<td>GraphColors(‘gcdatal’)</td>
<td>COLORS=</td>
<td></td>
</tr>
<tr>
<td><strong>GraphReference</strong></td>
<td>Attributes related to horizontal and vertical reference lines</td>
<td>CREF=, CREFBRK=</td>
<td>COLORS=</td>
</tr>
<tr>
<td>Color</td>
<td>GraphColors(‘greferencelines’)</td>
<td>COLORS=</td>
<td></td>
</tr>
<tr>
<td><strong>GraphDataText</strong></td>
<td>Attributes related to general text</td>
<td>CTEXT=</td>
<td>CTEXT=</td>
</tr>
<tr>
<td>Color</td>
<td>GraphColors(‘gtext’)</td>
<td>CTEXT=</td>
<td></td>
</tr>
<tr>
<td>Font</td>
<td>GraphFonts(‘GraphDataFont’)</td>
<td>FTEXT=</td>
<td></td>
</tr>
</tbody>
</table>
Table 9.7 (continued)

<table>
<thead>
<tr>
<th>Element/Attributes</th>
<th>Description</th>
<th>NETDRAW Option</th>
<th>GOPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GraphTitleText</strong></td>
<td>Attributes related to title text</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td>GraphColors(’gtextt’)</td>
<td></td>
<td>CTITLE=</td>
</tr>
<tr>
<td>Font</td>
<td>GraphFonts(’GraphTitleFont’)</td>
<td></td>
<td>FTITLE=</td>
</tr>
<tr>
<td><strong>GraphTitle1Text</strong></td>
<td>Attributes related to the first title text</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td>GraphColors(’gtextt’)</td>
<td></td>
<td>CTITLE=</td>
</tr>
<tr>
<td>Font</td>
<td>GraphFonts(’GraphTitleFont’)</td>
<td></td>
<td>FTITLE=</td>
</tr>
<tr>
<td><strong>GraphLabelText</strong></td>
<td>Attributes related to annotation text</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td>GraphColors(’glabel’)</td>
<td></td>
<td>CTEXT=</td>
</tr>
<tr>
<td>Font</td>
<td>GraphFonts(’GraphLabelFont’)</td>
<td></td>
<td>FTEXT=</td>
</tr>
<tr>
<td><strong>GraphDataDefault</strong></td>
<td>Default values for the attributes specified in Table 9.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td>GraphColors(’gdata’)</td>
<td></td>
<td>COLORS=</td>
</tr>
<tr>
<td><strong>GraphBackground</strong></td>
<td>Attributes related to graph background</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>Background image</td>
<td></td>
<td>CBACK=</td>
</tr>
</tbody>
</table>

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

Figure 9.9 demonstrates features of the NETDRAW graphical style. The diagram in the figure is the first output from Example 9.11.
Default Values

If the SAS system option GSTYLE is in effect (this is the default), then the default values of certain PROC NETDRAW options can depend on the current ODS style template. Table 9.8 lists these PROC NETDRAW options and lists the order in which PROC NETDRAW 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>
<thead>
<tr>
<th>Option</th>
<th>Search Order for Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARCS=</td>
<td>1. GraphConnectLine(Color)</td>
</tr>
<tr>
<td></td>
<td>2. The fourth color in the COLORS= list in the GOPTIONS statement</td>
</tr>
<tr>
<td>CAXIS=</td>
<td>1. GraphAxisLines(Color)</td>
</tr>
<tr>
<td></td>
<td>2. GraphDataDefault(Color)</td>
</tr>
<tr>
<td></td>
<td>3. The first color in the COLORS= list in the GOPTIONS statement</td>
</tr>
<tr>
<td>CREF=</td>
<td>1. GraphReference(Color)</td>
</tr>
<tr>
<td></td>
<td>2. GraphDataDefault(Color)</td>
</tr>
<tr>
<td></td>
<td>3. The first color in the COLORS= list in the GOPTIONS statement</td>
</tr>
<tr>
<td>CREFBRK=</td>
<td>1. GraphReference(Color)</td>
</tr>
<tr>
<td></td>
<td>2. GraphDataDefault(Color)</td>
</tr>
<tr>
<td></td>
<td>3. The first color in the COLORS= list in the GOPTIONS statement</td>
</tr>
<tr>
<td>CTEXT=</td>
<td>1. The value specified for the CTEXT= option in the GOPTIONS statement</td>
</tr>
<tr>
<td></td>
<td>2. GraphDataText(Color)</td>
</tr>
<tr>
<td></td>
<td>3. GraphDataDefault(Color)</td>
</tr>
<tr>
<td></td>
<td>4. The first color in the COLORS= list in the GOPTIONS statement</td>
</tr>
<tr>
<td>FONT=</td>
<td>1. The value specified for the FTEXT= option in the GOPTIONS statement</td>
</tr>
<tr>
<td></td>
<td>2. GraphDataText(Font)</td>
</tr>
<tr>
<td></td>
<td>3. The default hardware font for the graphics output device</td>
</tr>
</tbody>
</table>
Examples: NETDRAW Procedure

This section contains 18 examples that illustrate several features of the NETDRAW procedure. Most of the examples use the data from the Widget Manufacturing Project described in Chapter 4, “The CPM Procedure.” Two tables, Table 9.9 and Table 9.10, at the end of this section list all the examples in this chapter and the options and statements in the NETDRAW procedure that are illustrated by each example.

Example 9.1: Line-Printer Network Diagram

This example uses the data set WIDGET that was used in Example 4.2 in Chapter 4, “The CPM Procedure,” to illustrate the Activity-on-Node representation of the project. The following program invokes PROC NETDRAW twice. First, the activity data set WIDGET is used as input to the procedure. The activity and successor information is identified using the ACTIVITY= and SUCCESSOR= options in the ACTNET statement. The LINEPRINTER option is specified, producing the line-printer network diagram shown in Output 9.1.1.

```sas
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 . .
;
title 'Widget Manufacture';
options ps=32 ls=78;
proc netdraw data=widget lineprinter;
  actnet / activity=task successor=(succ1 succ2 succ3);
run;
```
Output 9.1.1  Line-Printer Network Diagram

**Widget Manufacture**
Next, PROC CPM is invoked to schedule the project, and the resulting Schedule data set is used as input to the NETDRAW procedure. In addition to the ACTIVITY= and SUCCESSOR= options, the DURATION= option is used in the ACTNET statement. The DURATION= option adds the values of the DURATION variable within each node of the network. The procedure also displays the values of the E_START, E_FINISH, L_START, L_FINISH, T_FLOAT, and F_FLOAT variables within each node. The network is displayed in Output 9.1.2.

```plaintext
proc cpm data=widget out=sched
date='1dec03'd;
activity task;
successor succ1 succ2 succ3;
duration days;
run;

options ps=45 ls=90;
title2 'Schedule Information';
proc netdraw data=sched lineprinter;
actnet / activity=task
   successor=(succ1 succ2 succ3)
   duration = days;
run;
```
Output 9.1.2  Project Schedule

Widget Manufacture
Schedule Information

---

<table>
<thead>
<tr>
<th>Drawings</th>
<th>Dur: 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES: 06DEC03</td>
<td>EF: 15DEC03</td>
</tr>
<tr>
<td>LS: 06DEC03</td>
<td>LF: 15DEC03</td>
</tr>
<tr>
<td>TF: 0</td>
<td>FF: 0</td>
</tr>
</tbody>
</table>
---

<table>
<thead>
<tr>
<th>Approve Plan</th>
<th>Dur: 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES: 01DEC03</td>
<td>EF: 05DEC03</td>
</tr>
<tr>
<td>LS: 01DEC03</td>
<td>LF: 05DEC03</td>
</tr>
<tr>
<td>TF: 0</td>
<td>FF: 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Write Specs</th>
<th>Dur: 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES: 06DEC03</td>
<td>EF: 10DEC03</td>
</tr>
<tr>
<td>LS: 11DEC03</td>
<td>LF: 15DEC03</td>
</tr>
<tr>
<td>TF: 5</td>
<td>FF: 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prototype</th>
<th>Dur: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES: 16DEC03</td>
<td>EF: 30DEC03</td>
</tr>
<tr>
<td>LS: 16DEC03</td>
<td>LF: 30DEC03</td>
</tr>
<tr>
<td>TF: 0</td>
<td>FF: 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Study Market</th>
<th>Dur: 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES: 06DEC03</td>
<td>EF: 10DEC03</td>
</tr>
<tr>
<td>LS: 05JAN04</td>
<td>LF: 09JAN04</td>
</tr>
<tr>
<td>TF: 30</td>
<td>FF: 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mkt. Strat.</th>
<th>Dur: 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES: 11DEC03</td>
<td>EF: 20DEC03</td>
</tr>
<tr>
<td>LS: 10JAN04</td>
<td>LF: 19JAN04</td>
</tr>
<tr>
<td>TF: 30</td>
<td>FF: 30</td>
</tr>
<tr>
<td>Materials</td>
<td>Dur:10</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>ES:31DEC03</td>
<td>EF:09JAN04</td>
</tr>
<tr>
<td>LS:31DEC03</td>
<td>LF:09JAN04</td>
</tr>
<tr>
<td>TF: 0</td>
<td>FF: 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Facility</th>
<th>Dur:10</th>
<th>Init. Prod.</th>
<th>Dur:10</th>
<th>Evaluate</th>
<th>Dur:10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES:31DEC03</td>
<td>EF:09JAN04</td>
<td>-&gt; ES:10JAN04</td>
<td>EF:19JAN04</td>
<td>-&gt; ES:20JAN04</td>
<td>EF:29JAN04</td>
</tr>
<tr>
<td>LS:31DEC03</td>
<td>LF:09JAN04</td>
<td>LS:10JAN04</td>
<td>LF:19JAN04</td>
<td>LS:25JAN04</td>
<td>LF:03FEB04</td>
</tr>
<tr>
<td>TF: 0</td>
<td>FF: 0</td>
<td>TF: 0</td>
<td>FF: 0</td>
<td>TF: 5</td>
<td>FF: 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marketing</th>
<th>Dur: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES:20JAN04</td>
<td>EF:20JAN04</td>
</tr>
<tr>
<td>LS:09FEB04</td>
<td>LF:09FEB04</td>
</tr>
<tr>
<td>TF:20</td>
<td>FF:20</td>
</tr>
</tbody>
</table>
Example 9.2: Graphics Version of PROC NETDRAW

The same network used in Example 9.1 is drawn here with the GRAPHICS option (which is also the default mode of output for the NETDRAW procedure). In this example, the network is drawn before scheduling the project with PROC CPM. The global options HPOS= and VPOS= are set to 100 and 70, respectively. These options control the number of character cell positions on the screen. The network is displayed in Output 9.2.1. Note that all the nodes are the same color as specified by the PATTERN statement, the color of the arcs is blue as specified by the CARCS= option, the width of all lines is 3 as specified by the LWIDTH= option.
Example 9.3: Spanning Multiple Pages

In this example, the Schedule data set produced by PROC CPM is displayed in a graphics network. As in the second part of Example 9.1, the procedure displays the duration as well as the early and late start and finish times and the total float and free float of each activity in the node corresponding to that activity. The network cannot fit on one page and is drawn across three pages, as shown in Output 9.3.1.

This example also illustrates several options for controlling the appearance of the network diagram. The pattern statements set a background fill color for all the nodes of the network (PATTERN1 and PATTERN2). The COUTLINE= and CCRITOUT= options set the outline colors for noncritical and critical activities, respectively. Recall that the procedure uses the values of the E_FINISH and L_FINISH variables to determine
if an activity is critical. The CARCS= and CCRITARCS= options color the regular arcs blue and the critical arcs (arcs connecting critical activities) red, respectively and the CTEXT= options sets the color of the text to blue. Finally, the LWIDTH= option sets the default width for all the lines in the network, and the LWCRIT= option further highlights the critical arcs by drawing them with thicker lines.

In this invocation of PROC NETDRAW, the SEPARATEARCS option is used so that the two parallel arcs leading into the activity ‘Test Market’ (one from ‘Mkt.Strat.’ and the other from ‘Init. Prod.’) are drawn along separate tracks instead of along a single track as in Example 9.2.

    /* 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 . .
    ;

    goptions hpos=80 vpos=50 border;
    pattern1 c=ltgray v=s;
    pattern2 c=ltgray v=s;
    title c=blue j=l h=1.5 ' Project: Widget Manufacture';
    title2 c=blue j=l h=1.5 ' Schedule Information';
    footnote c=blue j=r h=1.5 'Spanning Multiple Pages ';
    proc netdraw data=sched graphics;
      actnet / act=task
        succ=(succ1 succ2 succ3)
        dur = days
        coutline=blue
        ccritout=red
        carcs=blue
        ccritarcs=red
        ctext=blue
        lwidth=1
        lwcrit=2
        separatearcs;
    run;
Output 9.3.1 Project Schedule

Project: Widget Manufacture
Schedule Information
Project: Widget Manufacture
Schedule Information

Materials
ES: 31DEC03
LS: 31DEC03
TF: 0
Dur: 10
EF: 09JAN04
LF: 09JAN04
FF: 0

Test Market
ES: 20JAN04
LS: 20JAN04
TF: 0
Dur: 15
EF: 03FEB04
LF: 03FEB04
FF: 0

Facility
ES: 31DEC03
LS: 31DEC03
TF: 0
Dur: 10
EF: 09JAN04
LF: 09JAN04
FF: 0

Init. Prod.
ES: 10JAN04
LS: 10JAN04
TF: 0
Dur: 10
EF: 19JAN04
LF: 19JAN04
FF: 0

Evaluate
ES: 20JAN04
LS: 25JAN04
TF: 5
Dur: 10
EF: 03FEB04
LF: 03FEB04
FF: 5

Marketing
ES: 20JAN04
LS: 09FEB04
TF: 20
Dur: 0
EF: 20JAN04
LF: 09FEB04
FF: 20

Spanning Multiple Pages
Example 9.4: The COMPRESS and PCOMPRESS Options

In Example 9.2, the number of character cell positions were specified so that the entire network fit on one page; in Example 9.3, the network spanned several pages. To force the network diagram to fit on one page automatically, you can use the COMPRESS or PCOMPRESS options. Both options use window transformations to fit the network on one page: the COMPRESS option treats the horizontal and vertical transformations independently of each other so that one direction may not be as compressed as the other; the PCOMPRESS option uses a proportional transformation so that each direction is compressed as much as the other. The following two invocations of PROC NETDRAW illustrate the differences in the diagrams produced.

In both network diagrams, PATTERN statements are used to color the nodes red or green, depending on whether the activities they represent are critical or not. The first PATTERN statement is for nodes corresponding to noncritical activities, and the second statement is for critical activities. The second invocation of PROC NETDRAW also uses the NOVCENTER option in the ACTNET statement to turn off centering in the vertical direction, so that the network is drawn immediately below the titles.
Project: Widget Manufacture
Schedule Information

Example 9.4: The COMPRESS and PCOMPRESS Options
Project: Widget Manufacture
Schedule Information

PCOMPRESS Option
Example 9.5: Controlling the Display Format

In addition to the COMPRESS and PCOMPRESS options and the HPOS= and VPOS= global options, you can control the amount of information displayed on a single page by

- turning off the default ID variable selection (using the NODEFID option) and using the ID= option to select only a few variables of interest in the Network data set
- setting the dimensions of each node using the BOXWIDTH= and the BOXHT= options
- specifying the horizontal and vertical distances between nodes using the XBETWEEN= and YBETWEEN= options, respectively

This example uses the data from Example 8.1 in Chapter 8, "The GANTT Procedure," and some of the options just mentioned to produce the network diagram shown in Output 9.5.1. The ID= and NODEFID options are used to display only the activity name and duration values within each node. The NOLABEL
option suppresses the display of the variable names within each node. Some of the activity names are truncated by the BOXWIDTH= option. Even with the restrictions imposed, the network is too large to fit on one page and spans two pages. Note that on devices with higher resolution, you can increase the values of HPOS and VPOS (still maintaining readability) to enable more of the network to be drawn on one page.

data ex;
  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 . . .
  finished 0 . . .
;
proc cpm finishbefore date='1jan04'd out=sched;
  activity activity;
  duration dur;
  successors success1-success4;
  run;
proc sort;
  by e_start;
  run;
pattern1 v=e c=green;
pattern2 v=e c=red;

title j=l h=3 ' Site: Multi-Story Building'
j=r ' Date: January 1, 2004';
footnote j=r h=2 'Controlling Display Format ';

proc netdraw data=sched graphics;
  actnet / act = activity
    succ = (success1-success4)
    id = ( activity dur )
    nolabel noundefaultid
    boxwidth = 6
    ybetween = 6
    separatearcs;
run;

Output 9.5.1 Controlling the Display Format
You can also control the format of the display by specifying the number of pages into which the network diagram should be split. You can do this by

- using the HPAGES= and VPAGES= options, which specify the number of pages in the horizontal and vertical directions, respectively
- setting the number of nodes in the horizontal and vertical directions using the NXNODES= and NYNODES= options, respectively

The following statements invoke PROC NETDRAW with some additional page control options. The HTEXT= option is also used to control the height of the text used in the diagram.

```
footnote j=r h=2 'Controlling Number of Pages';

proc netdraw data=sched graphics;
  actnet / act = activity
    succ = (success1-success4)
    id = ( activity dur )
    nolabel nodefaultid
    boxwidth = 6
    ybetween = 6
    separatearcs
    htext=2
    nopagenumber
    hpages=3 vpages=1;
run;
```
Output 9.5.2 Controlling the Number of Pages

Site: Multi-Story Building  Date: January 1, 2004

Controlling Number of Pages
Example 9.6: Nonstandard Precedence Relationships

This example illustrates the use of the LAG= option to indicate nonstandard precedence relationships between activities. Consider the widget manufacturing project described in the earlier examples. Some of the precedence constraints between the activities may be nonstandard or have a nonzero lag value. For example, the activity ‘Init. Prod.’ may not be able to start until 2 days after the completion of the activity ‘Facility.’ The Network data set is displayed in Output 9.6.1. The variable lagdur indicates the type of relationship between the activities specified in the variables task and succ.

The following statements invoke PROC NETDRAW with the LAG= option. The resulting network diagram is shown in Output 9.6.2.
Example 9.6: Nonstandard Precedence Relationships

pattern1 v=e c=green;

title h=3 'Widget Manufacture';
title2 h=2 'Nonstandard Precedence Constraints';

proc netdraw graphics data=widglag;
  actnet / act=task
    succ=succ
    lag=lagdur
    pcompress
    htext=3 boxht=3 arrowhead=2
    xbetween=7 ybetween=9
    centerid
    separatearcs;
run;

Output 9.6.1 Network with Nonstandard Precedence Constraints

<table>
<thead>
<tr>
<th>Obs</th>
<th>task</th>
<th>days</th>
<th>succ</th>
<th>lagdur</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Approve Plan</td>
<td>5</td>
<td>Drawings</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Approve Plan</td>
<td>5</td>
<td>Study Market</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Approve Plan</td>
<td>5</td>
<td>Write Specs</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Drawings</td>
<td>10</td>
<td>Prototype</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Study Market</td>
<td>5</td>
<td>Mkt. Strat.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Write Specs</td>
<td>5</td>
<td>Prototype</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Prototype</td>
<td>15</td>
<td>Materials</td>
<td>ss_9</td>
</tr>
<tr>
<td>8</td>
<td>Prototype</td>
<td>15</td>
<td>Facility</td>
<td>ss_9</td>
</tr>
<tr>
<td>9</td>
<td>Mkt. Strat.</td>
<td>10</td>
<td>Test Market</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Mkt. Strat.</td>
<td>10</td>
<td>Marketing</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Materials</td>
<td>10</td>
<td>Init. Prod.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Facility</td>
<td>10</td>
<td>Init. Prod.</td>
<td>fs_2</td>
</tr>
<tr>
<td>13</td>
<td>Init. Prod.</td>
<td>10</td>
<td>Test Market</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Init. Prod.</td>
<td>10</td>
<td>Marketing</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Init. Prod.</td>
<td>10</td>
<td>Evaluate</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Evaluate</td>
<td>10</td>
<td>Changes</td>
<td>ss_6</td>
</tr>
<tr>
<td>17</td>
<td>Test Market</td>
<td>15</td>
<td>Changes</td>
<td>ff_3</td>
</tr>
<tr>
<td>18</td>
<td>Changes</td>
<td>5</td>
<td>Production</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Production</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Marketing</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 9.7: Controlling the Arc-Routing Algorithm

This example illustrates the use of the DP and HTRACKS= options to control the routing of the arcs connecting the nodes. The project is a simple construction project with the following data. A Schedule data set produced by PROC CPM is input to PROC NETDRAW. The first invocation of the procedure illustrates the default layout of the network. As explained in the section “Layout of the Network” on page 692, the NETDRAW procedure uses a simple heuristic to route the arcs between the nodes. In the resulting diagram displayed in Output 9.7.1, note that the specification of BOXHT=3 limits the number of rows within each node so that the float values are not displayed.
data exmpl;
  input task $ 1-16
    duration
    succesr1 $ 21-35
    succesr2 $ 36-50
    succesr3 $ 51-65;
datalines;
Drill Well        4 Pump House
Pump House        3 Install Pipe
Power Line        3 Install Pipe
Excavate          5 Install Pipe Install Pump Foundation
Deliver Material  2 Assemble Tank
Assemble Tank     4 Erect Tower
Foundation        4 Erect Tower
Install Pump       6
Install Pipe       2
Erect Tower        6
;
proc cpm data=exmpl date='1jan04'd out=sched;
  activity task;
    duration duration;
    successor succesr1 succesr2 succesr3;
  run;

pattern1 v=e c=green;
pattern2 v=e c=red;

title j=l h=3 ' Site: Old Well Road';
title2 j=l h=2 ' Date: January 1, 2004';
footnote j=r h=2 'Default Layout '; proc netdraw data=sched graphics;
  actnet / act = task
    dur = duration
    succ = (succesr1-succesr3)
    boxht = 3 xbetween = 10
    separatearcs
    htext=2
    pcompress;
  run;
Next, a different routing of the arcs is obtained by specifying the DP and the HTRACKS= options. As a result of these options, the NETDRAW procedure uses a dynamic programming algorithm to route the arcs, limiting the number of horizontal tracks used to 1. The resulting network diagram is shown in Output 9.7.2. Notice that at most one arc is drawn in each horizontal track. Recall that, by default, the procedure uses a dynamic programming algorithm for arc routing if the number of tracks is restricted to be less than the maximum number of successors. Thus, for this example, the default routing option will be DP, even if it is not explicitly specified (because HTRACKS = 1 and the maximum number of successors is 3).

```plaintext
footnote j=r h=2 'Controlled Layout '; proc netdraw data=sched graphics;
  actnet / act = task
dur = duration
succ = (succesr1-succesr3)
boxht = 3 xbetween = 10
separatearcs
htracks=1
htext=2
pcompress
dp;
run;
```
Example 9.8: PATTERN and SHOWSTATUS Options

As a project progresses, in addition to the criticality of the activities, you may also want to display the status of the activity: whether it is in progress, has been completed, or is still to be scheduled. The SHOWSTATUS option in the ACTNET statement displays this additional information. In the current example, the same progress data as shown in Example 4.13 in Chapter 4, “The CPM Procedure,” are used to illustrate the SHOWSTATUS option. The following program shows the necessary code. First, PROC CPM schedules the project with the SHOWFLOAT option; this enables activities that are already in progress or completed also to show nonzero float. Following this, a DATA step sets the variable \textit{style} to '3' for activities that are completed or in progress; the remaining activities have missing values for this variable.

PROC NETDRAW is then invoked with the SHOWSTATUS option, which draws two diagonal lines across nodes referring to completed activities and one diagonal line for in-progress activities. The PATTERN= option in the ACTNET statement identifies the variable \textit{style} containing the pattern information. Thus, the third pattern statement is used for in-progress or completed activities; the other activities (which have missing values for the variable \textit{style}) use the second or the first pattern statement according to whether or not they are critical. However, since the first two PATTERN statements have EMPTY fill patterns specified, the nodes...
representing activities that have not yet started are in fact colored on the basis of the COUTLINE= and CCRITOUT= options. The resulting network diagram is shown in Output 9.8.1.

data holidays;
   format holiday holifin date7.;
   input holiday & date7. holifin & date7. holidur;
   datalines;
24dec03 26dec03 4
01jan04 . .
;
* actual schedule at timenow = 19dec03;
data actual;
   format task $12. sdate fdate date7.;
   input task & sdate & date7. fdate & date7. pctc rdur;
   datalines;
Approve Plan 01dec03 05dec03 . .
Drawings 06dec03 16dec03 . .
Study Market 05dec03 . 100 .
Write Specs 07dec03 12dec03 . .
Prototype . . . .
Mkt. Strat. 10dec03 . . 3
Materials . . . .
Facility . . . .
Init. Prod. . . . .
Evaluate . . . .
Test Market . . . .
Changes . . . .
Production . . . .
Marketing . . . .
;
* merge the predicted information with network data;
data widgact;
   merge actual widget;
run;

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

/* Set patterns for activities that have started */
data netin;
   set widgupd;
   if a_start ^= . then style = 3;
run;
goptions hpos=120 vpos=70 border;
pattern1 c=green v=e;
pattern2 c=red v=e;
pattern3 c=ltgray v=s;

title  j=l h=3 ' Project: Widget Manufacture';
title2 j=l h=2 ' Date: December 19, 2003';
footnote1 j=l h=2 ' Activity';
footnote2 j=l h=2 ' Start';
footnote3 j=l h=2 ' Finish'
   j=r h=2 ' PATTERN and SHOWSTATUS Options ';
proc netdraw data=netin graphics;
   actnet / act=task
      succ=(succ1 succ2 succ3)
      ybetween = 10
      separatearcs
      pcompress
      id=(task e_start e_finish)
      nodefid nolabel
      carcs=cyan
      ccritarcs=red
      coutline = green
      ccritout = red
      showstatus
      pattern = style
      htext=2;

run;
Example 9.9: Time-Scaled Network Diagram

This example illustrates the use of the TIMESCALE and ALIGN= options to draw time-scaled network diagrams. The Schedule data set WIDGUPD, produced by PROC CPM in the previous example, is used. First, PROC NETDRAW is invoked with the TIMESCALE option without any ALIGN= specification. By default, the procedure aligns the nodes to coincide with the early start times of the activities. The network spans two pages (HPAGES=2 VPAGES=1), as shown in Output 9.9.1. The HMARGIN= and VMARGIN= options add extra space around the margins.

```
pattern1 v=e c=green;
pattern2 v=e c=red;

title j=l h=3 ' Project: Widget Manufacture';
title2 j=l h=2 ' Date: December 19, 2003';
footnote j=l h=2 ' Task Name / Early Finish Within Node'
   j=r h=2 'Time Scaled: Default Alignment ';
proc netdraw data=widgupd graphics;
   actnet / act=task
      succ=(succ1 succ2 succ3)
      ybetween = 8
```
Example 9.9: Time-Scaled Network Diagram

```
separatearcs
novcenter
id=(task e_finish) nodefid
nolabel
showstatus
carcs=darkblue
ccritarcs=red
vmargin=5
hmargin=5
timescale
htext=2 pcompress
hpages=2 vpages=1
nopagenumber;
run;
```

Output 9.9.1 TIMESCALE Option: Default Alignment

---

**Project: Widget Manufacture**

**Date:** December 19, 2003

1. **Approve Plan** (05DEC03)
2. **Study Market** (09DEC03)
3. **Drawings** (16DEC03)
4. **Write Specs** (12DEC03)
5. **Mkt. Strat.** (21DEC03)
6. **Prototype** (04JAN04)

---

Task Name / Early Finish Within Node  |  Time Scaled: Default Alignment
Output 9.9.1 continued

Project: Widget Manufacture
Date: December 19, 2003

<table>
<thead>
<tr>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>05 JAN 04</td>
</tr>
</tbody>
</table>

Facility 14 JAN 04  
Init. Prod. 24 JAN 04  
Evaluate 03 FEB 04  
Materials 14 JAN 04  
Test Market 06 FEB 04  
Changes 13 FEB 04  
Production 14 FEB 04  
Marketing 25 JAN 04

Next, PROC NETDRAW is invoked with several of the time-scale options:

- The ALIGN= option requests that the activities be placed according to the L_START times.
- The FRAME option produces a border around the network diagram.
- The AUTOREF option draws reference lines at every tick mark.
- The LREF= and CREF= options specify the line style and color for the reference lines.
- The SHOWBREAK option requests that breaks in the time axis be indicated by breaks before the corresponding tick marks.

```
footnote j=l h=2 ' Task Name / Late Finish Within Node'
    j=r h=2 'Time Scaled: Align = Late Start ';
proc netdraw data=widgupd graphics;
    actnet / act=task
        succ=(succ1 succ2 succ3)
        ybetween = 10
        separatearcs
```
Example 9.9: Time-Scaled Network Diagram

Output 9.9.2 Timescale Option: ALIGN= L_START

Project: Widget Manufacture
Date: December 19, 2003

Task Name / Late Finish Within Node

Time Scaled: Align = Late Start
Example 9.10: Further Time-Scale Options

In this example, the construction project described in Example 9.7 is used to illustrate some more time-scale options. First, the REFBREAK option indicates breaks in the time axis by drawing a zigzag line down the diagram before each tick mark corresponding to a break. The CREFBRK= and LREFBRK= options control the color and line style for these lines. The network diagram is shown in Output 9.10.1.

```latex
\begin{verbatim}
proc netdraw data=sched graphics;
  actnet / act = task
    dur = duration
    succ = (sucessr1-succesr3)
    dp compress separatearcs
    font=swiss htext=2
    timescale refbreak lrefbrk = 33
    carcs=cyan crefbrk = blue;
  run;
\end{verbatim}
```

Next, PROC NETDRAW is invoked with the LINEAR option so that there is no break in the time axis. The BOXWIDTH= option limits the size of each node. The diagram is drawn in Output 9.10.2.

```latex
\begin{verbatim}
proc netdraw data=sched graphics;
  actnet / act = task
    dur = duration
    succ = (sucessr1-succesr3)
    dp
    pcompress
    novcenter
    vmargin = 10
    separatearcs
    htext=2
    carcs=cyan
    id=(task)
    nodefid
    nolabel
    boxwidth=7
    timescale
    linear
    frame;
  run;
\end{verbatim}
```
Output 9.10.1 Time-Scaled Network: Reference Breaks
Output 9.10.2  Time-Scaled Network: LINEAR Option
Example 9.11: Zoned Network Diagram

This example illustrates zoned network diagrams. The Widget Manufacturing project is used to illustrate some aspects of this feature. The data set DETAILS contains a variable phase, which identifies the phase of each activity in the project. This data set is merged with the Activity data set from Example 9.1, WIDGET, to produce the data set NETWORK that is input to PROC NETDRAW. The ZONE= option divides the network diagram into horizontal zones based on the project phase. The ZONEPAT option causes the activities in each zone to be drawn using a different pattern. The resulting network diagram is shown in Output 9.11.1.

data details;
  format task $12. phase $13. descrpt $30.;
  input task & phase $ descrpt & ;
  datalines;
  Approve Plan    Planning    Develop Concept
  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     Test Product in Sample Market
  Changes        Engineering Engineering Changes
  Production     Manufacturing Begin Full Scale Production
  Marketing      Marketing   Begin Full Scale Marketing
;
  data network;
    merge widget details;
    run;

  pattern1 v=e c=green;
  pattern2 v=e c=red;
  pattern3 v=e c=magenta;
  pattern4 v=e c=blue;
  pattern5 v=e c=cyan;

  title j=l h=1.5 'Project: Widget Manufacture';
  title2 j=l h=1.5 'Date: December 1, 2003';
  footnote j=r h=1.5 'Zoned Network Diagram';

  proc netdraw data=network graphics;
    actnet / act=task succ=(succ1 succ2 succ3)
      separatearcs
      zone=phase
      zonepat
      pcompress
      htext=2;
    label phase = 'Department';
    run;
Next, the project is scheduled with PROC CPM, and PROC NETDRAW is invoked with the ZONE= and TIMESCALE options. The nodes are placed in different zones as dictated by the ZONE variable, phase, and are aligned along the time axis as dictated by the default ALIGN variable, E\_START. The MININTERVAL= option produces one tick mark per week for the duration of the project. The LREF= option identifies the linestyle of the reference lines and the dividing lines between zones. The nodes are colored red or green according to whether or not the corresponding activities are critical (PATTERN statements 1 and 2 from the previous invocation of PROC NETDRAW are still valid).

```
proc cpm data=network interval=weekday
  out=sched date='1dec03'd;
  activity task;
  succ   succ1 succ2 succ3;
  duration days;
  id phase;
run;
```

```
title j=l h=1.5 ' Project: Widget Manufacture';
title2 j=l h=1.5 ' Date: December 1, 2003';
footnote j=r h=1.5 'Zone and Timescale ';
proc netdraw data=sched graphics;
  actnet / act=task succ=(succ1 succ2 succ3)
    pcompress
    carcs = blue ccritarcs=red
    cref = cyan
    caxis = magenta
    lref = 33
    id = (task)
    nodefid
    nolabel
    boxwidth = 8
    htext=2
    separatearcs
    timescale
    mininterval=week
    autoref
    linear
    zone=phase
    zonespace;
  label phase = 'Department';
run;
```

**Example 9.12: Schematic Diagrams**

You can use PROC NETDRAW to determine node placement and arc routing for any network depicting a set of nodes connected by arcs. If you want the procedure to determine the node placement, the network must be acyclic. This example illustrates the use of PROC NETDRAW to draw two networks that represent different schematic flows. The first network does not contain any cycles, while the second one has one cycle; to draw the second network, you need to use the BREAKCYCLE option.

First, a schematic representation of the data flow going in and out of the three procedures (CPM, GANTT, and NETDRAW) is drawn using PROC NETDRAW. (See Chapter 3, “Introduction to Project Management,”
Output 9.11.1  Zoned Network Diagram
Output 9.11.2  Zoned Network Diagram with Time Axis
for a detailed discussion of such a data flow.) The PATTERN= option is used to specify the variable in the
data set that identifies the color that is to be used for each node. Nodes representing SAS/OR procedures are
colored red, the ones representing output data sets are colored green, and all other nodes (representing the
use of other parts of the SAS System) are colored blue. Three ID variables are used to specify the text that is
to be placed within each node. The flow diagram is shown in Output 9.12.1.

data dataflow;
  format id1 $18. id2 $14. id3 $19. ;
  input a $ b $ id1 & id2 & id3 & style;
  datalines;
  A B Data Definition: PROC FSEDIT, SAS/AF, etc. 2
  B C Data Manipulation: Sort, Merge, Concatenate, etc. 2
  B D Data Manipulation: Sort, Merge, Concatenate, etc. 2
  D C . PROC NETDRAW . 1
  C E PROC CPM . PROC PM 1
  C F PROC CPM . PROC PM 1
  E H Resource Usage . Schedule Data 3
  F G . . Schedule Data . 3
  G I Data Manipulation: Sort, Merge, Subset, etc. 2
  G J Data Manipulation: Sort, Merge, Subset, etc. 2
  H K Data Manipulation: Sort, Merge, Subset, etc. 2
  I . Other Reporting PROC's: PRINT, CALENDAR, etc. 2
  J . PROC GANTT . PROC NETDRAW 1
  K . Reporting PROC's: PLOT, CHART, GPLOT, GCHART, etc. 2
;
  pattern1 v=s c=red;
  pattern2 v=s c=blue;
  pattern3 v=s c=green;
  goptions hpos=110 vpos=70;
  title h=3 'A Typical Project Management System';
  title2 h=2.5 'Schematic Representation of Data Flow';
  proc netdraw data=dataflow graphics;
    actnet / act=a succ=b id = (id1-id3)
      nodefaultid
      nolabel
      pattern=style
      carcs=black coutline=black ctext=white
      hmargin = 2
      ybetween = 15
      rectilinear
      noarrowfill
      pcompress htext=2;
  run;

Next, a typical sequence of procedures followed at the scheduling of a nuclear power plant
outage is shown using the NETDRAW procedure. Such a schematic diagram is illustrated in
Chapter 3, “Introduction to Project Management.” In Figure 3.6, there is a cycle that is not normally
allowed in a Network data set that is input to PROC NETDRAW. However, you can draw such networks by
specifying the BREAKCYCLE option. (Note that you can also draw cyclic networks by specifying explicitly
the node coordinates or an ALIGN= variable that fixes the x coordinates for each node.)
Output 9.12.1  Schematic Representation of Data Flow
Output 9.12.2  Scheduling a Power Outage
In this example, the data set OUTAGE contains the network representation. The variable style is used to color nodes appropriately. The resulting diagram is shown in Output 9.12.2.

```plaintext
data outage;
  input a $ b $ id1 $20. id2 $20. style;
datalines;
A  B  Project Definition 1
B  C  CPM Schedule 2
C  D  Gantt Chart Network 3
D  E  Start Power Outage 4
E  F  Project Update 1
F  G  Schedule Update 2
G  E  Gantt Chart Network 3;
;
goptions hpos=110 vpos=70;
title h=3 'Scheduling an Outage';
title2 h=2.5 'Project Cycle';
pattern1 v=s c=green;
pattern2 v=s c=blue;
pattern3 v=s c=blue;
pattern4 v=s c=red;
proc netdraw data=outage graphics;
  actnet / act=a succ=b id = (id1 id2)
    breakcycle
    nodefaultid centerid
    vmargin = 5 hmargin = 0
    nolabel novcenter
    pattern=style
    carcs=black coutline=black ctext=white
    ybetween = 15 xbetween=3
    noarrowfill
    pcompress htext=2;
run;
```

**Example 9.13: Modifying Network Layout**

This example uses the SURVEY project described in Chapter 3, “Introduction to Project Management,” to illustrate how you can modify the default layout of the network. The data set SURVEY contains the project information. PROC NETDRAW is invoked with the GRAPHICS option. The network diagram is shown in Output 9.13.1.
data survey;
  format id $20. activity succ1-succ3 $8. phase $9. ;
  input id &
    activity &
    duration
    succ1 &
    succ2 &
    succ3 &
    phase $ ;
  datalines;
Plan Survey     plan sur 4 hire per design q .     Plan
Hire Personnel  hire per 5 trn per .       Prepare
Design Questionnaire design q 3 trn per select h print q Plan
Train Personnel trn per 3 cond sur .       Prepare
Select Households select h 3 cond sur .     Prepare
Print Questionnaire print q 4 cond sur .     Prepare
Conduct Survey  cond sur 10 analyze .      Implement
Analyze Results analyze 6 .               Implement
;

pattern1 v=s c=green;

title j=l h=3' Project: Market Survey';
title2 j=l h=2 ' Changing Node Positions';
footnote j=r h=2 'Default Layout ';

proc netdraw data=survey graphics out=network;
  actnet / act=activity
    succ=(succ1-succ3)
    id=(id) nodefid nolabel
    carcs = blue
    ctext = white
    coutline=red
    centerid
    boxht = 3
    htext=2
    pcompress
    separatearcs
    ybetween=8;
  run;

footnote;
title2 'NETWORK Output Data Set';
proc print data=network;
  run;
The Layout data set produced by PROC NETDRAW (displayed in Output 9.13.2) contains the \( x \) and \( y \) coordinates for all the nodes in the network and for all the turning points of the arcs connecting them.

Suppose that you want to interchange the positions of the nodes corresponding to the two activities, ‘Select Households’ and ‘Train Personnel.’ As explained in the section “Controlling the Layout” on page 697, you can invoke the procedure in FULLSCREEN mode and use the MOVE command to move the nodes to desired locations. In this example, the data set NETWORK produced by PROC NETDRAW is used to change the \( x \) and \( y \) coordinates of the nodes. A new data set called NODEPOS is created from NETWORK by retaining only the observations containing node positions (recall that for such observations, \_SEQ_ = ‘0’) and by dropping the \_SEQ_ variable. Further, the \( y \) coordinates (given by the values of the \_Y_ variable) for the two activities ‘Select Households’ and ‘Train Personnel’ are interchanged. The new data set, displayed in Output 9.13.3, is then input to PROC NETDRAW.
Output 9.13.2 Layout Data Set

Project: Market Survey

<table>
<thead>
<tr>
<th>Obs</th>
<th>FROM</th>
<th>TO</th>
<th>X</th>
<th>Y</th>
<th>SEQ</th>
<th>PATTERN</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>plan sur</td>
<td>hire per</td>
<td>1.0</td>
<td>2</td>
<td>0</td>
<td>Plan Survey</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>plan sur</td>
<td>hire per</td>
<td>1.5</td>
<td>2</td>
<td>1</td>
<td>Plan Survey</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>plan sur</td>
<td>hire per</td>
<td>1.5</td>
<td>3</td>
<td>2</td>
<td>Plan Survey</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>plan sur</td>
<td>design q</td>
<td>1.0</td>
<td>2</td>
<td>0</td>
<td>Plan Survey</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>hire per</td>
<td>tm per</td>
<td>2.0</td>
<td>3</td>
<td>0</td>
<td>Hire Personnel</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>hire per</td>
<td>tm per</td>
<td>2.5</td>
<td>3</td>
<td>1</td>
<td>Hire Personnel</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>hire per</td>
<td>tm per</td>
<td>2.5</td>
<td>1</td>
<td>2</td>
<td>Hire Personnel</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>design q</td>
<td>tm per</td>
<td>2.0</td>
<td>2</td>
<td>0</td>
<td>Design Questionnaire</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>design q</td>
<td>tm per</td>
<td>2.5</td>
<td>2</td>
<td>1</td>
<td>Design Questionnaire</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>design q</td>
<td>tm per</td>
<td>2.5</td>
<td>1</td>
<td>2</td>
<td>Design Questionnaire</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>design q</td>
<td>select h</td>
<td>2.0</td>
<td>2</td>
<td>0</td>
<td>Design Questionnaire</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>design q</td>
<td>select h</td>
<td>2.5</td>
<td>2</td>
<td>1</td>
<td>Design Questionnaire</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>design q</td>
<td>select h</td>
<td>2.5</td>
<td>3</td>
<td>2</td>
<td>Design Questionnaire</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>design q</td>
<td>print q</td>
<td>2.0</td>
<td>2</td>
<td>0</td>
<td>Design Questionnaire</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>tm per</td>
<td>cond sur</td>
<td>3.0</td>
<td>1</td>
<td>0</td>
<td>Train Personnel</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>tm per</td>
<td>cond sur</td>
<td>3.5</td>
<td>1</td>
<td>1</td>
<td>Train Personnel</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>tm per</td>
<td>cond sur</td>
<td>3.5</td>
<td>2</td>
<td>2</td>
<td>Train Personnel</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>select h</td>
<td>cond sur</td>
<td>3.0</td>
<td>3</td>
<td>0</td>
<td>Select Households</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>select h</td>
<td>cond sur</td>
<td>3.5</td>
<td>3</td>
<td>1</td>
<td>Select Households</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>select h</td>
<td>cond sur</td>
<td>3.5</td>
<td>2</td>
<td>2</td>
<td>Select Households</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>print q</td>
<td>cond sur</td>
<td>3.0</td>
<td>2</td>
<td>0</td>
<td>Print Questionnaire</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>cond sur</td>
<td>analyze</td>
<td>4.0</td>
<td>2</td>
<td>0</td>
<td>Conduct Survey</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>analyze</td>
<td></td>
<td>5.0</td>
<td>2</td>
<td>0</td>
<td>Analyze Results</td>
<td>1</td>
</tr>
</tbody>
</table>

```plaintext
data nodepos;
  set network;
  if _seq_ = 0;
  drop _seq_;
  if _from_ = 'select h' then _y_=1;
  if _from_ = 'tm per' then _y_=3;
  run;

title2 'Modified Node Positions';
proc print data=nodepos;
  run;
```
Note that the data set NODEPOS contains variables named _FROM_ and _TO_, which specify the (activity, successor) information; hence, the call to PROC NETDRAW does not contain the ACTIVITY= and SUCCESSOR= specifications. The presence of the variables _X_ and _Y_ indicates to PROC NETDRAW that the data set contains the x and y coordinates for all the nodes. Because there is no variable named _SEQ_ in this data set, PROC NETDRAW assumes that only the node coordinates are given and uses these node positions to determine how the arcs are to be routed. The resulting network diagram is shown in Output 9.13.4.

```plaintext
proc netdraw data=nodepos graphics;
  actnet / id=(id) nodefid nolabel
carcs = blue
certext = white
coutline = red
centerid
boxht = 3
htext=2
pcompress
separatearcs
ybetween=8;
run;
```
This example uses a typical problem in network flow optimization to illustrate how you can use PROC NETDRAW to draw a network by specifying completely all the node positions. Consider a simple two-period production inventory problem with one manufacturing plant (PLANT), two warehouses (DEPOT1 and DEPOT2), and one customer (CUST). In each period, the customer can receive goods directly from the plant or from the two warehouses. The goods produced at the plant can be used to satisfy directly some or all of the customer’s demands or can be shipped to a warehouse. Some of the goods can also be carried over to the next period as inventory at the plant. The problem is to determine the minimum cost of satisfying the customer’s demands; in particular, how much of the customer’s demands in each period is to be satisfied from the inventory at the two warehouses or from the plant, and also how much of the production is to be carried over as inventory at the plant? This problem can be solved using PROC NETFLOW; the details are not discussed here. Let PLANT_1 represent the production at the plant in period 1, DEPOT1_1 represent the inventory at DEPOT1 in period 1, DEPOT2_1 represent the inventory at DEPOT2 in period 1, and CUST_1 represent the customer’s demand in period 1 (i =1, 2). These variables can be thought of as nodes in a network with the following data representing the COST and CAPACITY of the arcs connecting them:
### Table: Network Flow Information

<table>
<thead>
<tr>
<th>FROM</th>
<th>TO</th>
<th>COST</th>
<th>CAPACITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANT_1</td>
<td>CUST_1</td>
<td>10</td>
<td>75</td>
</tr>
<tr>
<td>PLANT_1</td>
<td>DEPOT1_1</td>
<td>7</td>
<td>75</td>
</tr>
<tr>
<td>PLANT_1</td>
<td>DEPOT2_1</td>
<td>8</td>
<td>75</td>
</tr>
<tr>
<td>DEPOT1_1</td>
<td>CUST_1</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>DEPOT2_1</td>
<td>CUST_1</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>PLANT_1</td>
<td>PLANT_2</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>DEPOT1_1</td>
<td>DEPOT1_2</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>DEPOT2_1</td>
<td>DEPOT2_2</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>PLANT_2</td>
<td>CUST_2</td>
<td>10</td>
<td>75</td>
</tr>
<tr>
<td>PLANT_2</td>
<td>DEPOT1_2</td>
<td>7</td>
<td>75</td>
</tr>
<tr>
<td>PLANT_2</td>
<td>DEPOT2_2</td>
<td>8</td>
<td>75</td>
</tr>
<tr>
<td>DEPOT1_2</td>
<td>CUST_2</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>DEPOT2_2</td>
<td>CUST_2</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>CUST_1</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>CUST_2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

Suppose that you want to use PROC NETDRAW to draw the network corresponding to the preceding network flow problem and suppose also that you require the nodes to be placed in specific positions. The following program saves the network information along with the required node coordinates in the Network data set ARCS and invokes PROC NETDRAW to draw the network diagram shown in Output 9.14.1. The Network data set also contains a variable named `_pattern`, which specifies that pattern statement 1 be used for nodes relating to period 1 and pattern statement 2 be used for those relating to period 2.

```sas
data arcs;
  input from $ to $ _x_ _y_ _pattern;
  datalines;
  PLANT_1 CUST_1 1 5 1
  PLANT_1 DEPOT1_1 1 5 1
  PLANT_1 DEPOT2_1 1 5 1
  DEPOT1_1 CUST_1 2 6 1
  DEPOT2_1 CUST_1 2 4 1
  PLANT_1 PLANT_2 1 5 1
  DEPOT1_1 DEPOT1_2 2 6 1
  DEPOT2_1 DEPOT2_2 2 4 1
  PLANT_2 CUST_2 4 2 2
  PLANT_2 DEPOT1_2 4 2 2
  PLANT_2 DEPOT2_2 4 2 2
  DEPOT1_2 CUST_2 5 3 2
  DEPOT2_2 CUST_2 5 1 2
  CUST_1 . 3 5 1
  CUST_2 . 6 2 2
;

  title c=blue 'Distribution Network';
  pattern1 v=s c=green;
  pattern2 v=s c=red;
  proc netdraw data=arcs graphics out=netout;
    actnet / act=from succ=to separatearcs
       ybetween = 4
       centerid
       ctext = white
```

Example 9.15: Organizational Charts with PROC NETDRAW

This example illustrates using the TREE option to draw organizational charts. The Network data set, DOCUMENT, describes how the procedures are distributed between two volumes of the SAS/OR documentation. The structure can be visualized easily in a tree diagram. The data set DOCUMENT contains the parent-child relationship for each node of the diagram. For each node, a detailed description is contained in the variable ID. In addition, the variable _pattern specifies the pattern to be used for each node. PROC NETDRAW is invoked with the TREE option, which illustrates the organization of the documentation in the form of a tree diagram drawn from left to right. The CENTERID option centers text within each node. Arrowheads are not necessary for this diagram and are suppressed by specifying ARROWHEAD=0. Output 9.15.1 shows the resulting diagram.
data document;
  format parent $8. id $24.;
  input parent $ child $ id & _pattern;
datalines;
OR MPBOOK Operations Research 1
OR PMBOOK Operations Research 1
PMBOOK CPM Project Management 2
PMBOOK DTREE Project Management 2
PMBOOK EVM Project Management 2
PMBOOK GANTT Project Management 2
PMBOOK NETDRAW Project Management 2
PMBOOK PM Project Management 2
MPBOOK OPTMODEL Mathematical Programming 3
MPBOOK OPTLP Mathematical Programming 3
MPBOOK OPTMILP Mathematical Programming 3
MPBOOK OPTQP Mathematical Programming 3
CPM . CPM Procedure 2
DTREE . DTREE Procedure 2
EVM . EVM Macros 2
GANTT . GANTT Procedure 2
NETDRAW . NETDRAW Procedure 2
PM . PM Procedure 2
OPTMODEL . OPTMODEL Procedure 3
OPTLP . OPTLP Procedure 3
OPTMILP . OPTMILP Procedure 3
OPTQP . OPTQP Procedure 3
;

pattern1 v=s c=blue;
pattern2 v=s c=red;
pattern3 v=s c=green;

title j=l h=3 'Operations Research Documentation';
title2 j=l h=2 'Procedures in Each Volume';
footnote j=r h=2 'Default Tree Layout ';
proc netdraw graphics data=document;
  actnet / act=parent
    succ=child
    id=(id)
    nodelid
    nolabel
    pcompress
    centerid
tree
    xbetween=15
    ybetween=3
    arrowhead=0
    rectilinear
    carcs=black
    ctext=white
    htext=3;
run;
Example 9.15: Organizational Charts with PROC NETDRAW

Output 9.15.1 Organization of Documentation: Default TREE Layout

The procedure draws the tree compactly with the successors of each node being placed to the immediate right of the node, ordered from top to bottom in the order of occurrence in the Network data set. The next invocation of PROC NETDRAW illustrates the effect of the SEPARATESONS and CENTERSUBTREE options on the layout of the tree (see Output 9.15.2).

footnote j=r h=1.5 'Centered Tree Layout ';  
proc netdraw graphics data=document;  
    actnet / act=parent  
        succ=child  
        id=(id)  
        nodefid  
        nolabel  
        pcompress  
        novcenter  
        centerid  
        tree  
        separatesons  
        centersubtree  
        xbetween=15  
        ybetween=3  
        arrowhead=0
Example 9.16: Annotate Facility with PROC NETDRAW

This example demonstrates the use of PROC NETDRAW for a nonstandard application. The procedure is used to draw a time table for a class of students. The days of the week are treated as different zones, and the times within a day are treated as different values of an alignment variable. The following DATA step defines a total of twenty activities, ‘m1’, . . . , ‘f5’, which refer to the five different periods for the five different days of the week. The variable \textit{class} contains the name of the subject taught in the corresponding period and day. Note that the periods are taught during the hours 1, 2, 3, 5, and 6; the fourth hour is set aside for lunch. The time axis is labeled with the format CLASSTIM, which is defined using PROC FORMAT. The USEFORMAT option in the ACTNET statement instructs PROC NETDRAW to use the explicit format specified for the time variable rather than the default format.

This example also illustrates the use of the Annotate facility with PROC NETDRAW. The data set \textit{ANNO} labels the fourth period ‘LUNCH.’ The positions for the text are specified using data coordinates that refer to
Example 9.16: Annotate Facility with PROC NETDRAW

the \((X, Y)\) grid used by PROC NETDRAW. Thus, for example \(X = '4'\) identifies the \(x\) coordinate for the annotated text to be the fourth period, and the \(y\) coordinates are set appropriately. The resulting time table is shown in Output 9.16.1.

```plaintext
/* Define format for the ALIGN= variable */
proc format;
    value classtim 1 = ' 9:00 - 10:00'
                    2 = '10:00 - 11:00'
                    3 = '11:00 - 12:00'
                    4 = '12:00 - 1:00'
                    5 = ' 1:00 - 2:00'
                    6 = ' 2:00 - 3:00';
run;

data schedule;
    format day $9. class $12. ;
    input day $ class & time daytime $ msucc $;
    format time classtim.;
    label day = "Day \ Time";
    datalines;
Monday Mathematics 1 m1 .
Monday Language 2 m2 .
Monday Soc. Studies 3 m3 .
Monday Art 5 m4 .
Monday Science 6 m5 .
Tuesday Language 1 t1 .
Tuesday Mathematics 2 t2 .
Tuesday Science 3 t3 .
Tuesday Music 5 t4 .
Tuesday Soc. Studies 6 t5 .
Wednesday Mathematics 1 w1 .
Wednesday Language 2 w2 .
Wednesday Soc. Studies 3 w3 .
Wednesday Phys. Ed. 5 w4 .
Wednesday Science 6 w5 .
Thursday Language 1 th1 .
Thursday Mathematics 2 th2 .
Thursday Science 3 th3 .
Thursday Phys. Ed. 5 th4 .
Thursday Soc. Studies 6 th5 .
Friday Mathematics 1 f1 .
Friday Language 2 f2 .
Friday Soc. Studies 3 f3 .
Friday Library 5 f4 .
Friday Science 6 f5 .
;

data anno;
    /* Set up required variable lengths, etc. */
    length function color style $8;
    length xsys ysys hsys $1;
    length when position $1;
    xsys = '2';
```
ysys = '2';
hsys = '4';
when = 'a';

function = 'label ';
x = 4;
size = 2;
position = '5';
y=5; TEXT='L'; output;
y=4; TEXT='U'; output;
y=3; TEXT='N'; output;
y=2; TEXT='C'; output;
y=1; TEXT='H'; output;
run;

pattern1 v=s c=pink;
title 'Class Schedule: 2003-2004';
footnote j=l h=2 ' Teacher: Mr. A. Smith Hall'
    j=r h=2 'Room: 107 ';
proc netdraw graphics data=schedule anno=anno;
    actnet / act=daytime
        succ=msucc
        id=(class)
        nodefid nolabel
        zone=day
        align=time
        useformat
        linear
        pcompress
        coutline=black
        hmargin = 2 vmargin = 2
        htext=2;
run;
Example 9.17: AOA Network Using the Annotate Facility

This example illustrates the use of the Annotate Facility to draw an Activity-on-Arc network. First, PROC NETDRAW is invoked with explicit node positions for the vertices of the network. The ALIGN= and ZONE= options are used to provide horizontal and vertical axes as a frame of reference. The resulting diagram is shown in Output 9.17.1.

```sas
data widgaoa;
    format task $12.;
    input task & days tail head _x_ _y_;
datalines;
Approve Plan  5  1  2  1  2
Drawings      10  2  3  4  2
Study Market  5  2  4  4  2
Write Specs   5  2  3  4  2
Prototype     15  3  5  7  1
     Mkt. Strat.  10  4  6 10  3
    Materials    10  5  7 10  1
Init. Prod.   10  5  7 10  1
```

Teacher: Mr. A. Smith Hall
Room: 107
Evaluate 10 8 9 16 1
Test Market 15 6 9 18 2
Changes 5 9 10 20 1
Production 0 10 11 23 1
Marketing 0 6 12 19 2
Dummy 0 8 6 16 1
.
.
11 . 26 1
.
12 . 22 3
;

pattern1 v=e c=red;
title j=l h=3 'Project: Widget Manufacture';
title2 j=l h=2 'Network in Activity-on-Arc Format';
footnote j=r h=2 'Initial Layout';

proc netdraw graphics data=widgaoa out=netout;
  actnet / act=tail
    succ=head
    id=(tail)
    align=_x_
    zone=_y_
    ybetween = 10
    nodefid
    nolabel
    pcompress
    htext=2;
  label _y_=' Y \ X '; run;

In Output 9.17.1, the arc leading from vertex ‘4’ to vertex ‘6’ has two turning points: (10.5, 3) and (10.5, 2). Suppose that you want the arc to be routed differently, to provide a more symmetric diagram. The next DATA step creates a data set, NETIN, which changes the x coordinates of the turning points to 16.5 instead of 10.5. Further, two Annotate data sets are created: the first one labels the nodes outside the boxes, either to the top or to the bottom, and the second one sets labels for the arcs. PROC NETDRAW is then invoked with the combined Annotate data set to produce the diagram shown in Output 9.17.2.

data netin;
  set netout;
  if _from_=4 and _to_=6 and _seq_>0 then _x_=16.5;
run;

data anno1;
  set netout;
  if _seq_=0;
    /* Set up required variable lengths, etc. */
    length function color style $8;
    length xsys ysys hsys $1;
    length when position $1;
    length TEXT $12;
    xsys = '2';
    ysys = '2';
    hsys = '4';
    when = 'a';
    function = 'label ';
    size = 2;
Output 9.17.1  Activity-on-Arc Format

Project: Widget Manufacture

Network in Activity-on-Arc Format
position = '5';
TEXT = left(put(tail, f2.));
x=_x_; if _y_ = 1 then y=_y_- .3; else y=_y_+.5; run;

data anno2;
/* Set up required variable lengths, etc. */
length function color style $8;
length xsys ysys hsys $1;
length when position $1;
length TEXT $12;
xsys = '2';
ysys = '2';
hsys = '4';
when = 'a';
function = 'label '; size = 2;
position = '5';
x=2.5; y=1.8; TEXT='Approve Plan'; output;
x=5.5; y=.8; TEXT='Drawings'; output;
x=5.7; y=1.4; TEXT='Write Specs'; output;
x=7; y=3.4; TEXT='Study Market'; output;
x=8.5; y=.8; TEXT='Prototype'; output;
x=11.5; y=1.4; TEXT='Facility'; output;
x=11.5; y=.8; TEXT='Materials'; output;
x=14.5; y=.9; TEXT='Init. Prod'; output;
x=13.5; y=3.4; TEXT='Mkt. Strat.'; output;
x=18; y=.8; TEXT='Evaluate'; output;
x=21.5; y=.8; TEXT='Changes'; output;
x=24.5; y=.8; TEXT='Production'; output;
x=20; y=3.4; TEXT='Marketing'; output;
position=6;
x=16.6; y=1.5; TEXT='Dummy'; output;
x=18.6; y=1.5; TEXT='Test Market'; output;
;
data anno;
set anno1 anno2;
run;

footnote j=r h=2 'Annotated and Modified Layout ';
pattern1 v=s c=red;

proc netdraw graphics data=netin anno=anno;
actnet / nodefid
nolabel
boxwidth=1
pcompress
novcenter
vmargin=20
xbetween=10;
run;
Example 9.18: Branch and Bound Trees

This example illustrates a nonstandard use of PROC NETDRAW. The TREE option in PROC NETDRAW is used to draw a branch and bound tree such as one that you obtain in the solution of an integer programming problem. Refer to Chapter 4, “The LP Procedure” (SAS/OR User's Guide: Mathematical Programming Legacy Procedures) for a detailed discussion of branch and bound trees. The data used in this example were obtained from one particular invocation of PROC LP.

The data set NET (created in the following DATA step) contains information pertaining to the branch and bound tree. Each observation of this data set represents a particular iteration of the integer program, which can be drawn as a node in the tree. The variable node names the problem. The variable object gives the objective value for that problem. The variable problem identifies the parent problem corresponding to each node; for example, since the second and the seventh observations have problem equal to ‘-1’ and ‘1’, respectively, it indicates that the second and the seventh problems are derived from the first iteration. Finally, the variable _pattern specifies the pattern of the nodes based on the status of the problem represented by the node.

```sas
data net;
  input node problem cond $10. object;
  if cond="ACTIVE" then _pattern=1;
  else if cond="SUBOPTIMAL" then _pattern=2;
  else _pattern=3;
datalines;
  1 0 ACTIVE 4
  2 -1 ACTIVE 4
  3 2 ACTIVE 4
  4 -3 ACTIVE 4.3333333
  5 4 SUBOPTIMAL 5
  6 3 FATHOMED 4.3333333
  7 1 ACTIVE 4
  8 -7 ACTIVE 4
  9 -8 FATHOMED 4.3333333
  10 8 FATHOMED 4.3333333
  11 7 FATHOMED 4.5
  12 -11 FATHOMED 4.3333333
  13 11 FATHOMED 4.5;
```

The next DATA step (which creates the data set LOGIC) uses this child-parent information to format the precedence relationships as expected by PROC NETDRAW. Next, the two data sets are merged together to create the Network input data set (BBTREE) for PROC NETDRAW. The ID variable in the data set BBTREE is formatted to contain both the iteration number and the objective value.

Finally, PROC NETDRAW is invoked with the TREE, ROTATE, and ROTATETEXT options to produce a branch and bound tree shown in Output 9.18.1. Note that the ROTATE and ROTATETEXT options produce a rotated graph with a top-down orientation.

```sas
/* set precedence relationships
   using child-parent information */
data logic;
  keep node succ;
```
Output 9.17.2  Activity-on-Arc Format: Annotated Diagram
Example 9.18: Branch and Bound Trees

```
set net(firstobs=2);
succ=node;
node=abs(problem);
run;

proc sort data=logic;
  by node;
run;
/* combine the logic data and the node data */
/* set ID values */
data bbtree;
  length id $ 9;
  merge logic net; by node;
  if node < 10 then id=put(node,1.)||put(object,f8.2);
  else id=put(node,2.)||put(object,f7.2);
run;

goptions border rotate=portrait;
pattern1 v=s c=green;
pattern2 v=s c=red;
pattern3 v=s c=blue;
title h=3 j=c 'Branch and Bound Tree';
title2 '';
footnote1 h=1.5 j=c c=red 'Optimal ' c=green ' Active ' c=blue ' Fathomed ';
footnote2 '';
footnote3 h=1.5 ' Node shows Iteration Number and Objective Value ';
proc netdraw data=bbtree graphics out=bbout;
  actnet /activity=node
    successor=succ
    id=(id)
nodelabel
    ctext=white
coutline=black
carcs=black
  xbetween=15
  ybetween=3
  compress
  font=swiss
  rectilinear
  tree
  rotate
  rotatetext
  arrowhead=0
  htext=2;
run;
```
In the next invocation, PROC NETDRAW uses a modified layout of the nodes to produce a diagram where the nodes are aligned according to the iteration number. The following program uses the Layout data set produced in the previous invocation of PROC NETDRAW. The same y coordinates are used; but the x coordinates are changed to equal the iteration number. Further, the ALIGN= specification produces a time axis that labels each level of the diagram with the iteration number. Each node is labeled with the objective value. The resulting diagram is shown in Output 9.18.2.

```
data netin;
  set bbout;
  if _seq_ = 0; drop _seq_;
  _x_ = _from_;  
  id = substr(id, 3);
run;
```

```
options rotate=landscape;
title h=3 'Branch and Bound Tree';
title2 h=2 'Aligned by Iteration Number';
footnotel h=1.5 j=c c=red 'Optimal '
  c=green ' Active '
  c=blue ' Fathomed '
footnote2 ' ';
```
Example 9.18: Branch and Bound Trees

footnote3 \textit{j=l h=1.5 ' Node shows Objective Value '};

pattern1 v=e c=green;
pattern2 v=e c=red;
pattern3 v=e c=blue;

proc netdraw data=netin graphics;
  actnet /id=(id)
    ctext=black
carcs=black
    align = _from_
    frame
    pcompress
    rectilinear
    arrowhead=0
    nodefid
    nolabel
    htext=2.5
    xbetween=8;
  run;

\textbf{Output 9.18.2} Branch and Bound Tree: Aligned by Iteration Number

\begin{verbatim}
\textbf{Branch and Bound Tree}
\textbf{Aligned by Iteration Number}
\end{verbatim}

\begin{table}
\centering
\begin{tabular}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\
\hline
\end{tabular}
\end{table}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{branch_and_bound_tree.png}
\end{figure}

Node shows Objective Value

Optimal \hspace{1cm} Active \hspace{1cm} Fathomed
The next two tables reference the options in the NETDRAW procedure that are illustrated by the examples in this section. Note that all the options are specified on the ACTNET statement.

**Table 9.9** Options Specified in Examples 5.1–5.9

<table>
<thead>
<tr>
<th>Option</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITY=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ALIGN=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARROWHEAD=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>AUTOREF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOXHT=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOXWIDTH=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARCS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CCRITARCS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CCRITOUT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CENTERID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>COMPRESS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>COUTLINE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CREF=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CTEXT=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DURATION=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FONT=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>FRAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>GRAPHICS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HMARGIN=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>HPAGES=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>HTEXT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HTRACKS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ID=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LAG=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LINEPRINTER=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LREF=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LWCRT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LWIDTH=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NODEFID</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NOLABEL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NOPAGENUMBER</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NOVCENTER</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PATTERN=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PCOMPRESS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SEPARATEARCS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SHOWBREAK</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SHOWSTATUS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Table 9.9  (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCCESSOR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>TIMESCALE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>VMARGIN=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VPAGES=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XBETWEEN=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YBETWEEN=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9.10  Options Specified in Examples 5.10–5.18

<table>
<thead>
<tr>
<th>Option</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITY=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ALIGN=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ANNOTATE=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARROWHEAD=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTOREF=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOXHT=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOXWIDTH=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BREAKCYCLE=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARCS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CAXIS=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCRITARCS=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CENTERID=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CENTERSUBTREE=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPRESS=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COUTLINE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CREF=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREFBRK=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTEXT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DATA=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DURATION=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FONT=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRAME</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRAPHICS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HMARGIN=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTEXT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ID=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LINEAR=</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LREF=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LREFBRK=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MININTERVAL=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOARROWFILL=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NODEFID=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NOLABEL=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
### Table 9.10 (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOVCENTER</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUT=</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PATTERN=</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCOMPRESS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RECTILINEAR</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFBREAK</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROTATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>ROTATETEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SEPARATEARCS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEPARATESONS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUCCESSOR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMESCALE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TREE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USEFORMAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>VMARGIN=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XBETWEEN=</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YBETWEEN=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZONE=</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZONEPAT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZONESPACE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### References


Subject Index

Activity data set
- as input to NETDRAW procedure, 690
activity status
  - indicating within node, 681, 735
ACTIVITY variable
  - Network data set (NETDRAW), 677, 691
A_FINISH variable
  - Network data set (NETDRAW), 678, 691, 695
ALIGN variable
  - Network data set (NETDRAW), 677, 680, 691, 697
Annotate data set
  - NETDRAW procedure, 666, 705
Annotate facility
  - coordinate systems (NETDRAW), 705
  - example (NETDRAW), 762, 765
  - NETDRAW procedure, 705, 706
arc routing, 692, 693
arrowheads
  - length of, 684
  - no fill, 686
  - suppress display (NETDRAW), 684, 759
A_START variable
  - Network data set (NETDRAW), 678, 691, 695
center
  - ID variables (NETDRAW), 684
  - turning off vertical (NETDRAW), 687, 695, 722
color specification
  - arcs (NETDRAW), 683, 684
  - axis (NETDRAW), 683, 684
  - break lines along time axis (NETDRAW), 683, 685
  - critical arcs (NETDRAW), 683, 684
  - nodes (NETDRAW), 702, 705
  - outline for all nodes (NETDRAW), 685
  - outline for critical nodes (NETDRAW), 684
  - reference lines (NETDRAW), 683, 685
  - text (NETDRAW), 685
compress in graphics mode
  - network diagram, 668, 684, 687, 722
computer resource requirements
  - NETDRAW procedure, 707
coordinate system
  - annotate processing (NETDRAW), 705
corners, rectangular
    - network diagram, 687
corners, rounded
network diagram, 687
CPU requirement, see computer resource requirements
critical activities
  - node pattern (NETDRAW), 705
cycles
  - in network diagrams, 667, 699, 749
display
  - information in network diagram, 694
displayed output
  - NETDRAW procedure, 694
draw and display networks, see network layout, NETDRAW procedure
DURATION variable
  - Network data set (NETDRAW), 678
dynamic programming algorithm, 678, 694, 734
E_FINISH variable
  - Network data set (NETDRAW), 678, 691, 695
draw and display networks, see network layout, NETDRAW procedure
events
  - NETDRAW procedure, 706
E_START variable
  - Network data set (NETDRAW), 678, 691, 695
draw and display networks, see network layout, NETDRAW procedure
examples
  - statement and option cross-reference tables (NETDRAW), 774
F_FLOAT variable
  - Network data set (NETDRAW), 678, 691, 695
font specification
  - text (NETDRAW), 685
format control options
  - NETDRAW procedure, 678, 680–682, 694, 696
_FROM_ variable
  - Layout data set (NETDRAW), 696, 697
  - Network data set (NETDRAW), 677
full-screen version
  - changing the scale (NETDRAW), 701
  - global commands (NETDRAW), 704
  - local commands (NETDRAW), 702
  - modifying the network layout, 697
  - NETDRAW procedure, 701, 702, 704
  - options specific to (NETDRAW), 676, 682, 683
  - use of the PATTERN variable (NETDRAW), 702
functional summary
  - NETDRAW procedure, 672
general options
  - NETDRAW procedure, 677–682
global commands
  full-screen version (NETDRAW), 704
graphics catalog
  NETDRAW procedure, 676
  segment description (NETDRAW), 685
  segment name (NETDRAW), 686
graphics version
  NETDRAW procedure, 704–706
  options specific to (NETDRAW), 676, 683–689
  use of the PATTERN variable (NETDRAW), 705
hierarchical charts, see organizational charts
horizontal banding, see zoned network diagrams
horizontal space
  around margin (NETDRAW), 685, 738
  between nodes (NETDRAW), 682, 694, 732
ID variables
  NETDRAW procedure, 694, 695
  Network data set (NETDRAW), 678, 691, 695
input data sets
  NETDRAW procedure, 666
LAG variables
  Network data set (NETDRAW), 679
layout controls, see network layout, NETDRAW procedure
Layout data set
  as input to NETDRAW procedure, 690
  NETDRAW procedure, 666, 677, 690, 696
L_FINISH variable
  Network data set (NETDRAW), 678, 691, 695
line style specification
  break lines along time axis (NETDRAW), 686
  reference lines (NETDRAW), 686
line width specification
  arcs and nodes (NETDRAW), 686
  critical arcs and nodes (NETDRAW), 686
  node outlines (NETDRAW), 686
line-printer version
  options specific to (NETDRAW), 676, 689
local commands
  full-screen version (NETDRAW), 676, 689
L_START variable
  Network data set (NETDRAW), 678, 691, 695
macro variable
  _ORNETDR, 706
missing values
  NETDRAW procedure, 692
multiple pages
  NETDRAW procedure, 695
NETDRAW examples
  branch and bound trees, 769
  compressing graphics version to fit on a page, 722
  controlling information within nodes, 725
  controlling the number of pages, 725
  controlling the routing of the arcs, 732
  displaying status of the activities, 735
  drawing an AOA network using the Annotate facility, 765
  drawing organizational charts, 759
  drawing schematic diagrams, 746
  graphics version, 717
  illustrating data flow, 746
  illustrating several time-scale options, 742
  illustrating the Annotate facility, 762
  line-printer version, 712
  modifying network layout, 752
  nonstandard precedence relationships, 730
  spanning several pages, 718
  specifying node layout through the Network data set, 757
  time-scaled network diagram, 738
  zoned network diagram, 745
NETDRAW procedure
  Annotate data set, 676, 683
  Annotate facility, 705, 706
  arc layout options, 678, 680–682
  arc routing, 692, 693
  backward arcs in networks, 699
  computer resource requirements, 707
  controlling the network layout, 697
  cyclic networks, 667, 693, 699
  default ID variables, 694
  details, 689
  dynamic programming, 694
  format control, 678, 680–682, 694, 696
  full-screen options, 676, 682, 683
  full-screen version, 697, 701, 702, 704
  functional summary, 672
  graphics catalog specification, 676
  graphics options, 676, 683–689
  graphics version, 704–706
  hierarchical charts, 700
  horizontal placement of nodes, 697
  HTML Imagemap data set, 676
  Imagemap data set, 676
  information within node, 694
  input data sets, 666, 689
  Layout data set, 677, 696
  layout specification, 697
  line-printer options, 676, 689
  missing values, 692, 695
  modes of display, 667, 676, 695
  modifying the network layout, 667, 697
  moving nodes, 701
  Network data set, 676
network layout, 667, 668, 692–698
network specification, 677, 681
node coordinates and layout, 692
node layout, 667
number of pages, 696
ODS style template, 707
options classified by function, 672
organizational charts, 700
_ORNETDR macro variable, 706
output data sets, 666
overview, 666
page format details, 695
restricting number of tracks, 694
rotating the network, 698, 701
saving network layout information, 690
showing activity progress, 695
splitting across pages, 695
syntax skeleton, 672
table of syntax elements, 672
time-axis format, 698, 699
time-scaled networks, 667, 677–682, 698, 699
top-down networks, 698, 701
topological ordering, 693
tree diagrams, 678, 681, 682, 700
using full-screen to modify layout, 697
variables, list of, 691
vertical placement of nodes, 697
zoned networks, 667, 680, 682, 700
Network data set
description, 690
list of variables, 691
NETDRAW procedure, 666, 668, 689
network layout, NETDRAW procedure
discussion, 692–698
modifications for time-scaled networks, 699
modifications for zoned networks, 700
use of dynamic programming, 694
nodes on network diagram
coordinates, 692
height, 678, 694, 732
layout, 692
width, 678, 694, 726
number of nodes
horizontal (NETDRAW), 687
vertical (NETDRAW), 687
number of pages
displaying network diagram, 680
horizontal (NETDRAW), 685
NETDRAW procedure, 696
vertical (NETDRAW), 688
organizational charts
NETDRAW procedure, 700, 759
_OUTNETDR macro variable, 706
outlines and arcs
network diagrams, 683, 689
output data sets
NETDRAW procedure, 666
Output Delivery System (ODS)
NETDRAW procedure, 707
NETDRAW style template, 707
overview
NETDRAW procedure, 666
page
drawing network on a single, 684, 687, 722
format of network diagram, 695
page numbers
in NETDRAW procedure, 687
suppress displaying (NETDRAW), 686
pages
limiting number of (NETDRAW), 680
pattern specification
NETDRAW procedure, 705
PATTERN variable
Network data set (NETDRAW), 683, 687, 700, 702, 705
_PATTERN variable
Network data set (NETDRAW), 683, 687, 697, 700, 705
placement of ID variables (NETDRAW), 695
project
graphical representation, 668
rectangular corners
network diagram, 687
reference lines
automatic at every tick mark, 677
break character, 682
character to represent, 683, 689
color, 683, 685
line style, 686
rotating the network diagram, 668, 688, 694, 698, 701, 769
used for top-down trees, 698, 701
rounded corners
network diagram, 687
SAS catalogs, see graphics catalog
SAS data sets, see input data sets, see output data sets
NETDRAW procedure, 666
Schedule data set
as input to NETDRAW procedure, 690
_SEQ_ variable
Layout data set (NETDRAW), 697
Network data set (NETDRAW), 691, 697
S_FINISH variable
Network data set (NETDRAW), 678, 691, 695
spacing between
nodes, 682
S_START variable
- Network data set (NETDRAW), 678, 691, 695
STATUS variable
- Network data set (NETDRAW), 695
  Schedule data set (CPM), 681
SUCCESSOR variables
- Network data set (NETDRAW), 681, 691
supercritical activities
  node pattern (NETDRAW), 705
syntax skeleton
  NETDRAW procedure, 672
text
  color (NETDRAW), 685
  font (NETDRAW), 685
  height (NETDRAW), 685
T_FLOAT variable
- Network data set (NETDRAW), 678, 691, 695
time axis
  formatting, 682, 698, 699, 764
  maximum and minimum values, 677
  NETDRAW procedure, 698, 699
  scaling, 679, 698, 699
  show break in, 681
  smallest interval represented, 679
  suppress printing, 680
  suppress printing on every page, 680
time-scaled network diagrams, 667, 671, 698, 738, 742
  default ALIGN variable, 698
  spacing of nodes, 698
  tick marks, 698
_TO_ variable
- Layout data set (NETDRAW), 696, 697
  Network data set (NETDRAW), 681
top-down hierarchical diagram, 687
tree diagrams
  NETDRAW procedure, 667, 700, 701, 759
variables
  list of, NETDRAW procedure, 691
  treatment of missing values (NETDRAW), 692
vertical space
  around margin (NETDRAW), 688, 738
  between nodes (NETDRAW), 682, 694, 726
Web-enabled network diagram, 676, 688, 706
WEB variable
- Network data set (NETDRAW), 688
work breakdown structure, 667
_X_ variable
- Layout data set (NETDRAW), 697
  Network data set (NETDRAW), 691, 697
_Y_ variable
- Layout data set (NETDRAW), 697
  Network data set (NETDRAW), 691, 697
ZONE variable
- Network data set (NETDRAW), 682, 691, 697
zoned network diagrams, 667, 700, 745
## Syntax Index

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Page References</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITY= option</td>
<td>ACTNET statement (NETDRAW)</td>
<td>677, 712</td>
</tr>
<tr>
<td>ACTNET statement (NETDRAW)</td>
<td></td>
<td>677</td>
</tr>
<tr>
<td>full-screen options</td>
<td></td>
<td>682, 683</td>
</tr>
<tr>
<td>general options</td>
<td></td>
<td>677–682</td>
</tr>
<tr>
<td>graphics options</td>
<td></td>
<td>683–689</td>
</tr>
<tr>
<td>line-printer options</td>
<td></td>
<td>689</td>
</tr>
<tr>
<td>ALIGN= option</td>
<td>ACTNET statement (NETDRAW)</td>
<td>677, 698, 740</td>
</tr>
<tr>
<td>ANNOTATE= option</td>
<td>ACTNET statement (NETDRAW)</td>
<td>683</td>
</tr>
<tr>
<td>PROC NETDRAW statement</td>
<td></td>
<td>676, 764</td>
</tr>
<tr>
<td>ARROWHEAD= option</td>
<td>ACTNET statement (NETDRAW)</td>
<td>684</td>
</tr>
<tr>
<td>AUTOREF option</td>
<td>ACTNET statement (NETDRAW), 677, 740</td>
<td>746</td>
</tr>
<tr>
<td>AUTOZONE option</td>
<td>ACTNET statement (NETDRAW)</td>
<td>677</td>
</tr>
<tr>
<td>BOTTOMUP keyword</td>
<td>CHILDORDER= option (NETDRAW), 678</td>
<td>681</td>
</tr>
<tr>
<td>BOXHT= option</td>
<td>ACTNET statement (NETDRAW), 678, 694, 732</td>
<td>732</td>
</tr>
<tr>
<td>BOXWIDTH= option</td>
<td>ACTNET statement (NETDRAW), 678, 694, 726</td>
<td>726</td>
</tr>
<tr>
<td>BREAKCYCLE option</td>
<td>ACTNET statement (NETDRAW), 677, 752</td>
<td>752</td>
</tr>
<tr>
<td>BRKCHAR= option</td>
<td>ACTNET statement (NETDRAW), 682, 689</td>
<td></td>
</tr>
<tr>
<td>CARCS= option</td>
<td>ACTNET statement (NETDRAW), 683, 684, 717, 719</td>
<td>719</td>
</tr>
<tr>
<td>CAXIS= option</td>
<td>ACTNET statement (NETDRAW), 683, 684, 746</td>
<td>746</td>
</tr>
<tr>
<td>CCNODEFILL= option</td>
<td>ACTNET statement (NETDRAW), 684</td>
<td></td>
</tr>
<tr>
<td>CCRITARCS= option</td>
<td>ACTNET statement (NETDRAW), 683, 684, 719</td>
<td></td>
</tr>
<tr>
<td>CCRITOUT= option</td>
<td>ACTNET statement (NETDRAW), 683, 684, 719</td>
<td></td>
</tr>
<tr>
<td>CENTERID option</td>
<td>ACTNET statement (NETDRAW), 684, 705, 719</td>
<td>719</td>
</tr>
<tr>
<td>CENTERSUBTREE option</td>
<td>ACTNET statement (NETDRAW), 678, 701, 762</td>
<td>762</td>
</tr>
<tr>
<td>CHILDORDER= option</td>
<td>ACTNET statement (NETDRAW), 678</td>
<td></td>
</tr>
<tr>
<td>CNODEFILL= option</td>
<td>ACTNET statement (NETDRAW), 684</td>
<td></td>
</tr>
<tr>
<td>COLORS= option, GOPTIONS statement</td>
<td>NETDRAW procedure, 685</td>
<td></td>
</tr>
<tr>
<td>COMPRESS option</td>
<td>ACTNET statement (NETDRAW), 684, 722</td>
<td></td>
</tr>
<tr>
<td>COUTLINE= option</td>
<td>ACTNET statement (NETDRAW), 685, 705, 719</td>
<td></td>
</tr>
<tr>
<td>CREF= option</td>
<td>ACTNET statement (NETDRAW), 683, 685, 740</td>
<td>740</td>
</tr>
<tr>
<td>CREFBRK= option</td>
<td>ACTNET statement (NETDRAW), 683, 685, 740</td>
<td></td>
</tr>
<tr>
<td>CT= option, see CTEXT= option</td>
<td>ACTNET statement (NETDRAW), 685</td>
<td></td>
</tr>
<tr>
<td>CTEXT= option, GOPTIONS statement</td>
<td>NETDRAW procedure, 685</td>
<td></td>
</tr>
<tr>
<td>DATA= option</td>
<td>PROC NETDRAW statement, 676, 712</td>
<td></td>
</tr>
<tr>
<td>DAY keyword</td>
<td>MININTERVAL= option (NETDRAW), 679</td>
<td>679</td>
</tr>
<tr>
<td>DES= option, see DESCRIPTION= option</td>
<td>ACTNET statement (NETDRAW), 685</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION= option</td>
<td>ACTNET statement (NETDRAW), 685</td>
<td></td>
</tr>
<tr>
<td>DP option</td>
<td>ACTNET statement (NETDRAW), 678, 694, 734</td>
<td>734</td>
</tr>
<tr>
<td>DURATION= option</td>
<td>ACTNET statement (NETDRAW), 678, 740</td>
<td>714</td>
</tr>
<tr>
<td>FF keyword</td>
<td>LAG variable (NETDRAW), 679</td>
<td>679</td>
</tr>
<tr>
<td>FILLPAGE option</td>
<td>ACTNET statement (NETDRAW), 685</td>
<td></td>
</tr>
<tr>
<td>FONT= option</td>
<td>ACTNET statement (NETDRAW), 685</td>
<td></td>
</tr>
<tr>
<td>FORMCHAR= option</td>
<td>ACTNET statement (NETDRAW), 683, 689</td>
<td></td>
</tr>
<tr>
<td>FRAME option</td>
<td>ACTNET statement (NETDRAW), 678, 740, 742</td>
<td></td>
</tr>
<tr>
<td>FS keyword</td>
<td>LAG variable (NETDRAW), 679</td>
<td>679</td>
</tr>
<tr>
<td>FS option, see FULLSCREEN option</td>
<td>ACTNET statement (NETDRAW), 685</td>
<td></td>
</tr>
<tr>
<td>FTEXT= option, GOPTIONS statement</td>
<td>NETDRAW procedure, 685</td>
<td></td>
</tr>
<tr>
<td>FULLSCREEN option</td>
<td>PROC NETDRAW statement, 667, 676, 701</td>
<td>701</td>
</tr>
<tr>
<td>GOPTIONS statement</td>
<td>COLORS= option, 685</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CTEXT= option, 685</td>
<td></td>
</tr>
</tbody>
</table>
FTEXT= option, 685
HPOS= option, 705, 717, 725
VPOS= option, 705, 717, 725
GOUT= option
   PROC NETDRAW statement, 676
GRAPHICS option
   PROC NETDRAW statement, 667, 676, 717
HBETWEEN= option, see XBETWEEN= option
HEIGHT= option
   ACTNET statement (NETDRAW), 685
HMARGIN= option
   ACTNET statement (NETDRAW), 685, 738, 749
HOUR keyword
   MININTERVAL= option (NETDRAW), 679
HPAGES= option
   ACTNET statement (NETDRAW), 685
   PROCT:Statement, 676
HTEXT= option, see HEIGHT= option
HTML= option, see WEB= option
HTRACKS= option
   ACTNET statement (NETDRAW), 678, 694, 734
ID= option
   ACTNET statement (NETDRAW), 678, 694
IMAGEMAP= option
   PROC NETDRAW statement, 676
LAG= option
   ACTNET statement (NETDRAW), 679
LEFTRIGHT keyword
   CHILDORDER= option (NETDRAW), 678
LINEAR option
   ACTNET statement (NETDRAW), 679, 698, 742
LINEPRINTER option
   PROC NETDRAW statement, 676
LP option, see LINEPRINTER option
LREF= option
   ACTNET statement (NETDRAW), 686, 740
LREFBRK= option
   ACTNET statement (NETDRAW), 686, 742
LWCRIT= option
   ACTNET statement (NETDRAW), 686, 719
LWIDTH= option
   ACTNET statement (NETDRAW), 686, 717, 719
LWOUTLINE= option
   ACTNET statement (NETDRAW), 686
MAXEMPTY= option, see MAXNULLCOLUMN= option
MAXNCOL= option, see MAXNULLCOLUMN= option
MAXNULLCOLUMN= option
   ACTNET statement (NETDRAW), 679, 698
MAXZCOL= option, see MAXNULLCOLUMN= option
MININTERVAL= option
   ACTNET statement (NETDRAW), 679, 698, 746
MINUTE keyword
   MININTERVAL= option (NETDRAW), 679
MONTH keyword
   MININTERVAL= option (NETDRAW), 679
NAME= option
   ACTNET statement (NETDRAW), 686
NETDRAW procedure, 672
   ACTNET statement, 677
   PROC NETDRAW statement, 676
NLEVELSPERCOLUMN= option
   ACTNET statement (NETDRAW), 679, 698
NOARROWFILL option
   ACTNET statement (NETDRAW), 686, 749
NODEFID option
   ACTNET statement (NETDRAW), 680, 694, 726
NODETRACK option
   ACTNET statement (NETDRAW), 680
NODISPLAY option
   ACTNET statement (NETDRAW), 680
NODIMENSION option
   ACTNET statement (NETDRAW), 680, 694, 726
NONARROWFILL option
   ACTNET statement (NETDRAW), 686, 749
NONODETRACK option
   ACTNET statement (NETDRAW), 680
NONODETRACK option
   ACTNET statement (NETDRAW), 680
NONUMBER option, see NOPAGENUMBER option
NOPAGENUMBER option
   ACTNET statement (NETDRAW), 686
NOREPEATAxis option
   ACTNET statement (NETDRAW), 680
NOTIMEAXIS option
   ACTNET statement (NETDRAW), 680
NOVCENTER option
   ACTNET statement (NETDRAW), 687, 695, 738
NOZONEDESCR option, see NOZONELABEL option
NOZONELABEL option
   ACTNET statement (NETDRAW), 680
NPERCOL= option, see NLEVELSPERCOLUMN= option
NXNODEx= option
   ACTNET statement (NETDRAW), 687
NXPAGES= option, see HPAGES= option
NYNODEx= option
   ACTNET statement (NETDRAW), 687
NYPAGES= option, see VPAGES= option
OUT= option
   PROC NETDRAW statement, 677
PAGENUM option, see PAGENUMBER option

PAGENUMBER option
ACTNET statement (NETDRAW), 687

PAGES= option
ACTNET statement (NETDRAW), 680

PATTERN= option
ACTNET statement (NETDRAW), 683, 687, 700, 702, 705, 736

PCOMPRESS option
ACTNET statement (NETDRAW), 687, 722

PROC NETDRAW statement, see NETDRAW procedure
statement options, 676

QTR keyword
MININTERVAL= option (NETDRAW), 679
QUITMISSINGALIGN option
ACTNET statement (NETDRAW), 680

RECTILINEAR option
ACTNET statement (NETDRAW), 687, 760

REFBREAK option
ACTNET statement (NETDRAW), 680, 742

REFCHAR= option
ACTNET statement (NETDRAW), 683, 689

RESTRICTSEARCH option
ACTNET statement (NETDRAW), 681, 712

RIGHTLEFT keyword
CHILDORDER= option (NETDRAW), 678

ROTATE option
ACTNET statement (NETDRAW), 687, 701, 769, 771

ROTATETEXT option
ACTNET statement (NETDRAW), 688, 698, 701, 769, 771

RSEARCH option, see RESTRICTSEARCH option
RTEXT option, see ROTATETEXT option

SECOND keyword
MININTERVAL= option (NETDRAW), 679

SEPARATEARCS option
ACTNET statement (NETDRAW), 688, 695, 719

SEPARATESONS option
ACTNET statement (NETDRAW), 681, 701, 762

SF keyword
LAG variable (NETDRAW), 679

SHOWBREAK option
ACTNET statement (NETDRAW), 681, 740

SHOWSTATUS option
ACTNET statement (NETDRAW), 681, 695, 736

SPANNINGTREE option
ACTNET statement (NETDRAW), 681

SS keyword

LAG variable (NETDRAW), 679

SUCCESSOR= option
ACTNET statement (NETDRAW), 681, 712

TIMESCALE option
ACTNET statement (NETDRAW), 681, 698, 738

TOTDOWN keyword
CHILDORDER= option (NETDRAW), 678

TREE option
ACTNET statement (NETDRAW), 682, 700, 760

TREELAYOUT option, see TREE option

USEFORMAT option
ACTNET statement (NETDRAW), 682, 764

VBETWEEN= option, see YBETWEEN= option
VMARGIN= option
ACTNET statement (NETDRAW), 688, 738

VPAGES= option
ACTNET statement (NETDRAW), 688

VPOS= option, GOPTIONS statement
NETDRAW procedure, 705, 717, 725

VTRACKS= option
ACTNET statement (NETDRAW), 682, 694

WEB= option
ACTNET statement (NETDRAW), 688

WEEK keyword
MININTERVAL= option (NETDRAW), 679

XBETWEEN= option
ACTNET statement (NETDRAW), 682, 694, 732

YBETWEEN= option
ACTNET statement (NETDRAW), 682, 694, 726

YEAR keyword
MININTERVAL= option (NETDRAW), 679

ZONE= option
ACTNET statement (NETDRAW), 682, 700, 746

ZONEDESC option, see ZONELABEL option
ZONELABEL option
ACTNET statement (NETDRAW), 682

ZONELEVELD option, see ZONESPACE option
ZONESPACE option
ACTNET statement (NETDRAW), 683, 689, 700, 746

ZONESPACE option
ACTNET statement (NETDRAW), 682, 746