

Managing A Project Schedule; SAS Can Help

Hui-Ping Chen, Myriad Pharmaceutical Inc., Salt Lake City, UT

ABSTRACT

It is easier to have “all” project plans displayed while trying to schedule workloads and/or reallocate resources. A Graphic schedule displays complex timelines in order and helps to rearrange plans and/or resources in order to complete the project(s) on time (especially when some timelines overlap). Microsoft Office Project’s “Gantt Chart” is one of the more powerful planning and scheduling tools. The chart has a number of visual functions to keep team-members updated. In addition, a Gantt chart may also be created in Microsoft Office Excel with simple functions. For those companies that do not purchase Microsoft Project or other Gantt chart software, we present an alternative option, which is SAS® procedure, PROC GANTT. The procedure allows the user to simply input schedules into SAS DATA step or Microsoft Excel.

INTRODUCTION

A Gantt chart is a bar chart that illustrates project timelines. The chart displays the start and end dates for each task and/or individual. It is easy to track schedules visually due to the fact that Gantt chart is a graphical planning tool that displays the duration of tasks against the progression of time. The chart is very efficient in monitoring multiple tasks at the same time, especially when work loads are highly overlapped. Gantt chart provides additional information as well. For example, how to reallocate resources in order to have all projects finished on time. A Gantt chart is also able to generate a logic schedule plan in order to determine which projects should have priority over other projects – and in turn which project is to follow. To start, prepare a schedule dataset that includes starting and ending dates and/or duration days. The next step is to decide what information to display in the Gantt chart.

Similar to other graphic procedures, annotation of text strings or symbols onto a Gantt chart can be achieved. However, if only the annotation of text strings or symbols is required, Automatic Text Annotation facility is better choice than Annotation facility for Gantt chart since it is built into PROC GANTT. LABDATA = can be used to link the label dataset created for annotating, which is similar to the ANNO = option.

The following is an example of project schedules and a simple illustration of PROC GANTT. Suppose there are 5 projects, all of which need to start programming tasks 2 months prior to data lock. The allocation of resources to the project tasks according to priority is required.

EXAMPLE 1: PROGRAMMING BEGINS 2 MONTHS PRIOR TO DATA LOCK

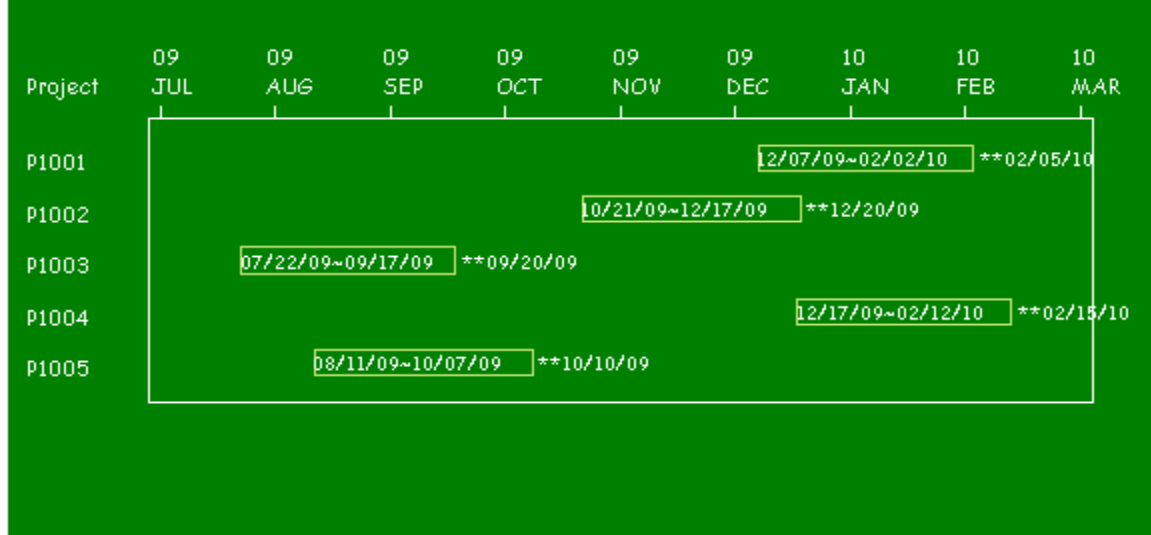
Project Schedule

Project	SAP	FPV	LPV	Lock	Programmer
P1001	2009/3/15	7/21/2009	10/30/2009	2/5/2010	A
P1002	2/1/2009	3/15/2009	9/15/2009	12/20/2009	B
P1003	1/15/2009	2/25/2009	6/20/2009	9/20/2009	C
P1004	3/10/2009	5/15/2009	11/1/2009	2/15/2010	C
P1005	2008/10/10	2009/1/15	2009/7/20	2009/10/10	A

Note: SAP = Statistical Analysis Plan; FPV = First Patient Visit; LPV = Last Patient Visit; Lock = Database Lock.

It is cumbersome to arrange the priority order of projects via a schedule such as the one above - especially if the completion date for program is not displayed. On the other hand, this is not an issue when using the SAS program to derive due date variables. In Example 1 we demonstrate how easy to apply PROC GANTT to our schedule data and display a visual graph. First, starting and ending dates need to be determined based on “requirements” that programming should **start** 2 months prior to data lock (variable: START) and be **completed** 3 days prior to data lock (variable: END).

Example 1: Start Programming 2 Months Prior to Data Lock Work Loads from 07/2009 to 03/2010



WHAT THE GANTT CHART DISPLAYS (EXAMPLE 1)

From this Gantt chart, we know there are **2 peaks** of workloads highly distributed in 2009 and 2010 – **the first** from early August 2009 to the end of September 2009 and **the second** from the middle of December 2009 to early February 2010. The implication is that more resources will be needed during these 2 peaks to implement the programming tasks on time. However, before programming is started on projects 1001, 1002, and 1004, all resources may be allocated to projects 1003 and 1005. The order of projects should be 1003, 1005, 1002, 1001, and 1004. At this point, there are 3 programmers, A, B, and C. Given that A is going to work on project 1003 while B works on project 1005, C is able to support validation on 1003 while B supports A on programming for 2 to 3 weeks. In turn, A performs validation on project 1005 after finishing programming on 1003.

PROC GANTT FOR EXAMPLE 1

```
PATTERN1 V = EMPTY C = LIY REPEAT = 10; ----- 1
PROC GANTT DATA = STUDY LABDATA = LABEL; ----- 2
  CHART / AS = START AF = END NOLEGEND NOJOBNUM ----- 3
    HEIGHT = 1.5 CAXIS = WHITE;
  ID PROJECT; ----- 4
RUN;
QUIT;
```

- 1: You can choose V = Solid or Empty to display annotated text strings. REPEAT = means to repeat V = EMPTY option 10 times. If you set up REPEAT = 3, the volume for first 3 projects will be empty, but with default solid color for the rest of projects.
- 2: LABDATA = is a data to label the text strings annotated into the graphs. In this graph we annotated 2 different text strings; one is to label duration of programming tasks (inside the empty bar), the other one is to label data lock date (outside the empty bar). Those variables beginning with underscore are reserved by SAS.

```
DATA LABEL;
  FORMAT _LVAR $10.; ----- a
  _Y = -1; ----- b
  _LVAR = "TEXT"; ----- c
  _XVAR = "START"; ----- d
```

```

_CLABEL = "WHITE"; ----- e
_HTEXT = .7;
_YOFFSET = .8; OUTPUT; ----- f
_LVAR = "LOCKDATE"; ----- g
_XVAR = "LOCK"; OUTPUT; ----- h
RUN;

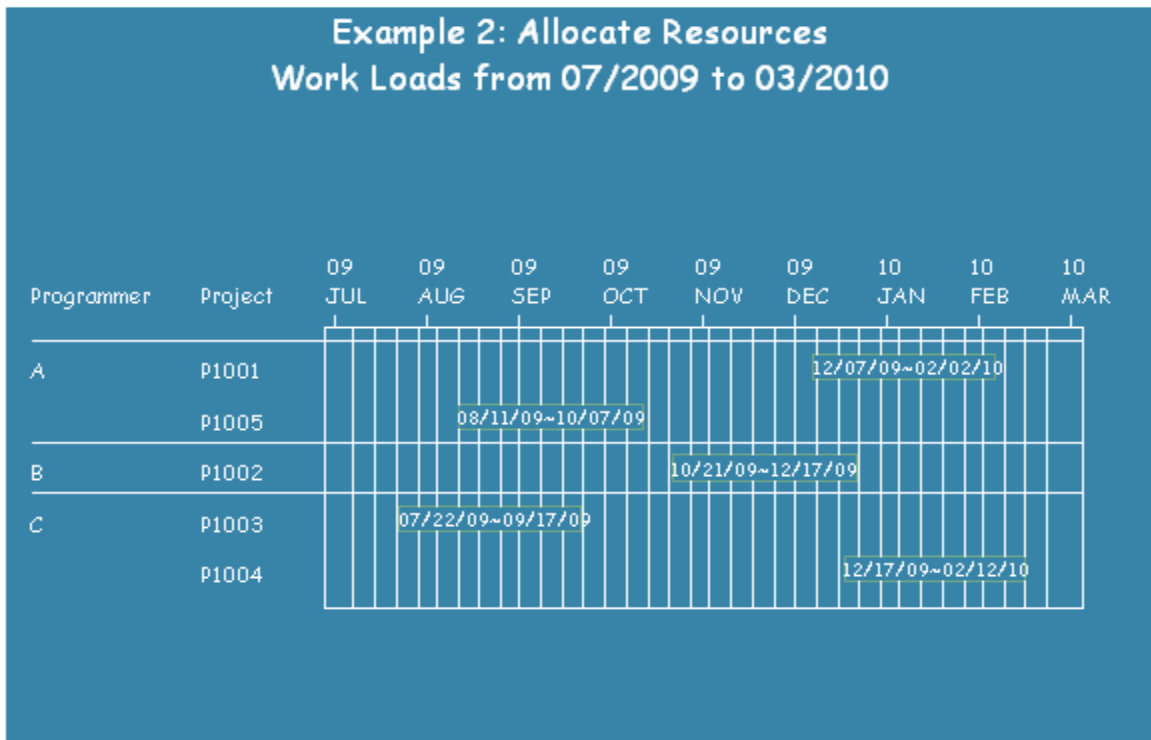
```

- a** it is important to ensure that the longer labeling variable name is not truncated since 2 text strings are annotated in the graph.
- b** `_Y` determines the vertical position of text strings while `_X` is for horizontal position. If the coordinate system is based on real data, `_Y` can be a positive number; otherwise, `_Y` should be set up to -1.
- c** **g** mean which variable is desired for labeling.
- d** **h** mean horizontal position of text strings is based on variables, START and LOCK, from the data.
- e** means how to color the text strings.
- f** means how to centralize text strings in the EMPTY bars.

- 3** : AS means actual start time of task and AF means actual finish time of task. Data Lock has been scheduled, if we plan to start programming 2 months prior to data lock, the variable START is equal to variable LOCK – 60 days. NOLEGEND and NOJOBNUM mean we do not want to display any Legend or number of Jobs in the graph.
- 4** : ID statement indicates what variable you want to display for each activity.

EXAMPLE 2: ALLOCATING RESOURCES

Continuing with the project schedule given above, 3 programmers are assigned to the 5 projects based on their daily workloads. Here is the Gantt chart which includes 3 programmers' information.



WHAT THE GANTT CHART DISPLAYS (EXAMPLE 2)

From Gantt chart Example 2, it is very clear that programmers A and C will not work on 2 different projects at the same time. Programmer B is able to mainly support validation for both projects 1005 and 1003 first. Programmer A should be able to support programming on project 1003 prior to project 1005 is started.

PROC GANTT FOR EXAMPLE 2

```
PATTERN1 V = EMPTY C = VIOLET REPEAT = 10;
```

```

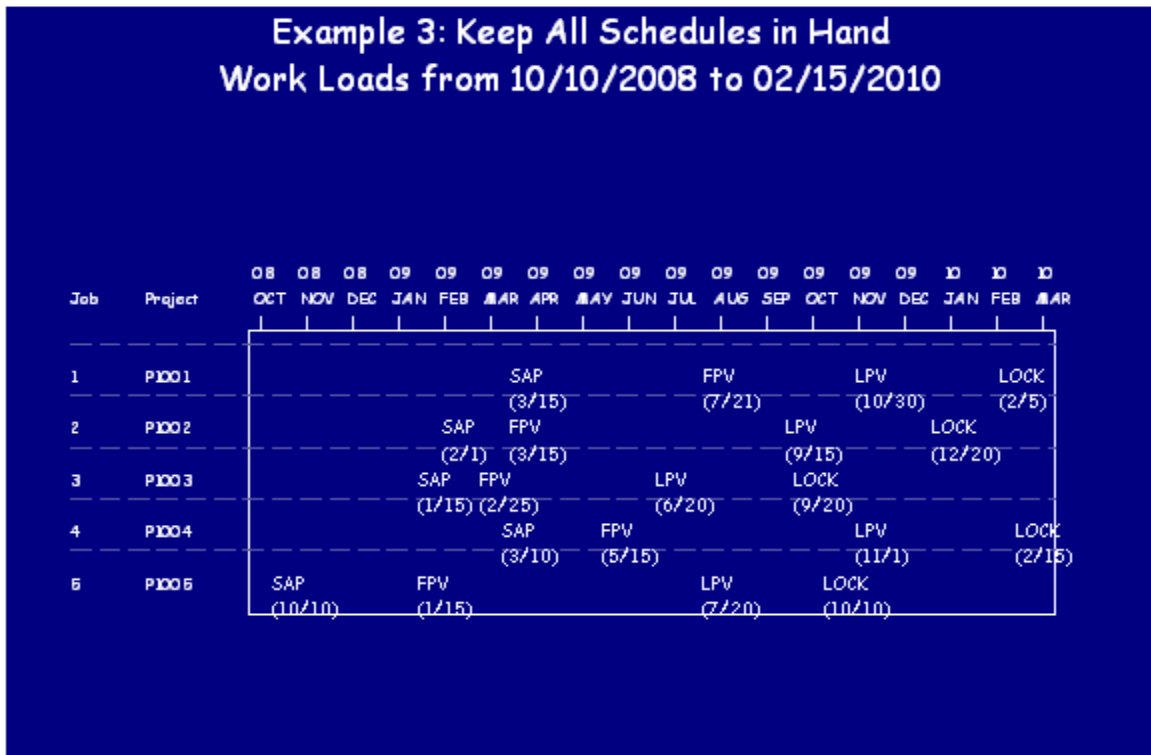
PROC GANTT DATA = STUDY LABDATA = LABEL;
  CHART / AS = START AF = END NOLEGEND NOJOBNUM
    HEIGHT = 1.1 CAXIS = WHITE DUPOK ----- 1
    ZONE = PROGRAMMER ONEZONEVAL CZONE = WHITE LZONE = 1 WZONE = 1 ----- 2
    REF = "07JUL09"d TO "02MAR10"d BY WEEK CREF = WHITE; ----- 3
  ID PROJECT;
RUN;
QUIT;

```

- 1 : Programmer A and C are assigned to 2 different projects, respectively. If we just want to display programmer information once, we can add the DUPOK function in the CHART statement.
- 2 : We can draw a line in white for each ZONE = PROGRAMMER, as this makes the Gantt chart easy to read. That is, ONEZONEVAL displays the value of the ZONE variable, PROGRAMMER, when beginning a new zone. We then can draw a line for each ZONE with the color we desired.
- 3 : A Reference Line BY Month or BY Week during the indicated time period.

EXAMPLE 3: KEEPING ALL SCHEDULES IN HAND

The schedule dataset contains the dates when the SAP is ready, First / Last Patient Visit, and Data Lock. It is easy to track the timelines of all the activities for each project when a graphical schedule is in hand – another benefit of Gantt chart



WHAT THE GANTT CHART DISPLAYS (EXAMPLE 3)

This Gantt chart simply displays all activities in order for each project. Shown on the top of the graph is the month and year while inside the graph is shown the DATE for each activity. It is easy to track and keep all schedules graphically.

PROC GANTT FOR EXAMPLE 3

```

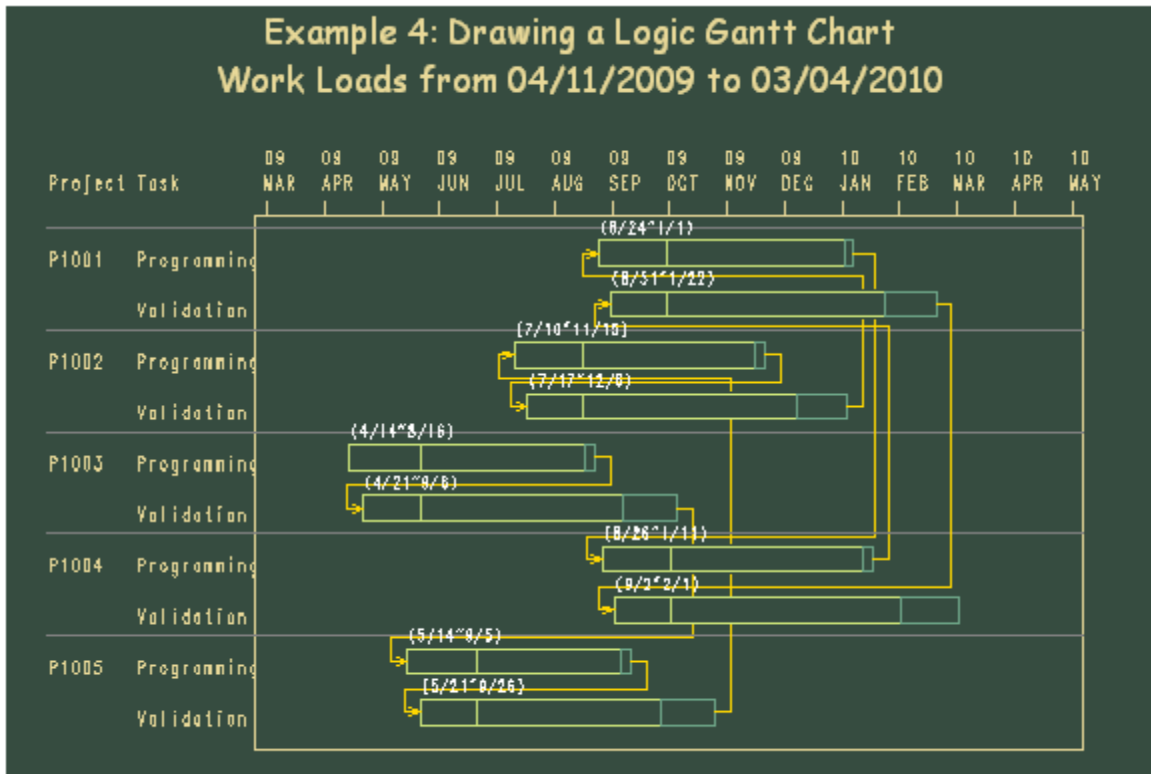
PROC GANTT DATA = TRAN LABDATA = LABEL; ----- 1
  CHART / AS = DATE NOLEGEND NOJOBNUM CAXIS = WHITE ----- 2
    ZONE = JOB HEIGHT = .8 CZONE = LIB LZONE = 3; ----- 3
  ID PROJECT;
RUN;
QUIT;

```

- 1: Since this Gantt chart only shows the time point for each activity instead of the duration of time, the finish date of the activity does not need to be considered. That is, we need to transfer the data structure from a horizontal to a vertical format.
- 2: AS means "actual start time". As mentioned above, the intent is to show the time point instead of the duration of time; therefore, AF = is **not** needed to display the actual finish time in this procedure. Only when it is desired to display actual start time, early start time, and late start time in the Gantt chart will there be a need to use variables for AS = , ES = , and LS, respectively.
- 3: It will be easy to read this graph if each project is split by line. This function is ZONE = as shown in the program. However, the difference between this example and Example 2 is there is no variable equal to ZONE in the dataset. If PROJECT as ID and ZONE variables is used, the information regarding PROJECT will be displayed in the graph twice. If the ZONE = function is not used, all events will be not be aligned on the same row. Therefore, the dummy variable "JOB" may be created as an identifier for each ZONE.

EXAMPLE 4: DRAWING A LOGICAL SCHEDULE PLAN

Given the goal to finish programming and validation 2 weeks prior to data lock, we should start both tasks, for example, 2 months before last patient visit. Sometimes programming may be started earlier or delayed due to unexpected reasons. Using the same data as an example with early start / finish and late start / finish variables, a Gantt chart can be created to show the order of tasks to work on.



WHAT THE GANTT CHART DISPLAYS (EXAMPLE 4)

In order to display the work flow, we need to derive Early Start / Finish and Late Start / Finish variables for Programming and Validation based on the plan to implement those tasks. The Gantt chart has 2 bars with different colors. The bar on the left displays the Early Start / Finish schedule. There is a vertical line inside the bar on the left, which is the Late Start date. The bar on the right displays the Late Finish date. In addition, this Gantt chart displays a logic flow showing the order for implementing those tasks.

The bar without an arrow flow is the first task we should work on first. Then, we simply follow the flow to the next task until all tasks are finished on time. The annotated text strings are Early Start / Finish dates.

PROC GANTT FOR EXAMPLE 4

```
PROC GANTT DATA = TASK LABDATA = LABEL;
```

```
CHART / INCREMENT = 1 NOJOBNUM NOLEGEND COMPRESS ----- 1
```

```
HEIGHT = .7 CAXIS = CREAM CPREC = GOLD ----- 2  
ZONE = PROJECT ONEZONEVAL CZONE = GRAY LZONE = 1  
HEAD = HEAD TAIL = TAIL; ----- 3
```

```
ID TASK;  
RUN;
```

1 3: Unlike the other examples discussed earlier, there are no ES = / EF = (Early Start / Finish) and LS = / LF = (Late Start / Finish) statements, but HEAD = and TAIL =. Since the HEAD = and TAIL = statements are used, it is not necessary to use ES, EF, LS, and LF. However, those variables have to be in the data with the SAS reserved variable names, E_START, E_FINISH, L_START, and L_FINISH. In other words, when we have those variables in our dataset, the next step is to specify the flow of implementing tasks, which are related to HEAD = and TAIL =. In this example, the schedule dataset is sorted by E_START date.

The option, COMPRESS, specifies how the Gantt chart will be drawn. The number of pages will be determined by the HPAGES = (Horizontal Pages) and VPAGES = (Vertical Pages) options. If HPAGES = and VPAGES = are not defined, the default value is 1 for each.

The option, INCREMENT =, specifies the minimal time interval for the time axis. In this example, the time axis is E_START and the minimal interval is one month.

2: CPREC specifies the color of Precedence Connection.

CONCLUSION

The GANTT procedure is a graphical scheduling tool for planning and tracking project tasks. Gantt chart is one type of bar chart that displays the relationship between project tasks and time. The actual schedule or the expected schedule can be plotted, as well as a Holiday schedule with logic flow. The Gantt chart makes managing projects much easier and more efficient.

REFERENCES

SAS® Institute (1999) Software: Reference, Version 9,
Copyright 1999 by SAS® Institute Inc., Cary, NC, USA

CONTACT INFORMATION

Your comments and questions are valuable and appreciated. Authors can be reached at

Hui-Ping Chen
Myriad Pharmaceutical Inc.
320 Wakara Way
Salt Lake City, UT 84108 U.S.A.
Email: hchen@myriad.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.