Statistics, Data Analysis, and Modeling

Multiproject Scheduling using SAS/OR® Software

Gehan A. Corea, SAS Institute Inc., Cary, NC

Abstract

This paper presents an overview of the multiproject capabilities in SAS/OR software. The new 6.11 features in the CPM, GANTT, and NETDRAW procedures, many of which facilitate multiproject scheduling, are introduced. This is followed by an illustration of how you would use these features in conjunction with the analysis and reporting tools available in the SAS® System to manage multiple projects, with special consideration given to the generation of resource utilization reports.

Introduction

The purpose of this paper is two-fold. Firstly, it introduces the new capabilities of the Project Management procedures in SAS/OR software. Secondly, and from a more practical perspective, it illustrates these features and the powerful reporting capabilities of the SAS System in a typical multiproject scenario faced by project managers. The scenario is one arising in functional organizations where much of the work effort is project oriented, but the management of the work effort must be correlated across projects (Levine 1993). The requirements of a project manager in such a situation are the capability to schedule multiple projects and report on the resulting schedules and their resource requirements, both by summary and by responsible function. This paper illustrates how you can use the SAS System for Project Management to satisfy these requirements.

The next section presents some of the new features in the SAS/OR Project Management procedures.

New in Release 6.11

Several new options have been added to the Project Management procedures in Release 6.11. For a detailed description of the new options, refer to SAS Software: Changes and Enhancements, Release 6.11. Some of the highlights are listed below by procedure.

CPM
- Multiproject scheduling capabilities
- WBS code generation
- Total resource usage reporting

GANTT
- Page control
- Bar pattern control at activity level
- Text color control at activity level

NETDRAW
- Page control
- Nonstandard precedence constraints
- Updated _ORNETDR macro variable

For further details on the preceding procedures, refer to SAS/OR Software: User’s Guide: Project Management, Version 6, First Edition and SAS Software: Changes and Enhancements, Release 6.10. The following section demonstrates some of the preceding features, as well as the powerful reporting capabilities available in the SAS System, by illustrating their application in a typical multiproject scenario (Levine 1993).

Multiproject Scheduling Scenario

You are the manager of the Engineering Department in a manufacturing firm, and your department is comprised of several divisions: Civil, Electrical, Environmental, Mechanical, etc. An enterprise wide project in the firm consists of several departmental projects possibly with interproject relationships. Each of these projects vie for the limited resources of your department. As manager of the Engineering Department, you are frequently subjected to the criticism of the different project leaders who accuse you of favoring one project over the other. In order to respond to these accusations as well as plan for future growth, you need to generate reports indicating how your personnel are being utilized across the various projects.

Table 1 through Table 4 define four basic projects that take place in the manufacturing firm which use the resources of your department. Each table contains the precedence relationships and resource requirements for the project.

<table>
<thead>
<tr>
<th>Table 1. Project 100</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Act</strong></td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>A6</td>
</tr>
<tr>
<td>A6</td>
</tr>
<tr>
<td>B6</td>
</tr>
<tr>
<td>C4</td>
</tr>
<tr>
<td>D10</td>
</tr>
<tr>
<td>E15</td>
</tr>
<tr>
<td>F12</td>
</tr>
<tr>
<td>G10</td>
</tr>
</tbody>
</table>

1353
All durations are in standard weekdays, and the resource requirements are given in terms of the rate of each Engineer type required throughout the duration of the activity. Notice that the activities corresponding to the project in Table 4 have nonstandard precedence relationships. The value 55 in the first row indicates a Start to Start lag of 10 days, which means that the successor A14 can begin 10 days after the activity A14 has begun.

Table 2. Project 120

<table>
<thead>
<tr>
<th>Act</th>
<th>Dur</th>
<th>Succ</th>
<th>Resource Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>A12</td>
<td>60</td>
<td>B10</td>
<td>0.5 0.25 0.25 0.25</td>
</tr>
<tr>
<td>B10</td>
<td>50</td>
<td>C5</td>
<td>0.5 0.25 0.25 0.5</td>
</tr>
<tr>
<td>C5</td>
<td>25</td>
<td>D20</td>
<td>0.5 0.25 0.5</td>
</tr>
<tr>
<td>D20</td>
<td>100</td>
<td>E5</td>
<td>0.5 0.25 0.25 0.25</td>
</tr>
<tr>
<td>E5</td>
<td>25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Project 130

<table>
<thead>
<tr>
<th>Act</th>
<th>Dur</th>
<th>Succ</th>
<th>Resource Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>A8</td>
<td>40</td>
<td>D16</td>
<td>0.5 1 0.25 0.5</td>
</tr>
<tr>
<td>B19</td>
<td>76</td>
<td>E5</td>
<td>0.75 0.5 0.25 0.5</td>
</tr>
<tr>
<td>B19</td>
<td>76</td>
<td>F11</td>
<td>1.0 1.0 0.5 1.5</td>
</tr>
<tr>
<td>D16</td>
<td>80</td>
<td>C4</td>
<td>0.75 0.5 0.25 0.5</td>
</tr>
<tr>
<td>E5</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F11</td>
<td>55</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Project 140

<table>
<thead>
<tr>
<th>Act</th>
<th>Dur</th>
<th>Succ</th>
<th>Lag</th>
<th>Resource Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>A14</td>
<td>70</td>
<td>B14</td>
<td>SS.10</td>
<td>0.25 0.5 0.5 0.5</td>
</tr>
<tr>
<td>B14</td>
<td>70</td>
<td>C14</td>
<td>SS.10</td>
<td>0.25 0.5 0.5</td>
</tr>
<tr>
<td>C14</td>
<td>70</td>
<td>D14</td>
<td>SS.10</td>
<td>0.25 0.5 0.5</td>
</tr>
<tr>
<td>D14</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Template Creation

The manufacturing firm has a large number of projects, each similar to one of the four described previously, which take place concurrently although possibly having different start dates. For this reason, it is convenient to set up project templates to ease the generation of additional projects. The following code illustrates the generation of projects PROJ100 and PROJ101 using templates.

```
data proj100t;
  input act: $ dur sucess: $ lag: $;
  cards;
  A6 30 B6 .5 0.5 .5 ;
  A6 30 B15 .  . . . ;
  B6 30 F12 .  . . 1.0 ;
  C4 20 D10 1.0 .25 ;
  D20 50 G10 .5 0.75 ;
  E15 75 G10 .5 .5 .5 ;
  F12 60 .  . . . 0.25 .5 ;
  run;
```

You first represent the data in Table 1 with the SAS data set PROJ100T. This is the only data set that you would explicitly create in order to generate projects similar to Project 100. The template extracts the data from PROJ100T. To enable you to distinguish between activities belonging to similar projects, the activity names are prefixed by the name of the project to which they belong. The ALDATE variable represents the date on which the project is scheduled to begin.

For the purpose of this example, suppose that you have one copy of each of the other three projects. You can similarly generate the data sets PROJ120, PROJ130, and PROJ140, which correspond to Projects 120, 130, and 140 as shown below.

```
data proj120t;
  set proj120t;
  label proj = 'Project';
  proc='P120'; aldate='18apr96'd;
  if act ne ' then act=proj||'||act;
  if succ ne ' then succ=proj||'||succ;
  run;
```

The five individual project data sets are now concatenated to form MULTPR, the multiproject data set. Thus far, all the precedence relationships are contained within each project.

Suppose you also have the restriction that activity C4 in PROJ101 cannot begin until ten weekdays after activity A4 in PROJ100 is completed. The following code illustrates how you can incorporate constraints of this nature during the creation of MULTPR.

```
data proj101t;
  set proj100t;
  label proj = 'Project';
  proc='P101'; aldate='11apr96'd;
  if act ne ' then act=proj||'||act;
  if succ ne ' then succ=proj||'||succ;
  run;
```

You first represent the data in Table 1 with the SAS data set PROJ100T. This is the only data set that you would explicitly create in order to generate projects similar to Project 100. The template extracts the data from PROJ100T. To enable you to distinguish between activities belonging to similar projects, the activity names are prefixed by the name of the project to which they belong. The ALDATE variable represents the date on which the project is scheduled to begin.

For the purpose of this example, suppose that you have one copy of each of the other three projects. You can similarly generate the data sets PROJ120, PROJ130, and PROJ140, which correspond to Projects 120, 130, and 140 as shown below.

```
data proj120t;
  set proj120t;
  label proj = 'Project';
  proc='P120'; aldate='18apr96'd;
  if act ne ' then act=proj||'||act;
  if succ ne ' then succ=proj||'||succ;
  run;
```

The five individual project data sets are now concatenated to form MULTPR, the multiproject data set. Thus far, all the precedence relationships are contained within each project.

Suppose you also have the restriction that activity C4 in PROJ101 cannot begin until ten weekdays after activity A4 in PROJ100 is completed. The following code illustrates how you can incorporate constraints of this nature during the creation of MULTPR.

```
data proj101t;
  set proj100t;
  label proj = 'Project';
  proc='P101'; aldate='11apr96'd;
  if act ne ' then act=proj||'||act;
  if succ ne ' then succ=proj||'||succ;
  run;
```

The five individual project data sets are now concatenated to form MULTPR, the multiproject data set. Thus far, all the precedence relationships are contained within each project.

Suppose you also have the restriction that activity C4 in PROJ101 cannot begin until ten weekdays after activity C4 in PROJ100 is completed. The following code illustrates how you can incorporate constraints of this nature during the creation of MULTPR.

```
data proj101t;
  set proj100t;
  label proj = 'Project';
  proc='P101'; aldate='11apr96'd;
  if act ne ' then act=proj||'||act;
  if succ ne ' then succ=proj||'||succ;
  run;
```

The five individual project data sets are now concatenated to form MULTPR, the multiproject data set. Thus far, all the precedence relationships are contained within each project.

Suppose you also have the restriction that activity C4 in PROJ101 cannot begin until ten weekdays after activity C4 in PROJ100 is completed. The following code illustrates how you can incorporate constraints of this nature during the creation of MULTPR.

```
data proj101t;
  set proj100t;
  label proj = 'Project';
  proc='P101'; aldate='11apr96'd;
  if act ne ' then act=proj||'||act;
  if succ ne ' then succ=proj||'||succ;
  run;
```

The five individual project data sets are now concatenated to form MULTPR, the multiproject data set. Thus far, all the precedence relationships are contained within each project.

Suppose you also have the restriction that activity C4 in PROJ101 cannot begin until ten weekdays after activity C4 in PROJ100 is completed. The following code illustrates how you can incorporate constraints of this nature during the creation of MULTPR.

```
data proj101t;
  set proj100t;
  label proj = 'Project';
  proc='P101'; aldate='11apr96'd;
  if act ne ' then act=proj||'||act;
  if succ ne ' then succ=proj||'||succ;
  run;
```

The five individual project data sets are now concatenated to form MULTPR, the multiproject data set. Thus far, all the precedence relationships are contained within each project.

Suppose you also have the restriction that activity C4 in PROJ101 cannot begin until ten weekdays after activity C4 in PROJ100 is completed. The following code illustrates how you can incorporate constraints of this nature during the creation of MULTPR.
You can also use XPROJMAN, the Graphical User Interface for Project Management with the SAS System, to create interproject relationships easily. Figure 1 shows a screen display from XPROJMAN showing the relationship between activities P100_C4 and P101_C4. The projects P100, P101, and P140 are shown exploded while projects P120 and P130 are shown in a collapsed state.

The following code illustrates how to use the NETDRAW procedure with the MULTPR data set to produce the Network Diagram in Figure 2, which shows the five individual projects and their activities and precedence relationships. The LAG= option in the ACTNET statement is used to accommodate the nonstandard precedence connections of project P140. Notice that the 'Start-to-Start' lags are represented by a connection originating from the left-hand side of the box corresponding to the activity. The ZONE= option groups the activities by project and separates projects with a horizontal line.

Output 1. Partial Listing of SCHED Data Set
Recall that the MULTPR data set does not have a separate observation for each of the individual projects. The ADDACT option specified in the PROC CPM statement requests that the procedure generate a record in the OUT= data set for each of the projects constituting the multiproject. This record contains schedule information and project duration as calculated by the procedure. The schedule data set generated by the CPM procedure contains two new variables when the PROJECT variable is specified that are very useful for reporting and filtering purposes. The PROJ.DUR variable contains the project duration for each parent activity, and it is set to missing for all leaf activities in the project. The PROJ.LEV variable specifies the depth of each activity from the root of the project hierarchy tree. A partial listing of the schedule data set is printed in Output 1.

You can produce a Gantt chart corresponding to this schedule by using the PROJ.LEV variable to indent the ID variables, as illustrated below. The resulting Gantt chart is displayed in Figure 3.

Notice that the ADDACT option in the PROC CPM statement produces separate observations for each subproject and appends them to the SCHED data set. For example, observation number 39 represents the multiproject in entirety and indicates a duration of 270 weekdays, and it is scheduled to finish on 16APR97. In order to sort the schedule by project, you need to use the ESO option in the PROJECT statement. This produces an index that can be used to order the schedule data set by E.START time within each project. Notice also that the late finish times for each subproject are the same. The backward pass, which computes the late start schedule, uses the early finish time of the master project as a starting point.

The following code determines the resource usage based on the early start schedule assuming that the only constraints are the time and precedence types. Specifying the ESO option in the PROJECT statement creates the ES_ASC variable in the OUT= data set for sorting purposes as described previously. The SEPCRIT option causes the late start schedule to be computed using separate critical paths for each project. This is in contrast to the default behavior of having one critical path for the entire project. Yet another possibility is the USEPROJDUR option, which computes the late start schedule based on the value of the duration variable for the subproject. The daily resource usage is stored in the RESUSG data set. The ESS option in the RESOURCE statement is used to request only an Early start usage profile. The schedule data set is then sorted by the ES_ASC variable, and a schedule for each of the five subprojects and the master project is listed in Output 2 by filtering on the PROJ.LEV variable. Notice the effect of the SEPCRIT option on the late start schedules.

Output 2. Schedule Summary Using SEPCRIT Option

Figure 3. Gantt Chart for Multiproject
Highlighting Summary Tasks

You can produce a Gantt chart highlighting the summary schedules by using the _PATIERN variable and the LABEL data set. The _PATIERN variable is a default numeric variable that you can use to override the default fill pattern for the schedule bars at the activity level. You can also use the PATTERN= option in the CHART statement to specify a different pattern variable. The value refers to the specifications in the generated PATTERN statements. For more information on using PATTERN statements, refer to SAS/GRAPH Software: Reference, Version 6, First Edition, Volume 1.

Multiproject Scheduling Example
Using the PATTERN Variable

The following code produces the Gantt chart in Figure 4 using a solid black fill pattern for the summary bars. The LABVAR=_PATIERN specification in the CHART statement is used to place the special symbols at the endpoints of the summary bars. Notice how the activities are sorted by early start time within each subproject.

```
pattern1 v=e e=black r=9;
pattern10 v=e e=black;
/* add a pattern variable */
data sched;
set sched;
  if proj_lev <= 1 then _patterns10;
run;

data labels;
  _pattern=10;
  _yoffset=1.0;
  _label='orfont'; _label='c';
  _label='2'; _xvar=e_start; output;
  _xvar='e_finish'; output;
run;
```

You can produce a plot of the resource requirements based on the early start schedule using the GPLOT procedure. For a detailed description of the GPLOT procedure, refer to SAS/GRAPH Software: Reference, Volume 2.

Multiproject Scheduling Example
Resource Usage based on Early Start Schedule

Scheduling with Resources

The Engineering department has a staff of 15 engineers, which is comprised of five civil, four electrical, two environmental, and four mechanical engineers. From the previous figure, it is evident that these values make the early schedule infeasible.

In order to schedule the project subject to these resource constraints, you invoke the CPM procedure with the RESIN= option and specify the resource availability data set. The WBS option in the PROJECT statement generates a WBS code for each activity in the project.

```
title2 'Resource Constrained Schedule';
proc cpm data=multpr out=schedc
  resinsav resourcsav
  interval=week
  addact;
  project proj / sepeart ess wbs;
  act act;
  dur dur;
  succ succ / lags;
  resource civ also env mech / periods;
  alidate aldata;
run;
```
The schedule for the subprojects is listed in Output 3. The S.START and S.FINISH variables give the resource constrained start and finish times for each subproject, and from the listing it is evident all the subprojects except PIOO have slipped due to resource constraints.

Output 3. Resource Constrained Schedule

<table>
<thead>
<tr>
<th>Activity</th>
<th>WBS</th>
<th>APR</th>
<th>MAY</th>
<th>JUN</th>
<th>JUL</th>
<th>AUG</th>
<th>SEP</th>
<th>OCT</th>
<th>NOV</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIOO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_01</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_02</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_03</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_05</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_06</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_07</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_08</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_09</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PIOO_10</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 6. Resource Constrained Schedule

The Gantt chart corresponding to this schedule is produced in Figure 6. The VDATES= option divides the Gantt chart into two pages in the vertical direction. The RBARHT= option specifies that the height for the resource constrained schedule bars be half the default height.

```
title b1.25 'multiproject scheduling' 'example';
title2 'resource constrained schedule';
proc gantt graphics data=schedc
 labdata=labl2sc;
 id actid wbs_code;
 chart type=compress
 nolegend
 mininterval=week
 mandate='01apr96' maxdate='01may97';
 label=proj
 vpages=2
 rbarcht=0.5
 patlevel=resource
 pagecum;
 run;
```

```
data actalgn;
 set schedc;
 aldate=s_start;
 altype="ms";
 keep proj act aldate acttype dur org ojiv alg ass;
 format aldate date7.;
 run;
```

```
%macro ensrg_agg(pname);
 proc cpm data=actalgn interval=weekday
 out=schedname resout=resource
 addact;
 where proj="p{name}";
 act act;
 dur dur;
 resource ojiv alg ass;
 aldate aldate;
 altype altype;
 run;
```

 Producing Tabular Resource Utilization Reports

For accounting purposes, suppose you wish to get a listing of the number of hours of support your department has provided for each of the projects broken down by Quarter, Month as well as a summary giving totals for each project. The RESOUT= data set, generated by the CPM procedure, gives the total usage and availability of each type of engineer by day. In order to obtain resource usage for a particular set of activities, you can invoke the CPM procedure on the selected set of activities and perform resource aggregation by using the schedule start time as a mandatory start time.
The preceding code illustrates this technique via the macro ENGR.AGG. Notice that the SUCI statement is not necessary since each of the activities has a mandatory alignment.

The following code illustrates the execution of the ENGR.AGG macro for each of the five subprojects and creation of the MULTUSG data set by concatenating the resulting resource usage data sets. The MULTUSG data set is next transposed using the TRANSPOSE procedure to create the NEWUSG data set. The PREFIX option specifies the prefix to use for constructing names for the transposed variables. For details on using the TRANSPOSE procedure, refer to SAS Procedures Guide: Version 6, Third Edition.

```
data multusg;
   set resp100 resp101 resp120 resp130 resp140;
run;
proc transpose data=multusg name=resource
cut=newusg(drop_label_) prefix=res;
    var $engrmmt
    run;
```

The NEWUSG data set contains a wealth of information for reporting purposes, and by using the NEWUSG data set with the TABULATE procedure, you can generate a wide spectrum of reports on resource utilization. Due to space considerations, this paper presents only the first page of each report. The following code, for example, produces the report in Output 4, which gives a monthly breakdown of Engineer days consumed by each project.

```
title2 'Monthly Usage of Engineers by Project';
proc tabulate data=newusg
   ordered data=multusg format=f7.2;
format _time_ monyy5.
   class proj _time_;
   table (_time_ all)
   resl*(proj sum='Total') / rts=14;
    var resl;
    label proj='Project';
    label resl='Usage in Engineer Days'
    _time_='Month';
    keylabel sum='';
run;
```

The class variables, which determine the categorization of observations, are PROJ and _TIME_ The analysis variable is RES1, which is a numeric variable that gives the resource usage for the time period defined by _TIME_ for each type of engineer. The TABLE statement describes the format of the table to be produced. The rows are defined by the _TIME_ variable, and the columns are defined by the crossing of the RES1 analysis variable with the PROJ class variable to produce a column for each project. For more details on using the TABULATE procedure, refer to the SAS Procedures Guide.

Output 4. Monthly Usage of Engineers by Project

```
+----------+----------+----------+----------+----------+----------+----------+
<table>
<thead>
<tr>
<th></th>
<th>Project</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P100</td>
<td>P101</td>
<td>P120</td>
<td>P130</td>
<td>P140</td>
<td>Total</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>APR26</td>
<td>52.251</td>
<td>23.001</td>
<td>13.251</td>
<td></td>
<td></td>
<td>84.501</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>MAY26</td>
<td>75.251</td>
<td>45.501</td>
<td>26.251</td>
<td>42.251</td>
<td></td>
<td>185.001</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>JUN26</td>
<td></td>
<td>82.501</td>
<td>21.001</td>
<td>78.501</td>
<td>21.001</td>
<td>155.001</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>JUL26</td>
<td>63.251</td>
<td>45.251</td>
<td>32.501</td>
<td>74.751</td>
<td>13.251</td>
<td>239.751</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>AUG26</td>
<td>59.501</td>
<td>61.501</td>
<td>33.001</td>
<td>76.251</td>
<td>22.251</td>
<td>232.251</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>SEP26</td>
<td>47.251</td>
<td>47.251</td>
<td>19.501</td>
<td>127.501</td>
<td>19.001</td>
<td>251.501</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>OCT26</td>
<td>25.501</td>
<td>49.751</td>
<td>18.751</td>
<td>125.251</td>
<td>55.001</td>
<td>302.251</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>NOV26</td>
<td></td>
<td>37.501</td>
<td>33.001</td>
<td>84.001</td>
<td>50.001</td>
<td>188.501</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>DEC26</td>
<td>4.251</td>
<td>32.251</td>
<td>23.251</td>
<td>42.251</td>
<td>23.501</td>
<td>95.501</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Proj</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>P100</td>
<td>7.501</td>
<td>23.001</td>
<td>43.251</td>
<td>57.501</td>
<td>19.501</td>
<td>102.751</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>P101</td>
<td></td>
<td>26.001</td>
<td>59.001</td>
<td>70.001</td>
<td></td>
<td>155.001</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>P120</td>
<td></td>
<td>21.401</td>
<td>52.001</td>
<td>73.501</td>
<td></td>
<td>147.001</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>P130</td>
<td></td>
<td></td>
<td>17.501</td>
<td>44.251</td>
<td></td>
<td>61.751</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>P140</td>
<td>1.501</td>
<td>21.501</td>
<td>22.001</td>
<td>84.501</td>
<td></td>
<td>130.501</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>RES1</td>
<td>699.001</td>
<td>404.001</td>
<td>322.501</td>
<td>754.251</td>
<td>350.001</td>
<td>2228.751</td>
</tr>
</tbody>
</table>
+----------+----------+----------+----------+----------+----------+----------+
```

Suppose you wish to produce a report showing the breakdown of the preceding figures across Engineer types. This is easily done by modifying the code as illustrated below.

```
proc format;
    value $engrmmt
        'CIVIL'='Civil'
        'ELEC'='Electrical'
        'ENV'='Environmental'
        'MECH'='Mechanical'
    run;
```
First, the FORMAT procedure is invoked to create a new format, $\text{ENGFRMT}$, for the RESOURCE variable. This helps improve the readability of the reports that you generate. Refer to the SAS Procedures Guide for details on the FORMAT procedure. Next, you define the RESOURCE variable as a class variable and cross it with the _TIME_ variable in the TABLE statement. The resulting report is presented in Output 5.

**Output 5. Monthly Usage of Engineer Types by Project**

In a similar fashion, it is possible to generate this type of report aggregating usage over any time period. The only change you need to make is in the _TIME_ variable format. For example, to generate a resource usage report aggregated by quarter, you simply change the _TIME_ format to yyq4. The following code generates the table in the next segment of code generates the table in Output 7, which summarizes the utilization of each division by project.

**Output 6. Quarterly Usage of Engineer Types by Project**

**Output 7. Total Usage of Engineer Types by Project**
Resource Breakdown Structure

Now suppose that you wish to generate a similar report for each division broken down by the engineers in the division. By way of example, this paper illustrates how you would generate the preceding reports for the Electrical Engineering division, which has four engineers: David, Katie, Michelle, and Ryan. A partial view of the resource hierarchy for the project is shown in Figure 7, which was produced using the NETDRAW procedure.

One way of scheduling the specific engineers is by using alternate resources when scheduling the multiproject. The four electrical engineers are added as alternate resources for the ELEC resource. The availability level of ELEC is set to zero and that of the individual engineers is set to one. The resource availability data set RESAVA is printed in Output 8. You schedule the projects as before with the exception that you now include the alternate resources in the RESOURCE statement as illustrated below.

```
data multpra;
  set multpra;
  David=.;
  Katie=.;
  Michelle=.;
  Ryan=.;
run;
```

Using a technique similar to that described for generating the previous reports, you can derive aggregate reports for each of the individual electrical engineers. Examples of quarterly usage and total usage reports for the four electrical engineers are presented in Output 9 and Output 10, respectively. Since the project is scheduled using alternate resources and the availability of ELEC is set to zero, the actual usage of electrical engineers is no longer given by the ELEC variable in the schedule data set. The usage of each alternate resource is given by the variable in the schedule data set that is named by prefixing the resource name with 'U'. When creating the ACTALGN data set, you regard the alternate resource usage variables as the resource requirement variables. The rest of the method is unchanged.

Finally, suppose you wish to generate a report giving the schedule for each engineer with respect to the five projects identifying the activities they have worked on. Using the schedule data set, you can easily generate a Gantt chart displaying such a schedule.

The following code produces the Gantt chart for Michelle shown in Figure 8. The schedule data set is first sorted by E_START time. The WHERE= option restricts the activities to those that Michelle works on. The LABELS data set specifies that the value of the UMICH.LE variable be printed left-justified at the S.START time for each activity (Y=1) at a depth of 3 cells. This translates to printing the rate at which Michelle works below the resource constrained schedule bar of each activity.

```
proc sort data=scheda;
  by s_start;
run;
```
Conclusions

This paper describes the new 6.11 features in the CPM, GANTT, and NETDRAW procedures of SAS/OR software and their application in a multiproject environment. The example illustrates several techniques which include the generation of subprojects via templates, scheduling of multiprojects, highlighting summary tasks in Gantt charts, creating a resource breakdown structure, and producing tabular resource utilization reports. It is evident that the variety of analysis and reporting tools available in the SAS System coupled with the wealth of data they generate make the SAS System for Project Management a very powerful tool in the management of multiple projects.

References


SAS, SAS/GRAPH and SAS/OR are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. * indicates USA registration.