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
- Updated _ORGANTT macro variable

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 subject 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 1. Project 100

			Resource Requirements			
Act	**Dur**	**Succ**	**Civ**	**Elec**	**Env**	**Mech**
A6	30	B6	0.5	0.5		0.5
A6	30	E15				
B6	30	F12				1.0
C4	20	D10	1.0		0.25	
D10	50	G10		0.5		0.75
E15	75	G10	0.5	0.5		0.5
F12	60			0.5	0.25	0.5
G10	50		0.5	0.5		

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 SS_10 in the first row indicates a Start to Start lag of 10 days, which means that the successor B14 can begin 10 days after the activity A14 has begun.

Table 2. Project 120

			Resource Requirements			
Act	Dur	Succ	Civ	Elec	Env	Mech
A12	60	B10	0.5	0.25	0.25	0.25
B10	50	C5	0.5	0.25	0.25	0.5
C5	25	D20	0.5	0.25		0.5
D20	100	E5	0.5	0.25		0.25
E5	25		0.5		0.25	0.5

Table 3. Project 130

			Resource Requirements			
Act	Dur	Succ	Civ	Elec	Env	Mech
A8	40	D16	0.5	1	0.25	0.5
B19	76	E5		0.75		0.5
B19	76	F11				
C4	20		1	0.5	0.5	1
D16	80	C4	0.75	0.5	0.25	0.5
E5	25	C4	1.0	1.0	0.5	1.5
F11	55		2.0	0.75	0.5	1.5

Table 4. Project 140

				Resource Requirements			
Act	Dur	Succ	Lag	Civ	Elec	Env	Mech
A14	70	B14	SS_10		0.25	0.5	0.5
B14	70	C14	SS_10	0.25		0.5	0.5
C14	70	D14	SS_10		0.25	0.5	0.5
D14	70			0.25		0.5	0.5

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 succ:$ lag:$
         civ elec env mech;
   cards;
   A6   30  B6   .  0.5  0.5  .     0.5
   A5   30  E15  .  .    .    .     .
   B6   30  F12  .  .    .    .     1.0
   C4   20  D10  .  1.0  .    0.25  .
   D10  50  G10  .  .    0.5  .     0.75
   E15  75  G10  .  0.5  0.5  .     0.5
   F12  60  .    .  .    0.5  0.25  0.5
   G10  50  .    .  0.5  0.5  .     .
run;
```

```
/* Generate projects P100 and P101 */
data proj100;
   set proj100t;
   label proj = 'Project';
   proj='P100'; aldate='04apr96'd;
   if act ne "" then act=proj||"_"||act;
   if succ ne "" then succ=proj||"_"||succ;
run;

data proj101;
   set proj100t;
   label proj = 'Project';
   proj='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 proj120;
   set proj120t;
   label proj = 'Project';
   proj='P120'; aldate='18apr96'd;
   if act ne "" then act=proj||"_"||act;
   if succ ne "" then succ=proj||"_"||succ;
run;

data proj130;
   set proj130t;
   label proj = 'Project';
   proj='P130'; aldate='16may96'd;
   if act ne "" then act=proj||"_"||act;
   if succ ne "" then succ=proj||"_"||succ;
run;

data proj140;
   set proj140t;
   label proj = 'Project';
   proj='P140'; aldate='05sep96'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 multpr;
   set proj100 proj101 proj120 proj130 proj140;
   output;
   /* add interproject constraints */
   if act='P100_C4' then do;
      succ='P101_C4';
      lag='FS_10';
      output;
   end;
run;
```

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.

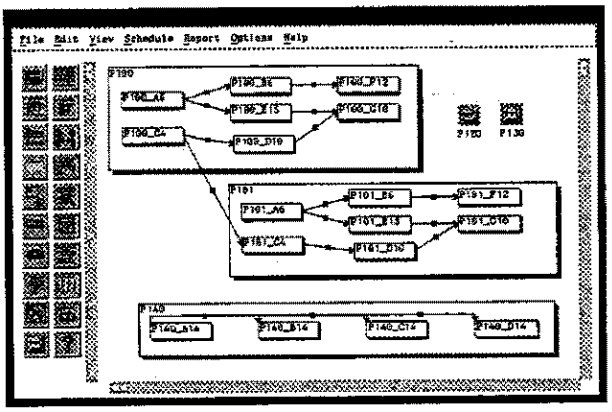


Figure 1. Using XPROJMAN to Create Interproject Relationships

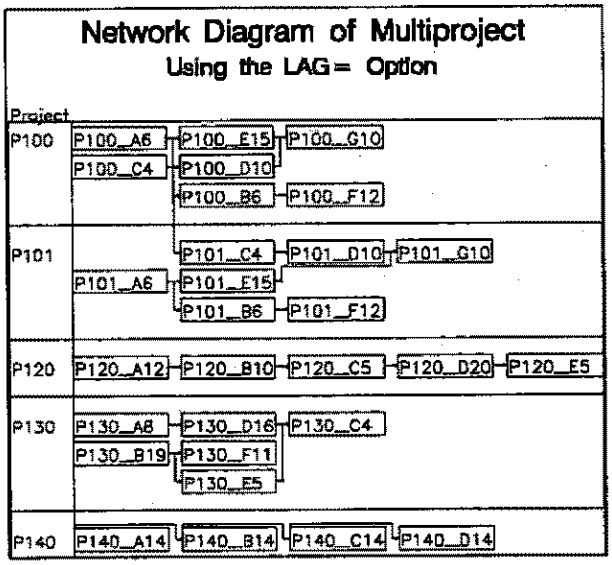


Figure 2. Network Diagram for Multiproject

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.

```
title h=1.25 'Network Diagram of '
             'Multiproject';
title2 'Using the LAG= Option';
proc netdraw data=multpr graphics;
   actnet / act=act succ=succ pcompress
            font=simplex
            zone=proj zonespace
            lag=(lag);
run;
```

Scheduling the Multiproject

In order to schedule the multiproject subject to time and precedence constraints, you invoke the CPM procedure with the PROJECT statement and specify the variable in the Activity data set that identifies the project to which an activity belongs. This variable defines the project hierarchy to the CPM procedure.

```
proc cpm data=multpr out=sched
      date='04Apr96'd interval=weekday
      addact;
   project proj;
   act act;
   dur dur;
   succ succ / lag=lag;
   aligndate aldate;
run;
```

Output 1. Partial Listing of SCHED Data Set

Schedule data set

O	A		D	F	P	P			R		L
B	C		U	R	R	R	E	L	E	L	A
S	T		R	J	O	O	_	A	_	A	G
					J	J	S	I	S	I	F
					_	_	T	N	T	N	I
					D	L	A	I	A	I	S
					U	S	R	S	R	S	E
					R	V	T	E	T	E	
20	P120_C5	25	P120	.	2	19SEP96	23OCT96	19SEP96	23OCT96		
21	P120_D20	100	P120	.	2	24OCT96	12MAR97	24OCT96	12MAR97		
22	P120_E5	25	P120	.	2	13MAR97	16APR97	13MAR97	16APR97		
23	P130_A8	40	P130	.	2	16MAY96	10JUL96	03OCT96	27NOV96		
24	P130_B19	76	P130	.	2	16MAY96	29AUG96	16OCT96	29JAN97		
25	P130_B19	76	P130	.	2	16MAY96	29AUG96	16OCT96	29JAN97		
26	P130_C4	20	P130	.	2	31OCT96	27NOV96	20MAR97	16APR97		
27	P130_D16	80	P130	.	2	11JUL96	30OCT96	28NOV96	19MAR97		
28	P130_E5	25	P130	.	2	30AUG96	03OCT96	13FEB97	19MAR97		
29	P130_F11	55	P130	.	2	30AUG96	14NOV96	30JAN97	16APR97		
30	P140_A14	70	P140	.	2	05SEP96	11DEC96	28NOV96	05MAR97		
31	P140_B14	70	P140	.	2	19SEP96	25DEC96	12DEC96	19MAR97		
32	P140_C14	70	P140	.	2	03OCT96	08JAN97	26DEC96	02APR97		
33	P140_D14	70	P140	.	2	17OCT96	22JAN97	09JAN97	16APR97		
34	P100	.		155	1	04APR96	06NOV96	12SEP96	16APR97		
35	P101	.		158	1	11APR96	13NOV96	12SEP96	16APR97		
36	P120	.		260	1	18APR96	16APR97	18APR96	16APR97		
37	P130	.		140	1	16MAY96	27NOV96	03OCT96	16APR97		
38	P140	.		100	1	05SEP96	22JAN97	28NOV96	16APR97		
39		.		270	0	04APR96	16APR97	18APR96	16APR97		

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.

```
pattern1  v=e c=black r=9;

data schedind;
   format actid $12.;
   set sched;
   if proj_lev !0 then do;
      actid=act;
      do i=1 to  proj_lev-1; /* indent actid */
         actid = "  " || actid;
      end;
   end;
run;

title h=1.25 'Multiproject Scheduling Example';
title2 'Time Constrained Schedule';
proc gantt graphics data=schedind;
   id proj actid;
   chart /pcompress nolegend nojobnum
         mininterval=week;
run;
```

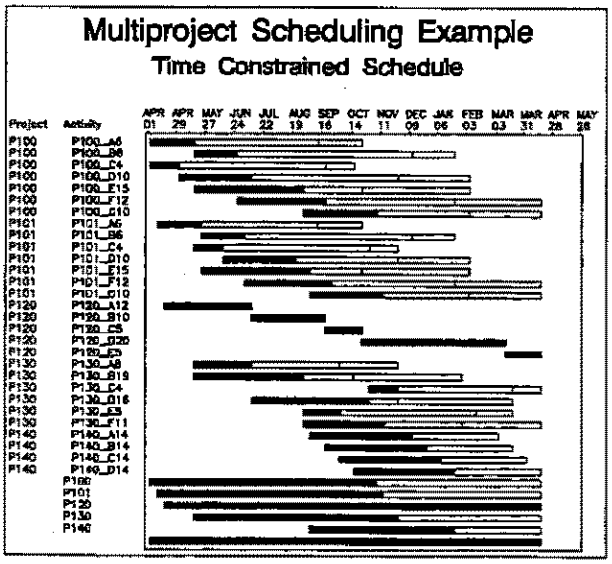


Figure 3. Gantt Chart for Multiproject

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 its 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.

```
title h=1.25 'Multiproject Scheduling '
             'Example';
title2 'Scheduling with the SEPCRIT option';
proc cpm data=multpr out=sched resout=resusg
     date='04Apr96'd interval=weekday addact;
   project proj / sepcrit eso;
   act act;
   dur dur;
   succ succ / lag=lag;
   resource civ elec env mech / ess;
   aligndate aldate;
run;

proc sort data=sched;
   by es_asc;
run;

proc print data=sched;
   where proj_lev<=1;
run;
```

Output 2. Schedule Summary Using SEPCRIT Option

Multiproject Scheduling Example
Scheduling with the SEPCRIT option

OBS	PROJ_DUR	PROJ_LEV	ACT	E_START	E_FINISH	L_START	L_FINISH
1	270	0		04APR96	16APR97	04APR96	16APR97
2	155	1	P100	04APR96	06NOV96	04APR96	06NOV96
12	155	1	P101	11APR96	13NOV96	11APR96	13NOV96
21	260	1	P120	18APR96	16APR97	18APR96	16APR97
27	140	1	P130	16MAY96	27NOV96	16MAY96	27NOV96
35	100	1	P140	05SEP96	22JAN97	05SEP96	22JAN97

Highlighting Summary Tasks

You can produce a Gantt chart highlighting the summary schedules by using the _PATTERN variable and the LABEL data set. The _PATTERN 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.*

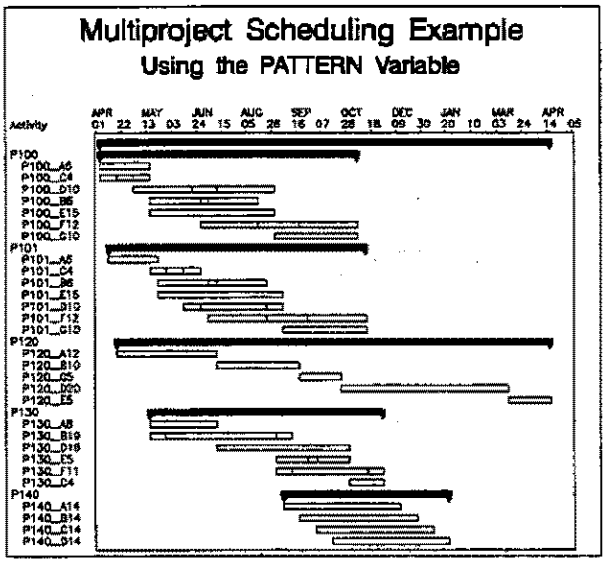


Figure 4. Gantt Chart Using the _PATTERN Variable

The following code produces the Gantt chart in Figure 4 using a solid black fill pattern for the summary bars. The LABVAR=_PATTERN 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 c=black r=9;
pattern10 v=s c=black;

/* add a pattern variable */
data schedp;
   set sched;
   if proj_lev <= 1 then _pattern=10;
run;

data labels;
   _pattern=10;
   _yoffset=1.0;
   _flabel='orfont'; _jlabel='c';
   _label='Z';
   _xvar='e_start '; output;
   _xvar='e_finish'; output;
run;

title h=1.25 'Multiproject Scheduling Example';
title2 'Using the PATTERN Variable';
```

```
proc gantt graphics data=schedp
           labdata=labels;
   id proj act;
   chart /pcompress nolegend nojobnum
          mininterval=week
          mindate='01apr96'd maxdate='01may97'd
          labvar=_pattern;
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.*

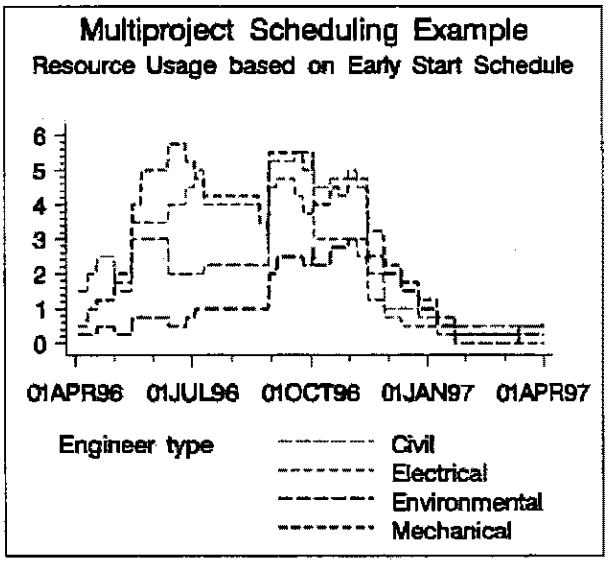


Figure 5. Early Start Resource Usage Profile

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
          resin=resav resout=resusg
          interval=weekday date='04Apr96'd
          addact;
   project proj / sepcrit sso wbs;
   act act;
   dur dur;
   succ succ / lag=lag;
   resource civ elec env mech / period=par;
   alignDate aldate;
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 P100 have slipped due to resource constraints.

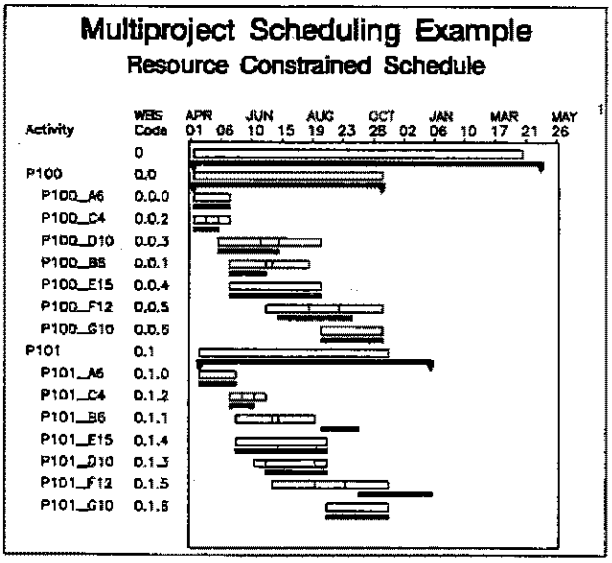
Output 3. Resource Constrained Schedule

Multiproject Scheduling Example
Resource Constrained Schedule

O	P D	P L	P A	S A	S R	E A	E R	L A	L R
	R U	R E		_ I	F _	_ I	F _	_ I	F _
	O R	O C		S I	I S	S I	I S	S I	I S
	J _	J _		T N	M T	N T	N T	T N	N T
1	286	0		04APR96	08MAY97	04APR96	16APR97	04APR96	16APR97
2	155	1	P100	04APR96	06NOV96	04APR96	05NOV96	04APR96	06NOV96
10	191	1	P101	11APR96	02JAN97	11APR96	13NOV96	11APR96	13NOV96
18	276	1	P120	18APR96	06MAY97	18APR96	16APR97	18APR96	16APR97
24	180	1	P130	16MAY96	22JAN97	16MAY96	27NOV96	16MAY96	27NOV96
31	150	1	P140	19SEP96	16APR97	05SEP96	22JAN97	05SEP96	22JAN97

The Gantt chart corresponding to this schedule is produced in Figure 6. The VPAGES= 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 h=1.25 'Multiproject Scheduling '
             'Example';
title2 'Resource Constrained Schedule';
proc gantt graphics data=schedc
           labdata=labelsc;
   id actid wbs_code;
   chart /pcompress nolegend nojobnum
         mininterval=week
         mindate='01apr96'd maxdate='01may97'd
         labvar=proj_lev
         vpages=2
         rbarht=0.5
         patlevel=resource
         pagenum;
run;
```



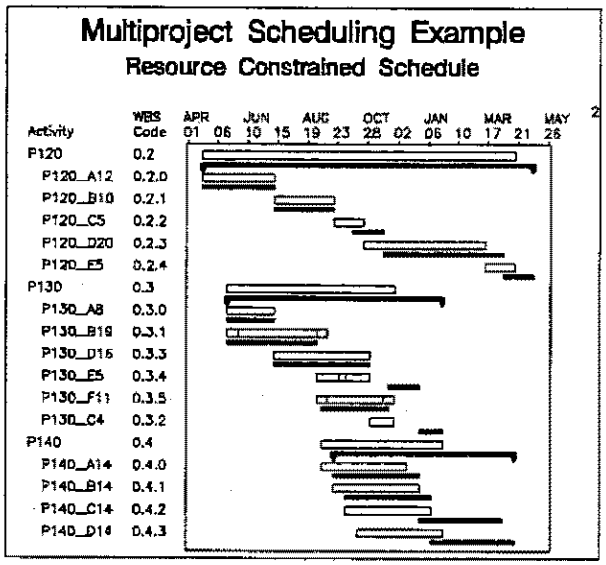


Figure 6. Resource Constrained Schedule

The PATLEVEL= option restricts application of the PATTERN variable to only the resource constrained schedule. By default it is applied to all the schedule bars corresponding to the activity. The LABELSC data set differs from LABELS in that the value of the YOFFSET variable is increased to correspond to the resource constrained schedule. The WBS code is printed as an ID variable.

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.

```
data actalgn;
   set schedc;
   aldate=s_start;
   altype="MS";
   keep proj act aldate altype dur
        civ elec env mech;
   format aldate date7.;
run;

%macro engr_agg(pname);
   proc cpm data=actalgn interval=weekday
        out=schd&pname resout=res&pname
        addact;
      where proj="&pname";
      act act;
      dur dur;
      resource civ elec env mech / ess;
      aligndate aldate;
      aligntype altype;
   run;
```

```
    data res&pname;
       set res&pname;
       proj="&pname";
    run;
%mend;
```

The preceding code illustrates this technique via the macro ENGR_AGG. Notice that the SUCC statement is not necessary since each of the activities has a mandatory align date.

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.*

```
%engr_agg(P100);
%engr_agg(P101);
%engr_agg(P120);
%engr_agg(P130);
%engr_agg(P140);

data multusg;
   set resp100 resp101 resp120 resp130 resp140;
run;

proc transpose data=multusg name=resource
     out=newusg(drop=_label_) prefix=res;
   by proj _time_;
   var eciv eelec eenv emech;
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
              order=data format=f7.2;
   format _time_ monyy5.;
   class proj _time_ ;
   table (_time_ all),
         res1*(proj sum='Total') / rts=14;
   var res1;
   label proj='Project'
         res1='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

Multiproject Scheduling Example
Monthly Usage of Engineers by Project

	Usage in Engineer Days					
	Project					
	P100	P101	P120	P130	P140	Total
Month						
APR96	52.25	21.00	11.25			84.50
MAY96	75.25	49.50	20.75	42.00		195.50
JUN96	73.00	42.50	25.00	70.00		210.50
JUL96	63.25	63.25	32.50	76.75		235.75
AUG96	59.50	61.50	33.00	70.25		224.25
SEP96	47.25	47.25	19.50	127.50	10.00	251.50
OCT96	25.50	49.75	18.75	153.25	55.00	302.25
NOV96	4.00	35.25	23.50	94.50	52.50	209.75
DEC96		27.50	22.00	84.00	55.00	188.50
JAN97		2.50	23.00	48.00	57.50	131.00
FEB97			20.00		50.00	70.00
MAR97			21.00		52.50	73.50
APR97			26.75		17.50	44.25
MAY97			7.50			7.50
ALL	400.00	400.00	312.50	766.25	350.00	2228.75

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 $engrfmt
   'ECIV'='Civil'
   'EELEC'='Electrical'
   'EENV'='Environmental'
   'EMECH'='Mechanical';
run;

title2 'Monthly Usage of Engineer Types by '
       'Project';
proc tabulate data=newusg
              order=data format=f7.2;
   format _time_ monyy5. resource $engrfmt.;
   class proj _time_ resource;
   table resource*(_time_ all),
         res1*(proj sum='Total') / rts=14;
   var res1;
   label proj='Project'
         res1='Usage in Engineer Days'
         _time_='Month' resource='Division';
   keylabel sum = ' ';
run;
```

First, the FORMAT procedure is invoked to create a new format, $ENGRFMT, 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

Multiproject Scheduling Example
Monthly Usage of Engineer Types by Project

		Usage in Engineer Days					
		Project					
		P100	P101	P120	P130	P140	Total
Division	Month						
Civil	APR96	28.50	7.00	4.50			40.00
	MAY96	12.50	23.50	11.50	6.00		53.50
	JUN96	10.00	18.00	10.00	10.00		48.00
	JUL96	11.50	11.50	11.50	15.25		49.75
	AUG96	11.00	11.00	11.00	16.50		49.50
	SEP96	10.50	10.50	6.50	51.75	0.00	79.25
	OCT96	11.50	11.50	7.50	62.50	5.25	98.25
	NOV96	2.00	4.50	10.50	35.00	5.25	57.25
	DEC96		0.00	11.00	22.00	5.50	38.50
	JAN97		0.00	11.50	16.00	5.75	33.25
	FEB97			10.00		5.00	15.00
	MAR97			10.50		5.25	15.75
	APR97			11.00		3.00	14.00
	MAY97			3.00			3.00
	ALL	97.50	97.50	130.00	235.00	35.00	595.00

(CONTINUED)

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 `monyy5.` format to `yyq4`. The following code generates the table in Output 6, which reports the quarterly usage by project broken down by type.

```
title2 'Quarterly Usage of Engineer Types '
       'by Project';
proc tabulate data=newusg
              order=data format=f8.2;
   format _time_ yyq4. resource $engrfmt.;
   class proj _time_ resource;
   table proj*(_time_ all) all,
         res1*(resource=' ' sum='Total')
         / rts=17;
   var res1;
   label proj='Project'
         res1='Usage in Engineer Days'
         _time_='Quarter';
   keylabel sum = ' ';
run;
```

Output 6. Quarterly Usage of Engineer Types by Project

Multiproject Scheduling Example
Quarterly Usage of Engineer Types by Project

		Usage in Engineer Days				
		Civil	Electrical	Environmental	Mechanical	Total
Project	Quarter					
P100	96Q2	51.00	52.00	5.00	92.50	200.50
	96Q3	33.00	66.00	14.50	56.50	170.00
	96Q4	13.50	14.50	0.50	1.00	29.50
	ALL	97.50	132.50	20.00	150.00	400.00
P101	Quarter					
	96Q2	48.50	29.50	5.00	30.00	113.00
	96Q3	33.00	57.00	0.00	82.00	172.00
	96Q4	16.00	45.00	14.50	37.00	112.50
	97Q1	0.00	1.00	0.50	1.00	2.50
	ALL	97.50	132.50	20.00	150.00	400.00
P120	Quarter					
	96Q2	26.00	13.00	13.00	13.00	65.00
	96Q3	29.00	14.50	14.50	27.00	85.00
	96Q4	29.00	14.50	0.00	20.75	64.25
	97Q1	32.00	16.00	0.00	16.00	64.00
	97Q2	14.00	0.75	6.25	13.25	34.25
	ALL	130.00	58.75	33.75	90.00	312.50

(CONTINUED)

The next segment of code generates the table in Output 7, which summarizes the utilization of each division by project.

Output 7. Total Usage of Engineer Types by Project

Multiproject Scheduling Example
Total Usage of Engineer Types by Project

	Usage in Engineer Days					
	Project					
	P100	P101	P120	P130	P140	Total
Division						
Civil	97.50	97.50	130.00	235.00	35.00	595.00
Electrical	132.50	132.50	58.75	213.25	35.00	572.00
Environmental	20.00	20.00	33.75	80.00	140.00	293.75
Mechanical	150.00	150.00	90.00	238.00	140.00	768.00
ALL	400.00	400.00	312.50	766.25	350.00	2228.75

```
title2 'Total Usage of Engineer Types by '
       'Project';
proc tabulate data=newusg
              order=data format=f7.2;
   format resource $engrfmt.;
   class proj resource;
   table resource all, resl*(proj sum='Total');
   var resl;
   label proj='Project'
         resl='Usage in Engineer Days'
         resource='Division';
   keylabel sum = ' ';
run;
```

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.

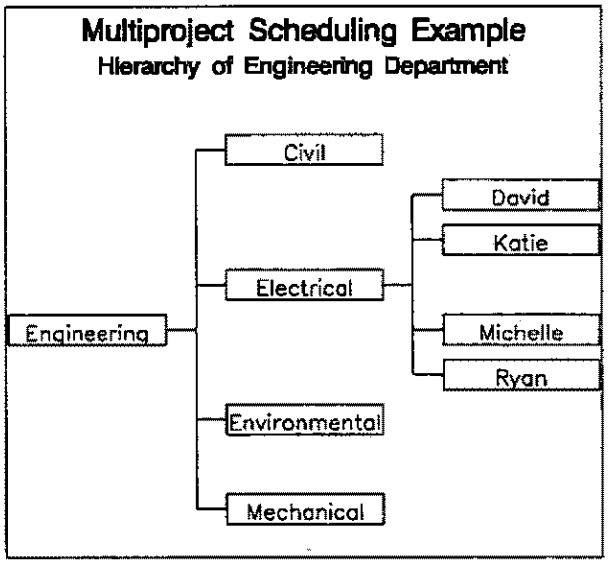


Figure 7. Partial View of Resource Breakdown Structure

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.

Output 8. RESAVA Data Set

Resource Availability Data Set
Defining a Resource Hierarchy

		C	E								M	
		E	N								I	
		S	S						D	K	C	
		T	E				E		M	A	A	E	R
O		Y	A		P	C	L	E	E	V	T	L	Y
B		P	M		N	I	E	N	C	I	I	L	A
S		E	E		R	V	C	V	K	D	E	E	N
1	alternate	elec			.	.	1	.	.	1	1	1	1
2	resLevel			04APR96	5	0	2	4	1	1	1	1	1

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

title h=1.25 'Multiproject Scheduling Example';
title2 'Scheduling with Alternate Resources';
proc cpm data=multpra out=scheda
     resin=resava resout=resusga
     date='04Apr96'd interval=weekday addact;
   project proj / sepcrit eso;
   act act;
   dur dur;
   succ succ / lag=lag;
   resource civ elec env mech
            David Katie Michelle Ryan
            / period=per obstype=obstype
              resid=resname;
   alignDate aldate;
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 ones that Michelle works on. The LABELS data set specifies that the value of the UMICHLLE 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;
```

```
data labels; /* prints rate below bar */
   _y=-1;
   _xvar='s_start';
   _lvar="uMichlle";
   _jlabel='l';
   _yoffset=3;
run;

title h=1.25 'Multiproject Scheduling Example';
title2 'Schedule for Michelle';
proc gantt graphics
   data=scheda (where=( uMichlle ^= .))
   labdata=labels;
   id act s_start s_finish;
   chart / pcompress mininterval=month
           ref='01apr96'd to '01jun97'd
            by month
           ebarht=.5 rbarht=.75;
run;
```

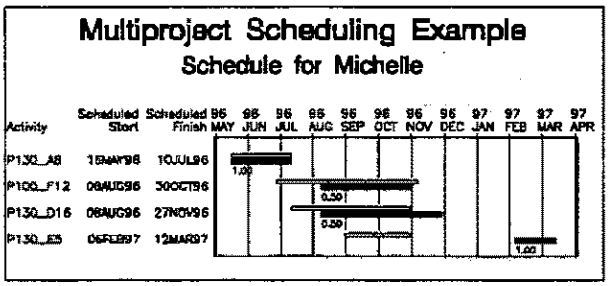


Figure 8. Gantt Chart Showing Schedule for Michelle

Output 9. Quarterly Usage of Electrical Engineers

Multiproject Scheduling Example
Quarterly Usage of Electrical Engineers by Project

		Usage in Engineer Days				
		David	Katie	Michelle	Ryan	Total
Project	Quarter					
P100	96Q2	31.00	21.00	0.00	0.00	52.00
	96Q3	21.50	4.00	19.00	11.00	55.50
	96Q4	0.00	0.00	11.00	14.00	25.00
	ALL	52.50	25.00	30.00	25.00	132.50
P101	Quarter					
	96Q2	28.50	0.00	0.00	0.00	28.50
	96Q3	35.50	29.00	0.00	0.00	64.50
	96Q4	18.50	21.00	0.00	0.00	39.50
	ALL	82.50	50.00	0.00	0.00	132.50
P120	Quarter					
	96Q2	0.00	13.00	0.00	0.00	13.00
	96Q3	2.00	14.50	0.00	0.00	16.50
	96Q4	16.50	0.00	0.00	0.00	16.50
	97Q1	12.75	0.00	0.00	0.00	12.75
	97Q2	0.00	0.00	0.00	0.00	0.00
	ALL	31.25	27.50	0.00	0.00	58.75

(CONTINUED)

Output 10. Total Usage of Electrical Engineers

Multiproject Scheduling Example
Total Usage of Electrical Engineers by Project

	Usage in Engineer Days					
	Project					
	P100	P101	P120	P130	P140	Total
Name						
David	52.50	82.50	31.25	10.00	17.50	193.75
Katie	25.00	50.00	27.50	41.25	17.50	161.25
Michelle	30.00	0.00	0.00	105.00	0.00	135.00
Ryan	25.00	0.00	0.00	57.00	0.00	82.00
ALL	132.50	132.50	58.75	213.25	35.00	572.00

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

Levine, H. A. (1993), "The Truth About Multiproject Scheduling: Now It Can Be Told," *PM Network*, 1, 22-26.

SAS Institute Inc. (1990), *SAS/GRAPH Software: Reference, Version 6, First Edition, Volume 1*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/GRAPH Software: Reference, Version 6, First Edition, Volume 2*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS Procedures Guide: Version 6, Third Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1993), *SAS/OR Software: User's Guide: Project Management, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1994), *SAS Software: Changes and Enhancements, Release 6.10*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1995), *SAS Software: Changes and Enhancements, Release 6.11*, Cary, NC: SAS Institute Inc.