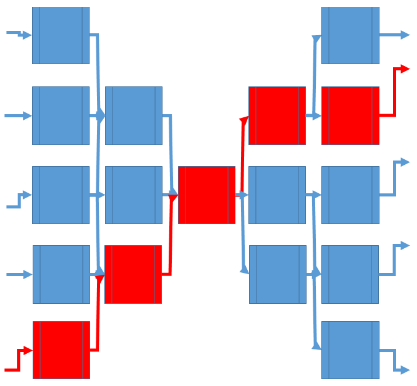
DRAG CALCULATION IN BPC LOGIC FILTER FOR MICROSOFT PROJECT



Prepared For:

General Release

Prepared By:

Thomas M. Boyle, PE, PMP, PSP

BPC Project No. 15-001
(Rev 4) 27-Sep-16

Boyle Project Consulting, PLLC

Project Management and Construction Support Services

4236 Chandworth Rd.
Charlotte, NC 28210

Phone: 704-916-6765
info@boyleprojectconsulting.com

TABLE OF CONTENTS

1.0 EXECUTIVE SUMMARY

Critical Path Drag is a rarely-used metric for allocating a projectôs potential for acceleration (and delay) among its various parts, namely its Critical Path activities. Despite its apparent usefulness, neither Drag nor the associated Drag Cost are natively computed by the dominant project schedule control tools in North America, including Microsoft Project. BPC Logic Filter ï a Microsoft Project Add-In product ï is a low-profile tool that generates sharply focused views of the logical relationships between various activities in a project. Although its core focus remains on visualization of the logical schedule network, the tool has been modified to calculate Critical Path Drag as a secondary function.

Drag can be possessed independently by five different logical components of the schedule Critical Path, including task durations, predecessor lags, task calendars, task date constraints, and in-progress task delays (splits). BPC Logic Filter analyzes four of the five Drag components and can save the most common one (task duration Drag) to a field in the schedule data file for later processing. The other three components are saved in text files only and must be manually retrieved.

This document describes the product background and motivation for this functionality, followed by appropriate descriptions of the pertinent schedule components, Drag calculation methodologies, and procedures for implementing the tool. The Appendix includes documentation for six sample projects.

2.0 BACKGROUND / MOTIVATION

Boyle Project Consulting originally developed BPC Logic Filter for Microsoft Project for our own use in monitoring and controlling large project schedules where Microsoft Project has been specified by the Client. The tool is presently (at the time of writing) undergoing testing in preparation for licensing for use by others.

For the professional planner, scheduler, project manager, or administrator working with Microsoft Project (2010+), the tool offers to fill the gaps between Projectôs native functionality and the requirements of logic-driven schedule management in complex projects. BPC Logic Filter includes simple tools for examining and tracing the schedule network(s) for any given task or collection of tasks. It also offers tools for standardized reporting of a projectôs Critical Path and near-critical paths when Microsoft Projectôs internal reporting does not suffice. Finally, BPC Logic Filter introduces and implements several tools for advanced schedule network analysis, including Path Relative Float, Bounded Network Analysis, and Cross-Project Links. The tool is implemented as a Microsoft Office Customization (i.e. Project Add-In) and is deployed using secure Microsoft-developed technology that does not require Administrative Privileges.

BPC Logic Filter also supports the calculation and reporting of Driving Path Drag, a generalization of the ñCritical Path Dragò metric introduced by Stephen Devaux in his 1999 book.[1] As described by Devaux, Drag essentially represents the amount of time each activity is adding to the overall completion of the project; consequently, only

[1] ñ*Devaux's Removed Activity Guage*ò (DRAG), from Devaux, Stephen A, Total Project Control: A Managerôs Guide to Integrated Project Planning, Measuring, and Tracking, John Wiley & Sons, 1999

Critical Path activities have it. The fact that the Critical Path is defined equally by ñthose tasks with non-zero Dragò and by ñthose tasks with zero Total Floatò means Drag and Total Float are closely related (though opposites). Unlike Total Float, Drag can be useful for allocating the overall costs of prospective project delays (and profits of project acceleration) to specific activities in the project schedule. This is the activity Drag Cost. In my experience, Drag and Drag Cost also tend to be more intuitively graspable by managers in non-project-oriented organizations, where Float and other Critical-Path-Method (CPM) concepts are poorly understood.

Computation of Drag Cost is beyond the scope of BPC Logic Filter. The purpose of this paper is to identify the various components of Drag that are computed and displayed by the tool for subsequent use in Drag Cost analysis by others. Readers are presumed to be trained and competent users of MS Project and other CPM project scheduling tools.

3.0 SCHEDULE NETWORK COMPONENTS

Within the context of a (logic-driven) CPM schedule network ï and specifically one prepared with Microsoft Project ï the overall project is comprised of a collection of activities to be accomplished in a prescribed logical sequence, from project Start to project Finish. The scheduled start and finish dates of any given activity (ñtaskò in MSP) are determined according to the following factors. *[Remarks in bracketed italics indicate features specific to MSP.]*

3.1 Duration

Duration is the amount of working time needed to complete the task once it has started. Duration can be computed automatically on the basis of total work (e.g. in person-hours or person-days) and applied resources (e.g. number of persons assigned), but this tends to be problematic for complex project organizations. Explicit specification of the duration is more common in larger project schedules.

> [*MSP stores all durations in minutes. Display in days (or weeks/months) can vary depending on the project's HoursPerDay (or HoursPerWeek/DaysPerMonth) properties.*
>
> *Like other CPM tools, MSP supports the use of project, task, and resource calendars to restrict the "working time" available for expending a task's duration. While not free of complications, applying variable calendars seems to be the most reliable and repeatable method to account for weekends, holidays, vacations, and even weather-related planned schedule interruptions. In addition to calendars, MSP allows for manually "splitting" the duration of a task into multiple parts on the bar chart's graphical user interface. Once imposed, however, such task splits are not easy to read, modify, remove, or analyze within the MSP user interface. Quality of life for the CPM-oriented scheduler is improved by leaving them out where possible.]*

3.2 Predecessor relationships

Predecessor relationships are logical sequential constraints precluding the start (or finish) of the successor task until any and all predecessor tasks have been started or finished. Typical relationship types include the following:

a. FF → Predecessor must finish before this successor task can finish.

b. FS → Predecessor must finish before this successor task can start.

c. SF → Predecessor must start before this successor task can finish.

d. SS → Predecessor must start before this successor task can start.

[*MSP stores all relationships as dependency objects that are members of the associated tasks. Unlike certain other tools, MSP allows ONLY ONE relationship between any two tasks. Thus more complex schedule models are required to represent common situations (e.g. sequential piecework tasks like "Clean 10,000 bricks"* → *"Lay 10,000 bricks" are ideally suited to concurrent SS and FF relationships, but these are not possible in MSP.)]*

3.3 Predecessor relationship lags and leads

Lags and leads are additional time offsets that are added to or subtracted from the relationship constraints. Thus:

a. FF+4d (lag) → The successor task may finish no sooner than 4 days after the predecessor finishes.

b. FS-3d (lead)→ The successor task may start no sooner than 3 days before the predecessor finishes.

c. SS+50% (percentage lag) → The successor task may start no sooner than [50% of the predecessorôs duration] after the predecessor starts.

[*MSP also stores lags, like durations, in minutes. Corresponding dates are computed based on the working-time calendar of the successor task. This is not user-selectable.*

Users should note that percentage lag is completely unrelated to "%Complete" of either task in the relationship. The lag of the "SS+50%" relationship above will commence with the start date (Actual or not) of the predecessor task and will conclude on a date and time that is (50% of the predecessor's total duration) later – ALL computed according to the SUCCESSOR'S CALENDAR.]

3.4 Calendar Mismatches

If a task and its predecessors share the same working time limitations (i.e. ñcalendarsò), then the governing logic is effected through the relationships alone. In the event of a calendar mismatch, however, the task may be prevented ï by its calendar ï from starting as soon as its predecessor relationships are satisfied.[2]

3.5 External Date Constraints

External constraints are hard-coded date limitations that are applied directly to a task and are generally of three types:

[2] If a task is delayed by calendar mismatch, then the ñdrivingò predecessor task possesses free float/slack, computed according to its own calendar. As a result, both free and total float/slack become unreliable as indicators of logical connection between the tasks.

a. Earliest Allowable Dates: These constraints are most often used to represent anticipated decisions, deliveries, access, or other interfaces with activities that are otherwise excluded from the schedule. Their date limitations are applied either alone or in addition to (never in opposition to) the predecessor logic, and computed dates are adjusted as needed to meet all conditions. The constraint is effective when it (rather than some more stringent predecessor logic) controls the taskôs scheduled dates. There are two variations:

   i. Start No Earlier Than (SNET)

   ii. Finish No Earlier Than (FNET)

b. Latest Allowable Dates: These constraints are typically used to represent project or contract commitments or other schedule obligations caused by external factors. Unlike the first type above, these date limitations are always applied in combination with and in opposition to the predecessor logic, resulting in the potential for logical conflicts. Consequently, they are used sparingly if at all, and some CPM practitioners bar their use completely. There are three variations listed here, although the third is a special case and (within MSP) not technically a ñconstraint.ò

   iii. Start No Later Than (SNLT)

   iv. Finish No Later Than (FNLT)

   v. Deadline

   [At the very least, imposing a late constraint on any task may directly affect the CPM late dates, and hence the Total Float/Slack calculation, for the task and its driving predecessors. If the scheduled task cannot satisfy all of its predecessor relationships and the imposed Late Constraint at the same time, then a logical conflict exists. In this case, if the "Tasks will always honor their constraint dates" schedule options checkbox is checked (the typical default condition), then MSP will override the predecessor logic and set both the early dates and late dates of the task to the same value as needed to satisfy the constraint.[3] (Unfortunately, this approach accepts and stores an untenable logical conflict in the schedule; it should be avoided.) If the checkbox is UNCHECKED, then the situation reverts to a more classical CPM approach: the task is scheduled to meet its predecessor relationships, and the constraint affects the backward-pass (late dates) only. Since the task's late dates precede its early dates, negative total slack is computed to reflect that the constraint cannot be met.

   For most practical purposes, a task Deadline behaves identically to a FNLT constraint in a classical CPM algorithm – that is, it ignores the "Tasks will always honor their constraint dates"

[3] MSP will then reset Total Slack to the same (negative) value that it would have been if the early dates had not been overwritten. The result is one of the few situations where Total Slack as computed in MSP does not agree with a classical CPM Total Float computation.

checkbox altogether.]

c. Must-Meet Dates: These rarely-used constraints represent a simultaneous combination of the previous two types. They are most often used to represent inevitable and invariable events, i.e. something that will ñhappenò on a given date ï neither earlier nor later ï ready or not. Examples include holiday celebrations and legally-mandated events like Election Day and the presidential inauguration in the USA. Project managers are tempted to (and sometimes do) impose must-meet constraints on key project commitments ï this is to avoid ñconfusionò over any forecast date that differs from the commitment (either earlier or later). Due to the possibility of logical conflicts with predecessor relationships, must-meet date constraints are not recommended for CPM schedules. There are two variations:

    vi. Must Start On (MSO)

    vii. Must Finish On (MFO)

3.6 Schedule Float Constraints

These constraints may be applied directly to a task to control its use of available schedule flexibility. There are generally two types, but I have included a third type that is defined as a ñConstraintò in MSP:

a. Zero-Free-Float (ZFF). This constraint forces the task to be delayed the maximum time possible without delaying any of its successors. It is commonly used (in other tools) to schedule just-in-time deliveries, resource mobilizations, and the like. [*This constraint is not available in MSP. Primavera tools now name this constraint "As Late as Possible."*]

b. Zero-Total-Float (ZTF). This constraint forces the task and all of its successors to be delayed until all of the available total float is used up (i.e. the maximum time without delaying the project.) It is not useful for CPM-oriented schedules. [*This constraint is not available in some other tools. MSP names this constraint "As Late as Possible" (ALAP).*]

c. As Soon as Possible (ASAP). This is the default scheduling approach for CPM scheduling tools. [*In MSP, "ASAP" is the default schedule constraint that basically indicates the absence of any other constraint.*]

[Unlike other tools, Microsoft Project allows only a single schedule constraint (out of all those listed above) for each task in the project. The exception: "Deadlines" are formally treated as separate and distinct from constraints, so it is possible for a task to have both a single constraint (SNLT for example) and a Deadline.]

3.7 Resource Leveling

Most modern CPM-based tools allow users to impose an additional ñleveling delayò as needed to avoid resource over-allocations. For example, if two tasks require the same resource at the same time, but there is only enough of the resource to execute one of them, then one of the tasks will be delayed. In BPC Logic Filter, the task that is not delayed is called a ñresource driverò for the delayed task. Some tools offer complex options for prioritizing tasks with conflicting resource demands.

[*Although detailed documentation appears sparse, MSP's "Standard leveling order" appears to assign leveling delay in order of task Total Slack. If the "Priority,Standard" leveling order is selected, then higher-priority tasks retain their scheduled dates regardless of their slack values. If the option is selected, MSP will also create splits in remaining work to avoid resource conflicts.]*

3.8 Schedule Progress Updating

In the real world, even the simplest projects rarely go according to plan, and complex project schedules require periodic progress updating and re-scheduling of uncompleted work to maintain an accurate forecast and a modicum of control.

[*This is not MSP's strong suit, and no progress updating practices seem to have been universally adopted in its user base. Within the program, however, here are some key behaviors:*

1. Marking a task as 100% complete (or equivalently entering an Actual Finish date) overrides the impacts of any predecessors or constraints. This behavior is commonly referred to as "progress override" in classic CPM tools.

2. Entering an Actual Start and a partial percent complete introduces a "Resume" date reflecting MSP's initial calculation of the date that the uncompleted work commences. If the actual start date is before the date allowed by its predecessors (i.e. out-of-sequence progress), AND the "split in progress tasks" scheduling option is checked, then the "Resume" date will inherit the effects of the start predecessors and constraints. The task may then be split into two (or more) non-contiguous pieces, with only the portion after the resume date meeting the start's logical constraints. This behavior is commonly referred to as "retained logic" in classic CPM tools. If "split in progress tasks" is not checked, then any logical restrictions on the start will have no impact in the presence of an Actual Start date, and the behavior reverts to standard "progress override."

3. When used as intended, the "Reschedule uncompleted work to start after…" method of the "Update Project" dialog routinely resets the "Resume" date of an in-progress task to the project's Status Date or to any other date selected by the user. (The same method imposes a new start constraint on all selected tasks.)

4. MSP includes no "Data Date" construct, and the project Status Date is not reliable for general analysis of task completion status.]

4.0 SCHEDULE NETWORK COMPONENTS – DRAG CALCULATIONS

4.1 What constitutes Drag?

Within the context of critical path management, the Drag of a schedule component represents the <u>potential</u> amount of project acceleration that might be achieved by removing the component itself while leaving the rest of the schedule unchanged. For example, it is possible to step through each task in a small schedule, set its duration to zero, and then re-calculate the schedule to observe the resulting acceleration of the project completion date (if any). The observed acceleration represents the Drag of the particular taskôs Duration. The core possessors of Drag in CPM schedules are the Critical Path work activities (i.e. the CP-tasks) ï measured through their (remaining) durations. Drag can also be possessed by other schedule components, which could be verified through a similar exercise of stepping through the project network and ñremovingò each component to observe the impact on the project completion date.

4.2 Schedule Components with Drag

4.2.1 Duration

As noted above, the remaining duration of each task on the projectôs Critical Path may possess Drag, and that Drag may be limited or otherwise constrained as follows:

1. (Duration) Drag may not exceed the remaining duration of the task. (Exception: see 4.2.4 below.)
2. A parallel non-critical path results in a Drag value no greater than the total float of the parallel path.
3. A non-effective finish constraint (FNET) may limit the duration drag if it is within the range of the taskôs remaining duration.
4. A parallel critical path results in a Drag of zero.
5. Where critical-path logic does not flow through the taskôs remaining duration (e.g. the taskôs start is both driven by its predecessors and driving its successors, or the taskôs finish is both driven and driving) then the Drag is zero.
6. Some tasks with reverse logic flow (i.e. driven finish and driving start) have an inverse duration relationship such that lengthening the task shortens the project, and shortening the task lengthens the project.[4] BPC Logic Filter assigns a negative sign to the Drag in this case, and the Drag is limited by non-driving relationships (if any), ineffective constraints, or parallel logic paths.
7. Resource-driving tasks in a leveled schedule possess Duration Drag which is never negative. BPC Logic Filter computes and displays this as long as the ñcheck resource driversò option is checked. [Note: Drag analysis is presently not compatible with (nor available for) resource leveled schedules. A future upgrade to improve compatibility is planned.]

[4] Such logic may appear perverse especially for effort-driven tasks in detailed schedules, but it is not uncommon in high-level schedules for information-intensive tasks like engineering and software development. For such tasks, concurrent start and finish predecessor relationships would be preferred, but they are not possible in MSP.

4.2.2 Predecessor relationships

Predecessor relationships themselves are not considered work activities and possess no Drag.

4.2.3 Predecessor relationship leads and lags

Predecessor relationship lags are considered to (typically) represent work activities, and they may or may not possess Drag that is limited or otherwise constrained as follows:

1. (Lag) Drag may not exceed the specified lag of the relationship.
2. A parallel path results in a (Lag) Drag value no greater than the total float of the parallel path.

[It can be argued that a percentage-based lag actually reflects the work of the predecessor task and that its drag should be included with the predecessor's duration-drag component. The argument appears valid as long as the predecessor task is not started, but problems arise once it starts. BPC Logic Filter converts all such percentage lags to days and includes their drag as the (Lag) Drag component for the successor task.]

4.2.4 Task Calendars

In the event of Calendar mismatches, the task calendar itself may possess significant Drag. BPC Logic Filter does not compute Calendar Drag directly. Rather, BPC Logic Filter allows the calculation of duration Drag using any one of three calendars (the local task, the selected/end task, or the project). The resulting duration Drag will be computed differently depending on the drag calendar used.

4.2.5 External Date Constraints

BPC Logic Filter presumes that there are no logical conflicts. Therefore, the ñTasks will always honor their constraint datesò checkbox must remain UNCHECKED, OR logical conflicts must have been manually found and removed by the scheduler.

4.2.5.1 Earliest Allowable Dates

Early date constraints are considered to (typically) represent external work activities, and they possess Drag which is limited or otherwise constrained as follows:

1. Only constraints that are effective (i.e. controlling the associated date in the schedule) have (Constraint) Drag.
2. (Constraint) Drag is limited by the non-driving predecessors of the task.
3. In the absence of non-driving predecessors, (constraint) Drag is limited by the project Status Date or the project Start Date, whichever is later.

4.2.5.2 Latest Allowable Dates

Late date constraints do not represent removable activities. They possess no Drag.

4.2.5.3 Must-Meet Dates

The must-meet date constraints are considered to possess drag identical to the Earliest Allowable Dates constraints above.

4.2.6 Schedule Float Constraints

Schedule Float constraints in MSP do not represent removable activities. They possess no Drag

4.2.7 Schedule Progress Updates

Schedule progress indicated by Actual (Start/Finish) dates and/or non-zero values for task percent complete can have direct and indirect impacts on the task (Duration) Drag, (Lag) Drag, and (Constraint) Drag as indicated below. In addition, using the ñUpdate Projectò method (or implementing similar methods manually) can introduce a new (Split/Resume) Drag value, especially when progress is extremely delinquent or out-of-sequence compared to the plan.

1. **Completed Tasks** have no Drag of any kind (Duration, Lag, Constraint, or Split.)
2. **In-Progress Tasks** have duration drag according to their remaining duration and other limitations above. In addition:
   a. In-progress tasks have no (Lag) Drag for any Start predecessors and no (Constraint) Drag for any Start Constraints. (Lag) Drags on successor tasks will still exist until those successors start.
   b. In-progress tasks continue to have drag associated with driving Finish Lags and effective Finish Constraints until they are completed.
   c. If rigorous methods for progress updating and re-scheduling/forecasting are in use[5] then the uncompleted part of in-progress tasks may be split and delayed compared to the completed part. This is common if the task progress is slower than planned. In that case, a (Split/Resume) Drag is introduced corresponding to the delay of the uncompleted part of the task.
   d. If an in-progress task has started <u>before</u> its predecessor relationships have been satisfied, then the uncompleted part of the work will be split from the completed part and delayed until its predecessor relationships are satisfied. In that case, a (Split/Resume) Drag is introduced corresponding to the delay of the uncompleted part of the task.
   e. Note: In case of multiple splits in a task, BPC Logic Filter will compute (Split/Resume) Drag only for the last split.
3. **Un-Started Tasks** have Drag (Duration, Constraint, and Lag) corresponding to the earlier paragraphs. In addition, if an un-started task has been manually split into multiple parts, BPC Logic Filter will compute a

[5] i.e. 1ï Project Status Date used; 2ï No Uncompleted work in the past; 3ï No Completed work in the future; 4 ï ñSplit in-progress tasksò Schedule Options Checkbox is checked

(Split/Resume) Drag for the last split only. This is not common.

4.3 Drag Calculation Methodology

4.3.1 "Brute Force" Approach

As mentioned previously (in 4.1), one conceivable approach to calculating Drag is to systematically step through the schedule, temporarily removing each potential Drag-possessing component, then ñre-calculatingò and observing the consequent impact on the project completion date. Any observed acceleration represents Drag. (Conversely, any observed slippage of the project completion date might represent negative Drag, though this is not universally accepted). ñRe-calculatingò in this case means having MSPôs scheduling engine re-compute all the schedule dates. This approach is theoretically correct and would lead to unquestioned results (subject only to the limitations of the scheduling engine itself). Unfortunately, the repeated re-calculations of the schedule network can place significant demands on computing resources, such that the time required to calculate Drag in a large schedule can be considered excessive. (Beginning with Release 1.1.1.14, this is the approach used by BPC Logic Filter.)

4.3.2 BPC Logic Filter Limitations

It is possible to construct MSP project schedules that do not comply with the scheduling assumptions underlying BPC Logic Filter, and the resulting Drag calculations may be inaccurate. The following situations and practices, in particular, are not specifically supported for Drag calculation.

1. Multiple logical paths terminating without successors. BPC Logic Filter anticipates that a single task (or completion milestone) accurately defines the completion of the project.
2. (Similar) Multiple logical paths terminating in tasks with no finish successors (i.e. dangling finishes). Although there is an implicit relationship between the completion of such tasks and the overall project completion, BPC Logic Filter will not recognize it.
3. Manually scheduled tasks.
4. Any persistent logical conflicts such as those caused by circular logic or by late constraints with the ñ<u>Tasks will always honor their constraint dates</u>ò checkbox checked.
5. Inactivated tasks as treated in MSP 2013.

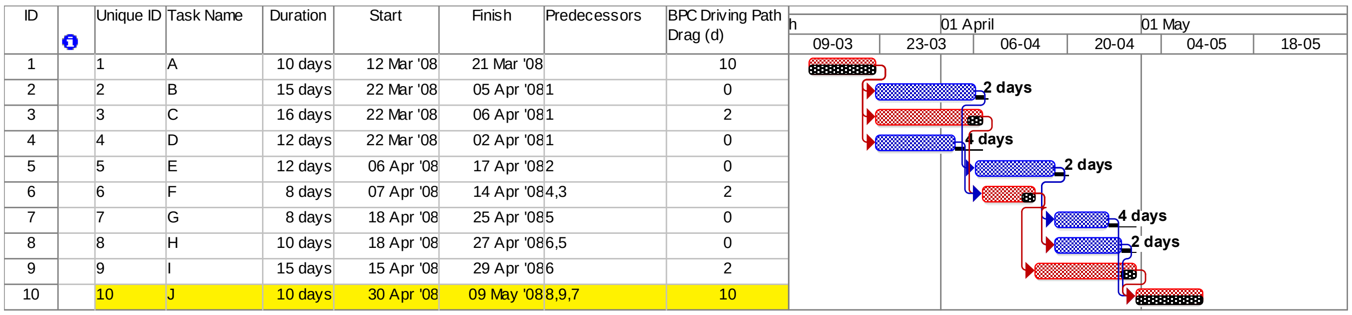
5.0 COMPUTING, DISPLAYING, AND STORING DRAG

5.1 "Driving Path" vs. "Critical Path" Drag

While the discussion of Drag in this paper has often referred to the projectôs Critical Path, the Critical Path Method (CPM) and Critical-Path tasks, these terms are formally <u>undefined</u> in BPC Logic Filter for Microsoft Project. This is because MSP defines ñCriticalò tasks (and hence, the Critical Path) solely in terms of the Total Slack task metric, which can be incorrect in the presence of late constraints, deadlines, and variable calendars. A fundamental and more accepted definition of the projectôs Critical Path ï not based on Total Float/Slack ï is ñthat collection of activities whose durations and relationships control the (earliest) completion date of the project.ò This is the projectôs Longest Path.

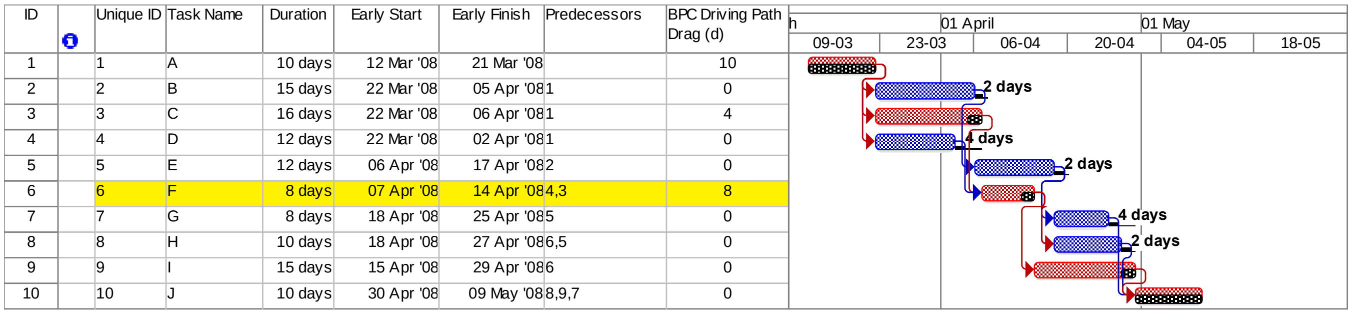
BPC Logic Filter ignores MSPôs slack calculations and examines logical predecessor relationships to identify the ñdriving pathò for any selected task in the project. The projectôs Longest Path (i.e. its Critical Path) represents a special case where the initial selected task happens to be the final task in the project. BPC Logic Filter includes a special method ï ñLongest Path Filterò ï for this special case. The resulting ñDriving Path Dragò is equivalent to the projectôs overall ñCritical Path Dragò. As an example, Figure 1 illustrates a simple project schedule and shows the Drag values corresponding to <u>Task Jôs</u> driving path. Task J ï the last task in the schedule ï had been automatically selected through the Longest Path Filter method.

Complex projects and programs can have more than one key deliverable, and the final activity of the schedule is not always the most important from a scheduling standpoint. That is, the project may have multiple key completion milestones, each with its own corresponding ñCritical Pathò. Starting from any selected completion task or milestone, BPC Logic Filter readily computes the corresponding ñDriving Path Dragò for any task in the project schedule. For example, Figure 2 shows the same simple schedule as before, but the analysis has instead begun from Task F, selected arbitrarily to represent another task of interest in the project. The indicated Drag values are for <u>Task Fôs</u> driving path. Since BPC Logic Filter is routinely used to examine logical driving and driven paths for any task in the schedule, computed Drag values must be expected to change every time the tool runs.



ID		Unique ID	Task Name	Duration	Start	Finish	Predecessors	BPC Driving Path Drag (d)
1		1	A	10 days	12 Mar '08	21 Mar '08		10
2		2	B	15 days	22 Mar '08	05 Apr '08	1	0
3		3	C	16 days	22 Mar '08	06 Apr '08	1	2
4		4	D	12 days	22 Mar '08	02 Apr '08	1	0
5		5	E	12 days	06 Apr '08	17 Apr '08	2	0
6		6	F	8 days	07 Apr '08	14 Apr '08	4,3	2
7		7	G	8 days	18 Apr '08	25 Apr '08	5	0
8		8	H	10 days	18 Apr '08	27 Apr '08	6,5	0
9		9	I	15 days	15 Apr '08	29 Apr '08	6	2
10		10	J	10 days	30 Apr '08	09 May '08	8,9,7	10

Figure 1: NLD59DRAG - Task J Selected (Longest Path Filter)



ID		Unique ID	Task Name	Duration	Early Start	Early Finish	Predecessors	BPC Driving Path Drag (d)
1		1	A	10 days	12 Mar '08	21 Mar '08		10
2		2	B	15 days	22 Mar '08	05 Apr '08	1	0
3		3	C	16 days	22 Mar '08	06 Apr '08	1	4
4		4	D	12 days	22 Mar '08	02 Apr '08	1	0
5		5	E	12 days	06 Apr '08	17 Apr '08	2	0
6		6	F	8 days	07 Apr '08	14 Apr '08	4,3	8
7		7	G	8 days	18 Apr '08	25 Apr '08	5	0
8		8	H	10 days	18 Apr '08	27 Apr '08	6,5	0
9		9	I	15 days	15 Apr '08	29 Apr '08	6	0
10		10	J	10 days	30 Apr '08	09 May '08	8,9,7	0

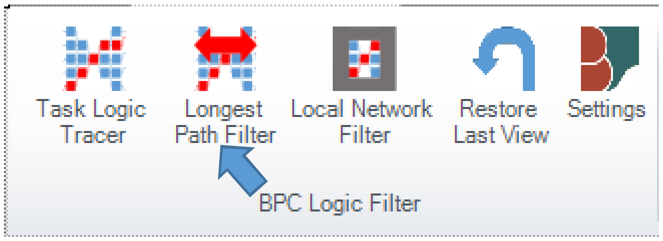
Figure 2: NLD59DRAG - Task F Selected (Task Logic Tracer)

5.2 Starting the Analysis

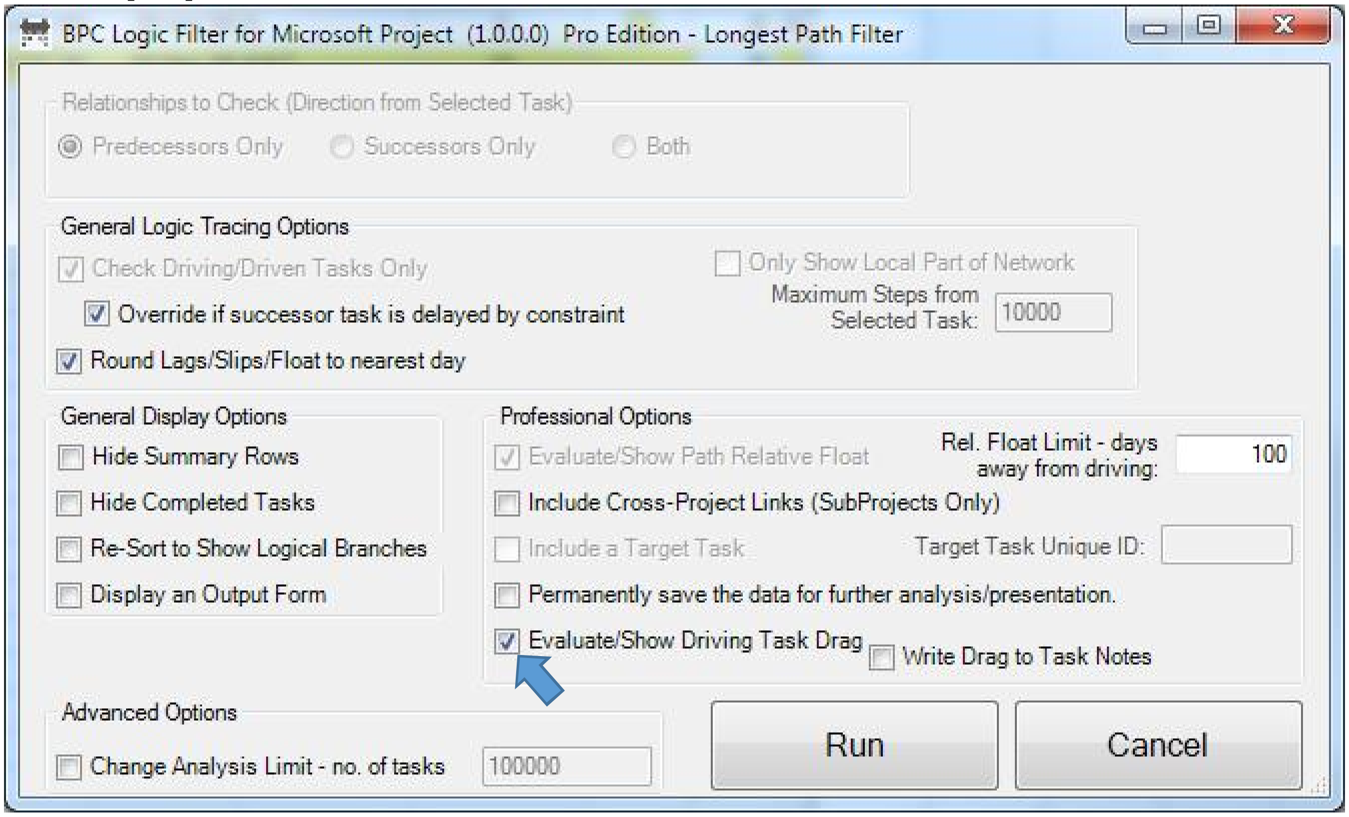
For general background and instructions on using the tool, the reader is referred to ñIntroduction to BPC Logic Filter for Microsoft Project,ò published on our website.

5.2.1 Longest Path Filter

For a relatively simple project, the quickest way to compute Critical Path Drag is to start with the Longest Path Filter method.



When the main form for the method appears, be sure to check the box for ñEvaluate/Show Driving Task Drag,ò then click [Run].


BPC Logic Filter for Microsoft Project (1.0.0.0) Pro Edition - Longest Path Filter

Relationships to Check (Direction from Selected Task)
[● Predecessors Only] ○ Successors Only ○ Both

General Logic Tracing Options
- [✓] Check Driving/Driven Tasks Only
- [✓] Override if successor task is delayed by constraint
- [✓] Round Lags/Slips/Float to nearest day
- [ ] Only Show Local Part of Network
- Maximum Steps from Selected Task: 10000

General Display Options
- [ ] Hide Summary Rows
- [ ] Hide Completed Tasks
- [ ] Re-Sort to Show Logical Branches
- [ ] Display an Output Form

Professional Options
- [✓] Evaluate/Show Path Relative Float
- Rel. Float Limit - days away from driving: 100
- [ ] Include Cross-Project Links (SubProjects Only)
- [ ] Include a Target Task
- Target Task Unique ID:
- [ ] Permanently save the data for further analysis/presentation.
- [✓] Evaluate/Show Driving Task Drag
- [ ] Write Drag to Task Notes

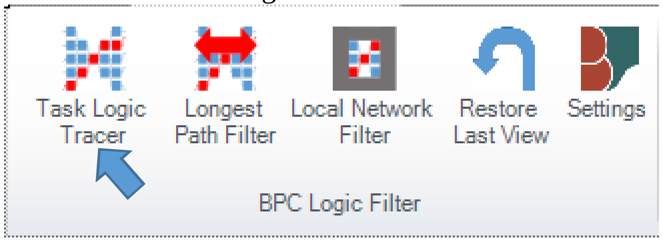
Advanced Options
- [ ] Change Analysis Limit - no. of tasks 100000

The options shown here will compute all available Drag components and then save the key steps and results of the calculations to a ñDragLogò text file in the default folder. In addition, final computed drag for all non-duration schedule components are written to the standard Run Log for BPC Logic Filter.

5.2.2 *Task Logic Tracer*

For a more complex project, including any project where the final task is not the preferred one for defining the Critical Path, then the general-purpose Task Logic Tracer should be used.

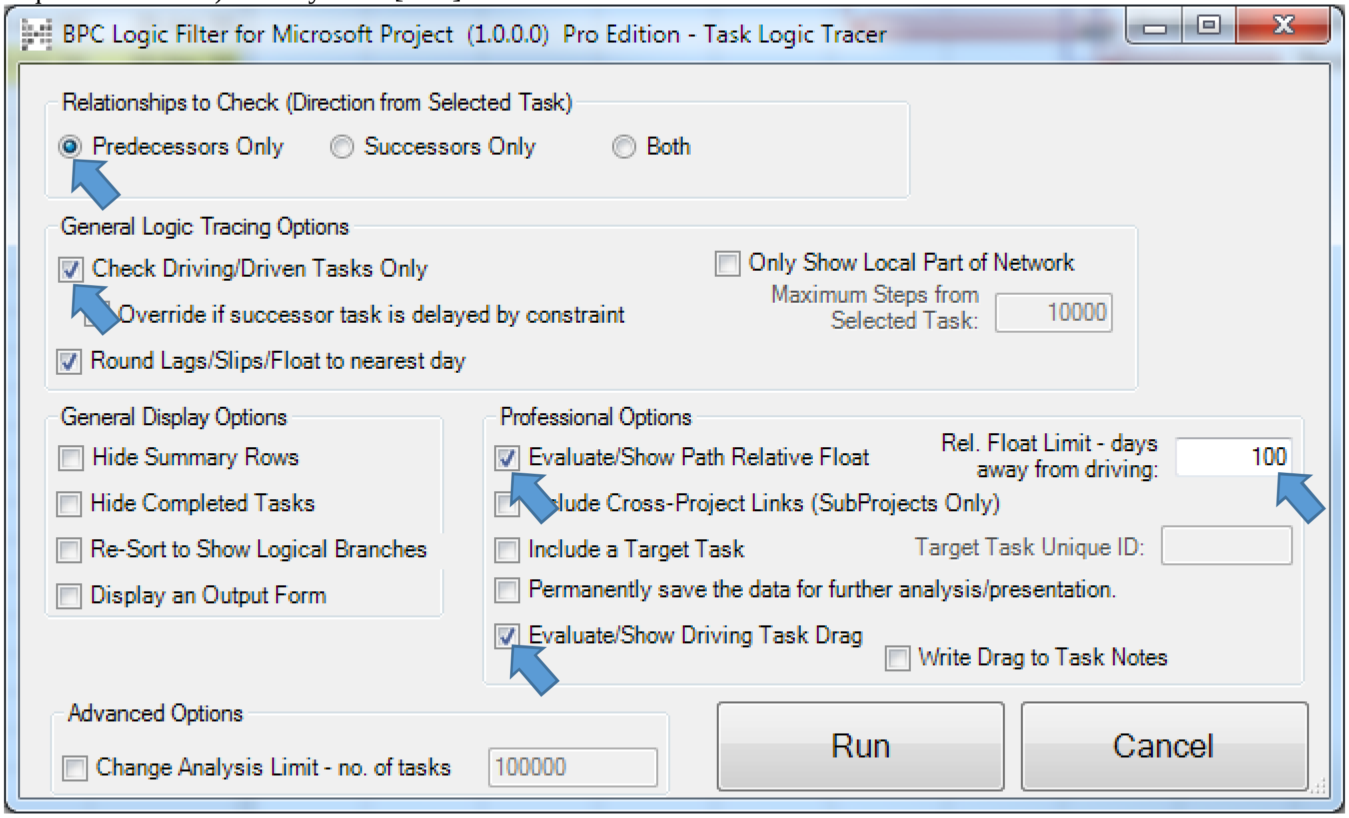
First, click in the task table to select the task that defines the completion milestone, then click the Task Logic Tracer button on the BPC Logic Filter Add-In Ribbon:



When the main form for the method appears, be sure that the following boxes are checked:

1. Predecessors Only
2. Check Driving/Driven Tasks Only
3. Evaluate/Show Path Relative Float
4. Evaluate/Show Driving Task Drag

In addition, the ñRel. Float Limit ï days away from driving:ò field may be modified if computed drags higher than the default value (100d) are expected. (Depending on the size of the project model, a higher value may increase the required run time.) Finally click [Run].



BPC Logic Filter for Microsoft Project (1.0.0.0) Pro Edition - Task Logic Tracer

Relationships to Check (Direction from Selected Task)
[● Predecessors Only] ○ Successors Only ○ Both

General Logic Tracing Options
- [✓] Check Driving/Driven Tasks Only
- [ ] Override if successor task is delayed by constraint
- [✓] Round Lags/Slips/Float to nearest day
- [ ] Only Show Local Part of Network
- Maximum Steps from Selected Task: 10000

General Display Options
- [ ] Hide Summary Rows
- [ ] Hide Completed Tasks
- [ ] Re-Sort to Show Logical Branches
- [ ] Display an Output Form

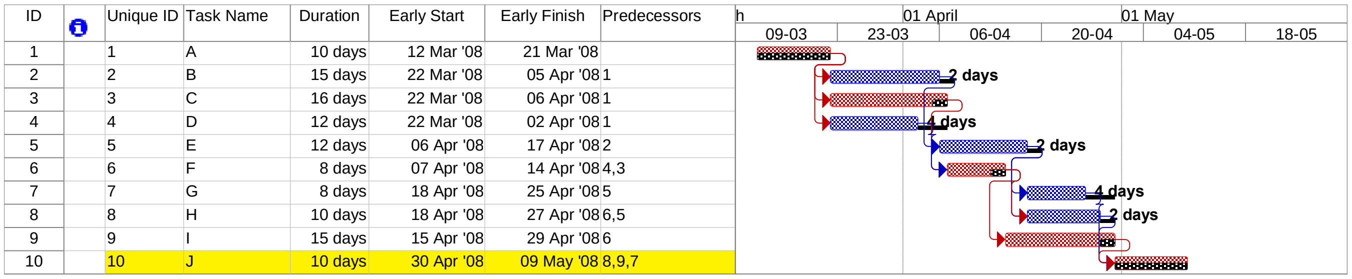
Professional Options
- [✓] Evaluate/Show Path Relative Float
- Rel. Float Limit - days away from driving: 100
- [ ] Include Cross-Project Links (SubProjects Only)
- [ ] Include a Target Task
- Target Task Unique ID:
- [ ] Permanently save the data for further analysis/presentation.
- [✓] Evaluate/Show Driving Task Drag
- [ ] Write Drag to Task Notes

Advanced Options
- [ ] Change Analysis Limit - no. of tasks: 100000

As for the Longest Path Filter, the options shown here will compute all available Drag components and then save the key steps and results of the calculations to a ñDrag Logò text file and the standard Run Log.

5.3 General Options for Displaying Drag Data

Whichever starting method is used, the analysis logs provide complete documentation of the results of the Drag calculations. In addition, BPC Logic Filter provides two general options for short term display and possible output (via printer, pdf-conversion, or image capture) of the Drag results. Using both options together provides full documentation of limited-scope analyses, where permanent storage of the results (in the MSP file) is not desired. The options are found as checkboxes on the General Display Options portion of either main entry form. Figures below are for the simple test schedule (NLD59DRAG) provided by Devaux.



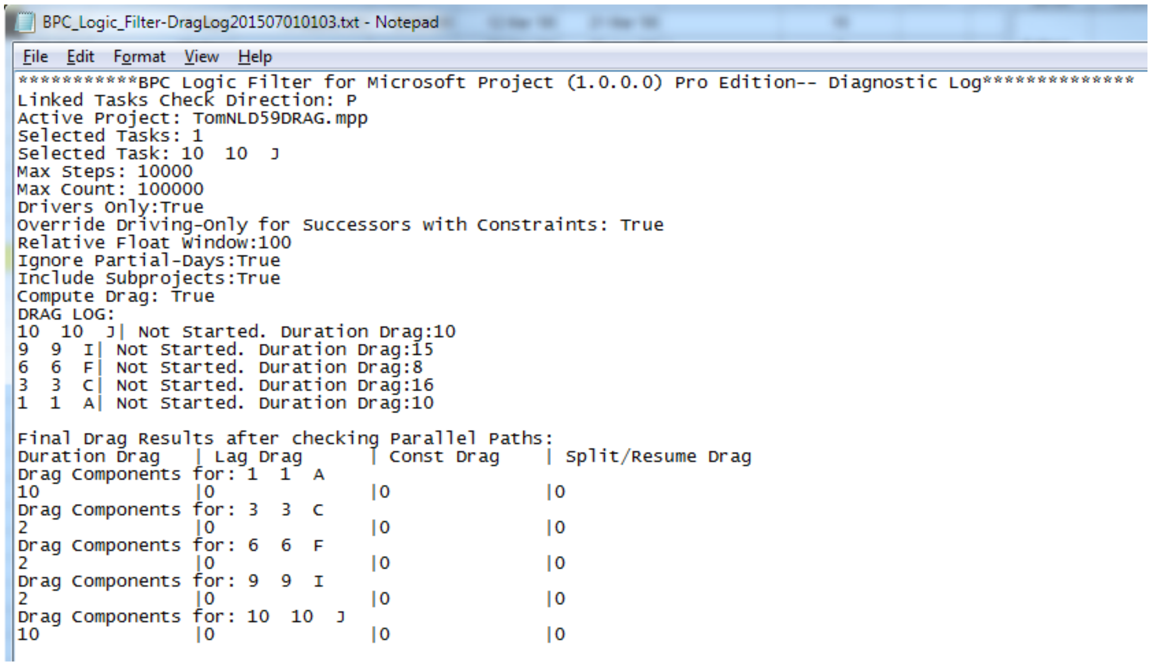
ID	ℹ	Unique ID	Task Name	Duration	Early Start	Early Finish	Predecessors
1		1	A	10 days	12 Mar '08	21 Mar '08	
2		2	B	15 days	22 Mar '08	05 Apr '08	1
3		3	C	16 days	22 Mar '08	06 Apr '08	1
4		4	D	12 days	22 Mar '08	02 Apr '08	1
5		5	E	12 days	06 Apr '08	17 Apr '08	2
6		6	F	8 days	07 Apr '08	14 Apr '08	4,3
7		7	G	8 days	18 Apr '08	25 Apr '08	5
8		8	H	10 days	18 Apr '08	27 Apr '08	6,5
9		9	I	15 days	15 Apr '08	29 Apr '08	6
10		10	J	10 days	30 Apr '08	09 May '08	8,9,7

Figure 3: NLD59DRAG - A Simple Schedule

5.3.1 Analysis Logs

The Drag Log is generated whenever Drag is computed and is stored as a separate, date/time-stamped text file in the default folder (normally [User]/My Documents/BPCLogicFilter). The Drag Log provides basic documentation of the Drag Analysis and is comprised of 3 main parts as seen in Figure 4:

1. An introductory section identifying the project, the selected completion task, and other main parameters.
2. The preliminary drag investigation, beginning with ñDRAG LOGò. Here the tasks are listed in reverse order, proceeding backward from the completion task, and preliminary findings are shown.
3. A tabular listing of the final Drag results after checking parallel paths. The list shows all four drag components: Duration, Lag, Constraint, and Split/Resume.

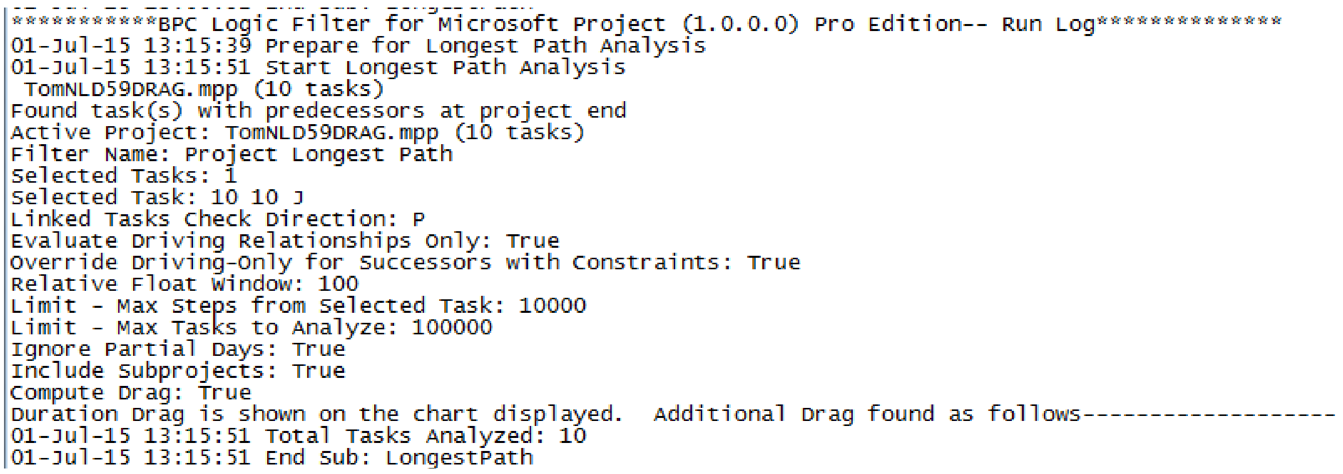


```
BPC_Logic_Filter-DragLog201507010103.txt - Notepad
File  Edit  Format  View  Help
***********BPC Logic Filter for Microsoft Project (1.0.0.0) Pro Edition-- Diagnostic Log**************
Linked Tasks Check Direction: P
Active Project: TomNLD59DRAG.mpp
Selected Tasks: 1
Selected Task: 10  10  J
Max Steps: 10000
Max Count: 100000
Drivers Only:True
Override Driving-Only for Successors with Constraints: True
Relative Float Window:100
Ignore Partial-Days:True
Include Subprojects:True
Compute Drag: True
DRAG LOG:
10  10  J| Not Started. Duration Drag:10
9  9  I| Not Started. Duration Drag:15
6  6  F| Not Started. Duration Drag:8
3  3  C| Not Started. Duration Drag:16
1  1  A| Not Started. Duration Drag:10

Final Drag Results after checking Parallel Paths:
Duration Drag    | Lag Drag       | Const Drag     | Split/Resume Drag
Drag Components for: 1  1  A
10               |0               |0               |0
Drag Components for: 3  3  C
2                |0               |0               |0
Drag Components for: 6  6  F
2                |0               |0               |0
Drag Components for: 9  9  I
2                |0               |0               |0
Drag Components for: 10  10  J
10               |0               |0               |0
```

Figure 4: NLD59DRAG Drag Log

The BPC Logic Filter Run Log is generated whenever the tool runs, but it is not stored as a separate file. Instead it is appended to a date-stamped DayLog text file that includes all the Run Logs for each day. The Run Log is designed as a brief summary of the analysis including noted exceptions, to be reviewed together with the filtered results. Drag possessed by Lags, Constraints, and Splits are considered exceptions and are therefore listed on the Run Log. Figure 5 shows the Run Log corresponding to the Drag Log above. Since this simple schedule included only Duration-related Drag, the Run Log shows no ñAdditional Drag foundé .ò

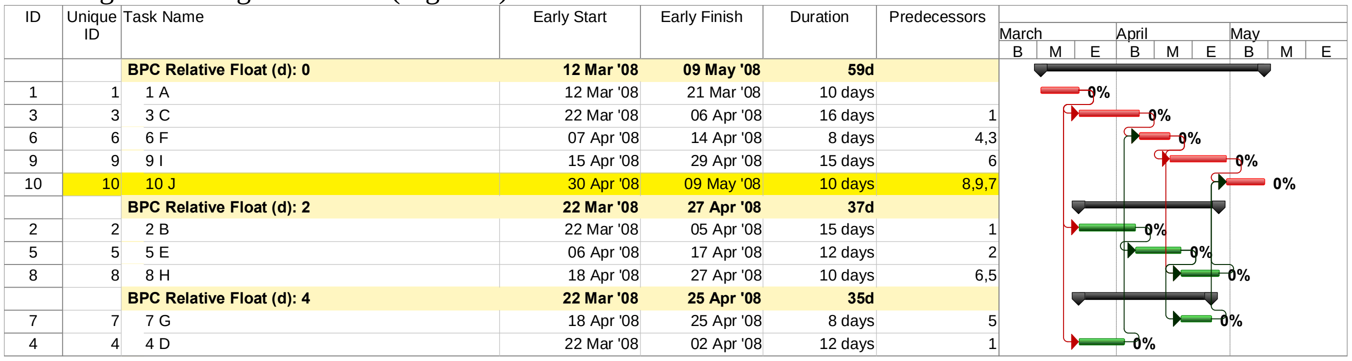


```
***********BPC Logic Filter for Microsoft Project (1.0.0.0) Pro Edition-- Run Log**************
01-Jul-15 13:15:39 Prepare for Longest Path Analysis
01-Jul-15 13:15:51 Start Longest Path Analysis
 TomNLD59DRAG.mpp (10 tasks)
Found task(s) with predecessors at project end
Active Project: TomNLD59DRAG.mpp (10 tasks)
Filter Name: Project Longest Path
Selected Tasks: 1
Selected Task: 10 10 J
Linked Tasks Check Direction: P
Evaluate Driving Relationships Only: True
Override Driving-Only for Successors with Constraints: True
Relative Float Window: 100
Limit - Max Steps from Selected Task: 10000
Limit - Max Tasks to Analyze: 100000
Ignore Partial Days: True
Include Subprojects: True
Compute Drag: True
Duration Drag is shown on the chart displayed.  Additional Drag found as follows-------------------
01-Jul-15 13:15:51 Total Tasks Analyzed: 10
01-Jul-15 13:15:51 End Sub: LongestPath
```

Figure 5: NLD59DRAG Run Log

5.3.2 Re-Sort to Show Logical Branches

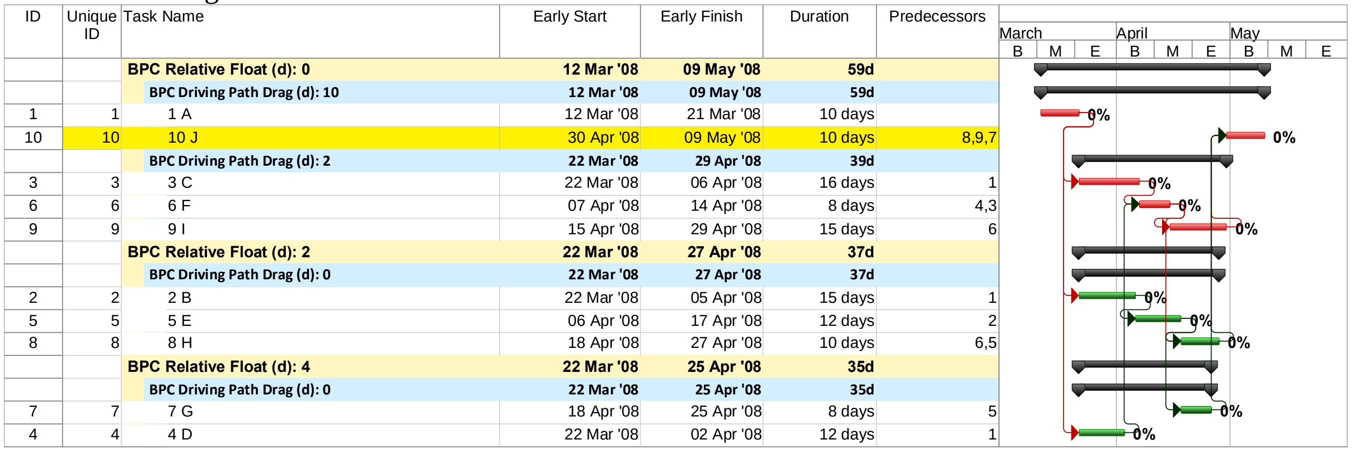
This is the default option for the general-purpose Task Logic Tracer. The option exists to provide a condensed view (ñBPC_MultipleFloatPathsò) of the driving and near-driving paths, which are grouped by Relative Float and sorted to distinguish the logic branches (Figure 6).



ID	Unique ID	Task Name	Early Start	Early Finish	Duration	Predecessors
		BPC Relative Float (d): 0	**12 Mar '08**	**09 May '08**	**59d**	
1	1	1 A	12 Mar '08	21 Mar '08	10 days	
3	3	3 C	22 Mar '08	06 Apr '08	16 days	1
6	6	6 F	07 Apr '08	14 Apr '08	8 days	4,3
9	9	9 I	15 Apr '08	29 Apr '08	15 days	6
10	10	10 J	30 Apr '08	09 May '08	10 days	8,9,7
		BPC Relative Float (d): 2	**22 Mar '08**	**27 Apr '08**	**37d**	
2	2	2 B	22 Mar '08	05 Apr '08	15 days	1
5	5	5 E	06 Apr '08	17 Apr '08	12 days	2
8	8	8 H	18 Apr '08	27 Apr '08	10 days	6,5
		BPC Relative Float (d): 4	**22 Mar '08**	**25 Apr '08**	**35d**	
7	7	7 G	18 Apr '08	25 Apr '08	8 days	5
4	4	4 D	22 Mar '08	02 Apr '08	12 days	1

Figure 6: NLD59DRAG - Default Logic Tracer View

When drag is calculated, this view is further grouped by ñBPC Driving Path Drag,ò which includes ONLY the taskôs Duration Drag component. Unfortunately, adding Drag has the side effect of disrupting the arrangement of logical branches. See Figure 7.

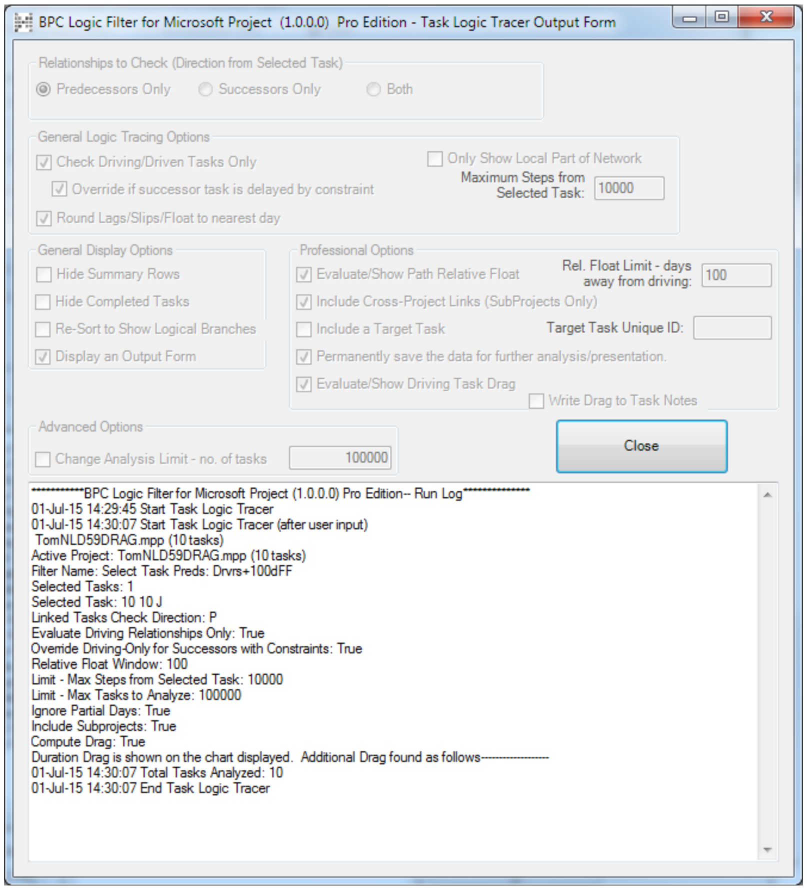


ID	Unique ID	Task Name	Early Start	Early Finish	Duration	Predecessors
		BPC Relative Float (d): 0	**12 Mar '08**	**09 May '08**	**59d**	
		BPC Driving Path Drag (d): 10	**12 Mar '08**	**09 May '08**	**59d**	
1	1	1 A	12 Mar '08	21 Mar '08	10 days	
10	10	10 J	30 Apr '08	09 May '08	10 days	8,9,7
		BPC Driving Path Drag (d): 2	**22 Mar '08**	**29 Apr '08**	**39d**	
3	3	3 C	22 Mar '08	06 Apr '08	16 days	1
6	6	6 F	07 Apr '08	14 Apr '08	8 days	4,3
9	9	9 I	15 Apr '08	29 Apr '08	15 days	6
		BPC Relative Float (d): 2	**22 Mar '08**	**27 Apr '08**	**37d**	
		BPC Driving Path Drag (d): 0	**22 Mar '08**	**27 Apr '08**	**37d**	
2	2	2 B	22 Mar '08	05 Apr '08	15 days	1
5	5	5 E	06 Apr '08	17 Apr '08	12 days	2
8	8	8 H	18 Apr '08	27 Apr '08	10 days	6,5
		BPC Relative Float (d): 4	**22 Mar '08**	**25 Apr '08**	**35d**	
		BPC Driving Path Drag (d): 0	**22 Mar '08**	**25 Apr '08**	**35d**	
7	7	7 G	18 Apr '08	25 Apr '08	8 days	5
4	4	4 D	22 Mar '08	02 Apr '08	12 days	1

Figure 7: NLD59DRAG - Resorted with Drag

5.3.3 Display an Output Form

When selected, this option displays an inactive form to identify the selected parameters as well as the standard Run Log for the analysis. If non-duration Drag components have been found, these computed values are included in the Run Log and hence on the output form (Figure 8). The form exists for quick (image) capture and subsequent insertion into an associated pdf file. It may also be attached to the completion task or any other task in the MSP file.



BPC Logic Filter for Microsoft Project (1.0.0.0) Pro Edition - Task Logic Tracer Output Form

Relationships to Check (Direction from Selected Task)
[✓ Predecessors Only] Successors Only Both

General Logic Tracing Options
- ✓ Check Driving/Driven Tasks Only
  - ✓ Override if successor task is delayed by constraint
- ✓ Round Lags/Slips/Float to nearest day
- ☐ Only Show Local Part of Network
- Maximum Steps from Selected Task: 10000

General Display Options
- ☐ Hide Summary Rows
- ☐ Hide Completed Tasks
- ☐ Re-Sort to Show Logical Branches
- ✓ Display an Output Form

Professional Options
- ✓ Evaluate/Show Path Relative Float
- Rel. Float Limit - days away from driving: 100
- ✓ Include Cross-Project Links (SubProjects Only)
- ☐ Include a Target Task
- Target Task Unique ID:
- ✓ Permanently save the data for further analysis/presentation.
- ✓ Evaluate/Show Driving Task Drag
- ☐ Write Drag to Task Notes

Advanced Options
- ☐ Change Analysis Limit - no. of tasks 100000

```
***********BPC Logic Filter for Microsoft Project (1.0.0.0) Pro Edition-- Run Log**************
01-Jul-15 14:29:45 Start Task Logic Tracer
01-Jul-15 14:30:07 Start Task Logic Tracer (after user input)
 TomNLD59DRAG.mpp (10 tasks)
Active Project: TomNLD59DRAG.mpp (10 tasks)
Filter Name: Select Task Preds: Drvrs+100dFF
Selected Tasks: 1
Selected Task: 10 10 J
Linked Tasks Check Direction: P
Evaluate Driving Relationships Only: True
Override Driving-Only for Successors with Constraints: True
Relative Float Window: 100
Limit - Max Steps from Selected Task: 10000
Limit - Max Tasks to Analyze: 100000
Ignore Partial Days: True
Include Subprojects: True
Compute Drag: True
Duration Drag is shown on the chart displayed.  Additional Drag found as follows----------------
01-Jul-15 14:30:07 Total Tasks Analyzed: 10
01-Jul-15 14:30:07 End Task Logic Tracer
```

Figure 8: NLD59DRAG Output Form

5.4 General Options for Saving Drag Data

In general, permanent storage (within project files) of the results generated by BPC Logic Filter is neither encouraged nor necessary for the purposes of the program. Nevertheless, there are two options for saving most of the computed Drag values for later integration with Drag Cost analysis. These options are found as checkboxes on the Professional Options portion of either main entry form.

5.4.1 Permanently save the data for further analysis/presentation

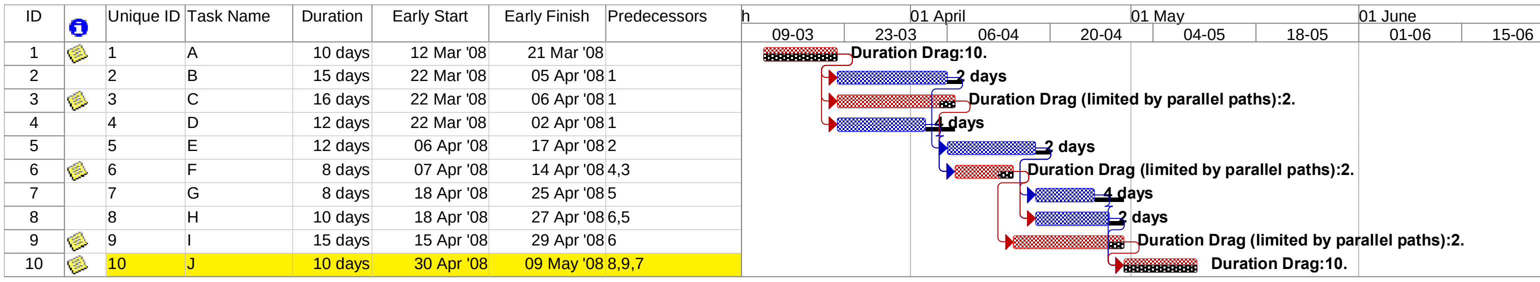
[This option is not available in the Trial version of BPC Logic Filter.]

If this option is selected, then the BPC Driving Path Drag field from the analysis is stored in the MSP file. (This field includes only the Duration-Drag component.) The data may then be copy-pasted or exported to excel or some other tool for drag cost integration. Figures in this paper that show a task-table column for ñBPC Driving Path Drag (d)ò were generated with this option enabled.

To permanently retain these data in the MSP file, the user must immediately rename the field to keep the values from being overwritten or deleted the next time that BPC Logic Filter is run.

5.4.2 Write Drag to Task Notes

If this option is selected, then the computed Drag values (including all components of the Drag) will be appended as text in the ñNotesò of the affected tasks. Figure 9 illustrates the consequence of this option for the simple project (after the ñCritical Taskò bar style has been modified to display the notes). This approach seems not optimal for most but may be useful for some. To reduce the chances of repetitive duplicate entries, the ñWrite Drag to Task Notesò selection does not persist in the entry forms ï the user must actively select it each time. It is also never noted on the output form.



ID	ℹ	Unique ID	Task Name	Duration	Early Start	Early Finish	Predecessors
1	📝	1	A	10 days	12 Mar '08	21 Mar '08	
2		2	B	15 days	22 Mar '08	05 Apr '08	1
3	📝	3	C	16 days	22 Mar '08	06 Apr '08	1
4		4	D	12 days	22 Mar '08	02 Apr '08	1
5		5	E	12 days	06 Apr '08	17 Apr '08	2
6	📝	6	F	8 days	07 Apr '08	14 Apr '08	4,3
7		7	G	8 days	18 Apr '08	25 Apr '08	5
8		8	H	10 days	18 Apr '08	27 Apr '08	6,5
9	📝	9	I	15 days	15 Apr '08	29 Apr '08	6
10	📝	10	J	10 days	30 Apr '08	09 May '08	8,9,7

h
01 April
01 May
01 June
09-03
23-03
06-04
20-04
04-05
18-05
01-06
15-06
Duration Drag:10.
2 days
Duration Drag (limited by parallel paths):2.
4 days
2 days
Duration Drag (limited by parallel paths):2.
4 days
2 days
Duration Drag (limited by parallel paths):2.
Duration Drag:10.

Figure 9: NLD59DRAG - Notes on Bars

5.5 Selection of Drag Calendar

Beginning with Release 1.1.1.14, BPC Logic Filter uses a brute force method to compute duration Drag based on actual impact on the selected/end task. By default, this impact is measured according to the calendar of the selected/end task. It is possible, however, to compute the impact using either the calendar of the local task or the project default calendar. The calendar to be used is selected in the analysis settings window.

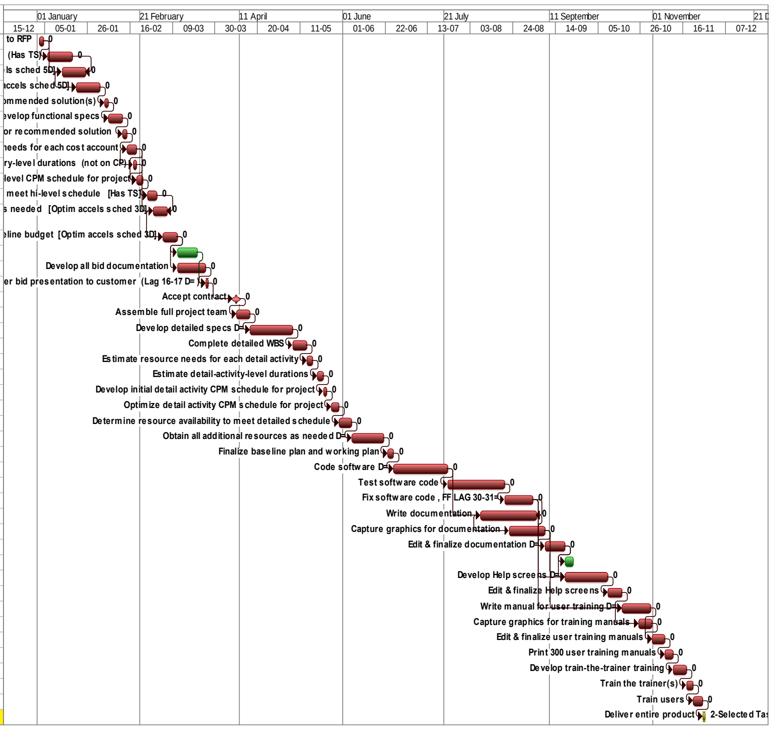
6.0 Appendix – Test Cases

The following pages display the results of running BPC Logic Filter with Drag analysis on additional test files provided by Mr. Devaux. Each example shows the unfiltered bar chart with tabulated data, followed by the Run Log and the Drag Log for each analysis. If included in the table, a ñDrag Specialò indicator is used to show that the taskôs duration drag has been limited compared to the task duration.

1. **RevisedOPTIMSAMPLEWALKTHRUDRAG&PROGRESS.mpp**
2. **Teamlayoutproject73DRAG.mpp**
3. **56dCOMPLEX EXER.mpp**
4. **5ACTss52.mpp**
5. **Reverse crit test.mpp**
6. **Test 2.mpp**

RevisedOPTIMSAMPLEWALKTHRUDRAG&PROGRESS.mpp

ID	ID	Unique ID	Task Name	Duration	Start	Finish	Predecessors	Successors	BPC Driving Path Drag (d)
1	1	1	Decide to respond to RFP	2 days	02-01-14 08:00	03-01-14 17:00		2,3	2
2	2	2	Assemble Bid & Proposal team (Has TS)	10 days	06-01-14 08:00	17-01-14 17:00	1	3FF+5 days	5
3	3	3	Brainstorm possible solutions [Optim accels sched 5D]	10 days	13-01-14 08:00	24-01-14 17:00	2FF+5 days,1	4SS+5 days	-5
4	4	4	Analyze products as possible solutions [Optim accels sched 5D]	10 days	20-01-14 08:00	31-01-14 17:00	3SS+5 days	5	10
5	5	5	Prioritize recommended solution(s)	2 days	03-02-14 08:00	04-02-14 17:00	4	6	2
6	6	6	Develop functional specs	5 days	05-02-14 08:00	11-02-14 17:00	5	7	5
7	7	7	Develop hi-level project WBS for recommended solution	2 days	12-02-14 08:00	13-02-14 17:00	6	8	2
8	8	8	Estimate resource needs for each cost account	3 days	14-02-14 08:00	18-02-14 17:00	7	9SS+1 day,10,12	0
9	9	9	Estimate summary-level durations (not on CP)	2 days	17-02-14 08:00	18-02-14 17:00	8SS+1 day	10	0
10	10	10	Develop hi-level CPM schedule for project	3 days	19-02-14 08:00	21-02-14 17:00	8,9	11	3
11	11	11	Determine resource availability to meet hi-level schedule [Has TS]	5 days	24-02-14 08:00	28-02-14 17:00	10	12FF+3 days	5
12	12	12	Arrange for acquiring the additional resources needed [Optim accels sched 3D]	5 days	27-02-14 08:00	05-03-14 17:00	11FF+3 days,8	13SS+3 days	-6
13	13	13	Develop baseline budget [Optim accels sched 3D]	5 days	04-03-14 08:00	10-03-14 17:00	12SS+3 days	14,15	5
14	14	14	Develop proposed contract	8 days	11-03-14 08:00	20-03-14 17:00	13	16	0
15	15	15	Develop all bid documentation	10 days	11-03-14 08:00	24-03-14 17:00	13	16	2
16	16	16	Deliver bid presentation to customer (Lag 16-17 D=)	1 day	25-03-14 08:00	25-03-14 17:00	14,15	17FS+10 days	1
17	17	17	Accept contract	0 days	08-04-14 17:00	08-04-14 17:00	16FS+10 days	18	0
18	18	18	Assemble full project team	5 days	09-04-14 08:00	15-04-14 17:00	17	19	5
19	19	19	Develop detailed specs D=	15 days	16-04-14 08:00	06-05-14 17:00	18	20	15
20	20	20	Complete detailed WBS	5 days	07-05-14 08:00	13-05-14 17:00	19	21	5
21	21	21	Estimate resource needs for each detail activity	3 days	14-05-14 08:00	16-05-14 17:00	20	22	3
22	22	22	Estimate detail-activity-level durations	3 days	19-05-14 08:00	21-05-14 17:00	21	23	3
23	23	23	Develop initial detail activity CPM schedule for project	2 days	22-05-14 08:00	23-05-14 17:00	22	24	2
24	24	24	Optimize detail activity CPM schedule for project	4 days	26-05-14 08:00	29-05-14 17:00	23	25	4
25	25	25	Determine resource availability to meet detailed schedule	4 days	30-05-14 08:00	04-06-14 17:00	24	26	4
26	26	26	Obtain all additional resources as needed D=	12 days	05-06-14 08:00	20-06-14 17:00	25	27	12
27	27	27	Finalize baseline plan and w orking plan	3 days	23-06-14 08:00	25-06-14 17:00	26	28	3
28	28	28	Code softw are D=	18.75 days	26-06-14 08:00	22-07-14 15:00	27	29,31	18.75
29	29	29	Test softw are code	20 days	22-07-14 15:00	19-08-14 15:00	28	30	12
30	30	44	Fix softw are code , FF LAG 30-31=	10 days	19-08-14 15:00	02-09-14 15:00	29	1FF+2 days,35,37	10
31	31	45	Write documentation	20 days	07-08-14 15:00	04-09-14 15:00	28,30FF+2 days	32SS+10 days,33	-2
32	32	46	Capture graphics for documentation	12 days	21-08-14 15:00	08-09-14 15:00	31SS+10 days	33,37	2
33	33	47	Edit & finalize documentation D=	8 days	08-09-14 15:00	18-09-14 15:00	32,31	34,35	8
34	34	48	Print 300 copies of documentation	2 days	18-09-14 15:00	22-09-14 15:00	33		0
35	35	49	Develop Help screens D=	15 days	18-09-14 15:00	09-10-14 15:00	33,30	36	15
36	36	50	Edit & finalize Help screens	5 days	09-10-14 15:00	16-10-14 15:00	35	37	5
37	37	51	Write manual for user training D=	10 days	16-10-14 15:00	30-10-14 15:00	32,36,30	38SS+6 days,39	0
38	38	52	Capture graphics for training manuals	5 days	24-10-14 15:00	31-10-14 15:00	37SS+6 days	39	1
39	39	53	Edit & finalize user training manuals	4 days	31-10-14 15:00	06-11-14 15:00	37,38	40	4
40	40	54	Print 300 user training manuals	2 days	06-11-14 15:00	10-11-14 15:00	39	41	2
41	41	55	Develop train-the-trainer training	5 days	10-11-14 15:00	17-11-14 15:00	40	42	5
42	42	56	Train the trainer(s)	3 days	17-11-14 15:00	20-11-14 15:00	41	43	3
43	43	57	Train users	3 days	20-11-14 15:00	25-11-14 15:00	42	44	3
44	44	58	Deliver entire product	1 day	25-11-14 15:00	26-11-14 15:00	43		1



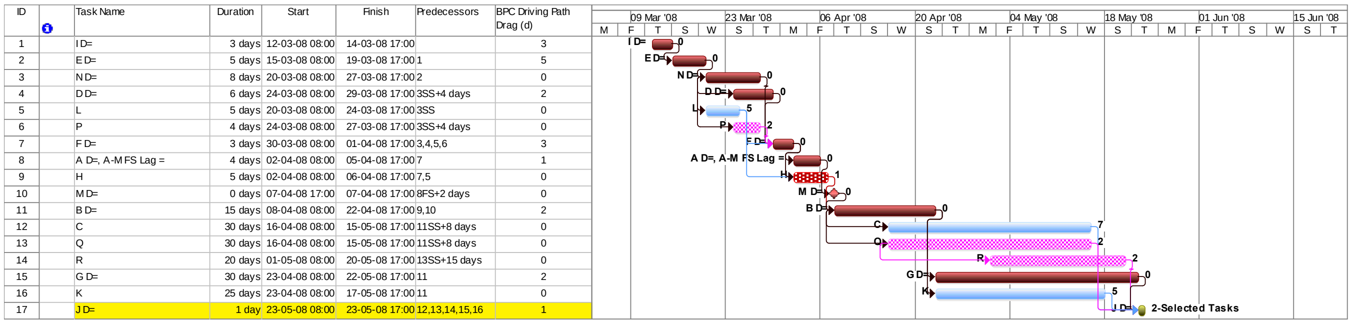
```
***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Run Log**************
Licensed to User:
Session Run: 1
20-Sep-16 14:57:51 Prepare for Longest Path Analysis
20-Sep-16 14:57:58 Start Longest Path Analysis
 Tom RevisedOPTIMSAMPLEWALKTHRUDRAG&PROGRESS.mpp (44 tasks)
Found task(s) with predecessors at project end
Active Project: Tom RevisedOPTIMSAMPLEWALKTHRUDRAG&PROGRESS.mpp (44 tasks)
Filter Name: BPC_ProjectLongestPath
Selected Tasks: 1
Selected Task: 44 58 Deliver entire product
Linked Tasks Check Direction: P
Evaluate Driving Relationships Only: True
Override Driving-Only for Successors with Constraints: True
Relative Float Window: 100
Limit - Max Steps from Selected Task: 10000
Limit - Max Tasks to Analyze: 1000001
Ignore Partial Days: True
Include Subprojects: True
Compute Drag: True Drag Calendar: 1
Check Resource Leveling Drivers: False
Check Resource Parallel Paths: False
Color Bars: True
In-Line Only: True
Duration Drag is shown on the chart displayed.  Additional Drag found as follows-------------------
20-Sep-16 14:58:02 Total Tasks Analyzed: 43
20-Sep-16 14:58:02 End Longest Path Filter.  Total Time: 00:00:11.8846798  Analysis Time: 00:00:04.2902454

***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Diagnostic Log**************
Licensed to User:
Session Run: 1
20-Sep-16 14:57:51 Start Sub LongestPath after license check.  Time Elapsed: 00:00:00.4080234
Tom RevisedOPTIMSAMPLEWALKTHRUDRAG&PROGRESS.mpp ClearDataFields Elapsed Time: 00:00:00.1180068
20-Sep-16 14:57:58Finding Last Task(s).
Found task(s) with predecessors at project end
Sorting Selected Tasks: 1
LastTasks Elapsed Time: 00:00:00.0260015
20-Sep-16 14:57:58 Starting Filter Controller.
DRAG LOG:
44  58  Deliver entire product | Not Started. Duration Drag:1
```

```
43  57  Train users | Not Started. Duration Drag:3
42  56  Train the trainer(s) | Not Started. Duration Drag:3
41  55  Develop train-the-trainer training | Not Started. Duration Drag:5
40  54  Print 300 user training manuals | Not Started. Duration Drag:2
39  53  Edit & finalize user training manuals | Not Started. Duration Drag:4
38  52  Capture graphics for training manuals | Predecessor Lag Drag:6
38  52  Capture graphics for training manuals | Not Started. Duration Drag:5
37  51  Write manual for user training D=| Start is both Driven and Driving. No Drag:0
36  50  Edit & finalize Help screens | Not Started. Duration Drag:5
35  49  Develop Help screens D=| Not Started. Duration Drag:15
33  47  Edit & finalize documentation D=| Not Started. Duration Drag:8
32  46  Capture graphics for documentation | Predecessor Lag Drag:10
32  46  Capture graphics for documentation | Not Started. Duration Drag:12
31  45  Write documentation | Predecessor Lag Drag:2
31  45  Write documentation | Driven Finish and Driving Start. Negative Drag: -1
30  44  Fix software code , FF LAG 30-31=| Not Started. Duration Drag:10
29  29  Test software code | Not Started. Duration Drag:20
28  28  Code software D=| Not Started. Duration Drag:18.75
27  27  Finalize baseline plan and working plan | Not Started. Duration Drag:3
26  26  Obtain all additional resources as needed D=| Not Started. Duration Drag:12
25  25  Determine resource availability to meet detailed schedule | Not Started. Duration Drag:4
24  24  Optimize detail activity CPM schedule for project | Not Started. Duration Drag:4
23  23  Develop initial detail activity CPM schedule for project | Not Started. Duration Drag:2
22  22  Estimate detail-activity-level durations | Not Started. Duration Drag:3
21  21  Estimate resource needs for each detail activity | Not Started. Duration Drag:3
20  20  Complete detailed WBS | Not Started. Duration Drag:5
19  19  Develop detailed specs D=| Not Started. Duration Drag:15
18  18  Assemble full project team | Not Started. Duration Drag:5
17  17  Accept contract| Predecessor Lag Drag:10
17  17  Accept contract| Not Started. Duration Drag:0
16  16  Deliver bid presentation to customer  (Lag 16-17 D= )| Not Started. Duration Drag:1
15  15  Develop all bid documentation | Not Started. Duration Drag:10
13  13  Develop baseline budget  [Optim accels sched 3D]| Predecessor Lag Drag:3
13  13  Develop baseline budget  [Optim accels sched 3D]| Not Started. Duration Drag:5
12  12  Arrange for acquiring the additional resources needed   [Optim accels sched 3D]| Predecessor Lag Drag:3
12  12  Arrange for acquiring the additional resources needed   [Optim accels sched 3D]| Driven Finish and Driving Start. Negative Drag: -1
11  11  Determine resource availability to meet hi-level schedule   [Has TS]| Not Started. Duration Drag:5
10  10  Develop hi-level CPM schedule for project| Not Started. Duration Drag:3
9  9  Estimate summary-level durations  (not on CP)| Predecessor Lag Drag:1
9  9  Estimate summary-level durations  (not on CP)| Not Started. Duration Drag:2
8  8  Estimate resource needs for each cost account | Start is both Driven and Driving. No Drag:0
7  7  Develop hi-level project WBS for recommended solution  | Not Started. Duration Drag:2
6  6  Develop functional specs | Not Started. Duration Drag:5
5  5  Prioritize recommended solution(s) | Not Started. Duration Drag:2
4  4  Analyze products as possible solutions  [Optim accels sched 5D]| Predecessor Lag Drag:5
4  4  Analyze products as possible solutions  [Optim accels sched 5D]| Not Started. Duration Drag:10
3  3  Brainstorm possible solutions  [Optim accels sched 5D]| Predecessor Lag Drag:5
3  3  Brainstorm possible solutions  [Optim accels sched 5D]| Driven Finish and Driving Start. Negative Drag: -1
2  2  Assemble Bid & Proposal team  (Has TS)| Not Started. Duration Drag:10
1  1  Decide to respond to RFP | Not Started. Duration Drag:2
3  3  Brainstorm possible solutions  [Optim accels sched 5D]| Negative Drag Constrained by Start predecessor float:-5
3  3  Brainstorm possible solutions  [Optim accels sched 5D]| DragCalc Driving Path Drag Branch: RootNode: 39.9 RelFlt:5 to:1  1  Decide to respond to
RFP  ReturnNode: 41.9
10  10  Develop hi-level CPM schedule for project| DragCalc Driving Path Drag Branch: RootNode: 33.1 RelFlt:0 to:8  8  Estimate resource needs for each
cost account  ReturnNode: 34.9
12  12  Arrange for acquiring the additional resources needed   [Optim accels sched 3D]| Negative Drag Constrained by Start predecessor float:-6
12  12  Arrange for acquiring the additional resources needed   [Optim accels sched 3D]| DragCalc Driving Path Drag Branch: RootNode: 30.9 RelFlt:6 to:8
8  Estimate resource needs for each cost account  ReturnNode: 34.9
16  16  Deliver bid presentation to customer  (Lag 16-17 D= )| DragCalc Driving Path Drag Branch: RootNode: 28.1 RelFlt:2 to:14  14  Develop proposed
contract ReturnNode: 0
14  14  Develop proposed contract| Driving Path Drag Return: RootNode:28.1 RelFlt:2 ReturnNode:29.9
31  45  Write documentation | Negative Drag Constrained by Start predecessor float:-12
31  45  Write documentation | DragCalc Driving Path Drag Branch: RootNode: 12.9 RelFlt:12 to:28  28  Code software D= ReturnNode: 15.9
33  47  Edit & finalize documentation D=| DragCalc Driving Path Drag Branch: RootNode: 11.1 RelFlt:2 to:31  45  Write documentation  ReturnNode: 13.1
35  49  Develop Help screens D=| DragCalc Driving Path Drag Branch: RootNode: 10.1 RelFlt:12 to:30  44  Fix software code , FF LAG 30-31=
ReturnNode: 13.9
37  51  Write manual for user training D=| DragCalc Driving Path Drag Branch: RootNode: 8.1 RelFlt:28 to:32  46  Capture graphics for documentation
ReturnNode: 11.9
37  51  Write manual for user training D=| DragCalc Driving Path Drag Branch: RootNode: 8.1 RelFlt:32 to:30  44  Fix software code , FF LAG 30-31=
ReturnNode: 13.9
39  53  Edit & finalize user training manuals | DragCalc Driving Path Drag Branch: RootNode: 6.1 RelFlt:1 to:37  51  Write manual for user training D=
ReturnNode: 7.9
```

```
Final Drag Results after checking Parallel Paths:
Duration Drag          | Lag Drag | Const Drag              | Split/Resume Drag| Leveling Drag (edays)
Implementing Brute Force Analysis for Duration Drag.
Drag Calendar: Selected/End Task
Decreasing Normal Logic: 2d 1  1  Decide to respond to RFP
Decreasing Normal Logic: 5d 2  2  Assemble Bid & Proposal team  (Has TS)
Re-Setting Drag Based on Parallel Paths:5d 2  2  Assemble Bid & Proposal team  (Has TS)
Reverse Logic Flow - Setting Negative Drag: -5d 3  3  Brainstorm possible solutions  [Optim accels sched 5D]
Decreasing Normal Logic: 10d 4  4  Analyze products as possible solutions  [Optim accels sched 5D]
Decreasing Normal Logic: 2d 5  5  Prioritize recommended solution(s)
Decreasing Normal Logic: 5d 6  6  Develop functional specs
Decreasing Normal Logic: 2d 7  7  Develop hi-level project WBS for recommended solution
Re-Setting Drag Based on Parallel Paths:0d 9  9  Estimate summary-level durations  (not on CP)
Decreasing Normal Logic: 3d 10  10  Develop hi-level CPM schedule for project
Decreasing Normal Logic: 5d 11  11  Determine resource availability to meet hi-level schedule   [Has TS]
Reverse Logic Flow - Setting Negative Drag: -6d 12  12  Arrange for acquiring the additional resources needed   [Optim accels sched 3D]
Decreasing Normal Logic: 5d 13  13  Develop baseline budget  [Optim accels sched 3D]
Decreasing Normal Logic: 2d 15  15  Develop all bid documentation
Re-Setting Drag Based on Parallel Paths:2d 15  15  Develop all bid documentation
Decreasing Normal Logic: 1d 16  16  Deliver bid presentation to customer  (Lag 16-17 D= )
Decreasing Normal Logic: 5d 18  18  Assemble full project team
Decreasing Normal Logic: 15d 19  19  Develop detailed specs D=
Decreasing Normal Logic: 5d 20  20  Complete detailed WBS
Decreasing Normal Logic: 3d 21  21  Estimate resource needs for each detail activity
Decreasing Normal Logic: 3d 22  22  Estimate detail-activity-level durations
Decreasing Normal Logic: 2d 23  23  Develop initial detail activity CPM schedule for project
Decreasing Normal Logic: 4d 24  24  Optimize detail activity CPM schedule for project
Decreasing Normal Logic: 4d 25  25  Determine resource availability to meet detailed schedule
Decreasing Normal Logic: 12d 26  26  Obtain all additional resources as needed D=
Decreasing Normal Logic: 3d 27  27  Finalize baseline plan and working plan
Decreasing Normal Logic: 18.75d 28  28  Code software D=
Decreasing Normal Logic: 12d 29  29  Test software code
Re-Setting Drag Based on Parallel Paths:12d 29  29  Test software code
Decreasing Normal Logic: 10d 30  44  Fix software code , FF LAG 30-31=
Reverse Logic Flow - Setting Negative Drag: -2d 31  45  Write documentation
Re-Setting Drag Based on Parallel Paths:-2d 31  45  Write documentation
Decreasing Normal Logic: 2d 32  46  Capture graphics for documentation
Re-Setting Drag Based on Parallel Paths:2d 32  46  Capture graphics for documentation
Decreasing Normal Logic: 8d 33  47  Edit & finalize documentation D=
Decreasing Normal Logic: 15d 35  49  Develop Help screens D=
Decreasing Normal Logic: 5d 36  50  Edit & finalize Help screens
Decreasing Normal Logic: 1d 38  52  Capture graphics for training manuals
Re-Setting Drag Based on Parallel Paths:1d 38  52  Capture graphics for training manuals
Decreasing Normal Logic: 4d 39  53  Edit & finalize user training manuals
Decreasing Normal Logic: 2d 40  54  Print 300 user training manuals
Decreasing Normal Logic: 5d 41  55  Develop train-the-trainer training
Decreasing Normal Logic: 3d 42  56  Train the trainer(s)
Decreasing Normal Logic: 3d 43  57  Train users
Decreasing Normal Logic: 1d 44  58  Deliver entire product
Diagnostic Log Elapsed Time: 00:00:00.0010001
```

Teamlayoutproject73DRAG.mpp



ID	ℹ	Task Name	Duration	Start	Finish	Predecessors	BPC Driving Path Drag (d)
1		I D=	3 days	12-03-08 08:00	14-03-08 17:00		3
2		E D=	5 days	15-03-08 08:00	19-03-08 17:00	1	5
3		N D=	8 days	20-03-08 08:00	27-03-08 17:00	2	0
4		D D=	6 days	24-03-08 08:00	29-03-08 17:00	3SS+4 days	2
5		L	5 days	20-03-08 08:00	24-03-08 17:00	3SS	0
6		P	4 days	24-03-08 08:00	27-03-08 17:00	3SS+4 days	0
7		F D=	3 days	30-03-08 08:00	01-04-08 17:00	3,4,5,6	3
8		A D=, A-M FS Lag =	4 days	02-04-08 08:00	05-04-08 17:00	7	1
9		H	5 days	02-04-08 08:00	06-04-08 17:00	7,5	0
10		M D=	0 days	07-04-08 17:00	07-04-08 17:00	8FS+2 days	0
11		B D=	15 days	08-04-08 08:00	22-04-08 17:00	9,10	2
12		C	30 days	16-04-08 08:00	15-05-08 17:00	11SS+8 days	0
13		Q	30 days	16-04-08 08:00	15-05-08 17:00	11SS+8 days	0
14		R	20 days	01-05-08 08:00	20-05-08 17:00	13SS+15 days	0
15		G D=	30 days	23-04-08 08:00	22-05-08 17:00	11	2
16		K	25 days	23-04-08 08:00	17-05-08 17:00	11	0
17		J D=	1 day	23-05-08 08:00	23-05-08 17:00	12,13,14,15,16	1

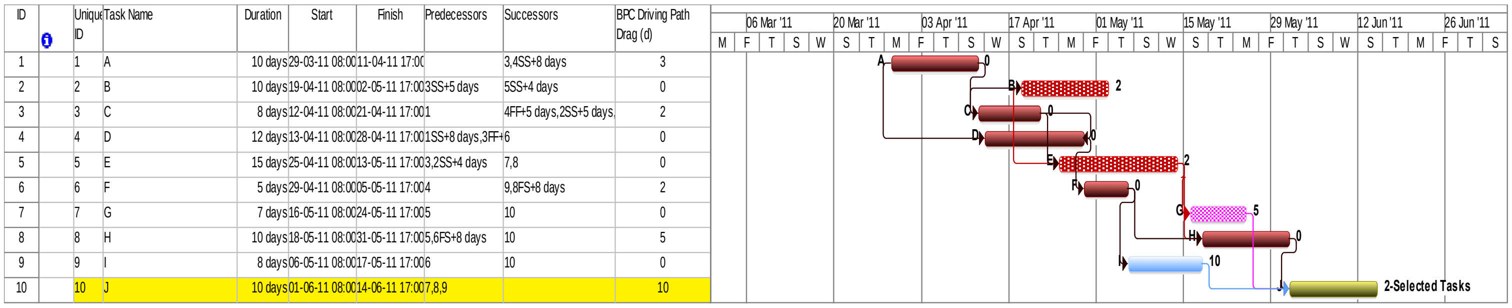
```
***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Run Log**************
Licensed to User:
Session Run: 2
20-Sep-16 15:03:45 Start Task Logic Tracer
20-Sep-16 15:03:56 Start Task Logic Tracer (after user input)
 Tom Teamlayoutproject73DRAG.mpp (17 tasks)
Active Project: Tom Teamlayoutproject73DRAG.mpp (17 tasks)
Filter Name: BPC_Select Task Preds: Drvrs+100dFF
Selected Tasks: 1
Selected Task: 17 17 J D=
Linked Tasks Check Direction: P
Evaluate Driving Relationships Only: True
Override Driving-Only for Successors with Constraints: False
Relative Float Window: 100
Limit - Max Steps from Selected Task: 10000
Limit - Max Tasks to Analyze: 1000001
Ignore Partial Days: True
Include Subprojects: True
Compute Drag: True Drag Calendar: 1
Check Resource Leveling Drivers: False
Check Resource Parallel Paths: False
Color Bars: True
In-Line Only: True
Duration Drag is shown on the chart displayed.  Additional Drag found as follows-------------------
20-Sep-16 15:03:57 Total Tasks Analyzed: 17
20-Sep-16 15:03:58 End Task Logic Tracer.  Total Time: 00:00:12.8507350  Analysis Time: 00:00:01.0810618
```

```
***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Diagnostic Log**************
Licensed to User:
Session Run: 2
20-Sep-16 15:03:45 Start Sub LinkedTasksControl
Tom Teamlayoutproject73DRAG.mpp ClearDataFields Elapsed Time: 00:00:00.0480027
20-Sep-16 15:03:57 Starting Filter Controller.
DRAG LOG:
17  17  J D=| Not Started. Duration Drag:1
15  15  G D=| Not Started. Duration Drag:30
11  11  B D=| Not Started. Duration Drag:15
10  10  M D=| Predecessor Lag Drag:2
10  10  M D=| Not Started. Duration Drag:0
8  8  A D=, A-M FS Lag = | Not Started. Duration Drag:4
7  7  F D=| Not Started. Duration Drag:3
4  4  D D=| Predecessor Lag Drag:4
4  4  D D=| Not Started. Duration Drag:6
3  3  N D=| Start is both Driven and Driving. No Drag:0
2  2  E D=| Not Started. Duration Drag:5
1  1  I D=| Not Started. Duration Drag:3
7  7  F D=| DragCalc Driving Path Drag Branch: RootNode: 6.1 RelFlt:2 to:6  6  P ReturnNode: 0
6  6  P| Driving Path Drag Return: RootNode:6.1 RelFlt:2 ReturnNode:8.1
7  7  F D=| DragCalc Driving Path Drag Branch: RootNode: 6.1 RelFlt:2 to:3  3  N D= ReturnNode: 7.9
7  7  F D=| DragCalc Driving Path Drag Branch: RootNode: 6.1 RelFlt:5 to:5  5  L ReturnNode: 0
5  5  L| Driving Path Drag Return: RootNode:6.1 RelFlt:5 ReturnNode:8.1
11  11  B D=| DragCalc Driving Path Drag Branch: RootNode: 3.1 RelFlt:1 to:9  9  H ReturnNode: 0
9  9  H| Driving Path Drag Return: RootNode:3.1 RelFlt:1 ReturnNode:5.9
17  17  J D=| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:2 to:14  14  R ReturnNode: 0
13  13  Q| Driving Path Drag Return: RootNode:1.1 RelFlt:2 ReturnNode:3.1
17  17  J D=| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:5 to:16  16  K ReturnNode: 0
16  16  K| Driving Path Drag Return: RootNode:1.1 RelFlt:5 ReturnNode:2.9
```

17 17 J D=| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:7 to:13 13 Q ReturnNode: 14.9
17 17 J D=| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:7 to:12 12 C ReturnNode: 0
12 12 C| Driving Path Drag Return: RootNode:1.1 RelFlt:7 ReturnNode:3.1

Final Drag Results after checking Parallel Paths:
Duration Drag | Lag Drag | Const Drag | Split/Resume Drag| Leveling Drag (edays)
Implementing Brute Force Analysis for Duration Drag.
Drag Calendar: Selected/End Task
Decreasing Normal Logic: 3d 1 1 I D=
Decreasing Normal Logic: 5d 2 2 E D=
Decreasing Normal Logic: 2d 4 4 D D=
Re-Setting Drag Based on Parallel Paths:2d 4 4 D D=
Decreasing Normal Logic: 3d 7 7 F D=
Decreasing Normal Logic: 1d 8 8 A D=, A-M FS Lag =
Re-Setting Drag Based on Parallel Paths:1d 8 8 A D=, A-M FS Lag =
Decreasing Normal Logic: 2d 11 11 B D=
Re-Setting Drag Based on Parallel Paths:2d 11 11 B D=
Decreasing Normal Logic: 2d 15 15 G D=
Re-Setting Drag Based on Parallel Paths:2d 15 15 G D=
Decreasing Normal Logic: 1d 17 17 J D=
Diagnostic Log Elapsed Time: 00:00:00

56dCOMPLEX EXER.mpp



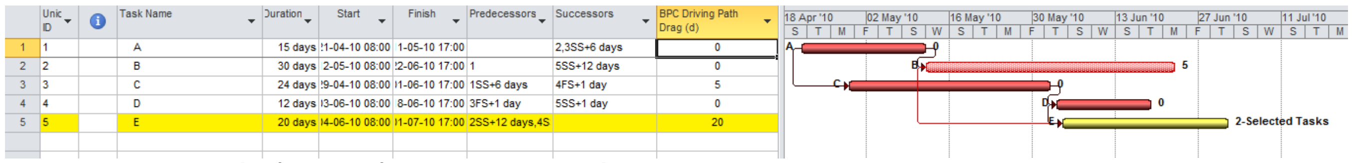
ID		Unique ID	Task Name	Duration	Start	Finish	Predecessors	Successors	BPC Driving Path Drag (d)
1		1	A	10 days	29-03-11 08:00	11-04-11 17:00		3,4SS+8 days	3
2		2	B	10 days	19-04-11 08:00	02-05-11 17:00	3SS+5 days	5SS+4 days	0
3		3	C	8 days	12-04-11 08:00	21-04-11 17:00	1	4FF+5 days,2SS+5 days,	2
4		4	D	12 days	13-04-11 08:00	28-04-11 17:00	1SS+8 days,3FF+	6	0
5		5	E	15 days	25-04-11 08:00	13-05-11 17:00	3,2SS+4 days	7,8	0
6		6	F	5 days	29-04-11 08:00	05-05-11 17:00	4	9,8FS+8 days	2
7		7	G	7 days	16-05-11 08:00	24-05-11 17:00	5	10	0
8		8	H	10 days	18-05-11 08:00	31-05-11 17:00	5,6FS+8 days	10	5
9		9	I	8 days	06-05-11 08:00	17-05-11 17:00	6	10	0
10		10	J	10 days	01-06-11 08:00	14-06-11 17:00	7,8,9		10

***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Run Log**************
Licensed to User:
Session Run: 5
20-Sep-16 15:17:21 Start Task Logic Tracer
20-Sep-16 15:17:25 Start Task Logic Tracer (after user input)
 Tom 56dCOMPLEX EXER.mpp (10 tasks)
Active Project: Tom 56dCOMPLEX EXER.mpp (10 tasks)
Filter Name: BPC_Select Task Preds: Drvrs+100dFF
Selected Tasks: 1
Selected Task: 10 10 J
Linked Tasks Check Direction: P
Evaluate Driving Relationships Only: True
Override Driving-Only for Successors with Constraints: False
Relative Float Window: 100
Limit - Max Steps from Selected Task: 10000
Limit - Max Tasks to Analyze: 1000001
Ignore Partial Days: True
Include Subprojects: True
Compute Drag: True Drag Calendar: 1
Check Resource Leveling Drivers: False
Check Resource Parallel Paths: False
Color Bars: True
In-Line Only: True
Duration Drag is shown on the chart displayed. Additional Drag found as follows-------------------
20-Sep-16 15:17:26 Total Tasks Analyzed: 10
20-Sep-16 15:17:26 End Task Logic Tracer. Total Time: 00:00:05.0412884 Analysis Time: 00:00:00.9190526
***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Diagnostic Log**************
Licensed to User:
Session Run: 5
20-Sep-16 15:17:21 Start Sub LinkedTasksControl
Tom 56dCOMPLEX EXER.mpp ClearDataFields Elapsed Time: 00:00:00.0320018
20-Sep-16 15:17:26 Starting Filter Controller.
DRAG LOG:
10 10 J| Not Started. Duration Drag:10
8 8 H| Predecessor Lag Drag:8

```
8  8  H| Not Started. Duration Drag:10
6  6  F| Not Started. Duration Drag:5
4  4  D| Predecessor Lag Drag:5
4  4  D| Not Started. Duration Drag:12
4  4  D| Finish is both Driven AND Driving.  No Drag:0
3  3  C| Not Started. Duration Drag:8
1  1  A| Not Started. Duration Drag:10
4  4  D| DragCalc Driving Path Drag Branch: RootNode: 4.1 RelFlt:3 to:1  1  A ReturnNode: 6.1
8  8  H| DragCalc Driving Path Drag Branch: RootNode: 2.1 RelFlt:2 to:5  5  E ReturnNode: 0
2  2  B| Driving Path Drag Return: RootNode:2.1 RelFlt:2 ReturnNode:5.1
5  5  E| Driving Path Drag Return: RootNode:2.1 RelFlt:3 ReturnNode:4.9
10  10  J| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:5 to:7  7  G ReturnNode: 0
10  10  J| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:10 to:9  9  I ReturnNode: 0
9  9  I| Driving Path Drag Return: RootNode:1.1 RelFlt:10 ReturnNode:2.9

Final Drag Results after checking Parallel Paths:
Duration Drag            | Lag Drag | Const Drag              | Split/Resume Drag| Leveling Drag (edays)
Implementing Brute Force Analysis for Duration Drag.
Drag Calendar: Selected/End Task
Decreasing Normal Logic: 3d 1  1  A
Re-Setting Drag Based on Parallel Paths:3d 1  1  A
Decreasing Normal Logic: 2d 3  3  C
Re-Setting Drag Based on Parallel Paths:2d 3  3  C
Decreasing Normal Logic: 2d 6  6  F
Re-Setting Drag Based on Parallel Paths:2d 6  6  F
Decreasing Normal Logic: 5d 8  8  H
Re-Setting Drag Based on Parallel Paths:5d 8  8  H
Decreasing Normal Logic: 10d 10  10  J
Diagnostic Log Elapsed Time: 00:00:00
```

5ACTss52.mpp



	Unic ID	ℹ	Task Name	Duration	Start	Finish	Predecessors	Successors	BPC Driving Path Drag (d)
1	1		A	15 days	ː1-04-10 08:00	1-05-10 17:00		2,3SS+6 days	0
2	2		B	30 days	2-05-10 08:00	ː2-06-10 17:00	1	5SS+12 days	0
3	3		C	24 days	ː9-04-10 08:00	ı1-06-10 17:00	1SS+6 days	4FS+1 day	5
4	4		D	12 days	ı3-06-10 08:00	8-06-10 17:00	3FS+1 day	5SS+1 day	0
5	5		E	20 days	ı4-06-10 08:00	ı1-07-10 17:00	2SS+12 days,4S		20

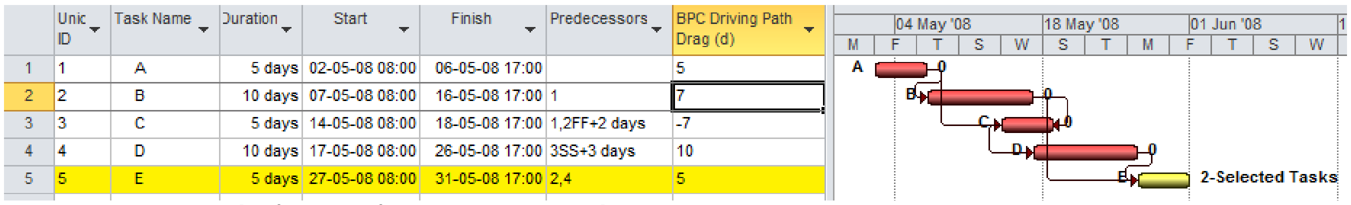
```
***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Run Log**************
Licensed to User:
Session Run: 6
20-Sep-16 15:24:46 Start Task Logic Tracer
20-Sep-16 15:24:53 Start Task Logic Tracer (after user input)
 Tom5ACTss52.mpp (5 tasks)
Active Project: Tom5ACTss52.mpp (5 tasks)
Filter Name: BPC_Select Task Links: Drvrs+100dFF
Selected Tasks: 1
Selected Task: 5 5 E
Linked Tasks Check Direction: B
Evaluate Driving Relationships Only: True
Override Driving-Only for Successors with Constraints: False
Relative Float Window: 100
Limit - Max Steps from Selected Task: 10000
Limit - Max Tasks to Analyze: 1000001
Ignore Partial Days: True
Include Subprojects: True
Compute Drag: True Drag Calendar: 1
Check Resource Leveling Drivers: False
Check Resource Parallel Paths: False
Color Bars: True
In-Line Only: True
The Selected Task has No Successors: 5 - E
Duration Drag is shown on the chart displayed.  Additional Drag found as follows-------------------
20-Sep-16 15:24:55 Total Tasks Analyzed: 6
20-Sep-16 15:24:56 End Task Logic Tracer.  Total Time: 00:00:10.3475919  Analysis Time: 00:00:03.1401796

***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Diagnostic Log**************
Licensed to User:
Session Run: 6
20-Sep-16 15:24:46 Start Sub LinkedTasksControl
Tom5ACTss52.mpp ClearDataFields Elapsed Time: 00:00:00.0110006
```

20-Sep-16 15:24:53 Starting Filter Controller.
DRAG LOG:
5 5 E| Predecessor Lag Drag:1
5 5 E| Not Started. Duration Drag:20
4 4 D| Predecessor Lag Drag:1
4 4 D| Start is both Driven and Driving. No Drag:0
3 3 C| Predecessor Lag Drag:6
3 3 C| Not Started. Duration Drag:24
1 1 A| Start is both Driven and Driving. No Drag:0
5 5 E| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:5 to:2 2 B ReturnNode: 0
2 2 B| Driving Path Drag Return: RootNode:1.1 RelFlt:5 ReturnNode:3.9

Final Drag Results after checking Parallel Paths:
Duration Drag | Lag Drag | Const Drag | Split/Resume Drag| Leveling Drag (edays)
Implementing Brute Force Analysis for Duration Drag.
Drag Calendar: Selected/End Task
Decreasing Normal Logic: 5d 3 3 C
Re-Setting Drag Based on Parallel Paths:5d 3 3 C
Decreasing Normal Logic: 20d 5 5 E
Diagnostic Log Elapsed Time: 00:00:00.0020001

Reverse crit test.mpp



	Unique ID	Task Name	Duration	Start	Finish	Predecessors	BPC Driving Path Drag (d)
1	1	A	5 days	02-05-08 08:00	06-05-08 17:00		5
2	2	B	10 days	07-05-08 08:00	16-05-08 17:00	1	7
3	3	C	5 days	14-05-08 08:00	18-05-08 17:00	1,2FF+2 days	-7
4	4	D	10 days	17-05-08 08:00	26-05-08 17:00	3SS+3 days	10
5	5	E	5 days	27-05-08 08:00	31-05-08 17:00	2,4	5

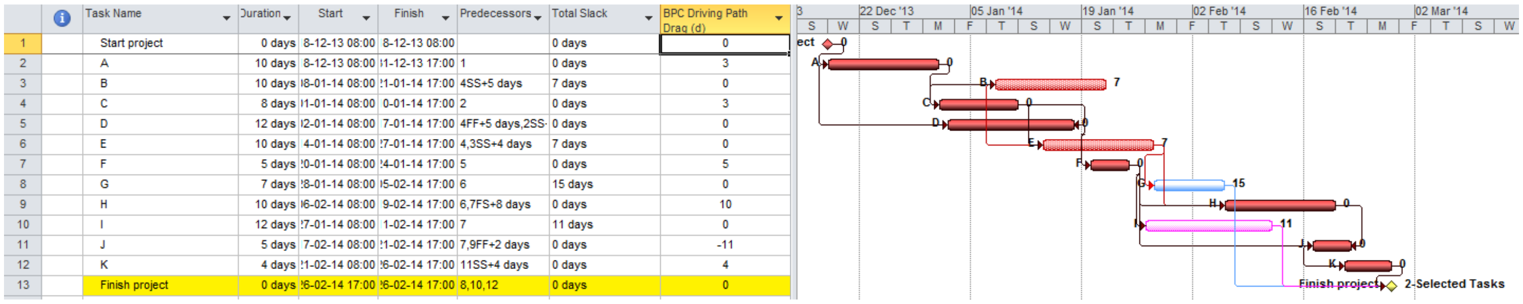
***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Run Log**************
Licensed to User:
Session Run: 7
20-Sep-16 15:31:58 Start Task Logic Tracer
20-Sep-16 15:32:11 Start Task Logic Tracer (after user input)
 Tom reverse crit test.mpp (5 tasks)
Active Project: Tom reverse crit test.mpp (5 tasks)
Filter Name: BPC_Select Task Preds: Drvrs+100dFF
Selected Tasks: 1
Selected Task: 5 5 E
Linked Tasks Check Direction: P
Evaluate Driving Relationships Only: True
Override Driving-Only for Successors with Constraints: False
Relative Float Window: 100
Limit - Max Steps from Selected Task: 10000
Limit - Max Tasks to Analyze: 1000001
Ignore Partial Days: True
Include Subprojects: True
Compute Drag: True Drag Calendar: 1
Check Resource Leveling Drivers: False
Check Resource Parallel Paths: False
Color Bars: True
In-Line Only: True
Duration Drag is shown on the chart displayed. Additional Drag found as follows-------------------
20-Sep-16 15:32:12 Total Tasks Analyzed: 5
20-Sep-16 15:32:12 End Task Logic Tracer. Total Time: 00:00:14.7918461 Analysis Time: 00:00:01.1710670

***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Diagnostic Log**************
Licensed to User:
Session Run: 7
20-Sep-16 15:31:58 Start Sub LinkedTasksControl
Tom reverse crit test.mpp ClearDataFields Elapsed Time: 00:00:00.0110006
20-Sep-16 15:32:11 Starting Filter Controller.
DRAG LOG:
5 5 E| Not Started. Duration Drag:5
4 4 D| Predecessor Lag Drag:3
4 4 D| Not Started. Duration Drag:10
3 3 C| Predecessor Lag Drag:2
3 3 C| Driven Finish and Driving Start. Negative Drag: -1
2 2 B| Not Started. Duration Drag:10
1 1 A| Not Started. Duration Drag:5

3 3 C| Negative Drag Constrained by Start predecessor float:-7
3 3 C| DragCalc Driving Path Drag Branch: RootNode: 2.9 RelFlt:7 to:1 1 A ReturnNode: 4.9
5 5 E| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:10 to:2 2 B ReturnNode: 3.9

Final Drag Results after checking Parallel Paths:
Duration Drag | Lag Drag | Const Drag | Split/Resume Drag| Leveling Drag (edays)
Implementing Brute Force Analysis for Duration Drag.
Drag Calendar: Selected/End Task
Decreasing Normal Logic: 5d 1 1 A
Decreasing Normal Logic: 7d 2 2 B
Re-Setting Drag Based on Parallel Paths:7d 2 2 B
Reverse Logic Flow - Setting Negative Drag: -7d 3 3 C
Decreasing Normal Logic: 10d 4 4 D
Decreasing Normal Logic: 5d 5 5 E
Diagnostic Log Elapsed Time: 00:00:00.0010001

Test 2.mpp



		Task Name	Duration	Start	Finish	Predecessors	Total Slack	BPC Driving Path Drag (d)
1		Start project	0 days	8-12-13 08:00	8-12-13 08:00		0 days	0
2		A	10 days	8-12-13 08:00	!1-12-13 17:00	1	0 days	3
3		B	10 days	!8-01-14 08:00	!1-01-14 17:00	4SS+5 days	7 days	0
4		C	8 days	!1-01-14 08:00	0-01-14 17:00	2	0 days	3
5		D	12 days	!2-01-14 08:00	7-01-14 17:00	4FF+5 days,2SS-	0 days	0
6		E	10 days	4-01-14 08:00	!7-01-14 17:00	4,3SS+4 days	7 days	0
7		F	5 days	!0-01-14 08:00	!4-01-14 17:00	5	0 days	5
8		G	7 days	!8-01-14 08:00	!5-02-14 17:00	6	15 days	0
9		H	10 days	!6-02-14 08:00	9-02-14 17:00	6,7FS+8 days	0 days	10
10		I	12 days	!7-01-14 08:00	1-02-14 17:00	7	11 days	0
11		J	5 days	7-02-14 08:00	!1-02-14 17:00	7,9FF+2 days	0 days	-11
12		K	4 days	!1-02-14 08:00	!6-02-14 17:00	11SS+4 days	0 days	4
13		Finish project	0 days	!6-02-14 17:00	!6-02-14 17:00	8,10,12	0 days	0

```
***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Run Log**************
Licensed to User:
Session Run: 8
20-Sep-16 15:36:29 Start Task Logic Tracer
20-Sep-16 15:36:39 Start Task Logic Tracer (after user input)
 Tom Test 2.mpp (13 tasks)
Active Project: Tom Test 2.mpp (13 tasks)
Filter Name: BPC_Select Task Preds: Drvrs+100dFF
Selected Tasks: 1
Selected Task: 13 15 Finish project
Linked Tasks Check Direction: P
Evaluate Driving Relationships Only: True
Override Driving-Only for Successors with Constraints: False
Relative Float Window: 100
Limit - Max Steps from Selected Task: 10000
Limit - Max Tasks to Analyze: 1000001
Ignore Partial Days: True
Include Subprojects: True
Compute Drag: True Drag Calendar: 1
Check Resource Leveling Drivers: False
Check Resource Parallel Paths: False
Color Bars: True
In-Line Only: True
Duration Drag is shown on the chart displayed.  Additional Drag found as follows-------------------
20-Sep-16 15:36:40 Total Tasks Analyzed: 13
20-Sep-16 15:36:40 End Task Logic Tracer.  Total Time: 00:00:11.7796738  Analysis Time: 00:00:01.5470885

***********BPC Logic Filter for Microsoft Project (1.1.1.14) Pro Edition-- Diagnostic Log**************
Licensed to User:
Session Run: 8
20-Sep-16 15:36:29 Start Sub LinkedTasksControl
Tom Test 2.mpp ClearDataFields Elapsed Time: 00:00:00.0270016
20-Sep-16 15:36:39 Starting Filter Controller.
DRAG LOG:
13  15  Finish project| Not Started. Duration Drag:0
12  14  K| Predecessor Lag Drag:4
12  14  K| Not Started. Duration Drag:4
11  12  J| Predecessor Lag Drag:2
```

```
11  12  J| Driven Finish and Driving Start. Negative Drag: -1
9  9  H| Predecessor Lag Drag:8
9  9  H| Not Started. Duration Drag:10
7  7  F| Not Started. Duration Drag:5
5  5  D| Predecessor Lag Drag:5
5  5  D| Not Started. Duration Drag:12
5  5  D| Finish is both Driven AND Driving.  No Drag:0
4  4  C| Not Started. Duration Drag:8
2  2  A| Not Started. Duration Drag:10
1  1  Start project| Not Started. Duration Drag:0
5  5  D| DragCalc Driving Path Drag Branch: RootNode: 6.1 RelFlt:3 to:2  2  A ReturnNode: 8.1
9  9  H| DragCalc Driving Path Drag Branch: RootNode: 4.1 RelFlt:7 to:6  6  E ReturnNode: 0
3  3  B| Driving Path Drag Return: RootNode:4.1 RelFlt:7 ReturnNode:7.1
6  6  E| Driving Path Drag Return: RootNode:4.1 RelFlt:8 ReturnNode:6.9
11  12  J| Negative Drag Constrained by Start predecessor float:-15
11  12  J| DragCalc Driving Path Drag Branch: RootNode: 2.9 RelFlt:15 to:7  7  F ReturnNode: 4.9
13  15  Finish project| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:11 to:10  10  I ReturnNode: 0
10  10  I| Driving Path Drag Return: RootNode:1.1 RelFlt:11 ReturnNode:4.9
13  15  Finish project| DragCalc Driving Path Drag Branch: RootNode: 1.1 RelFlt:15 to:8  8  G ReturnNode: 0

Final Drag Results after checking Parallel Paths:
Duration Drag           | Lag Drag | Const Drag              | Split/Resume Drag| Leveling Drag (edays)
Implementing Brute Force Analysis for Duration Drag.
Drag Calendar: Selected/End Task
Decreasing Normal Logic: 3d 2  2  A
Re-Setting Drag Based on Parallel Paths:3d 2  2  A
Decreasing Normal Logic: 3d 4  4  C
Re-Setting Drag Based on Parallel Paths:3d 4  4  C
Decreasing Normal Logic: 5d 7  7  F
Decreasing Normal Logic: 10d 9  9  H
Reverse Logic Flow - Setting Negative Drag: -11d 11  12  J
Re-Setting Drag Based on Parallel Paths:-11d 11  12  J
Decreasing Normal Logic: 4d 12  14  K
Diagnostic Log Elapsed Time: 00:00:00.0010001
```