Understanding the P6 Scheduling Options "When scheduling activities use, Retained Logic, Progress Override or Actual Dates"

1   Introduction

The Oracle Primavera P6 Scheduling Options "When scheduling activities use" are usually not understood and this paper will outline how these options operate and demonstrate the issues with some of the options.

These options are similar the Microsoft Project Schedule option "Split in-progress tasks" and Asta Power Project "Relink around completed tasks".

The examples below were complete using P6 Version 18 PPM and para 4 which was added on 22 January 2023 uses P6 Version 22 PPM.

2   Understanding "Out of Sequence Progress"

"Out of Sequence Progress" occurs when an activity starts before a predecessor defined by a relationship has finished. Therefore, the relationships have not been acknowledged and the successor activity has started out of sequence.

There are three options in P6 for calculating the finish date of a successor when the successor activity has started before the predecessor activity is finished:
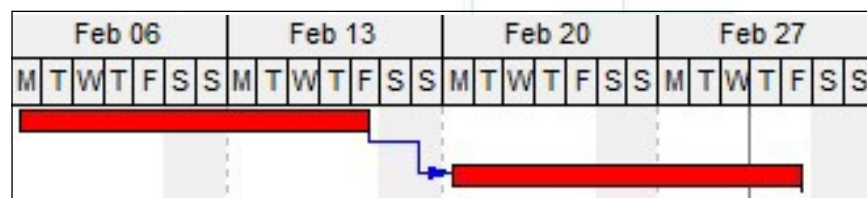
- Retained Logic
- Progress Override
- Actual Dates

The selected option is applied to all activities in a schedule when it is calculated. Open the **Schedule Options** form by selecting **Tools**, **Schedule...** and clicking on the [▶ Options...] icon where the options are found under **When scheduling progressed activities use**:


The picture below represents the status of the activities before updating the schedule:

2.1 Retained Logic.

In the example following, the relationship is maintained between the predecessor and successor for the unworked portion of the activity (the Remaining Duration) and continued after the predecessor has finished. The relationship forms part of the critical path and the predecessor has no float.

NOTE: This is the recommended option:
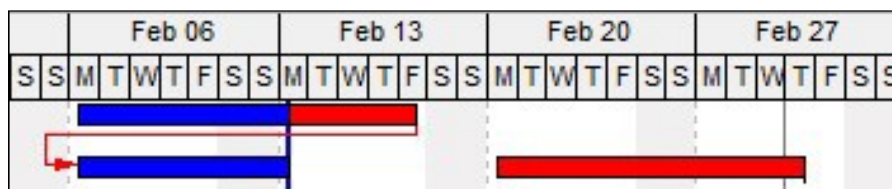

2.2 Progress Override.

In the example following, the Finish-to-Start relationship between the predecessor and successor is disregarded, and the unworked portion of the activity (the Remaining Duration) continues before the predecessor has finished.

NOTE: The relationship is not a driving relationship and DOES NOT form part of the critical path in the example below and the predecessor has float:
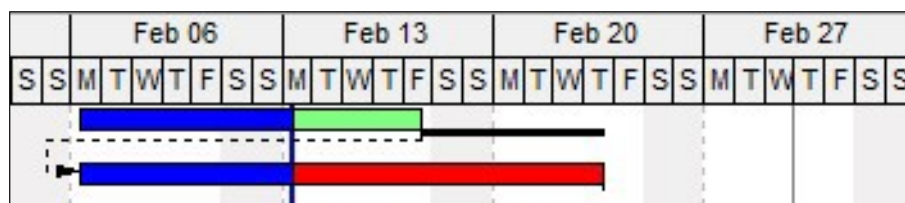
2.3 Actual Dates.

This function operates when there is an activity with Actual Start Dates in the future, which is not logical. With this option the remaining duration of an in-progress activity is calculated after the activity with actual start and finish in the future:
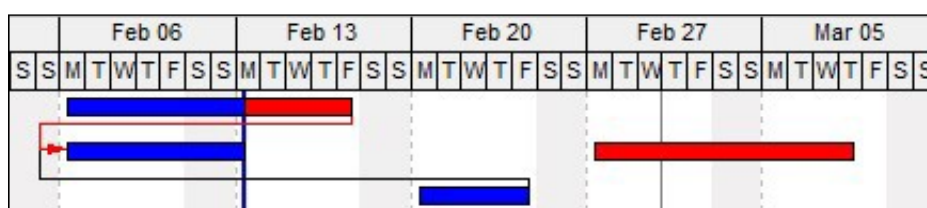
When there are no Actual Dates in the future this option calculates as Retained Logic.

In a properly updated schedule, there should be no Actual Dates in the future. The situation with Actuals in the future may happen when two projects are opened together and have different Data Dates. This situation is best avoided, and it is best to make the Data Dates of all projects the same.

3   Issues with "Out of Sequence Progress" options

3.1   Retained Logic

In the example following, which has:

- Activity A1060 has Actual Start and Actual Finish dates in the future, which is not logical, the relationships to this task are ignored. Activities A1050 and A1070 are connected around this activity by retaining the logic.
- Activity A1020 has Actual Start and Actual Finish dates in the past, which is not logical, and the relationships to this task are ignored. Activities A1010 and A1030 are connected around this activity by retaining the logic.
- Activity A1030 is in progress and the activity is split. Activities A1010, the incomplete portion of A1020 and A1030 are connected by retaining the logic, but the complete portion of A1030 is ignored.

This method of calculation results in the most conservative result and the scheduler may change any logs if required:
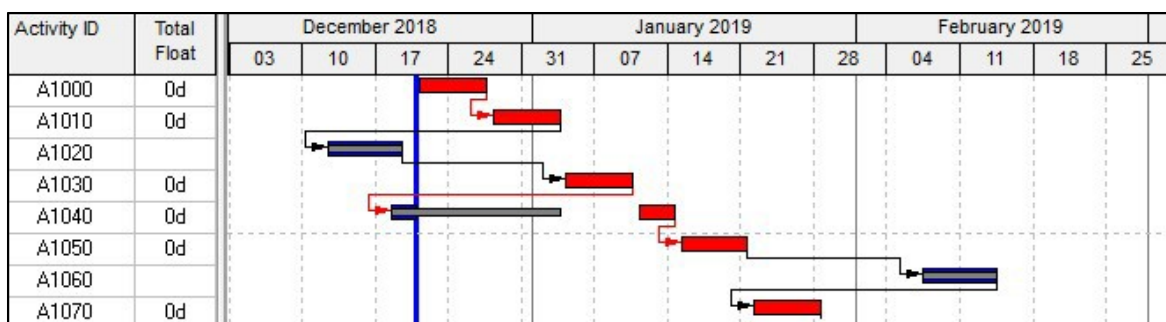
3.2   Retained Logic with Looping Logic

When a loop is inserted in the logic and the schedule is recalculated the loop is identified:
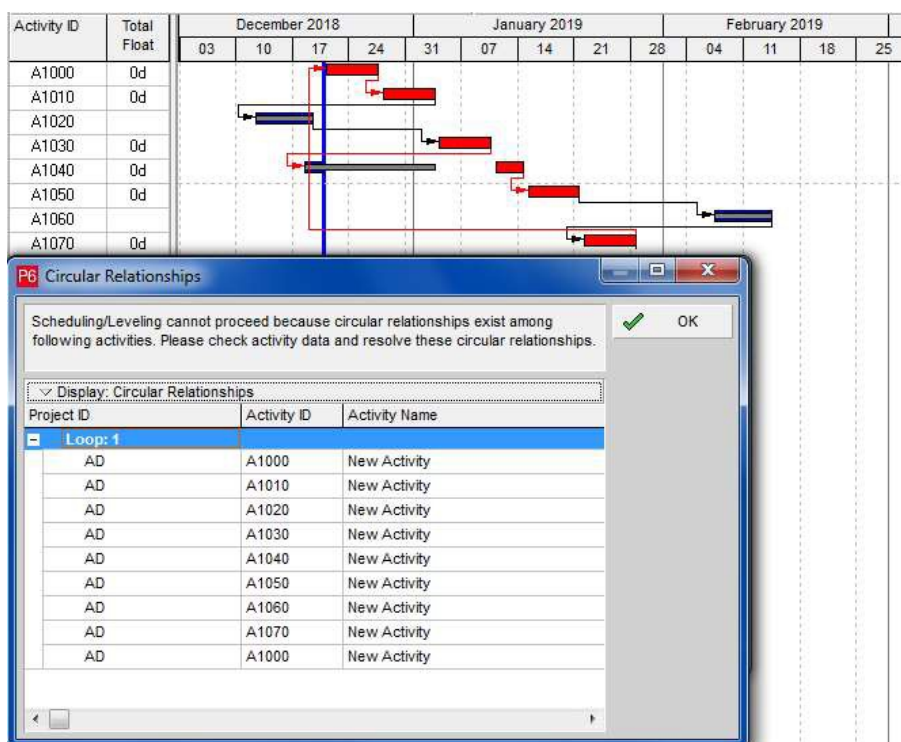
3.3   Progress Override

With Progress override:

- No successors are acknowledged for complete or in progress activities,
- Any activity with a complete or in-progress activity as a predecessor has the predecessor is ignored, and in turn it is scheduled on the Data Date:
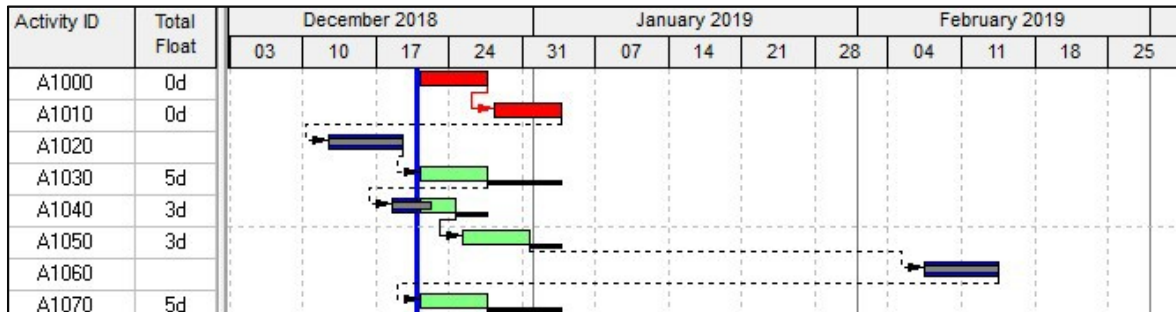
3.4   Progress Override with Looping logic

When a loop is inserted then the following issues are presented when scheduling:

- The Circular relationship is not identified and

- The schedule is recalculated, acknowledging the illogical looping logic.
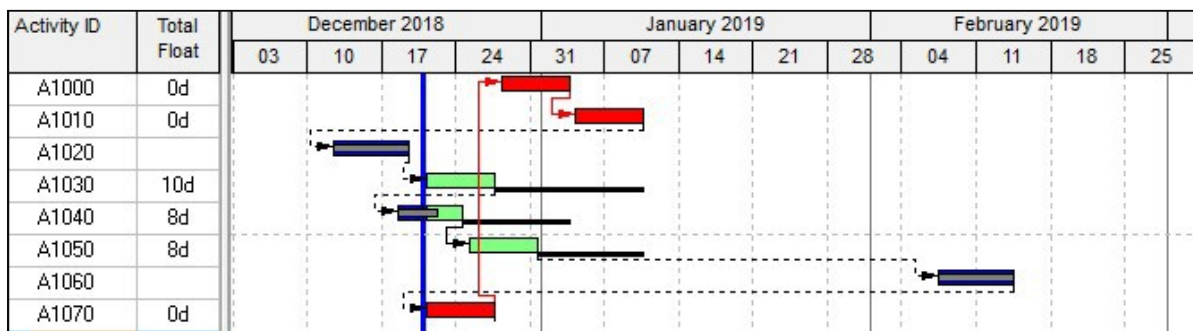
3.5   Actual dates

When the option of Actual Dates is selected:

- A task with a complete activity as a predecessor has the predecessor relationship acknowledged,
- Negative float is calculated which should not happen when a schedule has no constraints:

Note: This option should be avoided as there should never be completed tasks in the future and the **Total Float** is not calculated as expected.

3.6   Actual dates with Looping logic

When a loop in the log is introduced then:

- The Circular relationship is not identified,
- The schedule is recalculated, acknowledging the illogical looping logic.
- Negative float is calculated which should not happen when a schedule has no constraints:

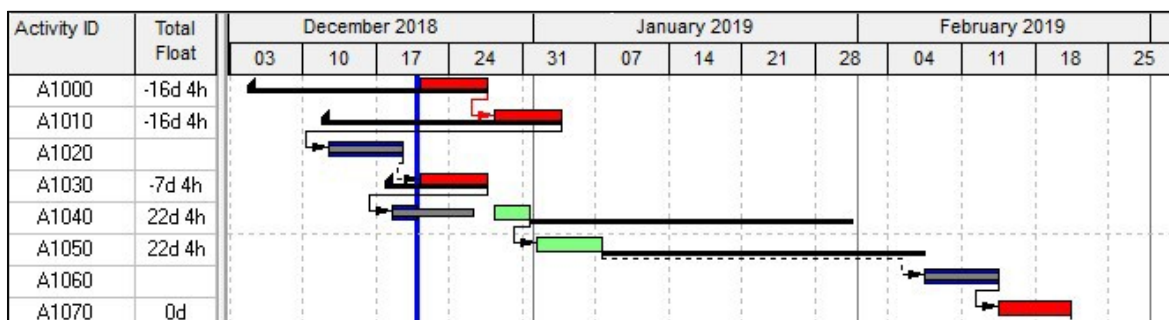**Note:** This option should be avoided as there should never be completed tasks in the future and the Total Float is not calculated as expected.

Discussion: There is a sound argument for the use of the **Actual Dates** option when there is a very large schedule that is being progressed over time. This option allows the scheduler to update activities that are in-progress or complete in the future and effectively give each activity its own approximate **Data Date**. The picture below shows how this works:

4   Free Float Calculation with Project Must Finish BY Date

When a **Project Must Finish By d**ate is set the things get interesting, I have set a **Project Must Finish By** date on 1 Dec 22 at 16:00 in the pictures below.

4.1 Retained Logic

The Total and Free Float calculate correctly:

4.2 Progress Override

In the picture below you will see that the Free Float is greater than Total Float which does not appear logical to me:
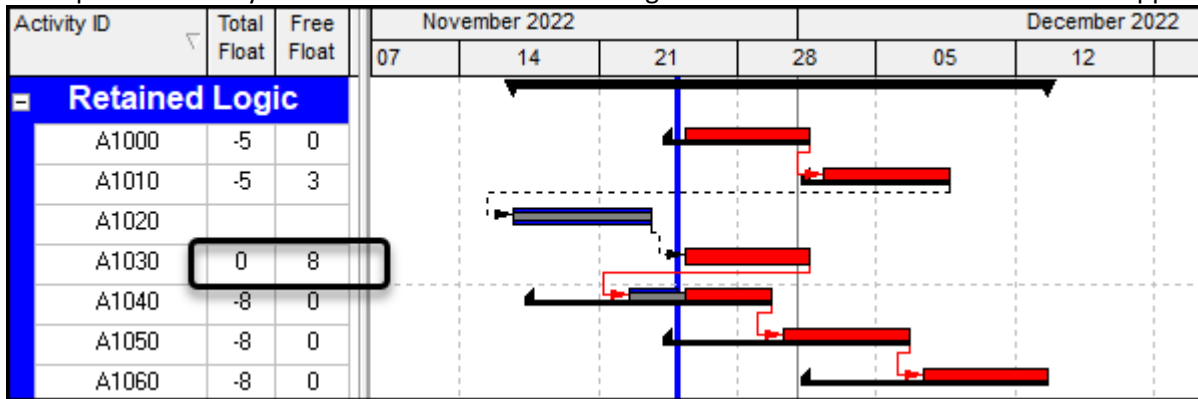
4.3 Actual Dates

Again, we have an example of positive Free Float against an activity with Negative Total Float which again does not seem logical to me:
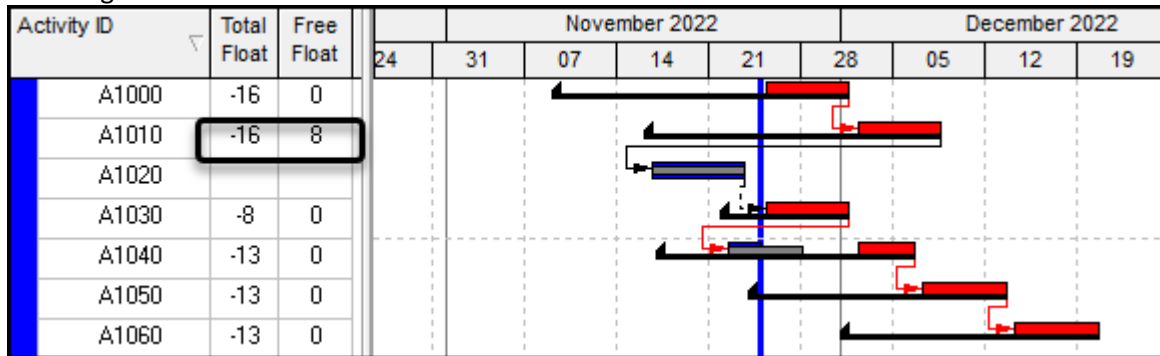
5   Recommendation

From the demonstration above it may be clearly seen that only the **Retained Logic** option should be used because it is the only option that identifies looping logic and allows Total and Free Float to be calculated as expected; it is fortunate that this is the default option.

Paul E Harris
Director Eastwood Harris Pty Ltd
16 April 2019
Para 4 added 27 January 2022

EASTWOODHARRIS

CREATE AND UPDATE AN UNRESOURCED PROJECT USING ELECOSOFT (ASTA) POWERPROJECT

VERSION 16

PLANNING AND CONTROL USING MICROSOFT® PROJECT 365 and 2021

Including Microsoft Project 2013, 2016 and 2019

PAUL E. HARRIS

99 TRICKS & TRAPS FOR MICROSOFT PROJECT 2013, 2016 & 2019

THE CASUAL USER'S SURVIVAL GUIDE!

For users who understand the basics and want a brief text to demonstrate the less intuitive functions.

PAUL E. HARRIS

PLANNING AND CONTROL USING MICROSOFT® PROJECT 365

Including Microsoft Project 2013, 2016 & 2019

PAUL E. HARRIS

PLANNING & CONTROL USING ORACLE® PRIMAVERA® P6

VERSION 8 TO 21 PPM PROFESSIONAL

PAUL E. HARRIS

99 TRICKS & TRAPS FOR ORACLE PRIMAVERA® P6 PPM PROFESSIONAL

THE CASUAL USER'S SURVIVAL GUIDE!

For users who understand the basics of P6 and want a brief text to demonstrate the less intuitive functions.

PAUL E. HARRIS