

## ELLIOTT 900 SERIES SIMULATOR

### ELLIOTT PERT PACKAGE.

The Elliott PERT Package is a standalone application for performing Critical path Analysis for a project defined in terms of a directed graph of activities (or tasks). Activities connect events (or milestones) and have a specified duration. All activities that precede a given activity in the graph are required to be completed before that activity can commence. Critical path analysis computes the shortest time in which the project can be completed by identifying the longest chain of activities from start events to finish events (the critical path). The analysis computes an earliest start and latest start for each activity. For activities on the critical path these are the same, for activities off the critical path they are separated by "free float". "Total float" by contrast is the amount by which an activity can be delayed without slipping the overall project.

### Summary of PERT input.

In essence, the Elliott PERT Package reads in a project graph, checks it for the absence of loops and other errors, then produces reports as requested by the user.

The input comprises two tapes:

1. A control tape
2. A data tape.

The control tape contains all the information required in a run and specified what outputs are required.

The data tape contains a specification of the activity graph for the project to be analysed.

Both tapes can be in either 900 or 903 telecode.

If the two tapes are punched on the same physical tape, they must be separated by runout. (This can be circumvented by patching location 76 to the instruction 8 80, which then allows the data to follow immediately after the control. To ensure the title on the data tape is correctly read in, the first characters of the date part should be \$ RETURN LINEFEED. This is illustrated in the demos: DEMO1 and DEMO2 use the < ! R!> facility to position blank tape before titles, DEMO3 uses

## ELLIOTT 900 SERIES SIMULATOR

an ENTER command to make the necessary patch to avoid the need to do this).

The program performs all the necessary calculations for the basic time analysis of the network producing earliest and latest "s for each event (with or without the overall project duration being set). The data is checked for the absence of loops and activity and event lists may be output to help with checking the network.

Once all the data is in an acceptable form it may be reviewed from time to time as the project proceeds, e.g., estimated durations may be replaced by actual durations. A binary tape of the accepted activities may be output during any run and re-input at the next run to save time.

The output required by the user is requested via the control tape.

Each run has, associated with it, a 'run number' and a 'date of run'. Run numbers must go in sequence, i.e., only run no. 4 can follow run no. 3.

### Control tape.

The control tape consists of a series of positive integers, terminated by a negative integer. (The positive integers can be signed or unsigned). The integers can be separated by any non-digit characters (and thus can contain commentary text, for example).

The first 9 integers are interpreted as follows:

1. Run number (used to distinguish this set of parameters from another set)
2. The number of working days in a calendar week (e.g., 5)
3. Day of run (dd)
4. Month of run (mm)
5. Year of run (yy)
6. Date of project start (dd)
7. Month of project start (mm)
8. Year of project start (yy)
9. Duration units: 0 = weeks, 1 = days.

The date of the run is shown in report headings.

The date of project start is used to compute milestone dates.

## ELLIOTT 900 SERIES SIMULATOR

The succeeding numbers in the steering file are interpreted as commands:

0. Read data from project tape and form activities lists
1. Read previous dump tape as produced by entry no. 2
2. Punch a dump tape of activities lists for current project, together with project title, date produced and run number
3. Calculate earliest and latest times for each event, checking for loops and outputting the start events, end events and overall project duration
4. Punch a summary report of events in ascending order of event numbers, with earliest and latest dates
5. Punch a summary report of the preceding event, succeeding event and duration of all the activities in preceding/succeeding event number order
6. Punch a report of all activities sorted by event number, showing start and end events, durations, earliest and latest start and finish dates, total and free float. Activities on the critical path are distinguished by an asterisk
7. All activities in earliest start date order
8. All activities in latest start date order
9. Critical path activities in preceding/succeeding event number order
10. Critical path activities in earliest start date order
11. Critical path activities in latest start date
12. All activities except dummies (which are not on the critical path) in event number order
13. All activities except dummies in earliest start date order
14. All activities except dummies in latest start date
15. Set project duration
16. All activities in total float order
17. All activities excluding dummies in total float order
18. All activities in free float order
19. All activities except dummies in free float order

PERT enforces a workflow. A project must be read in (by either command 0 or command 1) and then analysed (by command 3) before any other command can be executed.

The 2 command produces a binary dump of the current project graph suitable for reloading subsequently using the 1 command.

A 15 command should be followed by a further integer representing the total project duration. If this is longer than the critical path duration, additional total float will be available to project activities. If it is less than the

## ELLIOTT 900 SERIES SIMULATOR

critical path duration, total float will be negative for some task that cannot be completed by the required deadline.

Dummies are activities with duration 0 not on the critical path. Generally these are introduced to make drawing the project graph easier. They do not represent actual activities in the project.

The run number must be in the range 0-99. When using the number of working days in a week must be in the range 1-7. Note that the years in dates are in the range 0-99 and that Elliott PERT is not Year 2000 compliant.

When a dump is reloaded, the steering tape must reference an equal or larger run number than that specified for the run where the dump was produced.

### Data tape.

The data tape consists of a title, followed by interleaved control symbol lines and activity data lines.

The title consists of up to 40 characters, including carriage return and non-printing codes, terminated by a full-stop (.) and line feed. A dollar (\$) character cancels the preceding characters, provided the terminating line feed has not been encountered.

During control symbol and activity input, runout, erase and carriage return characters are ignored. All others are significant. Each control symbol or activity must start on a new line.

The control symbols specify how subsequent data lines will be interpreted, as follows:

- A Set mode to "add" (for adding additional activities to a project)
- C Set mode to "correct" (for changing the duration of activities in a project)
- R Set mode to "remove" (for deleting activities from a project)
- Dn Set project duration to n, where n is a positive integer
- T0 Data lines with single time duration
- T1 data lines with three time durations

## ELLIOTT 900 SERIES SIMULATOR

) Page separator (This can be used to separate data lines into a sequence of 'pages' to simplify the location of errors)

( End of data symbol

HALT Stop code, used if data is on more than one tape.

In "add" mode, activities are added to the project graph and an error is reported if a duplicate activity is encountered. This mode is the default. There can be at most 511 activities in a project.

In "correct" mode, subsequent activities must match the start and end events of previously entered activities, and the duration for the activity is updated. An error is reported if no matching previously entered activity can be found.

In "remove" mode subsequent activities must match the start and end events of previously entered activities, and the activity is deleted. An error is reported if no matching previously entered activity can be found.

Each control symbol should be on a new line. Their effects last until cancelled by another control symbol (except for D).

The D directive acts in the same way as the control tape 15 command, namely sets the total duration of the project independently of the critical path duration.

The T directive determines how activity durations are input. T0 specifies durations should be encoded as a single positive integer, T1 specifies durations should be encoded as a three integers representing an optimistic, most expected and pessimistic estimate of the activity's duration respectively. (In the second format the duration is the calculated as  $(O+4M+P)/6$ ). T0 is the default.

A closing round bracket ')' causes the page number given in error reports to be incremented.

An opening round bracket '(' signals the end of the project data.

Data lines take the form of a single line of text describing an activity, as follows:

start finish duration\*activity=comment

start is the number of event preceding the activity

## ELLIOTT 900 SERIES SIMULATOR

finish is the number of the event succeeding the activity  
start and finish cannot be the same event number.

Event numbers must be in the range 1-510.

The duration of the activity is represented as either a single number, or a triple as appropriate. The duration must be in the range 0-127.

Either or both of the event numbers and any or all of the numbers comprising the duration can be followed by a \$ in which case the entire line is cancelled.

Spaces and/or colons (:) should be used to separate the numbers in a data line.

Asterisk (\*) introduces an optional activity name terminated by an equal (=) character. (Note: the text can include line feed characters).

The optional comment following equal (=) can contain arbitrary text terminated by a line feed.

The asterisk (\*) and activity name can be omitted, leaving the activity to be terminated by equal (=) and comment. If neither activity name, nor comment is required the activity can be terminated by a line feed IMMEDIATELY following the final number of the duration.

In "remove" mode only the start and finish event numbers of the activity to be deleted should be specified or errors will arise.

(Warning: if an activity is not terminated by any of the \*...=..., \*...=, =..., =, or line feed endings it will be silently ignored. Therefore it is recommended each activity be terminated with one of the "=" endings).

## ELLIOTT 900 SERIES SIMULATOR

### PERT entry points.

- 32 Read and check control tape
- 9 Read data tape

If the control tape calls for a binary tape to be read, it should be loaded and PERT re-entered at 9. Similarly if multiple data tapes are to be read in (the last of which has the terminating ( symbol), re-enter at 9 for each additional tape.

The PERT package pauses after reading the steering tape and entry at 9 is required to start processing commands (to allow for a project tape or dump tape to be loaded as required). There is also a pause after a dump tape has been read in and entry at 9 is required to continue command processing.

### Error indications.

Errors are reported to the teletype in the form of a message "\*ERROR NUMBER n" where n is the error number. Some errors give additional information to help with diagnosis. Errors when reading the project tape report the page and line number at which the error was detected. Counting starts at page 0, line 1. A new page is started using the close round bracket ')' directive.

1. Checksum failure reading in dump tape. Re-enter at 9 to try re-input
2. Preceding and succeeding events of an activity are the same. (Data line is ignored)
3. No matching activity found in correct or remove mode. (Data line is ignored)
4. Attempt to add more than 511 activities to a project. (Program halts)
5. Attempt to add a duplicate activity, i.e., another activity with same preceding and succeeding events has previously been added. (Duration of activity read replaces old duration)
6. Event number not in range 1-511 (data line is ignored)
7. Duration larger than 127. (Data line is ignored)
8. Unexpected character encountered, e.g., data missing or syntax error etc. (Data line is ignored)

## ELLIOTT 900 SERIES SIMULATOR

9. Overall duration read from tape is out of range. (Data line is ignored)
10. Activity earliest start time is greater than 511 time units and/or overall project duration too large. (Program halts)
11. Activity latest start time is greater than 511 time units. (Program halts)
12. Steering commands not in correct work flow sequence. (Program halts)
13. Attempt to input more than 15 commands. Re-enter at 9 to obey first 15 entries.
14. Cycle found in project graph. (Program outputs list of loops and stops)
15. Run number in steering tape does not equal or exceed run number in dump file. (Re-enter at 9 to continue regardless or re-enter at 10 to re-input binary tape)
16. Error in steering tape, e.g., numbers missing or out of range. (Program halts)

### Demonstration programs.

#### DIRECTORY PERT

##### PERT(AJH).900:

A version of PERT Issue 2 modified by the author to correct layout errors in headings.

PERT(AJH).BIN: An assembled binary of PERT(AJH).900 package suitable for loading by initial instructions.

ASSEMBLE.DAT: A script to build PERT(AJH).BIN from PERT(AJH).900.

There are three demos:

DEMO1.DAT: Inputs a simple project and produces reports. Note the use of <! R !> to insert run out before the project heading.

DEMO2.DAT: Inputs a simple project and produces a dump file (DEMO2.BIN). The script then reads back in the dump file and produces a report. Note the use of <! R !> to insert run out before the project heading.



## ELLIOTT 900 SERIES SIMULATOR

DEMO3.DAT: A more complex project taken from the Elliott Programming Information Manual. Note the use of a patch to circumvent the need to insert run out before the project heading.