The Critical Path Method of Modeling Project Execution Strategy

A Technology Book

by Murray B. Woolf

ICS-Compendium Volume 1

ICS-Publications
A Division of the International Center for Scheduling, Inc.
Library of Congress Cataloging-in-Publication Data

Woolf, Murray B.

CPM Mechanics: The Critical Path Method of Modeling Project Execution Strategy / Murray B. Woolf
Includes Bookmarked and Hyperlinked Index, Table of Contents, List of Figures, List of Formulas and Rules


ICS-Publications books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to the Managing Director, ICS-Publications, P.O. Box 80915, Rochester, MI 48308.

CPM Mechanics: The Critical Path Method of Modeling Project Execution Strategy

Copyright 2012 by ICS-Publications, a division of the International Center for Scheduling, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

File Version 7: 20120618

Terms abstracted from the Wideman Comparative Glossary of Project Management Terms are © R. Max Wideman http://www.maxwideman.com 2006 and reproduced with permission. For the original source of individual terms, please refer to http://www.maxwideman.com/pmglossary/PMGloss_Sources.htm

These terms are offered to individual readers who may use them freely in connection with their project work. They may not be used by commercial or non-commercial organizations without permission.

The information contained within this book has been obtained by ICS-Publications from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, ICS-Publications or others, ICS-Publications does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from the use of such information.
Note: CPM Mechanics is Volume 1 of the ICS-Compendium and forms the basis for both the Dominant Project Management Series and the Cognitive Project Management Series. To learn more about CPM Mechanics, we encourage you to investigate it at www.CpmMechanics.com.

* The ICS-Compendium is a five-year project that began in January 2011. The primary eight volumes are scheduled for issuance in six-month increments, starting with CPM Mechanics, which was released in May 2012. Release Dates for the other volumes are posted at the ICS-Publications website. Simply go to www.ics-publications.com.

Terms abstracted from the Wideman Comparative Glossary of Project Management Terms are © R. Max Wideman http://www.maxwideman.com 2006 and reproduced with permission. For the original source of individual terms, please refer to http://www.maxwideman.com/pmglossary/PMGloss_Sources.htm

These terms are offered to individual readers who may use them freely in connection with their project work. They may not be used by commercial or non-commercial organizations without permission.
The ICS-Compendium is a five-year project that began in January 2011. The primary eight volumes are scheduled for issuance in six-month increments, starting with CPM Mechanics, which was released in May 2012. Release Dates for the other volumes are posted at the ICS-Publications website. Simply go to www.ics-publications.com.
The ICS-Compendium is a five-year project that began in January 2011. The primary eight volumes are scheduled for issuance in six-month increments, starting with CPM Mechanics, which was released in May 2012. Release Dates for the other volumes are posted at the ICS-Publications website. Simply go to www.ics-publications.com.
ICS-Global was founded in 2006 with the singular goal of improving Construction Project Time Management in North America. This, in turn, led to finding better ways to perform its parent business function, Project Management — in order to assure Owners, Contractors, Design Professionals, financiers and sureties, Project Managers, and Project Team Members that their Projects' temporal outcomes can be, and will be, achieved consistently, predictably, confidently, and repeatedly.

**Cognitive Project Management versus Dominant Project Management**

An early realization was that the predominant Project Management model (herein called Dominant Project Management) was not especially suited for the Construction Industry. This observation necessitated the creation of a very new and refreshingly robust and practical Project Management model, one that we call Cognitive Project Management.

Because it was designed “from the bottom up” with the informational needs of the Field Superintendent and Subcontractors (those who actually build the Projects) at the very epicenter of the Project Management model, we humbly contend that Cognitive Project Management is unquestionably superior to the Dominant Project Management model, at least within the Construction Industry.

**Enter — the ICS-Compendium**

The ICS-Compendium Development Team was tasked with writing a set of documents that would fully explain the Technology, Ideology, Methodology, and Ecology (T.I.M.E.) of Cognitive Project Management. In order to justify Cognitive Project Management and distinguish it from the current alternative, Dominant Project Management, the conceptual scope of the ICS-Compendium evolved into two essentially independent reference sets – contained within a single collection of seminal material.

To be sure, one can find scores of isolated academic and intellectual works on library shelves, but they are almost certain to use different terms, or similar terms with different meanings, backed by different ideologies, recommending differing “best" or “recommended" practices, and so forth. This convolution makes it difficult, if not utterly impossible, for those new to Project Time Management to make sense of what these experts have to say.
If the reader is knowledgeable enough to decipher the conflicting viewpoints, then he or she doesn't really need the advice all that much. But if the reader is not especially versed in the details of Project Time Management, then how can he or she know which “best practices” to accept, and which to reject?

The ICS-Compendium is intended to addresses this decades-old problem by offering a comprehensive treatise on Construction Project Time Management, using one set of well-defined terms, embracing a common Ideology that emanates from a single globally-recognized authority on Project Time Management, and written in a conversational style that is dripping with common sense and practicality.

ICS-Compendium Organization

The ten-volume ICS-Compendium is divided into three major groupings, reflective of three distinct informational objectives.

☐ The **DOMINANT PROJECT MANAGEMENT SERIES** is designed to provide everything that a Construction Contractor needs to know in order to practice Dominant Project Management as well as one can.

- **Volume 1: CPM MECHANICS**  
  The Critical Path Method of Modeling Project Execution Strategy
- **Volume 2: UNDERSTANDING PROJECT TIME MANAGEMENT**  
  The Critical Path Method of Managing Project Execution Strategy
- **Volume 3: CONSTRUCTION PLANNING AND SCHEDULING**  
  The Critical Path Method of Scheduling Project Execution Strategy
- **Volume 4: UNDERSTANDING CONSTRUCTION PROJECT MANAGEMENT**  
  Achieving Consistent and Reliable Project Time Management

☐ The **COGNITIVE PROJECT MANAGEMENT SERIES** provides knowledge and guidance for the Constructor organization that wishes to embrace the most ground-breaking Project Time Management innovations of the 21st century, including Momentum Management, Dilemma Forecasting, Project Administration and Guidance System (PAGUSYS), Performance Intensity, and so much more.

- **Volume 5: FUNDAMENTALS OF MOMENTUM SCIENCE**  
  Introducing Performance Intensity, Dilemma Forecasting, Momentum Checkpoints
- **Volume 6: INTRODUCTION TO MOMENTUM MANAGEMENT**  
  Applied Cognitive Project Management Using Practical Tools and Processes
- **Volume 7: APPLIED MOMENTOLOGY: INTRODUCTION TO PAGUSYS**  
  Cognitive Project Management’s Project Administration and Guidance System
- **Volume 8: PRINCIPLES OF COGNITIVE PROJECT MANAGEMENT**  
  Improvisational Management, Construction Currency, Neutralizing the Zero Sum Game Effect
About the ICS-Compendium

- The ICS-Reference Series provide foundational reference material, essential for every Project Time Management library.
  
  - **Volume 9:** Project Time Management: Essential Desk Reference
    - ICS-Dictionary, ICS-Encyclopedia, ICS-Glossatree
  
  - **Volume 10:** Project Time Management: Critical Thinking
    - The ICS-Compendium White Papers

**Note:** The ICS-Compendium is a five-year project that began in January 2011. The primary eight volumes are scheduled for issuance in six-month increments, starting with CPM Mechanics, which was released in May 2012. Release Dates for the other volumes are posted at the ICS-Publications website. Simply go to www.ics-publications.com.

About the Author

Murray B. Woolf is president of the International Center for Scheduling, Inc. and author of the best-selling "FASTER Construction Projects with CPM Scheduling." He founded ICS-Global as his response to the number of challenges facing the vital, and yet essentially misunderstood, field of Project Time Management — as specifically applied to Construction Project Management. “I wanted to give something back to a professional discipline that I love so passionately and that has served me and my family so well.”

Murray has more than 37 years of experience in Construction Project Management/Project Controls production, training, consulting, and testimonial expertise. Across his career he provided hands-on Project Time Management services on over 250 projects worldwide, with combined value estimated at around $28 billion.

Murray is a globally-recognized authority, speaker, and writer on Construction Project Time Management topics. Viewed by his peers as a Visionary, he is the inventor of numerous Project Time Management innovations including, Cognitive Project Management, Momentum Management, PAGUSYS, and Dilemma Forecasting.

In 2004, Murray was hand-picked by the chairman of the Project Management Institute's College of Scheduling to serve as the founding director for it Scheduling Excellence Initiative, a name Murray gave to this international, 250-volunteer effort to develop “Best Practices and Guidelines" for Project Scheduling. SEI's final work product is expected to be released in early 2013.

Murray has also served on Advisory Panels for numerous other international professional bodies concerned with Project Time Management issues. He is married, has two grown daughters, and enjoys living and working in Rochester Hills, Michigan. He may be reached at CpmMechanics@ics-global.com.
# Table of Contents

*About the ICS-Compendium* ........................................................................................................... v

*Table of Contents* ......................................................................................................................... ix

*List of Figures and Illustrations* ................................................................................................. xvii

*Expressions of Gratitude* ................................................................................................................. xxvii

## Chapter 1: CPM, A Modeling Method .................................................................................. 1

1A: Introducing the ICS-Compendium ...................................................................................... 1

1B: Exclusively for Construction Project Management ................................................................. 6

1C: The Three Meanings of the Expression, “CPM Scheduling” ................................................ 8

1D: The Book’s Organization and Structure ............................................................................... 11

1E: The Book’s Information Transfer Strategy ........................................................................... 14

1E1: The Book’s Approach to Technical Subject Matter .............................................................. 15

1E2: The Book’s Unique Formatting Symbolism ........................................................................ 16

1E3: The ICS-Dictionary ................................................................................................................. 17

## Chapter 2: Static Project Time Management Tools ......................................................... 21

2A: Introducing the Critical Path Method of Modeling ................................................................. 21

2A1: CPM from the Project Manager’s Perspective .................................................................... 21

2A2: CPM from the Project Facilitator’s Perspective .................................................................. 26

2B: Static Project Time Management Tools ................................................................................ 28

2B1: What is a Schedule? .............................................................................................................. 28

---

Copyright, 2012 International Center for Scheduling, Inc. (ICS-Global) All rights reserved.
Chapter 3: Dynamic Project Time Management Tools

3A: Introducing Dynamic Project Time Management Tools
3A1: Characterized by Energy or Effective Action
3A2: Affected by the Passage of Time
3A3: Affected by the Presence or Absence of Power
3A4: Motivates or Affects Development

3B: Activity Relationships: Two VERY Different Perspectives
3B1: Relationship, Beyond Dependence
3B2: Cognitive's Three Relationship Categories

3C: From Activities to Logic Diagrams
3C1: The Terminology of Performance Restrictions
3C2: Activity Path: Through Immediately-Restricting Ties

3D: CPM’s Four Restriction Linkages
3D1: Four Performance Restriction Linkages
3D2: Performance Restriction’s Three Key Attributes
3D3: CPM Dependency Types: Traditional Clarifications
3D4: Performance Restrictions: Per Cognitive Project Management

Chapter 4: How to Draw CPM Logic

4A: Introducing the Network-Based Project Schedule
4A1: Using the Word “Date” a Bit Loosely
4A2: Gregorian Dates and Ordinal Dates
4A3: Calculating Ordinal Dates during Logic Development Sessions
4A4: Ordinal Dates Are Much Faster and Simpler
Table of Contents

4A5: Dates Come Last, Not First ................................................................. 119
4B: Key Elements of a CPM Schedule ...................................................... 121
4B1: Provided versus Calculated Schedule Elements ......................... 121
4B2: Mandatory versus Optional Schedule Elements ......................... 121
4B3: Key Elements are Mandatory; Whether Provided or Calculated .......... 122
4C: CPM Method: Algorithms or Graphics? ............................................ 123
4C1: Distinguishing State of Being and Functional Process .................. 124
4C2: Dissecting the Activity “Box” ......................................................... 125
4C3: The Connecting Arrow; the Logic Tie ............................................. 127
4C4: How to Draw a Default Restriction ............................................... 128
4C5: How to Draw a Start Restriction ................................................. 130
4C6: How to Draw a Finish Restriction .............................................. 134
4D: The Point-of-Day Perspective ............................................................ 135

Chapter 5: Calculating Primary CPM Dates ......................................................... 137
5A: Introducing the Concept of Earliest Dates ........................................ 137
5A1: Defining Forward Pass and Backward Pass Processes ................... 137
5A2: Each Date Calculation Process has Four Elements ...................... 138
5A3: Date Calculation Arithmetic Formula Identifiers .......................... 139
5A4: Forward Pass Calculations through Default Restrictions ............. 140
5A5: Forward Pass Calculations through Start Restrictions .................. 149
5A6: Forward Pass Calculations through Finish Restrictions ............... 155
5A7: Forward Pass thru Combined Start/Finish Restrictions ............... 156
5A8: Forward Pass: Putting It All Together ....................................... 159
5B: Introducing the Concept of Latest Dates .......................................... 161
5B1: Calculating Latest Dates thru Default Restrictions ...................... 164
5B2: Calculating Latest Dates thru Start Restrictions ......................... 166
5B3: Calculating Latest Dates thru Finish Restrictions ..................... 171
5B4: Backward Pass: Putting It All Together ...................................... 174
Chapter 6: All About Total Float

6A: Five Key Elements to All CPM Schedules
6B: Schedule Float: Two Most Prevalent Types
   6B1: Total Float: A General Understanding
   6B2: Free Float: A General Understanding
6C: Understanding Total Float
   6C1: Total Float Calculations
   6C2: Calculating Total Float across Sequentially-Tied Activities
   6C3: Calculating Total Float within Overlapped/Fragmented Activities
6D: Total Float: Ideological Discussions
   6D1: Total Float is an Activity Path Value, Not an Activity Value
   6D2: About Most Critical Total Float
   6D3: Total Float Quantities are Unreasonably Optimistic
   6D4: Total Float Grossly Unstable
   6D5: Total Float Even Misunderstood by the “Experts”

Chapter 7: All About Critical Path

7A: Understanding the Concept of a Critical Path
7B: Critical Path: In a Nutshell
7C: Two Schools of Thought (Maybe Three!)
   7C1: A Cursory Introduction to Date Constraints
   7C2: Longest Path Definition
   7C3: Least Total Float Path Definition
   7C4: Driving Path Definition
7D: Critical Path Misunderstood by the “Experts”
   7D1: What We Have Learned about Critical Path
   7D2: Common Mistakes in Definitions of Critical Path
   7E: Critical Path: Ideological Discussions
   7E1: Cognitive Recommends the Paramount Path, But…
   7E2: One Paramount (Critical) Path per Finish Milestone
## Table of Contents

7E3: What is an Activity Path? 238  
7F: Tracing the Critical/Paramount Path 241  
7F1: Critical/Paramount Path Across Default Restrictions 241  
7F2: Tracing a Critical Path across Overlapped Activities 247  

### Chapter 8: Free Float and Symbiotic Networks 251

8A: Understanding Free Float 251  
8A1: Free Float Calculations 251  
8A2: Free Float: Ideological Discussions 255  
8A3: Free Float Defined 261  
8B: More about a Symbiotic and Progressive Relationships 263  
8B1: More about Symbiotic Relationships 263  
8B2: More about Progressive Relationships 266  
8B3: Phenomenon of Multi-Path Residency 267  

### Chapter 9: Supplemental Schedule Elements 273

9A: Essential, Supplemental, and Exploitive Schedule Elements 273  
9B: Elements that Provide Advanced Modeling Capabilities 274  
9B1: Understanding Date Constraints 275  
9B2: Understanding Work Performance Calendars 293  
9B3: Understanding Software Settings 299  
9C: Elements that Facilitate Schedule Data Management 308  
9C1: Understanding Activity Codes 309  
9C2: Understanding a Work Breakdown Structure (WBS) 311  
9C3: Activity-Resident Notations and Logs 314  

### Chapter 10: Exploitive Schedule Elements 317

10A: Work Breakdown Structure (WBS) 319  
10A1: What a WBS Is 320  
10A2: How a WBS is Used 320  
10B: Activity-Resident Project Cost Information 322
Chapter 11: Putting Reality into the Project Schedule

11A: The Project Schedule Must Model Reality
11A1: Planning, Scheduling … and Reality
11A2: Reflecting Reality Makes the Project Schedule Really Dynamic
11A3: “Reality” is Performance, Not Just Ingredients
11B: How Time is Understood in a Project Schedule
11B1: About Schedule Editions and Schedule Iterations
11B2: About Schedule Tense
11B3: Introducing the Data Date
11B4: Time Passage and Performance Achievement
11C: Temporal Context of Schedules: Schedule Segments
11C1: Performance Periods: Their Dimensions and Boundaries
11C2: Schedule Segment Data: Contextual Perspective
11C3: Inherent Nature of Different Schedule Segment Data Sets

Chapter 12: Schedule Data Credibility Profile

12A: The Schedule Data Credibility Profile
12A1: Credibility Profile Locates Data Date on Project Timeline
12A2: Schedule Data Integrity Improves Across the Project Length
12A3: Credibility Profile Explains Project Model Gradual Usefulness
12A4: Understanding the Schedule Data Credibility Profile
12A5: The Schedule Data Credibility Profile on Construction Projects
<table>
<thead>
<tr>
<th>Chapter 12: Making a CPM Execution Model Really Dynamic!</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12A6: What GRASP Tells Us about Dominant Project Management</td>
<td>375</td>
</tr>
<tr>
<td>12B: Making a CPM Execution Model Really Dynamic!</td>
<td>378</td>
</tr>
<tr>
<td>12B1: CPM Schedule is Actually THREE Project Execution Models</td>
<td>378</td>
</tr>
<tr>
<td>12B2: Secret to Most Effective Execution Model: Gleaned Data</td>
<td>380</td>
</tr>
<tr>
<td>12B3: GRASP the True Meaning of a Dynamic Project Schedule</td>
<td>381</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 13: CPM Calcs During Schedule Statusing</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13A: Understanding Schedule Updates</td>
<td>383</td>
</tr>
<tr>
<td>13A1: Schedule Updating Process, in General</td>
<td>383</td>
</tr>
<tr>
<td>13A2: Schedule Updating Considerations, in General</td>
<td>386</td>
</tr>
<tr>
<td>13A3: CPM Terms Associated with Schedule Updating</td>
<td>386</td>
</tr>
<tr>
<td>13B: Data-Adjusting Calculations during Schedule Update</td>
<td>388</td>
</tr>
<tr>
<td>13B1: Two Essential Variables Important to Schedule Updating</td>
<td>389</td>
</tr>
<tr>
<td>13B2: Understanding Out-of-Sequence Work</td>
<td>394</td>
</tr>
<tr>
<td>13B3: How to Interpret a Schedule Update</td>
<td>403</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 14: Project Time Management Alternatives</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14A: Arrow Diagramming Method: Original CPM Format</td>
<td>411</td>
</tr>
<tr>
<td>14B: PERT: Program Evaluation and Review Technique</td>
<td>417</td>
</tr>
<tr>
<td>14B1: PERT’s More Complex Duration Estimating Technique</td>
<td>417</td>
</tr>
<tr>
<td>14B2: PERT’s Flowcharting Format</td>
<td>419</td>
</tr>
<tr>
<td>14B3: PERT’s Legacy: Living on in the Critical Path Method</td>
<td>419</td>
</tr>
<tr>
<td>14C: Linear Scheduling and Line of Balance Scheduling</td>
<td>419</td>
</tr>
<tr>
<td>14D: Building Information Modeling (BIM)</td>
<td>421</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 15: Interpreting the Project Execution Model</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15A: What the Critical Path Method of Modeling is Not!</td>
<td>423</td>
</tr>
<tr>
<td>15A1: Virtually All Project Model Content is Speculative</td>
<td>423</td>
</tr>
<tr>
<td>15A2: Best Project Execution Model Never More than a Good Guess</td>
<td>427</td>
</tr>
<tr>
<td>15A3: Project Model: A Reflection of Compromise</td>
<td>427</td>
</tr>
<tr>
<td>15A4: Project Model: While Not Flawless, Better than the Alternative</td>
<td>427</td>
</tr>
</tbody>
</table>
ICS-Compendium Volume 1: CPM Mechanics

15B: What the Critical Path Method of Modeling Is! 428
15C: Not All Project Execution Models are Schedules 430
15C1: Project Execution Modeling versus Project Scheduling 431
15C2: Project Time Management Lacks Comprehensive Training 433
15D: Where Do You Go From Here? 435

Alphabetical Index .................................................................................................437
List of Figures and Illustrations

Chapter 1: CPM, A Modeling Method
F0102: Bottom Up ........................................................... 11

Chapter 2: Static Project Time Management Tools
F0202: Cognitive’s F.A.C.E. Diamond ........................................... 27
F0204: To Do List. ................................................................. 29
F0206: What Constitutes an Activity? ........................................... 33
F0208: Simple Timetable ....................................................... 35
F0209: The Sequence of Activities Matters .................................... 37
F0210: Simple Bar Chart ....................................................... 41
F0212: Start/Finish "Moments in Time" ........................................ 42
F0214: Bar Graph A ............................................................ 44
F0216: Bar Graph B ............................................................ 44
F0218: Bar Graph C ............................................................ 45
F0220: Bar Graph D ............................................................ 45
F0222: Bar Graph E ............................................................ 46
F0224: Continuous versus Impacted Durations ............................. 47
F0226: Combined Timetable and Bar Chart. ............................... 55
F0228: Two "Leapfrogging" Crews ............................................ 56
F0230: Computer-Generated "Confusing" Logic ............................ 57
F0232: Project Execution Commitments (Events and Milestones) ...... 61
F0234: Shipping Routes ...................................................... 65
Chapter 3: Dynamic Project Time Management Tools

F0302: Cognitive's Three Relationship Categories .................................................................74
F0304: Restriction Pair and Restriction Linkage ................................................................. 84
F0306: Restriction Linkage and Restriction Code ............................................................. 85
F0310: Example of Activity Subnetwork .............................................................................. 86
F0308: A Simple Activity Path ............................................................................................. 86
F0312: Example of Activity Network .................................................................................. 87
F0314: CPM's Four Performance Restriction Linkages ...................................................... 89
F0316: Restriction Code and Restriction Flow (in Logic Diagram) ..................................... 93
F0318: Finish-to-Start Dependency (Default Restriction) .................................................... 98
F0320: Start-to-Start Dependency (Start Restriction) ............................................................ 99
F0322: Finish-to-Finish Dependency (Finish Restriction) .................................................... 100
F0324: Start-to-Finish Dependency (Holdback Restriction), Possibility #1 ....................... 101
F0326: Start-to-Finish Dependency (Holdback Restriction), Possibility #2 ....................... 101
F0328: Start-to-Finish Dependency (Holdback Restriction), Possibility #3 ....................... 102
F0330: Start-to-Finish Dependency (Holdback Restriction), Possibility #4 ....................... 103
F0332: Passage of Time v. Work Performed ....................................................................... 107
F0334: Problem with Negative Start-to-Start (Start Restriction) ........................................ 109
F0336: Alternative to Negative Start-to-Start (Start Restriction) ........................................ 110
F0338: Negative Finish-to-Start (Default Restriction) ......................................................... 110
F0340: Alternative to Negative Finish-to-Start (Default Restriction) ................................... 111
F0342: Positive Finish-to-Start (Default Restriction) ............................................................ 112

Chapter 4: How to Draw CPM Logic

F0402: Calculating Gregorian Dates ...................................................................................... 118
F0404: Calculating Ordinal Dates ......................................................................................... 119
F0406: Key Schedule Elements in CPM Network-Based Schedule .................................... 122
F0408: State of Being - Open .............................................................................................. 124
F0410: State of Being - Closed ........................................................................................... 124
F0412: Work Performance ................................................................................................... 124
F0414: An Activity ............................................................................................................... 125
F  List of Figures and Illustrations

F0416: Not Just a Box .................................................................................................................................125
F0418: Anatomy of a Box ...............................................................................................................................126
F0422: The "End" Zone ...................................................................................................................................128
F0424: The "Finished" State .............................................................................................................................128
F0420: The Arrows of CPM ..............................................................................................................................128
F0426: After the Activity has Finished ...........................................................................................................128
F0428: "Not Yet Started" State ..........................................................................................................................129
F0430: Before the Activity's Start .....................................................................................................................129
F0432: Finish-to-Start Tie (Default Restriction) .............................................................................................129
F0434: Positive Default Restriction (Finish-to-Start) Tie .............................................................................130
F0436: Possible Arrow Connection Points ......................................................................................................131
F0438: Improper SS Hook Point .......................................................................................................................131
F0440: More Improper SS Hook Points ...........................................................................................................131
F0432: Ideal SS Hook Points .............................................................................................................................132
F0444: Possible Connection Points ..................................................................................................................133
F0446: Proper SS Ties ........................................................................................................................................133
F0448: Standard SS Logic Tie ...........................................................................................................................134
F0450: Standard FF Logic Tie ...........................................................................................................................134
F0452: Discerning Restriction Type by Sight? .................................................................................................135

Chapter 5: Calculating Primary CPM Dates

F0502: Three Activities; Forward Pass Exercise ............................................................................................141
F0504: What's the Earliest Start? ......................................................................................................................142
F0508: Another Earliest Finish .........................................................................................................................143
F0506: The Earliest Finish Is? ...........................................................................................................................143
F0512: Determine the Earliest Dates ...............................................................................................................145
F0510: Solved Earliest Finish ............................................................................................................................145
F0514: Earliest Dates (Solved) ..........................................................................................................................146
F0516: Two Restricting Activities ....................................................................................................................147
F0518: FP Multiple Restricting Rule ................................................................................................................147
F0520: Forward Pass thru Subnet, Default Restrictions Only (Problem) ...................................................149
F0522: Forward Pass thru Subnet, Default Restrictions Only (Solution) ............................................................... 150
F0524: FP thru Single Start Restriction ...................................................................................................................... 150
F0526: Finding ES for Activity B ......................................................................................................................................... 151
F0528: Multiple Start Restrictions (Problem) .................................................................................................................. 152
F0530: Multiple Start Restrictions (Solution) .................................................................................................................. 152
F0532: Multiple Arrows Reminder ...................................................................................................................................... 154
F0534: What Restriction Delays Refer To? .................................................................................................................................. 156
F0536: Combined SS/FF Ties (Problem) ............................................................................................................................ 156
F0538: Earliest Finish Arrow Count Rule ........................................................................................................................ 157
F0540: Solution to Figure F0536 ........................................................................................................................................ 158
F0542: Calculate Earliest Dates ........................................................................................................................................ 158
F0544: Calc Earliest Dates (Solved) .................................................................................................................................... 159
F0546: Complicated Forward Pass of Subnetwork (Problem) .............................................................................................. 160
F0548: Complicated Forward Pass of Subnetwork (Solution) .............................................................................................. 162
F0550: Simple BP thru Default Ties (Problem) .................................................................................................................... 163
F0552: Simple BP thru Default Ties (Solution) ................................................................................................................... 164
F0554: BP Multiple Restricted Rule .................................................................................................................................. 165
F0556: Simple Backward Pass ............................................................................................................................................... 165
F0558: BP thru Default Restrictions (Problem) .................................................................................................................. 166
F0560: Backward Pass thru Start Restriction ........................................................................................................................ 167
F0562: BP thru Default Restrictions (Solution) ................................................................................................................... 168
F0564: Latest Start Arrow Count Rule ..................................................................................................................................... 169
F0566: Backward Pass thru SS and FF Ties (Problem) ........................................................................................................... 171
F0568: Backward Pass thru SS and FF Ties (Solution) ............................................................................................................ 172
F0574: BP thru S/F Restrictions (Solution) .......................................................................................................................... 173
F0572: BP thru S/F Restrictions (Problem) .......................................................................................................................... 173
F0570: FF Affects Activity End .................................................................................................................................................. 173
F0576: Forward Pass/Backward Pass Comprehensive Exercise (Problem) ............................................................................. 175
F0578: Forward Pass/Backward Pass Comprehensive Exercise (Solution) ............................................................................. 177
## List of Figures and Illustrations

### Chapter 6: All About Total Float

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0604</td>
<td>Getting to School</td>
<td>182</td>
</tr>
<tr>
<td>F0602</td>
<td>Schedule Tabular Report</td>
<td>182</td>
</tr>
<tr>
<td>F0606</td>
<td>Birthday Call, Scenario 1</td>
<td>184</td>
</tr>
<tr>
<td>F0608</td>
<td>Birthday Call, Scenario 2</td>
<td>185</td>
</tr>
<tr>
<td>F0610</td>
<td>Simple Logic Problem</td>
<td>187</td>
</tr>
<tr>
<td>F0612</td>
<td>Forward/Backward Pass</td>
<td>187</td>
</tr>
<tr>
<td>F0616</td>
<td>Start/Finish Total Float</td>
<td>189</td>
</tr>
<tr>
<td>F0614</td>
<td>Path Total Float Law</td>
<td>189</td>
</tr>
<tr>
<td>F0618</td>
<td>Forward Pass Changes Can Affect Total Float</td>
<td>191</td>
</tr>
<tr>
<td>F0620</td>
<td>Backward Pass Changes Can Affect Total Float</td>
<td>191</td>
</tr>
<tr>
<td>F0622</td>
<td>Tracing Least Float Path (Problem)</td>
<td>192</td>
</tr>
<tr>
<td>F0624</td>
<td>Tracing Least Float Path (Solution)</td>
<td>193</td>
</tr>
<tr>
<td>F0626</td>
<td>Forward Pass View of Fragmented Activities</td>
<td>194</td>
</tr>
<tr>
<td>F0628</td>
<td>Backward Pass View of Fragmented Activities</td>
<td>195</td>
</tr>
<tr>
<td>F0630</td>
<td>Different STF and FTF Values</td>
<td>196</td>
</tr>
<tr>
<td>F0632</td>
<td>Different TF = Different Routes</td>
<td>196</td>
</tr>
<tr>
<td>F0634</td>
<td>Analyzing Activity Fragmentation</td>
<td>198</td>
</tr>
<tr>
<td>F0636</td>
<td>Recognizing Fragmented Activities (Problem)</td>
<td>201</td>
</tr>
<tr>
<td>F0638</td>
<td>Recognizing Fragmented Activities (Solution)</td>
<td>203</td>
</tr>
<tr>
<td>F0640</td>
<td>FS Only ... Means Matching STF/FTF Values</td>
<td>204</td>
</tr>
<tr>
<td>F0642</td>
<td>Total Float is Redundantly Stated in the Typical CPM Printout</td>
<td>206</td>
</tr>
<tr>
<td>F0644</td>
<td>Comparing Total Float Options</td>
<td>208</td>
</tr>
<tr>
<td>F0646</td>
<td>Most Critical Total Float</td>
<td>209</td>
</tr>
<tr>
<td>F0648</td>
<td>Start Total Float</td>
<td>209</td>
</tr>
<tr>
<td>F0650</td>
<td>Finish Total Float</td>
<td>210</td>
</tr>
<tr>
<td>F0652</td>
<td>Five Day Workweek versus Six Day Workweek</td>
<td>211</td>
</tr>
<tr>
<td>F0654</td>
<td>How Float Works</td>
<td>214</td>
</tr>
</tbody>
</table>
Chapter 7: All About Critical Path

F0702: Impact of Multiple Date Constraints ......................................................................................... 224
F0704: Multiple Date Constraints ............................................................................................................ 226
F0706: Different Date Constraints ............................................................................................................ 226
F0708: Finding the Longest Path through the Subnet ............................................................................... 227
F0710: Paramount (Critical) Path Leading to Finish Milestone Z ............................................................... 229
F0712: Paramount (Critical) Path Leading to Finish Milestone M from Start Milestone A ....................... 242
F0714: Paramount (Critical) Path Leading to Finish Milestone U from Start Milestone A ....................... 244
F0716: Paramount (Critical) Path Leading to Finish Milestone X from Start Milestone N ....................... 246
F0718: Paramount (Critical) Path Leading to Finish Milestone Z from Start Milestone N ....................... 248

Chapter 8: Free Float and Symbiotic Networks

F0802: Two Restricting Activities ............................................................................................................. 252
F0804: Three Restricting Activities .......................................................................................................... 252
F0808: Understanding Start Free Float ..................................................................................................... 253
F0806: Understanding Free Float ............................................................................................................. 253
F0810: Understanding Finish Free Float .................................................................................................. 254
F0812: Understanding Most Critical Free Float .......................................................................................... 255
F0814: Two Overlapping Activity Paths .................................................................................................. 256
F0816: Activity R Starts Late ...................................................................................................................... 257
F0818: Activity R Takes Three Times Longer .............................................................................................. 258
F0820: Activity S Takes Longer ................................................................................................................... 259
F0822: Activity T Delayed ........................................................................................................................... 260
F0824: Schedule Editions ... and Free Float Changes ............................................................................... 264
F0826: How Total Float is "Nested" .......................................................................................................... 266
F0828: In Overlapping Activity Paths, Total Float is a Cumulative Value .................................................. 269
F0830: Understanding Total Float in Overlapping Activity Paths .............................................................. 270

Chapter 9: Supplemental Schedule Elements

F0902: Start-No-Earlier-Than Date Constraint .......................................................................................... 276
F0904: Finish-No-Earlier-Than Date Constraint ......................................................................................... 277
List of Figures and Illustrations

F0906: Start-No-Later-Than Date Constraint .................................................................277
F0908: Finish-No-Later-Than Date Constraint .................................................................278
F0910: Expected Finish Date Constraint ........................................................................279
F0912: Mandatory Start Date Constraint .........................................................................280
F0914: Mandatory Finish Date Constraint .......................................................................281
F0916: Start On Date Constraint .....................................................................................281
F0918: Two Start-No-Earlier-Than Date Constraints ......................................................282
F0920: Solution to Problem at Figure F0918 .................................................................283
F0922: Start-No-Earlier-Than AND Finish-No-Later-Than ...........................................284
F0924: Our Solution to Figure F0922 .............................................................................285
F0926: Finding Critical Path Through Date Constraints .................................................287
F0928: Our Solution to Problem in F0926 .................................................................289
F0930: Finding the Longest Path .....................................................................................291
F0932: Three Different Total Float Values ....................................................................292
F0934: C1: 7-Day Work Calendar ................................................................................293
F0936: C2: 6-Day Work Calendar ................................................................................293
F0938: C3: 5-Day Work Calendar ................................................................................294
F0942: Convert Ordinal to Gregorian Dates .................................................................294
F0940: C4: 5-Day Work Calendar with/Holidays ..........................................................294
F0944: Converting Ordinal to Gregorian Dates, using Seven-Day Workweek ..............295
F0946: Converting Ordinal to Gregorian Dates, using Five-Day Workweek ..................295
F0948: Ordinal to Gregorian Date Conversion Problem ..............................................296
F0950: Solution to Figure F0948 ....................................................................................296
F0952: How to Convert Dates .......................................................................................297
F0954: Date Conversion Exercise ................................................................................298
F0956: Solution to Figure F0954 ................................................................................298
F0958: Example of Interruptible Duration .....................................................................300
F0960: Example of Contiguous Duration ......................................................................301
F0962: Effect of Contiguous Durations on Earliest Dates (Before) ...............................302
F0964: Effect of Contiguous Durations on Earliest Dates (After) .................................303
Chapter 10: Exploitive Schedule Elements

F1002: WBS: Excavation Budget Line Items ................................................................. 325
F1004: Earned Value Calculations ............................................................................. 329

Chapter 11: Putting Reality into the Project Schedule

F1102: Schedule Editions and Schedule Iterations ..................................................... 344
F1104: Project Length and Schedule Periods ............................................................ 350

Chapter 12: Schedule Data Credibility Profile

F1202: How the Five Segments of Data Fit Together ............................................... 368
F1204: Importance of Gleaned Data per Cognitive Project Management ............... 370
F1206: Schedule Data Credibility Profile .................................................................. 371
F1208: Use of Gleaned Data per Dominant Project Management ......................... 375
F1210: One Schedule with Three Schedule Segments of Data .............................. 379

Chapter 13: CPM Calcs During Schedule Statusing

F1302: How to Incorporate Remaining Durations ...................................................... 389
F1306: Updating Basic Steps .................................................................................... 390
F1304: Project Manager’s Update Email ................................................................. 390
F1308: Forward/Backward Pass Calculations ....................................................... 391
F1310: Schedule Update Exercise .......................................................................... 392
F1312: Previous Schedule Edition .......................................................................... 393
F1314: Solution to Figure F1312 Exercise ............................................................... 394
F1316: Skipped/Performed Work on SAME Path .................................................. 395
F1318: Analyzing Figure F1316 ............................................................................. 395
F1320: Skipped/Performed Work on DIFFERENT Path ...................................... 397
F1322: Analyzing Figure F1320 ............................................................................. 398
F1324: How Out-of-Sequence Really Works ............................................................ 400
F List of Figures and Illustrations

F1326: Applying Progress Override .................................................................402
F1328: A Schedule Edition ..................................................................................404
F1330: Information Needed for
Exercise in Figure F1332 ..................................................................................405
F1332: Update Exercise ......................................................................................406
F1334: Solution to Figure F1332 .......................................................................408

Chapter 14: Project Time Management Alternatives

F1402: Simple Timetable .....................................................................................412
F1404: Piping Example ......................................................................................413
F1406: Subnet Drawn in Precedence Diagramming Method (PDM) ....................414
F1408: Subnet Drawn in Precedence Diagramming Method (PDM) ....................416
F1410: Example of Line of Balance Scheduling ................................................420
In Loving Memory

Barry Burton
Eva Ruth Burton
Jamie Kay Burton
Otto and Evadna Burton
David and Sarah Kahan
Mary Ramona Phillips
Willis Harold and Bertha Mae Phillips
Mina Elizabeth Robbins
Benjamin and Mollie Woolf
Sandra Lee Woolf
Simon and Josephine Woolf

We thought of you with love today,
but that is nothing new
We thought about you yesterday,
and days before that too.

We think of you in silence;
we often speak your name.
Now all we have are memories,
and your picture in a frame.

But your Legacy's our Inspiration,
with which each new day starts.
God surely has you in His keeping,
But we have you safely in our hearts.

"When someone you love becomes a memory,
the memory becomes a treasure."

Page xxviii
Copyright, 2012 International Center for Scheduling, Inc. (ICS-Global) All rights reserved.
The Compendium: The Complete Collection on Construction Project Time Management

All writing, if done well, is a laborious affair — some more taxing than others. Scholarly papers require extensive research and documentation, while fictional pieces can pour from the writer’s mind faster than one can type. But level of effort is a poor way to explain the challenge. Since each writer has his or her own approach to writing, there really is no such thing as a single best process.

Now imagine writing not just one book, but a ten-volume set of books, with some being highly technical while others are mainly ideological and philosophical. Consider further that the “writer” is not a single individual, but instead a team of writers, each with his or her own writing style, points of view, and unique growth of practical experiences.

Perhaps more challenging than any other factor is the dual objectives of the ICS-Compendium: to simultaneously describe how things are, as well as how much better things could be.

- **How Things Are:** The first four volumes of the ICS-Compendium provide a complete guide to Dominant Project Management’s approach to Construction Project Time Management. Needless to say, bookstores are filled with many other lauded works that cover the same subject matter. If the hope for the ICS-Compendium is for it to improve upon all other treatments of these topics, surely it must do so by providing a consistent set of terms with precise and interrelated definitions, highest quality graphics, an easy-to-understand writing style, and a completely accurate and comprehensive treatment of all related topics.

- **How Much Better Things Could Be:** The second four-volume set within the ICS-Compendium provides the foundation for a new approach to Construction Project Time Management. The hundreds of revolutionary ideas that collectively comprise Cognitive Project Management in large part came out of thin air — that is, they were inspired. To be sure, each member of the ICS-Compendium Development Team came to the table with a rich history of past experience, an incredible wealth of practical knowledge, and a deep passion for “The ICS-Compendium Project,” itself.

Expressions of Gratitude
The Silent Partner: During the writing of this book, it was commonplace for ICS-Compendium Development Team members to report having “an epiphany” upon awakening. These incidents of Inspiration happened so often that we soon began to feel as though we were part of something much bigger than any one of us, or even of all of us combined — as if we had been selected to ghost write a literary work that was somehow being designed, crafted, and assembled by a Silent Partner.

To be sure, all members of the ICS-Compendium Development Team have a strong sense of Spirituality. And how could we not? So many of the ideas that we brought to the Morning Meetings were as new and confusing to the person explaining them as they were to the rest of us hearing them for the first time. It was as if we were all “tapped into” a Universal Mind.

Talk of Spirituality and Universal Energy may feel a bit odd or uncomfortable for many construction types, who are anything but touchy-feely. I realize that I run the risk of “losing them” if I say much more about ephemeral energies in the cosmos. But I would be dishonest if I did not acknowledge at least the perception, if not the reality, that development of the ICS-Compendium was being guided by something much greater than any one of us.

How else can you explain how ideas that came from complete strangers and that sort of “dawned on us” across a period of five years, somehow seemed to connect with one another like so many finely-cut pieces of a jigsaw puzzle? How do you explain that, once assembled, the puzzle pieces revealed a Big Picture that not one of us had the slightest previous Vision of, in our own minds? I just wish you could have been with us when, as the last puzzle piece was laid in place, we all stepped back and just stood there in complete silence and amazement, dumb-founded as we were with how it all “fit so perfectly.”

So, in a short piece entitled **EXPRESSIONS OF GRATITUDE**, the first such expression goes to our Silent Partner, to this Universal Mind that selected and gathered the members of the ICS-Compendium Development Team in the first place, and then fed them with daily doses of Inspiration.

**Thought Contributors:** The next Expression of Gratitude goes to my colleagues who have contributed to this Body of Work. While we have each commented at one time or another that with respect to the ICS-Compendium the Work is its own reward, the truth is that we have additionally benefited by the effort itself. We have learned so much, we have trusted completely, and we have grown into lifelong friends.

It is difficult to even know, let alone recall, all of the individuals who across the decades have contributed to the ICS-Compendium. Over the years I have maintained a **List of People Who Have Inspired and Guided Me**. The List is simply too long to include in its
entirety. And so I decided that, with each new book that I write, I will draw from that continually growing List the names those who were most influential to whatever work product I might be working on. The extent of their influence ranges from a single idea or action to the most overpowering deluge of insights, effort, wisdom, and inspiration.

That said, I would like to thank the following individuals for their contribution to this book. Some will understand why their name appears; many others will not. All I can say is that each of you has made an indelible mark on me, and I am so very grateful.

<table>
<thead>
<tr>
<th>Name</th>
<th>Company/Position</th>
<th>Institution/Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rey Diaz-Acevedo</td>
<td>Digital Project Solutions</td>
<td>Catholic University of America</td>
</tr>
<tr>
<td>Delbert Bearden</td>
<td>Bear3 Consulting</td>
<td>General Electric Energy</td>
</tr>
<tr>
<td>Jeffrey Huneycutt</td>
<td>US Army Corps of Engineers</td>
<td>ProjectControlsOnline.com</td>
</tr>
<tr>
<td>Mark Dochtermann</td>
<td>Microsoft; Advisicon</td>
<td>ADR Consulting</td>
</tr>
<tr>
<td>Paul McCool</td>
<td>Black &amp; Veatch</td>
<td>Alpha Corporation</td>
</tr>
<tr>
<td>Ronny Warren</td>
<td>University of North Texas</td>
<td>Brasfield &amp; Gorrie</td>
</tr>
<tr>
<td>Max Wideman</td>
<td>Shaw Group</td>
<td>Constellation Energy</td>
</tr>
<tr>
<td>Amy Hazuda</td>
<td>Touchstone Construction</td>
<td>Kitchell</td>
</tr>
<tr>
<td>Patrick Weaver</td>
<td>Mosaic Project Services</td>
<td>CPM Schedules</td>
</tr>
<tr>
<td>Hany ElBanna</td>
<td>HSE Scheduling Consultants</td>
<td>Crane Technologies</td>
</tr>
<tr>
<td>David Tyerman</td>
<td>Athena Project Services</td>
<td>L.G. Barcus &amp; Sons</td>
</tr>
<tr>
<td>John Herbrig</td>
<td>Barton Malow Company</td>
<td>Andersen Construction</td>
</tr>
<tr>
<td>Tony Bernard</td>
<td>Bernard Consulting</td>
<td>Hill International</td>
</tr>
<tr>
<td>Jose Noe</td>
<td>BHP Billiton</td>
<td>WPL Publishing</td>
</tr>
<tr>
<td>Gary Ehrlich</td>
<td>Bluegreen Corporation</td>
<td>Boeing</td>
</tr>
<tr>
<td>Stephen Wilchek</td>
<td>Amtrak</td>
<td>Brix Corporation</td>
</tr>
<tr>
<td>Bob Blaskowski</td>
<td>Caterpillar (Bucyrus)</td>
<td>Burns &amp; McDonnell</td>
</tr>
<tr>
<td>Fred Plotnick</td>
<td>EnProMac</td>
<td>US Army Corp of Engineers</td>
</tr>
<tr>
<td>Thomas Vandemars</td>
<td>Northrop Grumman</td>
<td>Kodak</td>
</tr>
<tr>
<td>Edward Middlewick</td>
<td>South African Airways</td>
<td>Georgia Department of Labor</td>
</tr>
<tr>
<td>Jan Wilson</td>
<td>Food Services of America</td>
<td>Boeing</td>
</tr>
<tr>
<td>Trevor Nelson</td>
<td>Nelson Project Consulting</td>
<td>Rutgers University</td>
</tr>
<tr>
<td>Chris Hebert</td>
<td>Taylor Brothers Construction</td>
<td>URS</td>
</tr>
<tr>
<td>Kevin Cranford</td>
<td>Project CPM</td>
<td>Barba Consulting</td>
</tr>
<tr>
<td>Bob Schreier</td>
<td>Florida Power &amp; Light</td>
<td>Highfill Consulting</td>
</tr>
<tr>
<td>John Scharf</td>
<td>University of California</td>
<td>Apprill Resources</td>
</tr>
<tr>
<td>Keith Howard</td>
<td>Motorola</td>
<td>Archibald Consultants</td>
</tr>
<tr>
<td>Sam Mayes</td>
<td>Management Solutions</td>
<td>CPM Mechanics</td>
</tr>
</tbody>
</table>
My Closest Colleagues: There is a special class of colleagues who have grown into close friends, and who deserve a special acknowledgement for the important role that each played in the creation of this book in particular. While each one’s contribution was unique, in common they shared a strong and driving conviction that the Construction Industry desperately needs the ICS-Compendium. I express my deepest gratitude:

- To Roger DeHondt, the first to climb aboard the ICS-Compendium train, for his daily partnership and deep friendship;
- To Kay DeCaro for the “lecture hall and cathedral;”
- To Andy Schack for weekly aligning me to my goals (enjoy your newborn!)
- To Amy Hazuda for her ever-inspiring positive outlook and cheerful demeanor;
- To Travis Neal, a selfless aide and good friend, willing to do anything and everything to “get the book out the door” — your spiritual strength inspires me;
- To Rey Diaz (mi hermano en Puerto Rico), my friend, confidante, and advisor;
- To Gunnar Lucko (the “Professor”) for providing the Academia Perspective
- To Jeff Huneycutt for providing the Government Perspective
- To Delbert Bearden for providing the Scheduler’s Perspective
- To Ronny Warren for providing the Trainer’s Perspective
- To Thomas Congleton for providing the Owner’s Perspective
- To Chad Lanier and Paul McCool for providing the Contractor’s Perspective
- And, to Anil Godhawale for providing the European Perspective.
Expressions of Gratitude

My Family: Last but not least, there is my family — whose steadfast support of “Murray's crazy ideas” were manifested in great acts of love and personal sacrifice, devotion that never escaped my awareness. I am indebted to all of you, for the small things you did that allowed me to single-mindedly focus on my life-long pursuit of a better Project Time Management system for the Construction Industry. You rode shotgun, cleared the rocks from my path, tended to the daily grind -- in a word, indulged me.

First and foremost, there is my dear wife, Vicki – you are the rock that anchors me. You keep me grounded when my drifting mind would take me aloft. Your wise counsel and uncluttered perspective repoint me in the right direction. Your wealth of knowledge, accumulated over a lifetime, has well earned you the nickname, “Vikipedia.” Your devotion, love, and friendship fuel my days. Your patience and gentle demeanor keep me level in the roughest of seas. I would be lost without you. I love you ... today!

To my daughters, who have sacrificed so much over the years so that their father might respond to persistent voices in his head that have called out to him for so long and drawn him (and his family) to so many remote places. Susan, congratulations on the engagement; I can’t wait. Lori, a special Expression of Gratitude to you for your deep wellspring of faith and inspiration, revealed by words that so often reassured me in my doubting hours.

To my friend of a lifetime and the mother of my children, Patti. You were with me when I started out in the construction business 37 years ago. You were the one who worked multiple jobs outside the home, in addition to being a mother and homemaker, so that I might go to college and get a degree in Construction Management. I would not be where I am now, if it were not for your faith, devotion, and steadfast support. You have my most heartfelt gratitude, always.

To my brother Bob, you helped incubate ICS-Global in its opening year of life, and gave so selflessly with only one question at the start of each new day: “What can I do to help?” And to my parents-in-law Gene and Jeanne Phillips, thank you for the moral support, and for never failing to inquire in each Sunday evening call from Florida, “How is the book coming?” I so very much appreciate you both — for your steady flow of love, your deep friendship, your wise counsel, and for opening your home to us each summer.

Whether it truly takes a village to raise a child I cannot say with certainty. But what I do know, beyond a shadow of a doubt, is that it takes much more than an author alone – to write a book. It also takes those who believe [in him and his cause], it takes those who support, those who assist, those who inspire, and those who love.

And then ... there is the Wizard behind the Curtain ... the Universal Mind that inspires and guides all that we do. To It – I am eternally indebted.
CHAPTER ONE

1A Introducing the ICS-Compendium
1B Exclusively for Construction Project Management
1C The Three Meanings of "CPM Scheduling"
1D The Book’s Organization and Structure
1E The Book’s Information Transfer Strategy
1E1 The Book’s Information Transfer Strategy
1E2 The Book’s Unique Formatting Symbolism
1E3 The ICS-Dictionary
**CPM, A Modeling Method**

Even before you begin this book, *CPM Mechanics*, there is something very important that you need to understand about it: *it has a dual purpose.*

1A: **Introducing the ICS-Compendium**

ICS-Global was founded in 2006 with a singular goal in mind: to enhance the Construction Industry's understanding of Project Time Management. Early studies by ICS-Research, its Think Tank division, discovered that a surprising amount of the prevailing Project Management dogma, as advocated by the world's leading Project Management authorities, was not especially well suited for the Construction Industry. This discovery seemed consistent with studies by other world-class organizations that reported Project failure rates on Construction Projects as high as 70%, in terms of Cost and Schedule goals not being met. [1]

After long thought and with certain hesitation, ICS-Global chose to authorize the creation of a Project Management model designed specifically for the Construction Industry, which we named Cognitive Project Management. In contrast, we dubbed the currently prevailing Project Management system Dominant Project Management, in recognition of its overwhelming influence around the world.

Once we began peeling away a few layers of the Dominant Project Management onion we found that, notwithstanding an overt name change from the previous *Planning and Scheduling* label to the new *Project Time Management* label, there was virtually nothing new or different about how Dominant Project Management envisioned the Time Management of Projects. It still came down to the basic three components: Schedule Development, Schedule Maintenance, and Schedule Control.

By contrast, Cognitive Project Management takes a much deeper and richer view of Project Time Management, especially as it relates to the Construction Industry. We insist that there is much more to Project Time Management than mere Schedule Development and Schedule Maintenance.

---

1 See Chaos Report 2009, Standish Group, as just one supporting study.
As for Schedule Control, Cognitive Project Management discourages its pursuit, since the very notion of control sends the wrong message to those that the Schedule is intended to support. Besides, true control is nearly impossible to attain and may very well work at cross-purposes to the overall objectives of the Project Team.

The profound importance of Project Time Management to overall Construction Project Management cannot possibly be overstated. As the ICS-Dictionary explains:

- **Project Time Management**: The central component of effective Project Management. A basic tenet of Cognitive Project Management is that effective Project Execution simply cannot be achieved without the correspondingly effective use of Time by the Project Execution Team. Project Time Management requires the development, maintenance, and use of products and services especially designed for this purpose.

What we realized at the outset was that there is a symbiotic relationship between Project Management and Project Time Management. From a practical perspective, if we were to change how Project Time Management was to work on Construction Projects, we would also have to change how Project Management itself works — for the latter constitutes the Operational Context (i.e., the Ecology) of the former.

As we saw it, one explanation for why Dominant Project Management does not work especially well on most Construction Projects is that its underlying Ideology is intentionally designed for mass appeal: to work for “most projects most of the time,” across any number of disparate Project Types and different industries. These generalities of principles and recommended practices render Dominant Project Management, as a coherent system, far too non-specific to support the intense operational demands of the typical Construction Project.

With a lump in our throats, we accepted that we would have to develop a new Project Management model, one specifically designed for the Construction Industry. And yet our initial charter and ultimate goal had not been to rewrite Project Management as a whole, but instead to discover and deliver the best Project Time Management system to the Construction Industry. There is much in Dominant Project Management that is good, valid, and beneficial to Construction Project Management. We did not want to throw the baby out with the bath water!

In short, we were quite clear among ourselves that we did not want to get lost in creating an entirely new Project Management model, but instead only those aspects of Project Management Ideology, Methodology, and Technology that could potentially enhance

---

2 This expression has permeated Dominant Project Management discussion for decades. These six words summarize an Ideology that believes that Projects across disparate industries are sufficiently similar to warrant a common set of Project Management "best practices" and "standards."
or inhibit effective Construction Project Time Management.

As a result, Cognitive Project Management (as of this writing) is an incomplete Project Management system. The extent of our developmental work can be summarized as these achievements:


- **When it comes to the temporal aspects of Project Management, ICS-Global spent five years in development of a completely new and different approach to Construction Project Time Management. Compatible with Cognitive's T.I.M.E. FRAMEWORK, ICS-Global developed a comprehensive and tightly interconnected set of Technological, Ideological, Methodological, and Ecological innovations which, when taken as a whole and practiced with sincere commitment, are capable of dramatically improving how Construction Projects ultimately turn out, especially in terms of budget and schedule.**

- **Cognitive's F.A.C.E. Diamond**: As a recommendation for a more responsive Construction Project Management organization, Cognitive Project Management recognizes four distinct Project Management Domains at play on each Construction Project. Each domain has its own organizational structure, required skill sets, functional processes, and success criteria. The acronym, F.A.C.E., derives from the names of the domains: Facilitation, Administration, Collaboration, and Execution.

- **Cognitive Ideology**: The Cognitive Project Management Ideology constitutes a radically different set of beliefs and values than are espoused by Dominant Project Management. These Ideological differences impact the choices and implementation of Project Time Management Methodologies and Technologies.

- **Momentum Management**: Borne out of the Cognitive Ideology, Momentum Management embraces both Technological and Methodological innovations aimed at facilitating Project Execution efforts that consistently result in greater temporal (and fiscal) successes. Momentum Management entails planning, measuring, monitoring, directing, and influencing the rate at which Work is being performed.

But beyond introducing these high-level innovations, ICS-Global leaves it to other Project Management interests and future generations of Cognitive Project Management proponents to put meat on the bone with respect to their specific areas of expertise.

---

3 These innovations are discussed more fully elsewhere in this book and within the ICS-Compendium.
We encourage those with a special passion for Cost Management, Risk Management, Human Resources Management, Communications Management, and other essential Project Management areas of focus to find new and better ways to implement Cognitive Project Management Ideology, Methodology, and Technology.

The ICS-Compendium was written with the typical North American Constructor squarely in mind. Our decision to zero in on such a specific target audience arose out of a belief that Construction Projects differ as much among themselves (affected by a host of distinguishing variables) as they do from Projects of other industries.

**Constructor:** The term, Constructor, refers to a variety of business entities that might have a vested interest in the timely performance of Construction Projects. Included would be Owners, Architects, Engineers, Construction Managers, Owner's Representatives, General Contractors, Specialty Contractors, and so forth.

As mentioned earlier, ICS-Global disagrees with the notion that a single set of Standards or Best Practices can be devised that will work equally well for “most projects most of the time.” And so, ICS-Global examined the innate characteristics of Project attributes, in conjunction with the temporal goals of Project Management, in an effort to identify the largest set of Projects with sufficient similarity to warrant a common "Solution."

While the target audience for the ICS-Compendium is Constructors of medium to large Construction Projects in the United States and Canada, this not to say that the contents of the ICS-Compendium might not benefit Constructor organizations outside of this defined group. For instance, stakeholders in small or mega Projects may find much value in ICS-Compendium content.

Likewise, Constructors around the world, regardless of Project size, might also find much within the ICS-Compendium to improve the temporal outcome of their Projects, although a certain amount of translation (of native language, regional terminology, methodological differences, and business culture nuances) might be warranted.

The ultimate expectation and intention is that the ICS-Compendium will find its way onto the shelves of the typical Constructor company library, where it will serve as a set of reference volumes that can inform and guide implementation of highly effective Project Time Management. Owners and Contractors alike will benefit from its advice.

To meet this goal, the ICS-Compendium “assumes nothing” about its readers, except that they know how to build Projects. When it comes to Project Management or Project Time Management, the ICS-Compendium “starts from scratch.” Its collective content across ten volumes covers all of Project Time Management, from soup to nuts.
This book, for instance, is one of the basic volumes in the set, and introduces the Critical Path Method as a mechanism for modeling Project Execution Strategy. Other volumes will explain how to employ the Critical Path Method to effectively manage the use of Time in the course of building a Project.

Back to the first sentence of this book, we stated that CPM MECHANICS has dual objectives. It all ties back to the current state of affairs in the world of Construction Project Management. On the one hand, Dominant Project Management currently dominates (hence, the name choice) Project Management literature and conventional wisdom, yet it really doesn't seem to work especially well in the Construction Industry. On the other, Cognitive Project Management is ideal for the Construction Industry but is not yet very well known.

Moreover, in order for Cognitive Project Management to be practiced on a given Project, certain changes in attitude and approach are required of the Project's Owner, such that contractual parameters are rewritten to allow for the innovative improvements recommended by Cognitive Project Management. These changes will take time to make their way into an industry that has a reputation for slow adoption of new ideas.

Meanwhile, the Projects go on – and their need for better approaches to Project Time Management has never been greater. If the ICS-Compendium was to be of any practical or real value in the short term, we reasoned, it would have to contain recommendations aimed at facilitating the best application of a currently practiced Dominant Project Management, not just argue for the different approaches of the far more promising Cognitive Project Management.

The ICS-Compendium Development Team grappled for quite some time with how best to satisfy two distinctly different informational needs. In the end we decided to present both perspectives throughout all volumes of the ICS-Compendium. The plan was that, any time we encountered a topic for which Dominant Project Management and Cognitive Project Management held different viewpoints, we would present both.

This decision, we felt, would allow the reader to have the best of both worlds. In the short term, thanks to the guidance provided by the ICS-Compendium, readers would be able to perform Project Time Management as well as one could, in concert with Dominant Project Management theory and recommended practices.

And yet, the reader could also acquire enough understanding of Cognitive Project Management to be able to lead its organization out of the quagmire of current thought toward a more robust and responsive Project Time Management model specifically intended for the Construction Industry in North America.
Consistent with the above decision, the ICS-Compendium contains two subordinate book series: the **Dominant Project Time Management Series** and the **Cognitive Project Time Management Series**. Each series contains four volumes, dealing in turn with one of the four **T.I.M.E. Framework** elements. [4]

For instance, this book, **CPM Mechanics**, deals with the Technological aspects of Dominant Project Time Management, which heavily relies upon the Critical Path Method. Its counterpart in the **Cognitive Project Time Management Series** is **Volume 5: Fundamentals of Momentum Science**, which is the Technological foundation for Cognitive Project Management.

The ten volumes of the ICS-Compendium are:

- **Dominant Project Time Management Series**
  - Volume 1: **CPM Mechanics** (T)
  - Volume 2: **Understanding Project Time Management** (I)
  - Volume 3: **Construction Planning and Scheduling** (M)
  - Volume 4: **Understanding Construction Project Management** (E)

- **Cognitive Project Time Management Series**
  - Volume 5: **Fundamentals of Momentum Science** (T)
  - Volume 6: **Introduction to Momentum Management** (I)
  - Volume 7: **Applied Momentology: Introduction to PAGUSYS** (M)
  - Volume 8: **Principles of Cognitive Project Management** (E)

- **ICS-Compendium Reference Materials**
  - Volume 9: **Project Time Management: Essential Desk Reference**
  - Volume 10: **Project Management Critical Thinking**

**1B: Exclusively for Construction Project Management**

As just explained, the entire ICS-Compendium was written for the exclusive benefit of the world of Construction Project Management. Actually, even within this world it is directed more specifically toward a subset of Constructors and Projects that fall in size somewhere between (exclusive of) small and mega.

Throughout the ICS-Compendium we use the term Construction Project Management very precisely. It is more specific than Project Management, yet more general than Construction Management.

---

Project Management refers to a systematic business approach to the achievement of a Project, regardless of industry. In this sense, Construction Project Management zeroes in on Project Management within the Construction Industry in particular.

Consider the ICS-Dictionary definitions for these three, similar-sounding terms:

- **Project Management:** The term Project Management has two distinct, yet interdependent meanings. As a discrete business entity, Project Management refers to a functional organization of specially-trained Project Team members who are tasked with achievement of a Project’s predefined Success Criteria. As an operational system, Project Management involves Technologies and Methodologies that are best suited to achieve Ideological objectives and address Ecological constraints. Project Management can be understood by its Operational Divisions as well as its Areas of Primary Attention.

- **Construction Management (CM):** A set of products and services provided to a Project on behalf of the Project’s Owner, and by a Constructor business entity that serves in a chiefly overseer role, ranging in authority from limited to extensive.

- **Construction Project Management:** The term Construction Project Management refers to Project Management, as performed in the Construction Industry. With this meaning, the word ‘construction’ is an adjective that identifies the industry in which Project Management is being performed. Accordingly, one might similarly refer to Software Project Management, Pharmaceuticals Project Management, and so forth.

For instance, where a CM has been employed by the Owner to act as its ‘eyes and ears’ or to provide advisory consulting to the Owner, the Owner might hold three separate contracts:

- The Owner-Designer Contract would employ a design firm to provide design services. The Design Professional(s) would design the facility.
- The Owner-Builder Contract would employ a construction company to perform as a General Contractor (GC) on the Project. The GC would construct the facility.
- The Owner-CM Contract would employ a construction company to monitor the performance of both the Design Professionals and the builders.

Even though the company providing Construction Management services is a qualified builder in its own right, its role on the Project (as CM) is mainly a supervisory one. That is, while acting in its CM role, the CM organization does not actually perform the work. Therefore, it does not manage the work in the ways that are generally understood to be the responsibility of a Project Manager for the construction company (GC).
This should clarify why we have chosen to use the compound expression Construction Project Management. By doing so, we are referring to the unique application of Project Management Methodologies and Technologies to Construction Projects.

10: The Three Meanings of the Expression, “CPM Scheduling”

This book is about the Critical Path Method (CPM), a Network-Based Technology used to model, schedule, and guide the performance of a Project Execution Strategy. Since this entire book is about the raw mechanics of the Critical Path Method, we will not stop to define it here. What few people appreciate, even those who have extensive experience in Project Management or Projects Controls, is that the expression, CPM Scheduling, has three distinctly different meanings.

The absence of such appreciation goes a long way to explain why there is so much confusion about what CPM Scheduling can or cannot do for the betterment of a Project, how CPM Scheduling works, who performs CPM Scheduling, when CPM Scheduling takes place in the Project Life Cycle, and so forth.

The three uses of the expression, CPM Scheduling, refer to increasing levels of Project Time Management that build upon one another.

- **Critical Path Method of Modeling:** At its most fundamental, the Critical Path Method is a computer-based Project Modeling Technology, complete with notational symbols and Arithmetic Calculations that can be used to simulate a Project Execution Strategy in notational form. Values derived from these mathematical processes include Earliest Dates and Latest Dates for each of the Activities in the Schedule. In turn, these Four Basic Calculated Dates yield a measure of Activity Criticality known as Total Float. For its part, Total Float is used to identify one or more Critical Paths that weave through the Schedule.

  For a mental comparison as to what we mean by the Critical Path Method of Modeling, think of the weather map on the evening news. It contains special

---

5 As discussed elsewhere in this and other ICS-Compendium volumes, the term Project Controls means different things to different people. Generally speaking, Project Controls refers to a set of administrative functions considered to be at the core of what Project Management is all about. While the precise list of component functions varies by organization, there is almost universal agreement that both Cost Control and Time Control are always part of Project Controls.

6 Technically, the Critical Path Method of Modeling could be performed manually, without the use of a computer. But given the size of the typical Construction Schedule, manual performance of CPM calculations would be costly, inefficient, and subject to error. For all practical purposes, the Critical Path Method of Modeling is rarely performed manually.
symbols and various key statistics (temperature, precipitation, wind velocity and direction, and on) that the fast-speaking meteorologist helps us to decipher.

Here is the ICS-Dictionary definition of this term:

**Critical Path Method of Modeling:** One of three possible meanings of the expression, CPM Scheduling. This interpretation refers to a modeling technology, complete with notational symbols and Arithmetic Calculations, that can be used to simulate a Project Execution Strategy in notational form. This is CPM at its most basic; a flowcharting system with inherent computational values, formulas, and processes that can be applied manually or with the aid of a computer.

**Critical Path Method of Scheduling:** But a computer model of a Project Execution Strategy offers little value to the Project unless it is ultimately put to some practical use. This is what we mean by the Critical Path Method of Scheduling, which refers to organizational processes that apply the fundamentals of the Modeling technology to design, develop, and maintain Project Schedules. It is these processes that constitute what many refer to as “Planning and Scheduling.”

Back to our meteorology comparison, the equivalent to the Critical Path Method of Scheduling would be the various practices performed by the meteorologist throughout the day leading up to and culminating in the live broadcast. Using the computer model, she would apply her special training to pre-show analysis of raw data, thus drawing conclusions about current and imminent weather conditions, which she would then explain during the broadcast.

In application, after its initial creation and acceptance as a credible and useful representation of Project Execution Strategy, a Project Schedule is routinely (most commonly, monthly) updated to reflect the latest changes in Project conditions, including achieved progress, Work Scope additions or deletions, changes in risk assessments, and so forth.

One popular report associated with the monthly update is a **Total Float Report**, which lists remaining work to be performed, sorted by Total Float (in order of decreasing Activity Criticality). The recalculation of key CPM variables (Earliest Dates, Latest Dates, Total Float) is part of the Critical Path Method of Modeling, while performing the monthly update or revising the Schedule to include new Scope, are aspects of the Critical Path Method of Scheduling.
Here is the ICS-Dictionary definition of this term:

**Critical Path Method of Scheduling**: One of three possible meanings of the expression, CPM Scheduling. Various organizational processes that apply the fundamentals of the Critical Path Method of Modeling to the design, development, and maintenance of Project Schedules. It is these processes that constitute what Dominant Project Management refers to as “CPM Planning and Scheduling.”

**Critical Path Method of Managing**: But there is more to Project Time Management than the mere development and maintenance of Project Schedules. The Critical Path Method of Managing refers to the use of temporal information produced through the Critical Path Method of Scheduling in order to inform Project Management as it struggles to take important decisions on a daily basis.

Back to the **Total Float Report** example (that was produced as a function of the Critical Path Method of Scheduling), the Project Manager may, for instance, choose to include the **Total Float Report** as part of the agenda for the **Weekly Coordination Meeting** with the Subcontractors, or in its **Monthly Progress Review Meeting** with the Owner. Use of a Scheduling report (generated through the Critical Path Method of Scheduling) represents an example of the Critical Path Method of Managing.

Here is the ICS-Dictionary definition of this term:

**Critical Path Method of Managing**: One of three possible meanings of the expression, CPM Scheduling. Critical Path Method of Managing refers to the use of temporal information produced through the Critical Path Method of Scheduling in order to inform Project Management as it struggles to take important decisions on a daily basis.

The end game for most Project Management stakeholders is the Critical Path Method of Managing. To be clear, however, the Critical Path Method of Managing utilizes the principles and processes of the Critical Path Method of Scheduling, which in turn is based on the core mechanics of the Critical Path Method of Modeling, which is the narrow subject of this book.

Adopting the above distinctions, our guess is that your interest doesn’t stop at the Critical Path Method of Modeling. Why would it? Since the Critical Path Method is rarely applied outside of a Project Management context, it is probably safe for us to assume that your interest in the topic aligns with how the Critical Path Method is, or can be, used to help manage Projects.
This book will only take you part of the way – but it is, in our humble opinion, the most important part. The long journey to fully appreciating why CPM is a favorite arrow in the Project Manager’s quiver begins with an understanding of how CPM works at its core: that is, as the Critical Path Method of Modeling.

10: The Book’s Organization and Structure

This book adopts a very logical approach, one that should strike the reader as extremely intuitive. Picture a check mark, like the one shown in Figure F0102. Our approach to the subject matter of the Critical Path Method will start at the top, moving quickly from general (down) into the specifics, and then slowly back (up) to general again. In other words, we will begin with a 30,000 foot view – briefly considering the universe in which Projects are managed, and then immediately orient our focus to the role of the Critical Path Method as one of many, albeit the most popular, tools for managing Construction Projects.

Hold your hard hat — because we will plummet from the dizzying heights of all Projects down into the esoteric subtleties of CPM’s most basic building block, the Schedule Activity [7] — all within the opening paragraphs of Chapter Two.

Then, from that starting place at the base of the Great CPM Mountain we will begin our slow, methodical ascent back to the initial vantage point, high atop the Project Management vista, from which perspective we will write the book’s closing commentaries and conclusions.

☐ The first chapter of course is this one, Chapter One: CPM, A Modeling Method, which provides a brief introduction to the overarching topics that form the foundation for understanding the Critical Path Method of Modeling Project Execution Strategy.

☐ After laying out a simple framework for the discussions to be had throughout this book, Chapter Two: Static Project Time Management Tools will begin our climb by telling us about the simplest tools for Project Coordination: Listings, Timetables, and Bar Charts. It is here in Chapter Two that we will first encounter the primary elements of an Activity: its Activity Identifier, Activity Description, and Activity Duration.

7 While the complete term is Schedule Activity, throughout this book and the ICS-Compendium we will mainly use its abbreviated name, Activity.
CHAPTER THREE: DYNAMIC PROJECT TIME MANAGEMENT TOOLS introduces the concept of a Network-Based Schedule, one that portrays a Schedule as an interwoven mass of Activities and their interlinking Progressive Relationships, called Logic. This mass of Activities is called a Logic Diagram. The chapter provides our first glimpse of Performance Restrictions, the feature that links Activities together and gives this category of Schedules its dynamic attribute.

CHAPTER FOUR: HOW TO DRAW CPM LOGIC uses the foundational elements of the Critical Path Method of Modeling to guide us as we draw our very first CPM Logic. As it does so, the chapter's discussions distinguish between Gregorian Dates and Ordinal Dates, Provided and Calculated Schedule Elements, and Mandatory versus Optional CPM Schedule Elements. One of the most important discussions in this chapter has to do with the concept of Point-of-Day Perspective.

CHAPTER FIVE: CALCULATING PRIMARY CPM DATES gives meaning to the word Dynamic, as it shows us precisely how to take a CPM Logic Diagram, add Calculated Dates, and thus create our first Network-Based CPM Schedule. We learn about the arithmetic behind the three major Performance Restrictions and how they are applied in Forward Pass and Backward Pass Date Calculations. Finally, we discover how Earliest Dates and Latest Dates are determined.

In CHAPTER SIX: ALL ABOUT TOTAL FLOAT we are at last taught how the infamous Total Float is calculated. We are instructed on how Performance Restrictions (and their Restriction Delays) can often create conditions whereby a single Activity may have differing Start Float and Finish Float values.

Following on the heels of the Total Float discussion, CHAPTER SEVEN: ALL ABOUT CRITICAL PATH presents a comprehensive dialogue about what constitutes a Schedule’s Critical Path, and informs us that most Schedules have more than one Critical Path at any given time.

CHAPTER EIGHT: FREE FLOAT AND SYMBIOTIC NETWORKS addresses two essentially unrelated matters. The chapter's first part teaches us all about Free Float and how it differs from Total Float. The chapter closes with a return to an earlier concept: that all of the Activities in a Network-tied Schedule are related to one another, not just the ones that share a common Logic Tie.

Beyond the fundamental elements found in every CPM Schedule (Activities, Activity Durations, and Logic Ties), there are a number of SUPPLEMENTAL SCHEDULE ELEMENTS, which is also the title of CHAPTER NINE. Among these are Date Constraints, Work Performance Calendars, Activity Codes (including the infamous Work Breakdown Structure), and a variety of Scheduling Software Settings.
Beyond the Supplement Schedule Elements of Chapter Nine that one might expect to find in every CPM Schedule, there are other Exploitive Schedule Elements that may or may not appear in a CPM Schedule. Chapter Ten discusses these Exploitive Schedule Elements, the three most prominent examples of which are Activity Cost-Loading, Activity Resource-Loading, and the Work Breakdown Structure.

It has been said that Project Scheduling is as much an art as it is a science. While the first ten chapters deal with CPM Scheduling at its most mechanical, in Chapter Eleven, we step away from the Critical Path Method of Modeling, and consider instead the manner of our procedural and mechanical actions with an eye toward Putting Reality into the Schedule, the chapter’s title.

A Project is an ongoing group effort that spans months and sometimes years. The Project Schedule is a management tool designed to simulate Project Execution Strategy – both planned and performed. There is a fascinating correlation between the momentary credibility of the Project Schedule and where the Project stands at any point along the Project Life Cycle.

Chapter Twelve introduces us to a concept conceived at ICS-Research, called the Schedule Data Credibility Profile, which helps us understand, measure, and depicts the relative reliability of a Schedule at any given time. It explains the G.R.A.S.P. concept, which provides a structured process for measuring and reporting Schedule Credibility.

It is not enough to simply create a Project Execution Model, it must be maintained throughout the Project Life Cycle. This is the subject of Chapter Thirteen: CPM Calculations During Schedule Statusing. We will examine an assortment of variables and calculations encountered in the course of a typical Schedule Update, such as Percent Complete, Remaining Duration, Data Dates, and Schedule Outcome Projections.

Despite this chapter's treatment of aspects of Schedule Maintenance, the Scope of this book does not extend to the Critical Path Method of Scheduling. To repeat a point made several times throughout the ICS-Compendium, the expression Critical Path Method actually has three different meanings: as a Modeling Method, as a Scheduling Method, and as a Project Management Method. Since this book is devoted to explaining the Critical Path Method of Modeling, Chapter Twelve is limited to discussing how, from a mechanical perspective, a Schedule is kept current and relevant – so that it can continue to reliably model the remainder of the uncompleted Project.
Elsewhere in the ICS-Compendium we will discuss this same topic, but from the perspective of the Critical Path Method of Scheduling. In this book, we will simply establish the underlying mechanical principles that will come into play later, including the need for cyclical adjustments, the concept of a Data Date, and the arithmetic processes that are involved in Activity Duration, Calculated Date, and Activity Path adjustments.

As just one example, in this chapter we discuss the Remaining Duration of an Activity, a value that plays no role during Schedule Design or Schedule Development or within a Baseline Schedule. It is only after the Project has commenced, and after the first Schedule Update has been performed, that the Remaining Duration assumes importance. By the end of Chapter Thirteen we will have sufficiently covered all aspects of the Critical Path Method of Modeling.

In Chapter Fourteen we step away from the Critical Path Method and instead consider various other Project Time Management Alternatives. Specifically we take a quick look at Arrow Diagramming (the original CPM Scheduling format), PERT (a contemporary to early CPM), Line of Balance/Linear Scheduling), and Building Information Modeling (BIM).

The book concludes with Chapter Fifteen, Interpreting the Project Execution Model. Only after completing the previous fourteen chapters is the reader able to fully appreciate the significance of these closing admonitions. Here, we are reminded of what the Critical Path Method of Modeling is, and is not—what it can, and cannot, do.

1E: The Book’s Information Transfer Strategy

Every book, no matter what its purpose, transfers information – even fictional novels that are written for entertainment. Throughout the course of writing CPM Mechanics, ever-present in our minds were two key points of focus: who the intended readers of this book would be and what information we felt that they would want (or need) to have.

Who: The first item was easy. This book was originally requested by the ICS-Institute for Construction Scheduling Studies and planned for their use as a primary textbook for its CAST Program. CAST is an educational program designed to “create the next generation of entry-level Construction Schedulers,” and stands for Construction Apprentice Scheduler Training.

Consistent with this objective, a basic assumption underlying the design of all CAST program elements, including this book, is that students entering CAST might not possess any prior knowledge or experience in either Project
Management, construction, or as matters to this book – the Critical Path Method (CPM) in any of its uses: Modeling, Scheduling, or the management of Projects. As a result, this book assumes that the reader will be starting from scratch, and so must this book.

- What: The second item followed neatly from the first. The reader would need to be educated on terminology and concepts, practices and processes, rationales and clarifications. We decided to make a list of every piece of information that we believed an entry-level Construction Scheduler ought to know and understand when he or she reports to work for the first time. Once we sat back and stared at the sixteen-page list that we had just amassed, it became apparent that, if we could somehow divide the subject matter into a few major groupings of information such that each new category of thought built upon the previous one, such an approach might be quite helpful to the reader.

1E1: The Book’s Approach to Technical Subject Matter

That is how the ICS-Compendium Development Team came to desire that the book follow a systematic, orderly morphing of content, starting with the simplest elements and ending with the most complex.

Specifically, we would start with Static Schedules (e.g., Listings, Timetables, and Bar Charts) that just “lie there and do nothing.” Eventually we would progress to Dynamic Schedules, such as full-blown CPM Networks — the ones that “live and breathe.”

We decided to use the increasing complexity of the different Schedule Forms as a structure for introducing key terminology and concepts. In other words, each time a new Schedule Form is introduced, we would stop to discuss any new elements which that particular Schedule Form contained, elements not found in simpler forms.

To clarify, consider each of these increasingly more complex Schedule Forms, and the Schedule Elements (concepts and components) introduced by way of discussion of each one:

- Listing Form
  - The concept of an Activity representing a Work Scope Element
  - Distinguishing an Activity from Action
  - The concept of Work Sequencing

- Timetable Form
  - Activity Identifier
  - Concept of Time
  - Concept of Start-of-Day Perspective and End-of-Day Perspective
Bar Chart Form
- Concept of Activity Duration
- Workdays versus Non-Workdays
- Continuous versus Impacted Activity Durations
- Concept of Continuous Crew Days
- Concepts of Activity Sequencing and Activity Dependency

Network Form
- Communal, Symbiotic, and Progressive Relationships
- CPM’s Four Performance Restrictions (Dependency Types)

You get the idea. The above outline is not exhaustive; there is no need to outline the entire book. We just want you to understand that as you read about a particular Schedule Form, we will take that opportunity to introduce Schedule Elements that first appear at that (complexity) level of Schedule Form.

1E2: The Book’s Unique Formatting Symbolism

The Critical Path Method is a technical subject and it comes with its own terminology, concepts, and processes. To make your reading experience a bit easier, we have adopted a somewhat unconventional use of capitalization and special fonts, going against the recommendation of professional publishers and editors, who insist that doing so might “clutter up the page.”

The use of capitalization, special fonts and selective hyphenation, intended to make it easier to decipher technical sentences, evolved from a pet peeve about technical books that we have read. Perhaps you have had the same experience as us, where procedural or technical terms mingle with common words in the same sentence, such that it is difficult, if not sometimes impossible, to understand what is being said.

Compare the following two versions of the same sentence. Which is easier to understand?

- “Note that with the start to start dependency between activity a and activity b, the predecessor activity’s late start is actually later than its early start, but still earlier than the successor’s early start.”

- “Note that with the Start-to-Start Dependency between Activity A and Activity B, the predecessor Activity’s Late Start is actually later than its Early Start, but still earlier than the successor’s Early Start.”

Accordingly, throughout this book (and the entire ICS-Compendium, as well) we have adopted the following formatting conventions:
1 CPM, A Modeling Method

☐ **Arcane Term:** Any time an arcane term or concept is used it will be capitalized, such as, “It is important to maintain the credibility of the Activity Duration.”

☐ **Proper Name:** Any proper name or title has its own font, such as the title of this book, CPM MECHANICS. Included in this group would be specific Activity Descriptions, such as INSTALL BOILER.

☐ **Data Values:** Data values will appear in a green font. This font will be applied to Figure Numbers, Durations, Restriction Delays, and other calculated or given data values. For example, “Activity A enjoys Total Float of \(TF +12\), even though it starts on Day 122 and ends on Day 132.”

☐ **Definitions and Quotes:** We want to make sure that you can find important definitions, quotes, and citations. These are indicated by a Talking Head, and will be shaded in pink, such as this:

![Talking Head]

* Resident in a CPM Network Diagram, an Activity is an artificial representation of its real life counterpart, a Project Execution Action.

☐ **Special Hints:** Every so often, we may think of some special hint or tip that we believe would enhance your understanding of the subject matter. We will be sure to point it out, with the Pointing Finger symbol and soft green background:

![Pointing Finger]

* This is an example of a special hint.

☐ **Figure Numbering:** This book contains over 225 figures to help you really understand the subject matter. Figure numbering is quite simple. For example, Figure F0412.

  - **Position 1:** The letter \(F\) refers to a figure.
  - **Positions 2 and 3:** Indicate the chapter number, such as 04.
  - **Positions 4 and 5:** The final two positions, such as 12, provide a unique sequential number, incremented by twos, through the length of a chapter. This unique number starts all over again with 02 for each new chapter.

1E3: The ICS-Dictionary

The ICS-Dictionary is fast becoming a globally-recognized collection of reliable definitions for key terminology relevant to Project Time Management specifically, and Project Management more broadly. Purchase of any volume of the ICS-Compendium includes a free copy of the ICS-Dictionary.
The ICS-Dictionary contains hundreds of carefully worded definitions. All terms are cross-referenced with related terms. Best of all, each of the ICS-Compendium volumes use the same terms, and assume the same meanings of those terms. This, in itself, constitutes a major improvement over the current literature in Dominant Project Management, where each separate author or authoritative body uses its own set of terms and choice of interpretations.

As you read through this, or any other, volume of the ICS-Compendium, you may wish to refer to the ICS-Dictionary to get a better understanding of the term and its relationship to related technical terms or concepts.

So let’s get started, shall we? We hope you like what we have put together for you.

Best regards,

Murray Woolf
CHAPTER TWO

2A Introducing the Critical Path Method of Modeling
   2A1 CPM from the Project Manager’s Perspective
      2A1a Distinguishing Projects by Industry
      2A1b Distinguishing Projects by Participant Perspective
      2A1c Projects Don’t Just Happen; They are Managed
      2A1d Project Time Management: Project Management’s Heart
   2A2 CPM from the Project Facilitator’s Perspective
   
2B Static Time Management Tools
   2B1 What is a Schedule?
   2B2 Listings
      2B2a An Activity Represents a Work Scope Element
      2B2b Distinguishing an Activity from an Action
      2B2c Every Activity has an Activity Description
      2B2d The Sequencing of Activities
   2B3 Timetables
      2B3a Activity Identifier
      2B3b Concept of Time
      2B3b-1 We Don’t Manage Time
      2B3b-2 Time and Money are Not Equivalent
      2B3b-3 Project Execution’s Use of Time
      2B3c Start-of-Day and End-of-Day Perspectives
   2B4 Bar Charts
      2B4a Concept of a Duration
      2B4b Workdays and Non-Workdays
      2B4c Continuous and Impacted Durations
      2B4d Concept of Continuous Crew Days
      2B4e Activity Sequencing Not the Same as Activity Dependency
   
2C Project Execution Commitments
   2C1 The Difference between Events and Milestones
   2C2 Points of Navigational Significance (PINS)
This book is about the Critical Path Method of Modeling. Said differently, it is not about the Critical Path Method of Scheduling or the Critical Path Method of Managing. Those two other ways of using the expression, CPM Scheduling, are treated in detail in other volumes of the ICS-Compendium. In this book, we are concerned with the most rudimentary mechanics of the underlying Critical Path Method, of how to depict any series of interrelated actions, human or otherwise, by using a set of carefully designed and applied symbols, arithmetic formulas, and operational steps.

2A: Introducing the Critical Path Method of Modeling

Obviously this will be a fairly technical book and it would be quite easy for us to get lost in a forest of details. But before we worry about maintaining our perspective, we must first and foremost realize that there are actually two different perspectives through which we will (or should) comprehend just about everything we encounter throughout the ICS-Compendium.

One interpretation is from the perspective of those who consume Project Time Management products and services. We would know them by labels such as Project Manager, Superintendent, Project Team, Project Executors, and the like.

The other perspective is the one held by those who provide those same Project Time Management products and services. The labels Planner, Scheduler, and Project Facilitator immediately come to mind.

2A1: CPM from the Project Manager’s Perspective

If we were to climb into the basket of a hot air balloon and drift skyward in an effort to
gain a 30,000 foot view of the landscape surrounding Project Time Management, what would we see? For starters, the very edge of the horizon would define the outer boundaries of the Business World itself; the basis of our economy, the source of all purchasable needs, and the place to which we venture when we head off to work in the morning. So let us use this outer boundary as our reference point and see where Project Time Management fits within it.

2A1a: Distinguishing Projects by Industry

We begin the narrowing of our focus by noting that, of all of the industries in the Business World, the Construction Industry is considered to be one of the (if not the) largest. To be sure, few would argue that it is one of the most important to the economy — both globally and locally.

It is virtually impossible to travel more than a few feet, or even live through a single day, without encountering the products of the Construction Industry. Included under this single category of business enterprise is so much of what makes our lives the grand and limitless ventures that they are.

Where would we be, and how happily or well might we live, if we did not have any (or even some) of the following?

- Let’s start with Buildings. Included in this group are hotels, offices, restaurants, retail shops, skyscrapers, houses of worship, commercial buildings (factories, warehouses, distribution centers, gas stations, fitness centers, spas), and residential edifices (apartments, condominiums, single-family homes, and even planned communities and subdivisions);

- Then there are Facilities, clusters of buildings that spread across developed land, as in a campus. These would include correctional facilities (low, medium, or high-security prisons), healthcare facilities (hospitals, walk-in clinics), primary schools (elementary, middle, and high schools), and secondary schools (college and university campuses);

- There are Public Gathering complexes, such as libraries, museums, performance halls, recreational centers (bowling alleys, miniature and 18-hole golf courses, paint ball centers, arcades), sports complexes (arenas, stadiums, ball fields), theme parks, and entertainment venues (stage and movie theaters, dance halls, clubhouses, lodges);
Supplying all of the above buildings, facilities, and venues are a host of support systems, often referred to as societal **Infrastructure**. Two categories of infrastructure come to mind: utilities and transportation.

- Under the **Utilities** heading would be included pipelines, power plants, sewage and water treatment plants, power and communication distribution towers and lines; storm drainage systems, flood control ponds and dams, and so forth.

- **Transportation** would encompass bus and train terminals, light rail systems, high-speed rail, airports, roads, bridges, tunnels and even shipping and marine ports, harbors, and channels.

Then there is **Industrial** construction, which is a special category of Commercial Projects, unique because of particular and extensive design and construction considerations. Included under this heading would be automotive plants (both manufacturing and assembly), general manufacturing, off-shore oil rigs, petrochemical plants, process plants, pulp and paper facilities);

Finally there is a catch-all group which we will call **Specialty**. These would include yet other types of Construction Projects, such as aerospace (commercial and fighter aircraft), clean rooms (e.g., microchips, pharmaceuticals, hospital surgery/operating rooms) and other dirt-free environments, environmental, high tech, military and defense, research, ship-building, mining equipment, and even interplanetary (space station).

**2A1b: Distinguishing Projects by Participant Perspective**

We have already begun to use the word Project, which is a structured and organized way in which the Construction Industry accomplishes virtually all that it does. According to MaxWideman.com, a frequently cited definition of the term, Project, is:

> **Project**: A temporary endeavor undertaken to create a unique product, service, or result.

In our opinion, this definition is a bit too broad. For instance, according to this definition, buttering toast would qualify as a Project. We think that a Project — at least one warranting the application of a formal Project Management system — must, at a minimum, require the performance of more than just one or two individuals. Technically, a single person could be faced with a Project, (for instance, researching one's genealogy to see if there is any connection to the American Revolution — or perhaps cleaning out the attic). Yet when we consider Construction Projects, these would surely require more than a few
individuals, working together in some concerted manner.

The ambiguity as to what constitutes a Project extends beyond the number of players involved. It also extends to what Cognitive Project Management refers to as Participant Perspective. For instance, depending on whom you ask among the primary Project Participants on a single Construction Project, their definition of a Project is likely to be different from the others.

Consider a Project involving the renovation of a hospital. The Design Professionals, such as an Architect and Engineer, would likely refer to the design phase of the Project as constituting “the Project.” By contrast, a Contractor would see the construction phase as “the Project.” But the Project's Owner, who would surely be concerned with both the design and construction phases, might also include additional Scope Elements that take place before, during, and after the collective Design/Construction phases.

2A1c: Projects Don’t Just Happen; They are Managed

Projects are managed. That is, hundreds or thousands of actions performed by hundreds or thousands of human participants utilizing hundreds or thousands of pieces of equipment or materials are precisely orchestrated by a specially-trained group of coordinators, collectively referred to as Project Management.

Project Management as a professional discipline can be divided into major areas of responsibility, concern, and focus. Again, the categories of Project Management differ, depending on which “authority” is speaking (or writing). For instance:


- By contrast, the Association for Project Management (APM) subdivides Project Management into two major groupings, Planning and Execution. Under Planning, they include Stakeholder Management, Value Management, Risk Management, Quality Management, and Health, Safety & Environment Management. They consider Execution to include Scope Management, Scheduling, Resource Management, Budgeting and Cost Management, Change Control Management, Earned Value Management, Information Management and Reporting, and Issue Management.

- A third global leader in Project Management, the International Project Management Association (IPMA) contends that Project Management when properly performed

1 For better understanding of what PMI espouses, visit its website at www.pmi.org.
requires 46 competence elements\(^2\) that are organized under three broad headings:

- **Behavioral Competence** includes Leadership; Engagement & Motivation; Self-Control; Assertiveness; Relaxation; Openness; Creativity; Results Orientation; Efficiency; Consultation; Negotiation; Conflict & Crisis; Reliability; Values Appreciation; and Ethics.

- **Technical Competence** includes Project Management Success; Interested Parties; Project Requirements & Objectives; Risk & Opportunity; Quality; Project Organization; Teamwork; Problem Resolution; Project Structures; Scope and Deliverables; Time & Project Phases; Resources; Cost & Finance; Procurement & Contract; Changes; Control & Reports; Information & Documentation; Communication; Start-Up; and Close-Out.

- **Contextual Competence** includes Project Orientation; Programme Orientation; Portfolio Orientation; Project, Programme and Portfolio Orientation; Permanent Organization; Business; Systems, Products & Technology; Personnel Management; Health, Security, Safety, & Environment; Finance; and Legal.

2A1d: **Project Time Management: Project Management's Heart**

Among the above areas of Project Management attention, none is more important or central to the success of any Project (no matter how one defines “success”) than Project Time Management. We say this because without the effective use of Time, it would be nearly impossible to meet the performance expectations of any of the other Project Management areas of concern – be they Scope, Cost, Risk, Safety, Procurement, Human Resources, Changes, Communications and so forth.

Without over-simplifying the subject, Project Time Management comprises practices that span the Project Life Cycle. In Dominant Project Management circles, these practices are often described with labels such as Project Planning, Project Scheduling, and Project Control.\(^3\)

The irrefutable thread that seems to link all three phases of Project Time Management processes is a common set of Project Time Management tools. These tools can be divided into Project Execution Plans and Project Execution Schedules.

- **Project Execution Plans:** These would include Feasibility Plans, Optimization Plans, Consensus Plans, and Strategic Plans.

---

\(^2\) Visit the IPMA's website at http://ipma.ch to learn more about their Project Management philosophy.

\(^3\) Alternatively, books on Construction Planning and Scheduling equally often refer to the third phase as “Schedule Control.”
Project Execution Schedules: These tend to evolve over Time, correlating with the natural progression of the Construction Project delivery process. Thus, a Design Schedule might lead to Procurement Schedule or Sourcing Schedule, which in turn might grow into a Bid/Proposal Schedule. Eventually, though, the Project Team develops a comprehensive Project Execution Schedule.

An undeniable morphing effect exists between Project Execution Plans and Project Execution Schedules, the latter evolving from the former. Among Scheduling practitioners there is widespread and passionate debate as to where Planning leaves off and Scheduling picks up. One thing is for sure: both practices employ the same modeling Technologies; by far the most common among these being the Critical Path Method of Modeling (CPM).

At the risk of mild irritation, let us repeat one last time that the term, CPM Scheduling, actually has three different connotations in Project Management circles. To be sure, the end game for most Project Management Team members is the Critical Path Method of Managing. However, the Critical Path Method of Managing utilizes the principles and processes of the Critical Path Method of Scheduling, which in turn is based on the core mechanics of the Critical Path Method of Modeling, which is of course the subject of this book.

2A2: CPM from the Project Facilitator’s Perspective

Before we go any further, let us clarify a few important labels. Currently, in Dominant Project Management literature, those charged with creating Project Schedules are referred to rather interchangeably as Planners or Schedulers, and their discipline is similarly known as Planning and Scheduling.

From the perspective of Cognitive Project Management, there is a significant difference between Project Time Management and the old-school discipline that it replaces, Planning and Scheduling. This perspective stands in stark contrast to the view of Dominant Project Management, whose literature mostly fails to note any difference whatsoever.

By contrast, Cognitive Project Management theory adopts the following understanding, as explained and upheld throughout the volumes of the ICS-Compendium:

Project Time Management: The central component of effective Project Management. A basic tenet of Cognitive Project Management is that effective Project Execution simply cannot be achieved without the correspondingly effective use of Time by the Project Execution Team. Project Time Management requires the development, maintenance, and use of products and services especially designed for this purpose.
Cognitive Project Management organizes the Project Management Team into four operational groups, arranged into a diamond shape with four vertices.

Each vertex represents the focal point of a separate Project Management Domain. We call the resultant image, as shown in Figure F0202, the Cognitive F.A.C.E. Diamond, a mnemonic reference to the names of the Domains.

- Those who develop and maintain Project Time Management products and services are known in Cognitive Project Management jargon as Project Facilitators.

- Those who provide capacity for Project Execution, by way of supplying necessary Resources and more, are called Project Administrators.

- At the top of the Cognitive F.A.C.E. Diamond is the Project Coordinator and his/her personal staff, who oversee all others on the Project Management Team, and liaise with Project-external customers.

- Those responsible for performing the Work of the Project, and who do so by utilizing Project Time Management products and services in the course of performing their duties, are known as Project Executors.

Delving a bit further into Cognitive Project Management's vision of an effective Construction Project Management organizational structure, three primary functional labels would be found under the umbrella title of Project Facilitator:

- Project Facilitation Planner, which replaces the current Project Planner label.

- Project Facilitation Scheduler, which replaces the current Project Scheduler label.

- Project Facilitation Analyst, for which there is no counterpart in Dominant Project Management.
Back to our earlier point, Project Time Management practitioners, immersed as they typically are in the nuances of their craft, nevertheless understand that there are many different options when it comes to effectively managing the use of Time during Project Execution. What they know, and you should as well, is that there are other options for Project Time Management Tools in addition to Critical Path Method Plans and Schedules. Some of the well-known of these alternatives are discussed in **CHAPTER FOURTEEN: OTHER TIME MANAGEMENT ALTERNATIVES**.

When it comes to the creation and use of Project Plans and Schedules, Project Facilitators are also well aware that the Critical Path Method is a choice, and not a mandate. To be sure, CPM is the most popular method for producing and utilizing Project Schedules, and is the prime example of a complex Scheduling Technology, but it is by no means the only Technology.

Planning and Scheduling tools can range from the very simple (*static* tools) to the most complex (*dynamic* tools). In the next sections, we will discuss the range of Planning and Scheduling Technologies most commonly called upon to manage the effective use of Time on Construction Projects.

### 2B: Static Project Time Management Tools

We begin our discussion by considering the simplest forms of Plans and Schedules, ones that we call Static Project Time Management Tools. Three popular types of Plans and Schedules fall under this heading: Listings, Timetables, and Bar Charts.

The road to understanding the Critical Path Method — from the Project Facilitator’s perspective – begins with an acknowledgement of the obvious: Project Facilitators work with Schedules. And that means that we can go no further until we define the word, Schedule.

### 2B1: What is a Schedule?

So, what is a Schedule? Dictionary.com provides several general definitions which, taken in combination, confirm that the word can have more than one meaning, at least in the common vernacular: [4]

- **Schedule:** A printed or written list of items in tabular form.
- **Schedule:** A list of times of departures and arrivals; a timetable; tabular listings.
- **Schedule:** A list of tasks to be performed, especially within a set period.

---

4 Throughout this book we cite definitions from Dictionary.com, which you can locate at www.dictionary.com.
While each definition is slightly different, we may note a few concepts that they seem to share in common. For instance, consider these observations:

- A Schedule presents a displayed Listing of some sort.

- While the majority of the above definitions refer to the concept of an Element of Time somewhere in their descriptions, two definitions suggest instead that a Schedule can be as simple as a basic Listing of things to do, without any reference to Time.

- Four of the six definitions fail to express any concern about the sequence of Events. Yet the last two definitions do include Sequencing as an element of a Schedule (i.e., “ordered list,” “specifying the order”).

- Interestingly (as well as somewhat semantically) we note that only one of the definitions openly speaks of the idea of Action (as in “tasks”) – although the expression “things are planned to occur” in another definition seems to hint at Action! Of course, the Planned “thing” could also be an Event or a Milestone, something that Cognitive Project Management calls a Project Execution Commitment.

From the last bullet, we can conclude that Listing Schedules can be divided into two Types, according to Cognitive Project Management.

- There are Commitment Schedules, which depict the essence of a Project in terms of the achievement of what Cognitive Project Management calls Project Execution Commitments. An example of a Commitment Schedule is a bus or train schedule that shows the arrival and departure times at different locations along various routes across town; the arrival and departure times being the Commitments.
The other Schedule Type is an Action Schedule, which portrays a Project principally in terms of Actions to be taken. A Weekend To Do List is an example of an Action Schedule, such as the one shown in Figure F0204. Suffice it to say that since the primary use of a Construction Schedule is to coordinate the efforts of hundreds or thousands of Project Participants, this book will mainly concentrate on Action Schedules (and not as much about Commitment Schedules).[5]

2B2: Listings

The above observations suggest a hierarchy of Static Schedule[6] forms, ranging from quite simple to moderately complex.

- A Performance Listing is just that -- a list of Actions that may, must, or will be performed … or of Commitments that may, must, or will occur. Figure F0204 is an example of a simple Performance Listing.
- Slightly more involved, a Timetable is a Performance Listing of Activities that additionally shows Times associated with the occurrence of those Activities.
- Still more elaborate, a Bar Chart is a graphical representation of a Timetable, wherein an Activity is depicted as a horizontal bar and drawn to scale against the backdrop of a Timeline bearing Unit of Time demarcations (e.g., decades, years, months, weeks, days, hours, minutes).

2B2a: An Activity Represents a Work Scope Element

Before we go any further we wish to remind you that, as explained in Chapter One, we will be using the increasing complexity of the Schedule Forms as a framework for presenting new terminology and concepts. We begin that practice now with the introduction of some basic elements of a Performance Listing:

- Activity versus Action
- Scope versus Work Scope
- Activity Description

2B2a-i: Activity versus Action

Without debate, the main element of every Performance Listing is this thing called an Activity; the Project Schedule Element to which statements about Time are associated (according to four of those six definitions). An Activity is the most basic, fundamental

---

[5] This is not to say that an Action Schedule cannot, or routinely does not, contain the tracking of key Project Execution Commitments (Milestones or Events). But, in the majority, the Project Schedule is comprised of Activities because Schedules primarily exist to coordinate Action.

[6] We have not defined the term Static Schedule yet; we will do so shortly.
building block of each and every Project Schedule, regardless of its type or format.

Cognitive Project Management considers the word, Task, to be an improper substitute for the term Activity. [7]

An Activity is a Schedule Element, and it represents a corresponding real-life Action out on the Project itself. Out on the job, such Action is, in reality, the Project Team’s effort to accomplish some portion of the Project's overall Work Scope, as defined by the Project Owner (usually in the form of a performance contract).

In theory, one could construct a Project Schedule with only one Activity – with an all-encompassing description such as PERFORM PROJECT. But that wouldn’t be a very helpful Schedule, would it? It would however be a legitimate Schedule, just not one that anyone would find much value in using. And so, it is our safe assumption that a Project Schedule will have multiple Activities, and therefore each Activity would separately represent some portion of the Project's overall Scope.

2B2a-ii: Scope versus Work Scope

What do we mean by the word Scope – and, in particular, Work Scope? Let’s go back to Dictionary.com where we learn that Scope has two primary interpretations: one related to the subject of observation, while the other relates to our ability, as humans, to observe.

- **Scope**: The range of one's perceptions, thoughts, or actions.
- **Scope**: Breadth or opportunity to function.
- **Scope**: The area covered by a given activity or subject.
- **Scope**: Extent or range of view, outlook, application, operation, effectiveness, etc.

From these definitions, we get the sense that Scope refers to whatever it is that we may be concentrating on, as we perform our portion of the Project. Our favorite definition is the last one, which comes closest to what we mean when we discuss the Work Scope of a Project – and, in particular, the Scope of a Project Schedule. It is the “extent or range of view, outlook, application, operation, [or] effectiveness” that a particular Schedule Element – such as an Activity has.

Generally speaking, the word Scope refers to the breadth and depth of treatment of...
“a given Activity or subject.” Work Scope, therefore, is a more specific application of this definition, and explains “treatment” as that which must be performed or worked. In Construction, “that which must be performed or worked” is detailed in a Contract.

From all of the above we can conclude that an Activity represents a Work Scope Element in the same way that the Project in its entirety constitutes the full Scope of Work to be performed. And since a typical Project Schedule has more than one Activity, each Activity must therefore represent some portion of the full Work Scope. The Construction Contract (between Owner and Constructor) defines the Work Scope. The Project Schedule defines the Scope of each separate Activity.

2B2a-iii: Activity Description

Actions out on the Project exist, whether or not they have names, or even if we notice them. The same holds true for Activities in the Schedule. Each Activity represents a portion of Work Scope, even if we were not to give an Activity a particular name – and even if we were to change that name. The name of the Activity is called an Activity Description.

Before we go any further, let's introduce the ICS-Dictionary definition of an Activity:

- **Activity:** The most basic building block of a Project Schedule, an Activity represents a discrete portion of the overall Scope of Work to be performed through Project Execution, with support from Project Administration. Resident in a CPM Network Diagram, an Activity is an artificial representation of its real life counterpart, a Project Execution Action. The dimensions of a single Activity’s included Work Scope are circumscribed by the Activity Description and Activity Duration, as well as the corresponding Action’s location, complexity, performers, and other limiting factors.

2B2b: Distinguishing an Activity from an Action

Let us detour for a short semantic discussion that will draw your attention to a fairly innocuous distinction in meaning that Cognitive Project Management gives to two seemingly synonymous words, Activity and Action. To be sure, the distinction we highlight is one of our own choosing; we are not suggesting that you will necessarily experience this distinction out in the workplace. Yet we make this distinction in the interest of highlighting what we believe to be an important (albeit, subtle) difference in perspective.

Let us recall that a Project Schedule is, after all is said and done, a model of a Project
Execution Strategy. That is, it is not the Project itself, it is just a representation of Project Performance. A good analogy would be a city map. The map depicts the city; it is not the city. The lines on the map depict roadways; they are not the actual roadways. You cannot drive on those lines with your car.

As we have already noted, a Project Schedule is comprised of Activities, each one representing some portion of the overall Project Scope. In other words, there is the actual Action out in the field or office, and then there is the representation of that Action within the Schedule (as a model). Cognitive Project Management has chosen to use the word Action to refer to what actually occurs as the Project is being performed, whereas the word Activity is used to refer to the artificial representation of that Action within the Schedule (as a model).

2B2c: Every Activity Has an Activity Description

For many students new to Project Time Management, the distinction between an Activity and an Activity Description is sometimes a little hard to grasp. After all, how can an Activity exist that does not also have an Activity Description? Aren’t the two things inseparable, one-and-the-same? Well, you tell us.

Take a look at Figure F0206. In Row A we see a string of three boxes that contain no textual content of any kind? Do these boxes constitute Activities? We would say “No!” They are just boxes, and even though there are Arrows that seem to link the Boxes together, we could just be looking at a flowchart of some kind.
But in **Row B**, we now see Activity Descriptions. Ah! It would now appear that these are indeed Activities, especially when we read the Descriptions and spot the Action verb, “do.” Judging solely from the second row, would we be correct in concluding that an Activity and an Activity Description are one and the same? We would answer, “Not so fast!”

Look at **Row C**. Here we find the same three Activities, but now the Activity Descriptions have become more specific. We’re beginning to detect how the student is using this string of Activities.

At first, at **Row A**, she simply chalked out placeholders for three Activities to be performed in her evenings. Next, at **Row B**, she has decided to spend her evenings doing homework and not partying with her friends. By **Row C**, she has realized that the homework assignments are to write papers, although the subjects of those papers are apparently not yet known to her. In **Row D**, she now has at last acquired a clear idea of each paper’s topic.

So, the question we are asking is this: can an Activity Description change from time to time – even though the Activity itself remains lodged between (before or after) other Activities? The answer is, obviously, “Yes.” Doesn’t that suggest that an Activity and an Activity Description are two different things? Think of when a person changes their name, such as when a woman gets married. They are still the same person; just having a different label. We state it this way: an Activity Description is one of the properties of an Activity, and every Activity has an Activity Description. The purpose of an Activity Description is to briefly describe the Work Scope Element that the Activity seeks to model.

*A less common name for an Activity Description is Activity Title. Also note that an Activity Description is not the same as an Activity Identifier.*

### 2B2d: The Sequencing of Activities

As we saw earlier, two definitions of the word, Schedule, mention the “order” of Events or Tasks as an attribute of a Performance Listing. We wish to point out here that the Sequencing (we think, a better word than “ordering”) of items in a Performance Listing is entirely optional. For this reason, we will wait until our discussion of Timetables to introduce the concept of Sequencing.

### 2B3: Timetables

Our discussion of Static Schedule Forms now moves to the consideration of Timetables. The main difference between a Performance Listing and a Timetable is the inclusion of some reference to Time, be they dates, hours, or such. A train schedule presents a
Timetable for the arrival and departure of trains at different stations.

Continuing our practice of exploring new terms and concepts as they reveal themselves with each new level of Schedule Form, let us now acknowledge certain properties contained in a Timetable that we might not necessarily have found associated with the simpler Schedule Form, Performance Listing. Specifically, let us talk about Activity Identifiers; the Concept of Time; the idea of Start/End of Day; and what we mean by the term, Sequencing.

2B3a: Activity Identifier

You may be wondering why we did not introduce the Activity Identifier earlier, as a property of the Performance Listing, instead of now as a property of Timetables. To explain our reasoning, let us first be clear on what an Activity Identifier is. If you have ever worked with a long list of any kind, whether electronic or physical, you know that each record must be uniquely distinguished from other similar entries.

What does a teacher do on the first day of school when she realizes that there are three TOM SMITHS in the room? After a little back and forth banter, an agreement is reached that one of them will be TOM, one will be TOMMY, and the third will be THOMAS.

In Schedules, similarly-worded Activity Descriptions can quite easily become confusing. The purpose for an Activity Identifier, therefore, is to provide a unique label for each distinct Activity. When the list of Activities is a manual one – such as the WEEKEND TO DO LIST in Figure F0204 – there is rarely a need for an Activity Identifier; most such lists are usually quite short.

Yet, Activity Identifiers might enhance even a handwritten Performance List in certain situations. For instance, if the list contains multiples of a single entity (e.g., trains on a train Schedule), identifying the trains by their route numbers could be clarifying.

Another condition where the use of Activity Identifiers in a manual list might be recommended is if the list is especially long. Most handwritten lists tend to be both manually-maintained and short in length, and thus the use of Activity Identifiers is not all that common among them.

On the other hand, Timetables tend to be more extensive in Scope, complex in

<table>
<thead>
<tr>
<th>SIMPLE TIMETABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BldgA-001 Excavate Trench 03JUN 04JUN</td>
</tr>
<tr>
<td>BldgA-002 Lay Pipe in Trench 05JUN 06JUN</td>
</tr>
<tr>
<td>BldgA-003 Inspect Pipe 07JUN 07JUN</td>
</tr>
<tr>
<td>BldgA-004 Backfill Trench 08JUN 08JUN</td>
</tr>
<tr>
<td>BldgB-001 Excavate Trench 05JUN 06JUN</td>
</tr>
<tr>
<td>BldgB-002 Lay Pipe in Trench 07JUN 08JUN</td>
</tr>
<tr>
<td>BldgB-003 Inspect Pipe 09JUN 09JUN</td>
</tr>
<tr>
<td>BldgB-004 Backfill Trench 10JUN 10JUN</td>
</tr>
<tr>
<td>BldgC-001 Excavate Trench 07JUN 08JUN</td>
</tr>
<tr>
<td>BldgC-002 Lay Pipe in Trench 09JUN 10JUN</td>
</tr>
<tr>
<td>BldgC-003 Inspect Pipe 11JUN 11JUN</td>
</tr>
<tr>
<td>BldgC-004 Backfill Trench 12JUN 12JUN</td>
</tr>
</tbody>
</table>

F0208: Simple Timetable
content, and frequently deal with similarly-labeled entities (e.g., bus Schedules, sports game Schedules, and even some simpler Project Schedules). Accordingly, Activity Identifiers are somewhat common in Timetables. A sample Timetable from a Project Schedule appears in Figure F0208; the first column contains Activity Identifiers, such as BldgA-001.

2B3b:  Concept of Time

A few quick words about this thing we call Time. You may be thinking that now we’re getting just a little too anal-retentive. After all, we all know what Time is. What more needs to be said? Well, there are a few subtle points that we want to make, which we think will be of help to you in your years ahead, whatever may be your particular involvement in the Project Team's achievement of effective Project Time Management.

2B3b-i:  We Don’t Manage Time

When you stop to think about the idea of Time Management, we humans don’t actually manage Time any more than we manage the wind. On a sailboat, with those white sheets billowing, we may manage our use of the wind – but we don’t actually manage the wind itself. Likewise, through clever and effective Project Management practices we humans seek to manage our use of Time. But we don’t actually manage Time itself.

2B3b-ii:  Time and Money Are Not Equivalent

For decades there has been this ongoing rivalry between Cost Engineers and Scheduling Engineers, each insisting that their respective discipline is the more important one. The Cost folks would argue that without money there would be no Projects altogether. Schedulers would argue that, no matter how much money or no matter how effective the Cost Control taken, without a Project Schedule to coordinate the work, the Project would likely never finish on Time -- or within budget!

We respectfully submit that this age-old standoff was pivoting on the wrong issues. Consider one very major difference between the sciences of Cost Management and Time Management: Dependency.

In mathematics there is a rule known as the Commutative Law of Addition, which states that a set of numbers can be added up in any possible order and the resultant sum will always be the same. Likewise, the infamous “bottom line” on any accounting table of costs will always yield the same total, no matter how the line items are sorted, organized, and summed.
But the same is not true of Schedules and their Logic, however. If you change the order of the Activities, you may very well (and quite often do) affect the earliest possible achievement dates for downstream Events or Milestones. In later chapters we will spend a considerable amount of time understanding why this is so.

For now, though, we ask you to trust us when we say that the ultimate order of the Activities does matter tremendously to all of the basic calculations of a Network-Based Schedule: Earliest Dates and Latest Dates, Total Float, Free Float, and even the route of the Schedule along one or more of its Critical Paths.

If you require a simple example, then consider that you are faced with two things to do, one of which is a prerequisite to something someone else has to do. See Figure F0209.

Assume that the goal is to complete all three Activities as soon as possible. Assume, further, that each of the three Activities will take one hour to perform. Let's see what happens if we perform the Restricting Activity first (Scenario 1), or if we perform it last (Scenario 2).

- **Scenario 1**: If you perform the Restricting Activity first, then while the other person performs the Restricted Activity, you can be performing your Unrestricting Activity in the same time frame. Between the two of you, all three Activities will complete in two hours!

- **Scenario 2**: If instead, however, you perform your Unrestricting Activity first and then proceed to the Restricting Activity second – while this sequence allows you to finish your two Activities in the same two hours the other person will not be able to commence their single Restricted Activity until you have finished your second Activity. Consequently, they will not finish their Activity until the end of the third hour!
From this simple example we see that, by altering the sequence of your two Activities, the combined length of the three Activities – the overall Schedule, so to speak – can swing by 50%!

The reason that we bring up this point here, under an esoteric discussion of Time, is that there is an important reality for you to appreciate – one that each of us has discovered in our own personal lives: Time stands still for no one! On a Construction Project, no matter whether we work efficiently, inefficiently, or don’t work at all, Time continues to flow by at a steady rate.

Consider your hometown’s central river — around which your city initially settled, and later straddled — that flows steadily and unrelentingly, as it has for eons. We can lower a water paddle into the river and harness its energy into electricity, or we can do nothing of the kind. In the same way, we can make our very best use of Time, or we can squander it. But the one thing we cannot do is slow or halt its steady slippage from our grasp.

And that is perhaps the biggest difference between money and Time. An Owner can decide to withhold funds, even suspend the Project itself. But even as he does, Time continues to trickle by. And should he decide to resume the Project at a later date, the frozen Funds can be thawed but circumstances may well be different – made different by that Passage of Time.

One can think of a Project as a plane in flight. It must constantly burn fuel in order to continuously propel forward and remain aloft. Should it run out of fuel, it will surely plummet to the earth, and there crash and burn. There is no pausing mid-air. Likewise, once a Project begins, there is no stopping … if the efficient use of Time is our measure.

2B3b-iii: Project Execution’s Use of Time

Let us make one further point about the use of Time, which has to do with whose use we are referring to. As briefly introduced earlier, Cognitive Project Management perceives the Project Management Team as operating simultaneously in four parallel Domains, corresponding visually to the four points of the Cognitive F.A.C.E. Diamond. Refer back to Figure F0202 on page 27.

The Ideology that Cognitive Project Management dares to assert is that the four Project Management Domains are not of equal importance to ultimate Project success (at least, from the Contractor’s perspective). It is our opinion that Project Execution is the primary Project Management Domain, and that the other three Domains are all secondary – performing in support roles. \[8\]

8 We hope you appreciate the political courage of this position. Contrast this stance with Dominant Project Management’s far more diplomatic assertion that all Knowledge Areas are equally important.
To better understand the previous two paragraphs, here are a few pertinent definitions from the ICS-Dictionary.

- **Project Stakeholder**: Any organization or individual that has a vested interest in how the Project is managed, performed, and accomplished.

- **Project Execution**: One of four Project Management Domains within the Cognitive Project Management model, Project Execution is where the rubber meets the road. This is the Project Management Domain where the work of the Project is performed. The Project Executor is Cognitive Project Management’s title for the role more commonly known as General Superintendent.

- **Project Administration**: One of four Project Management Domains within the Cognitive Project Management model, Project Administration provides the capacity for Project Execution, by supplying the necessary resources and operating conditions.

- **Project Coordination**: One of four Project Management Domains within the Cognitive Project Management model, Project Coordination is poised at the top of the Cognitive Project Management F.A.C.E. Diamond, where it provides a vital integration function by acting as a liaison between and among External and Internal Project Participants.

- **Internal Project Participants**: A Project Stakeholder group comprised of organizational structures and individuals that are members of the Project Team, and with whom the Project Coordinator provides a liaison role. Internal Project Participants are scattered across the four Project Management Domains of the Cognitive Project Management’s F.A.C.E. Diamond: Project Facilitation, Project Administration, Project Coordination, and Project Execution.

- **External Project Participants**: A Project Stakeholder group that is external to the Project Team and with whom the Project Coordinator provides a liaison role. External Project Participants include the General Contractor’s home office management layers as well as the Project Owner (directly, or through its representatives and agents).

- **Project Facilitation**: One of four Project Management Domains within the Cognitive Project Management model, Project Facilitation provides information and logistic guidance, and is dedicated to facilitating achievement of Project Coordination, Project Administration and Project Execution objectives.
All four Project Management Domains perform necessary Project work, and since all work consumes Time, all four domains use Time. But only Project Execution's Work Performance is primary to the ultimate purpose of the Project altogether. Therefore, the efficient use of Time in the execution of the work is what we mainly refer to, when we speak of Project Time Management.

2B3c: Start-of-Day and End-of-Day Perspectives

Time, as a concept, intrigues us in that, while it is not visible, it actually has a length. And anything that has a length therefore must have a starting point and an ending point. Any Unit of Measure has a definable start and stop. In temporal terms, we know what a Day is, just as we know what we mean by the terms Month or Week.

So when we read a date on a Timetable, we understand not just the date itself, but also the implication as to whether we are speaking about the start of that date, or the end of that date. For instance, back on Figure F0208 (page 35) we see that Building C’s Excavate Trench Activity will take two days to complete, and will span days June 7 and June 8.

But we interpret the two dates differently, don’t we? We naturally recognize the context of the situation. We understand that while the Activity will not commence before the start of June 7, it will not complete before the end of June 8. It may be helpful to think of the Action of every Activity occurring between two Points in Time that act like temporal book ends. Let us call these two Points in Time the Start-of-Day and End-of-Day, respectively.

While we will discuss this concept at greater length in Chapter Four, for now it is adequate that you simply understand that there is an implied context for any Calculated Dates that you read in a Project Schedule -- that is more specific than the Calculated Dates themselves. There is an implied understanding with respect to the start or finish of any Unit of Time – whether that happens to be a year, month, week, day, or even hour. Just keep that distinction in mind, for it will come up in our discussions, a little later on when we start working with Date Calculations in Chapter Three.

2B4: Bar Charts

At its simplest, a Bar Chart is a graphical depiction of a Timetable. Now, let’s get a little more precise. Figure F0210 presents a Bar Chart that depicts the very same information contained in the Timetable shown in Figure F0208.

☐ A Bar Chart is a graphical form of Communication.

[9] That is, the instant that an Activity can, may, or does start and the instant when it can, may, or does finish.
Activities are shown as horizontal Activity Bars.

The Activity Bars are usually plotted to scale, against the backdrop of a constant Timeline appearing at the top or bottom of the sheet.

Most often an Activity Description (and sometimes an Activity Identifier) will appear to the left of the Activity Bar.

Nothing prevents the Activity Description and Activity Identifier from being presented along with the Activity Bar, in the graphics area.

We’d like to draw your attention to some important distinctions between a Bar Chart and a Timetable, in terms of how the same information is presented.

### F0210: Simple Bar Chart

#### 2B4a: Concept of a Duration

The first observation, a very important one for you to understand, is that the Activity Bar is telling us something more than just where the Activity starts and ends in Time. Compare Figure F0210 to Figure F0212, which shows only the Activity start and end Moments in Time. Do you see what’s missing?

The Activity Bar of course! And that horizontal mass of ink, which would have connected these two Moments in Time, symbolizes the Activity Duration. An Activity Duration is a numerical value that expresses the length of Time required to perform the Work Scope of that Activity. The Activity Bar is a graphical representation of that length of Time, the Activity Duration.
If you are really thinking about what we are saying, you should be struggling with a mental question, which is: *if the Activity Bar represents a numerical value, then where is the numerical value?*

The answer, which gets us to the very important point we want to make under this heading, is that the Activity Bar *implies* a Duration, whether or not the Bar Chart’s creator had a precise numerical value in mind. The truth is that, far more often than not, the person drawing the Bar Chart does not have any *precise* sense of how long (in terms of a particular numerical value) an Activity may, must, or will take. Instead, he has this *general* sense of when the Activity can, will, or must start — as well as when it can, will, or must end.

And so, with this notion in mind, he simply lowers his pencil to the paper at a point vertically beneath a Point in Time on the Timeline when he thinks the Activity will, should, or must start. Then he drags the pencil to the right until he arrives under another Point in Time at which he thinks the Activity will, should, or must end. Does he have a precise, numerical Duration in mind? Not likely.

The ability to create a Bar Chart without actually having a numeric value (that represents the Activity Duration) in mind is one of the most attractive features of a Bar Chart Schedule … and one of its greatest weaknesses. As we will discuss in depth in Chapter Five, a Project Schedule — of any form (even a Bar Chart Schedule) — is only as good as the Activity Durations assigned to its Activities. If even one Activity Duration is bogus, the integrity of the entire Schedule may be brought into question. That is why it is a far
better practice to produce Bar Charts as a by-product of a Logic-Based Schedule, than independent of underlying Logic.

**2B4b: Workdays and Non-Workdays**

Here is another concept to wrap your mind around, which suggests yet another way that the accuracy or credibility of a Bar Chart can be eroded. As a matter of practice, Activity Bars are commonly drawn in a manner that is somewhat oblivious to the Workdays and Non-Workdays that they span.

In the world of Scheduling, the term Workday refers to a period of Time allotted for performance of Work Scope; that is, there are no explicit prohibitions on Work being performed during this period of Time. A good example of a Workday is what is commonly referred to as “business days” (e.g., Monday through Friday in the United States). By contrast, the two days of the weekend typically would be considered Non-Workdays.

As a matter of practice, most of the time when a Bar Chart is being drawn out on a construction job site, the one doing the drawing (perhaps a Field Superintendent or Crew Foreman or, more often the case, a Project Manager who, despite a Scheduling course in college, doesn't know better) draws the Activity Bar straight across Workdays and Non-Workdays, with no regard for the implications!

Imagine yourself driving cross-country on a major interstate highway. Your vehicle floats along at 75 miles per hour on what is a completely flat slab of concrete--- except, the continuous pavement is at one point lying on solid ground … and then suddenly it is atop the elevated deck of a bridge, spanning beams and columns hundreds of feet above a deep canyon below … and then it is once again back on solid ground. If you hadn’t looked down through your side window, you would never have known that you had gone from ground to air and back to ground again. That is how it seems to us, when we detect an Activity Bar drawn across Workdays, Non-Workdays, and then back to Workdays again.

“What’s the big deal about whether the days underneath the Activity Bar are Workdays or not,” you might wonder. The answer to this lies in how the Bar Chart Schedule will be used. If, on the one hand, it is simply a Communication device intended to convey a general time frame and Plan of Execution for the Project, then the damage may be minor. But if, on the other hand, the Schedule is intended to aid in coordinating and directing Action in the field, then the problematic aspect of such a Bar Chart (where its Activity Bars span Workdays and Non-Workdays indiscriminately) is potentially immense. The best way to explain the problem is to point to another analogy. (Have you noticed that we love analogies?)
In drafting, there is this concept called Scaling. That is, drawings are drawn to a specific scale; for instance, a half-inch on the drawing equals one foot in reality. The advantage of a scaled diagram is that, even if no numerical dimensions are provided in the drawing, one can calculate the length of a depicted wall simply by measuring the length of the line representing that wall (in inches) and then converting this to actual length. For example, at a scale of one-half inch per foot, a pencil line 3.5 inches long would represent a wall seven feet in length.

Most Bar Charts are drawn beneath a Timeline, which means that they are Time-Scaled Bar Charts. And this means that someone reading a Bar Chart may use the length of an Activity Bar to calculate the Activity Duration of the Activity, if no numerical value is given.\[10\]

Take a look at Figure F0214, Bar Graph A. Please tell us if those three Activities have the same Duration. Whether your answer is “yes” or “no,” you would be giving the wrong answer. The right answer is, “I can’t say for sure.”

Here is why you cannot tell for sure. If this is a Contractor’s Schedule, is he working weekends, or not? If he is working a seven-day week, then those three Activities would seem to have different Durations. But, what happens when we account for weekends?

Take a look at Bar Graph B, in Figure F0216. Now what is your answer? Do the three Activities have the same Duration? Well, it appears that at least Activity A and Activity C have the same, two-day Duration, while Activity B seems to have a four-day Duration.

But what if we told you that the weekend in the middle of the graph is preceded by a two-day holiday — and it is actually a four-day Thanksgiving Weekend, as shown in Figure F0218, Bar Graph C?

10 This is known as “scaling” the Activity Duration.
We can clearly see that, in the previous paragraph, “appears” is the operative word. It appeared that Activity B had a four-day Duration. But with the long Holiday weekend revealed, it is now clear that all three Activities actually assume the same Duration – two days each.

![Bar Graph C](image1.png)

**F0218: Bar Graph C**

So what is the practical effect of Bar Charts comprised of Activity Bars that disregard whether they are spanning Workdays or Non-Workdays? Well, quite often … confusion! But more importantly – possible misrepresentation and inaccuracy!

And it happens all of the time! Someone scales off the length of an Activity Bar (even though the Non-Workdays are showing, as we did with the shading in Figure F0218), and makes the quick (and erroneous) interpolation across periods that have a different combination of Workdays and Non-Workdays.

![Bar Graph D](image2.png)

**F0220: Bar Graph D**

To illustrate this point, suppose the Project Superintendent announces at a Weekly Coordination Meeting that the job site will be closed at the end of business on Wednesday, the day before Thanksgiving … and won’t be open for business until the Monday after the Holiday weekend. He then turns to you, the Project Scheduler, and instructs you to “show Activity A, Activity B, and Activity C as postponed until after the Holiday break.”

If you are not careful with what you are doing, you might be inclined to simply slide the bars over to the right, as shown in Figure F0220, Bar Graph D. And, of course, you would be erring … big time! We repeat: this kind of mistake happens all of the time.

Now look at Figure F0222, Bar Graph E, which shows how the Schedule should look — if, that is, you take into account both Workdays and Non-Workdays. Notice that we had to change the Activity Bar lengths to reflect the underlying Activity Durations of two-days each — and not simply slide the previously-drawn Activity Bars (with their varying lengths) to the right.
The point of this small discussion is to help you better understand the difference between Workdays and Non-Workdays, and the Bar Chart is the first Schedule Form (as we move from simple to complex) where these two terms and concepts come into play. [11]

<table>
<thead>
<tr>
<th>BAR GRAPH E</th>
<th>T</th>
<th>F</th>
<th>S</th>
<th>S</th>
<th>M</th>
<th>T</th>
<th>W</th>
<th>T</th>
<th>F</th>
<th>S</th>
<th>S</th>
<th>M</th>
<th>T</th>
<th>W</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2B4c: Continuous and Impacted Durations

Now let’s deal with two others terms that also have subtle yet important differences in meaning: Continuous Durations and Impacted Durations. First we will explore what the differences are; then we will get into why those differences are important to you, as a Project Facilitator. We’ll contrast the differences through two sports analogies.

2B4c-i: The Difference between Continuous and Impacted Durations

As the word suggests, a Continuous Duration refers to the Duration of an Activity that represents a corresponding Action expected to continue un中断, once it has begun and until it has completed. Think of a first-time parachutist who is about to jump out of an airplane at 12,000 feet. He asks his trainer, “How long will the drop take?” The trainer answers, “At this height, about six minutes.” This is an example of a Continuous Duration. Once the student steps out of the aircraft, the drop begins and it does not end until he arrives on terra firma, hopefully safely.

Now consider the typical American football game. Officially, the game’s Continuous Duration is 60 minutes. But the Impacted Duration of the game is always considerably longer than 60 minutes. Here is what Wikipedia has to say about the length of a football game:

- Football: A standard football game consists of four 15-minute quarters (12-minute quarters in high-school football and often shorter at lower levels), with a 12 minute half-time intermission after the second quarter. The clock stops after certain plays; therefore, a game can last considerably longer (often more than three hours in real time), and if a game is broadcast on television, TV time-outs are taken at certain intervals of the game to broadcast commercials outside of game action.

We want to be clear about what we mean by “impact.” We are talking about impacts

[11] Later, in Chapter Nine, when we discuss Work Performance Calendars, we will learn more about the differences between Workdays versus Non-Workdays.
due to factors *external* to the Activity itself. In a Schedule, this means the impacts due to other Activities. To understand this better, take a look at Figure F0224.

**Activity B** shows a Continuous Duration of *four-days*. But **Activity B** is tied, logically, back to **Activity A**, with both Start-to-Start and Finish-to-Finish Restrictions. We realize that you have not yet been introduced to Logic Ties and Restriction Delays, but we think you will still be able to follow the Logic as we interpret it. What the SS:2 says is that **Activity B** cannot start any earlier than *two-days* after **Activity A** starts.

Likewise, the FF:2 says that **Activity B** cannot finish any earlier than *two-days* after **Activity A** finishes. Since **Activity A** has an Activity Duration of *eight-days*, and since (thanks to the Logic Ties) **Activity B** will both start and end no earlier than **Activity A** starts and ends, then **Activity B** has an Impacted Duration of *eight-days*.

You may be wondering why we wouldn't just assign a Continuous Duration of *eight-days* to **Activity B**, since it is clearly going to take at least *eight days*, based on the current Logic? The answer is in those last five words, “*based on the current Logic.*”

What would happen if we were to postpone the start of **Activity B** until after **Activity A** had finished; thus, starting on *Day 9*? When would **Activity B** now finish, *at the earliest*? Well, if the work of **Activity B** can actually be accomplished in *four-days*, as we were originally told, then **Activity B** could finish as early as *Day 12*. But if we had artificially changed the Continuous Duration to *eight-days*, then wouldn't we be computing an Earliest Finish of *Day 16*? Wouldn't that be wrong?

Okay, now let's go back to our original point. If an Impacted Duration includes the
effects of external influences, but a Continuous Duration assumes uninterrupted workflow, would we be safe in saying that an Impacted Duration is always longer than a Continuous Duration? Well, not if there are no impacts. It comes down to whether the Activity is impacted and, if so, by how much. One thing is certain: an Impacted Duration can never be shorter than the Continuous Duration.

Now that you understand the difference between Continuous Durations and Impacted Durations, we can present their definitions as found in the ICS-Dictionary:

- **Continuous Duration:** A Continuous Duration is an Activity Duration that represents a corresponding Action which is expected, once it has begun, to continue uninterrupted until it has completed.

- **Impacted Duration:** The effective Activity Duration of an Overlapped Restricted Activity when the effects of intervening (impacting) Performance Linkages (e.g., Default Restrictions, Start Restrictions, and Finish Restrictions) are taken into consideration.

2B4c-ii: Continuous and Impacted Durations: Why They Matter

Now let’s get to our second point: this topic’s importance to you as a Project Facilitator. Consider this question: In the typical Construction Schedule, what percent of the Activities do you suppose are impacted by one thing or another? Or, more to the point, – and recalling that a Project Schedule is a simulation model of real Project Execution – on a typical Construction Project, what percent of the Actions do you suppose are impacted by other Actions? In the inverse, what percent of the Activities/Actions usually continue, unimpacted, once they commence and continue straight on through until they finish?

If you said that most are impacted, you would be correct. And that is because the average Construction Project is wrought with people, conditions, processes, and Actions that all challenge the deliberate prosecution of even the simplest undertaking. If this makes sense to you, who may be someone who is possibly completely green to Construction, then surely the experienced Project Manager, Superintendent, or Crew Foreman knows this just as well.

So when you, as a Project Facilitator, ask one of these three contributors to the Schedule Development process to provide an Estimated Duration for a given Activity, you can bet dinner on the fact that they will be wanting to give you an Impacted Duration! The problem, of course, is that we don’t want to assign Impacted Durations to Activities. Instead, we want to apply Continuous Durations. Just why we want Continuous Durations and not Impacted Durations is something fully discussed in Chapter Nine.
The reason we are bringing up the difference between Continuous Durations and Impacted Durations here in **CHAPTER TWO** is because this distinction actually matters to how we evaluate a simple Bar Chart! When you spot an Activity Bar drawn across a period of Time, do you suppose that the Activity Duration implied by that Activity Bar represents a Continuous Duration or Impacted Duration?

*Impacted* … of course! And now we have a second justification for our growing opinion that Bar Chart Schedules may be great for communicating a general Plan for the work, but they are prone to being unreliable as a Project coordination tool … unless, that is, pains are taken to account for Non-Workdays and Impacted Durations. Most Bar Charts don't (except, perhaps, very detailed, **LEVEL 4 LOOK-AHEAD Schedules**). [12]

Early in your career as a Project Facilitator you will master the art of explaining the subtle difference between Continuous Durations and Impacted Durations to field personnel from whom you will be eliciting Activity Duration estimates. If you do not make them aware of this important difference, then the Activity Duration estimates you receive will surely be a mixture of Continuous Durations and Impacted Durations, and the resultant Schedule that depends on these Activity Duration estimates, as a consequence, may well lose much of its accuracy and credibility. Our message: be careful!

**2B4d:** **Concept of Continuous Crew Days**

This is a term that Cognitive Project Management uses to help explain to field personnel the difference between Continuous Durations and Impacted Durations. So you may not run into too many others who are familiar with this term.

Let us be clear: the term Continuous Crew Day is not synonymous with Continuous Duration. Let’s not forget that a Duration, regardless of its basis (e.g., continuous, impacted -- or others not yet discussed), is a numeric value and as such is a quantity or **measurement.** By contrast, a Continuous Crew Day is a Unit of Measure.

Think of a road map. It shows the distances between cities and towns, right? For instance,
the distance between Toledo, Ohio and Detroit, Michigan is sixty (60) miles. The distance of 60 miles is a Measurement, whereas the Unit of Measure is a mile. Likewise, the Activity Duration is a temporal quantity or Measurement, whereas a Continuous Crew Day is a temporal Unit of Measure.

Let's talk a minute about Crew Configurations. The ICS-Dictionary offers this definition:

- **Crew Configuration**: The compositional make-up of a work crew: both the number of workers and type of workers.

Each of the following constitutes a different Crew Configuration:

- **Crew Configurations, Differing by Quantity**
  - One laborer, one carpenter
  - One laborer, three carpenters
  - Two laborers, one carpenter

- **Crew Configurations, Differing by Type**
  - One laborer, one electrician
  - One laborer, one pipe fitter

- **Crew Configurations, Differing by Quantity and Type**
  - One laborer, one electrician, one pipe fitter
  - Two laborers, one electrician, three carpenters

What is really neat (and powerful) about Continuous Crew Days is that they absorb (account for) differences in Crew Configurations. Let’s say we have a large pile of used bricks that need to have the mortar scraped off of them. Let’s suppose that a laborer can scrape **15 bricks** in an hour. In an **eight-hour** Workday, she can scrape **120 bricks**. Now let’s suppose that the pile contains **2,160 bricks**.

If the **Brick Scraping Crew** is comprised of only one laborer, then the Continuous Duration, based on Continuous Crew Days, would be **18 days** \( \frac{2,160}{120} \). But if we add a second laborer, thus reconfiguring the crew to a **two-person crew**, the Continuous Duration for the Activity would reduce by **50%**, and become **nine** Continuous Crew Days.

If we assumed a **three-man crew**, the Continuous Duration would become **six** Continuous Crew Days. By adding the word Crew, we make it easier for a field person to understand what we are talking about when we ask for Continuous Durations.

Let's take another example to see how Cognitive Project Management recommends that you **begin** a Logic Development Session with a field person, where you are about to be asking for a lot of Activity Durations.
“Mac,” you begin, “we are about to go through this set of Activities (pointing to a sheet of paper on
the table), and I need for you to estimate a Continuous Duration for each Activity. Let me explain
what I mean by “continuous.”
The man sitting in front of you looks apprehensive, if not also a little scared. You continue. “Okay,
let’s start with this first Activity, **Excavate Perimeter Footings.** Tell me about the crew you will
be using for this Activity.”
You pause, dramatically waiting with pencil in hand. Not just a little confused, he begins: “Well, I
don’t know what you are asking me … but,” he pauses, “… I’ll be using a backhoe and a laborer.”
“Okay,” you respond, “so you’ll have one machine operator and one laborer?” You emphasize the word
“one.” He nods in agreement. “I assume the laborer is hand-shoveling spillage back into the hole?”
“No, not really. He’s hand-digging the small corners and other places where it is more cost-effective
to do it by hand than by machine?”
“Oh,” you say, “so how long will it take, with just the two of them, to completely excavate the
perimeter footings around this building?”
“Probably a week … or more.”
“Why so long? I thought it would have been one day; two at the most.”
“Well, yeah, scooping out the trench is nothing for a machine. But that ground’s full of stones that
the Owner wants saved and reused for pavers on the driveway later in the job.”
“Go on…”
“So, I’m gonna have the laborer hand-pick the best stones from the bucket of each scoop.”
“Okay. I get it,” you respond. “So let me ask you – couldn’t you add a second or third laborer to the
task?”
“Well, yeah, I could. But why do I care? It don’t matter none if I get the trench cut sooner, ‘cause the
pipe ain’t expected for another week!”
“Okay. I understand your thinking.” You now need to teach him a few things, but you worry about
maintaining the relief that is beginning to appear on his face. You don’t want to come across as some
nerd in a white shirt. Right now, he feels confident, like he knows something that you don’t. You’d
prefer to keep it that way for as long as possible. When a person feels confident they tend to be more
talkative, and to provide better information.
“Mac,” you begin, “that really helped. Thanks.” Then you purposely pause, as if to collect your
thoughts. This dramatic pause sends him the distinct message that you are about to take the microphone
back, so to speak. “Here’s the thing….” He waits, carefully studying you, watching to see if you are
about to challenge or argue with him. You continue.
“You see, we need to build this Schedule in such a way that it can be easily adjusted no matter what
the changing circumstances are. You’re spending a lot of time in this trailer with me, creating this
Schedule. We want to make the most of your time, by making a Schedule that can contract or expand
as you need it to. Does that make sense?”
He nods. You read his face: he is still not sure where you are headed, but at least for the moment
he’s willing to let you keep talking. “For instance,” you move on, “what if Billy\textsuperscript{13} decides to pay
premium dollars to get that pipe here overnight, so we don’t have to wait a week?”

\textsuperscript{13} The Project Manager; Mac’s boss
Mac thinks for a moment and begins to speak ... but then quickly stops himself. Then he starts again: “Well, if I knew now that the pipe won’t be a problem – uh, a late delivery – then I’d wanna work that crew faster.”
“And how would you do that?”
“Well, the holdup ain’t the machine,” he blurs out somewhat sarcastically.
“It’s the one laborer, right?” you ask, fairly confident you were following his thinking.
“You betcha! Hell, we can put three laborers around that bucket.”
“And if you...” you start to say.
“Two days!” He cuts you off, with a smile.
“Okay, good. So, the answer to my original question then, Mac, is two-days. Right?”
“Yeah,” he responded, “but remember ... this assumes a four-man crew: one operator and three laborers.” He stops, as if he just thought of something else. “But wait ... what if we have to wait for that pipe after all?”

Now it is your turn to cut him off. “No problem! The answer is this thing we Schedulers call ‘Float.’ We’ll get into that later Mac, but for now just know that each Activity will have a value, called Float, that will tell us whether we need to rush or not. If the pipe delivery is escalated by Billy, then the Float will be less. If it isn’t, then the Float will be more. If you see that you have a lot of Float, then you can cut back on the manpower. Okay?”

Mac looks slightly confused. But you can also see that he is beginning to trust you, so you let it go at that, and move on to the next Activity.

“Okay, now let’s look at the INSTALL SEWER PIPE Activity. Again ... let’s start with defining the Crew Configuration that you have in mind. And based on that Crew Configuration, and assuming no interruptions ... how many Continuous Crew Days will it take…”

And off you go, for the next few hours. We hope you now understand what the term Continuous Crew Day means and how it is used. Most especially, we hope you appreciate why it is so important. Just remember this: a Schedule is only as credible as the Activity Durations within it.

The main message of this chapter, and indeed of this book, is this: regardless of what assumptions are being made as Activity Durations are being estimated, it is vital that those assumptions be consistently applied. You just can’t have some Activities expressing Impacted Durations, while others reflect Continuous Durations – or, some with Continuous Crew Days, while others with interrupted crew days.

Cognitive Project Management recommends that the following script, or something like it, be recited to each person from whom you will be seeking a Duration estimate:

“John, (insert your audience’s name) I’d like you to assume whatever Crew Configuration you desire -- and tell me what that configuration is, because I need to write it down. I also want you to assume that:
It is extremely important for you to insure that everyone has the same understanding about the basis of Activity Durations, before you spend any amount of time in Schedule Development. And now, having examined Activity Durations from every conceivable angle, let us present the ICS-Dictionary definition of an Activity Duration:

**Activity Duration:** Expressed in Continuous Crew Days, the Activity Duration represents an estimate of the amount of time required by an assumed Crew Configuration to perform the Scope of Work of an Activity, as briefly described by the Activity Description. An Activity Duration is one of two Date Calculation Variables used to determine the appropriate Date Calculation Procedures, Computational Rules, and Arithmetic Formulas.

**2B4e: Activity Sequencing Not the Same as Activity Dependency**

Before we leave our discussion of Static Project Time Management Tools, let us draw deep lines of distinction between three seemingly similar terms: Sequencing, Timing, and Activity Dependency.

Sequencing (synonymous with Order) refers to the arrangement of objects, in time or space, relative to one another. Two instructive definitions from Dictionary.com may shed some light.

- **Order:** The disposition of things following one after another, as in space or Time; succession or sequence.
- **Sequence:** The following of one thing after another; succession; a continuous or connected series.

So, when we speak of the Sequence or order of things, we are considering the comparative arrangement of those things, relative to one another. Slow-walking participants making their way down the aisle in a particular order during a wedding procession are an example of Sequencing.

Notice that, according to the first definition at least, Sequencing may be an expression of order against a backdrop of a Time – but it doesn’t have to be. It could be just a spatial arrangement, such as shoppers in line at the checkout counter, or cars in line at a traffic
light.

So, a numbered Performance Listing provides a sequence devoid of a Time reference. Now, compare that to a Timetable, which includes both a Performance Listing and clear references to Time. A Timetable provides us with information about the timing of Activities, not just their Sequencing.

The third concept, Activity Dependency, goes beyond Sequencing and Timing, and speaks to the causative aspect of sequencing between Activities, such that the performance of the Restricting Activity to some extent restricts the performance of the Restricted Activity.

Let’s return to Dictionary.com:

- **Dependent:** conditioned or determined by something else.

Cars (all but the first one, of course) waiting in line for a traffic light to turn to green are each dependent on any cars in front of them to roll forward before they can do likewise. By contrast, runners in separate lanes, side by side, about to begin a foot race around an athletic track are not dependent on one another for their own performance.

So now, before we leave our examination of Bar Charts, let us ask you one final question. **Does a Bar Chart indicate Dependency?** We know that a Bar Chart always indicates Sequencing and it can indicate Timing (if drawn beneath a Timeline), but -- does it also tell us something about the dependency of Activities, one upon another?

Take a look at Figure F0226, which combines Figure F0208 and Figure F0210, from earlier in this chapter. Viewed side by side, it is easy to see how a Bar Chart form so coherently conveys the intended Sequencing of Activities. And in conjunction with the two Date columns, we clearly have information about the timing of the Activities, as well. But what do we know about how the Activities relative to one another?

Based on what you see in Figure F0226, would you say that, for any of the three buildings, the start of **Lay Pipe in Trench** is dependent upon the prior completion of its corresponding **Excavate Trench** Activity? If you said “Yes,” then on what did you base your opinion?

You have three possible answers to our question: “I formed my opinion based on….”

- … the position of the Activity Bars relative to the Timeline and relative to one another.”
- … the Start and Finish Dates in the included Timetable.”
- … the wording of the Activity Descriptions.”

What would you say if we told you that none of these are a legitimate reason to conclude that there necessarily exists any dependent relationship between these Activities? Here’s why.
Relative Position of Bars: The first reason is a visual one. Seeing one Activity Bar starting at the same Point in Time beneath the Timeline that another Activity Bar ends, you automatically assume that the start of the latter Activity is somehow dependent on the completion of the former Activity. But this may not be true.

For instance, why can’t the first two Activities overlap? That is, why couldn’t the Contractor start to **Lay Pipe in Trench** when **Excavate Trench** is roughly 50% completion … rather than waiting for **Excavate Trench** to finish completely? The answer may be that the Contractor is using one crew to perform both Activities, and so that crew can’t start the second Activity until it has finished the first one. But what if the Contractor was able to use two different crews, one for **Excavate Trench** and one for **Lay Pipe in Trench**?

“Okay,” you say, “I see your point. Obviously, the Contractor is using one crew for the first two Activities.” But you would be making yet another hasty and possibly erroneous assumption, since the Bar Chart tells us nothing definitively about Crew Configurations.

What if we were to tell you that we have inside information and happen to know that the Contractor is using different crews to perform **Excavate Trench** and **Lay Pipe in Trench** and that, from a labor availability perspective, the two Activities could be overlapped?

“So, why aren’t the two Activities overlapped,” you ask? The answer, we have been told by the Contractor, is the delivery of pipe (not shown on this Bar Chart), is not expected to arrive until first thing in the morning, **June 5th**. The point we are making is that, just because one Activity starts at the same time another

<table>
<thead>
<tr>
<th>Building A</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>BldgA-001</td>
<td>Excavate Trench</td>
<td>03JUN</td>
<td>04JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgA-002</td>
<td>Lay Pipe in Trench</td>
<td>05JUN</td>
<td>06JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgA-003</td>
<td>Inspect Pipe</td>
<td>07JUN</td>
<td>07JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgA-004</td>
<td>Backfill Trench</td>
<td>08JUN</td>
<td>08JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Building B</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>BldgB-001</td>
<td>Excavate Trench</td>
<td>05JUN</td>
<td>06JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgB-002</td>
<td>Lay Pipe in Trench</td>
<td>07JUN</td>
<td>08JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgB-003</td>
<td>Inspect Pipe</td>
<td>09JUN</td>
<td>09JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgB-004</td>
<td>Backfill Trench</td>
<td>10JUN</td>
<td>10JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Building C</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>BldgC-001</td>
<td>Excavate Trench</td>
<td>07JUN</td>
<td>08JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgC-002</td>
<td>Lay Pipe in Trench</td>
<td>09JUN</td>
<td>10JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgC-003</td>
<td>Inspect Pipe</td>
<td>11JUN</td>
<td>11JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BldgC-004</td>
<td>Backfill Trench</td>
<td>12JUN</td>
<td>12JUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Activity ends, you cannot leap to the conclusion that the latter Activity is necessarily dependent on the former one.

Applying this warning, what do you say about what appears to be a clear Sequencing of similar work, from one building to the next? Based on the Start Dates and Finish Dates, doesn’t it look like the Contractor plans to use one EXCAVATE TRENCH crew, that will move from BUILDING A to BUILDING B, and finally on to BUILDING C?

We would agree with you that it does appear that way. But we would have to stop at the word “appear.” We would not conclude that the three buildings are necessarily tied together in a dependent way. And just what might give us sufficient confidence to conclude that the buildings are tied together? Answer: an explicit indicator of Dependency or Restriction.

The traditional graphical symbol for Dependency or Restriction in a Bar Chart is a Restriction Arrow. Take a look at Figure F0228, and see if this reinforces our point.

Using Restriction Arrows, the author of the Bar Chart can explicitly indicate the Performance Restrictions that are part of the assumptions upon which the Schedule is based. From the Restriction Arrows, plus perhaps a couple of annotations, we see that the Contractor plans to deploy two crews in what is known among Project Facilitators as a Leapfrog arrangement. Leapfrogging is often used in high-rise construction, where one crew might take the even-numbered floors, while another crew attacks the odd-numbered floors.

In Figure F0228, we see that one crew is scheduled to perform EXCAVATE TRENCH and LAY PIPE IN TRENCH Activities at BUILDING A before moving on to BUILDING C.
Another crew, scheduled to start two days after the first crew, will tend to **Building B**.

- **Start and Finish Dates:** Do you see how we would have been wrong to make assumptions about the Dependency or Restriction between Activities simply based on the physical orientation of Activity Bars on a Bar Chart? Well, the same holds true for misreading Timetables, where we assume -- just because **Lay Pipe in Trench** at **Building A** is Scheduled to begin on **June 5th** and **Excavate Trench** is expected to complete on **June 4th** -- that those two Activities have a Dependency between them. You now know that such an assumption, while reasonable (especially when our thought process further takes into account the Activity Descriptions), doesn’t make it necessarily so.
Wording of Activity Descriptions: We hope you see that even the wording of the Activity Descriptions might lead us to the wrong conclusion.

In closing, Static Project Time Management Tools are not designed or intended (and should not be expected) to express the Dependency or Restriction among Activities in a Schedule. And you have been warned against drawing inferences about Dependencies or Restrictions from Static Schedules, even from Bar Charts which sometimes seem to be shouting, “This Activity is clearly waiting on that Activity before it can start!” Unless the Bar Chart contains Restriction Arrows specifically included to represent explicit Dependencies, don’t draw any assumptions about any causal Relationships between the Activities.

Finally, we want to note that Dependencies or Restrictions are never included in Performance Listings or Timetables, and only rarely in Bar Charts. When they are included in Bar Charts, most of the time they are generated by Scheduling Software. Only in the rarest of occasions will you find custom-drawn Dependencies or Restrictions, such as in Figure F0228.

Unfortunately, most software-drawn Dependencies or Restrictions in Bar Charts are difficult to decipher and are, as a result, less than useful altogether – as Figure F0230 demonstrates. This is because such software programs provide merely an all-or-none-at-all option with respect to depicting Dependencies or Restrictions. Either all Dependencies or Restrictions are shown, or none are shown. And when all are shown, the resulting illustrations are often so congested that they are next to impossible to read.

Bottom Line: Do not associate (the depiction of) Dependencies or Restrictions with Static Project Time Management Tools. Consideration of the interdependencies among Project Actions is the very hallmark of the Dynamic Project Time Management Tool. When you think of the Critical Path Method of Modeling, one appreciates its ability to describe how thousands of Project Actions interact with one another, when they are dependent upon one another.

2C: Project Execution Commitments

A Project Execution Commitment is one of those terms that falls into a category that we mentioned earlier; it does not exist in the Dominant Project Management vernacular, but rather is a creation of Cognitive Project Management. In order to explain what a Project Execution Commitment is, we first need to consider the meaning of the word Event, as used in Dominant Project management circles.
2C1: The Difference between Events and Milestones

As is the case with almost every term that is important to Project Time Management, the term Event is inconsistently defined and used by Dominant Project Management. There is no point looking up Dictionary.com’s definition of the word, Event, because it has so many common meanings.

2C1a: Dominant Project Management’s View of Events and Milestones

Let’s get right to the question of what that word means to seasoned Project Time Management practitioners in the Dominant Project Management world. While there are many different interpretations of the word, we think that two prevailing uses sufficiently frame the debate that rages on among Scheduling experts.

- **Event as a Milestone:** One camp thinks of an Event as a significant Milestone in the life of a Project. The Owner’s issuance of a **NOTICE TO PROCEED** document is an example of a Significant Milestone. A final Activity in a Schedule, called **PROJECT COMPLETE**, might constitute another Significant Milestone.

- **Event as Moment in Time:** The other camp, though, puts a far more technical spin on the term, Event. In the context of a Project Schedule, they construe an Event as an imaginary Moment in Time that unmistakably positions itself *between* the Activities in the Schedule.

Here are a few popular definitions of the word, Event, found at the extremely helpful website of Max Wideman: [14]

- **Event:** A point in time when certain conditions have been fulfilled, such as the start or completion of one or more activities.
- **Event:** A happening or occurrence; outcome of an activity or decision point between activities.
- **Event:** In CPM networks, the end state for one or more activities that occurs at a specific point in time.
- **Event:** Something that happens at a point or moment in time. A significant event is often called a “milestone.”
- **Event:** Something that happens; an occurrence, an outcome.
- **Event:** A point in time representing the intersection of two or more arrows. The event has no time duration. It can be a milestone. Junction between two or more activities in a logical network.

[14] Definitions found in the PM Glossary at www.maxwideman.com
We won’t spend much time analyzing the above definitions. But we do want to make two observations:

- The term, Event, is considered by some to be synonymous with another important Scheduling term, Milestone. For others, though, the two terms have different meanings.
- There seems to be general consensus that Events occur between Activities.

Now (if you’re not confused enough already) let’s review a few popular definitions of the term, Milestone, from the same source:

- **Milestone:** A point in time representing a key or important intermediate event in the life of a project.
- **Milestone:** A key or important intermediate goal in the network.
- **Milestone:** A major event in a project, typically one requiring the customer or upper management to approve further work.
- **Milestone:** An activity with zero duration (usually marking the end of a period).
- **Milestone:** A significant point or event in the project.
- **Milestone:** Significant event in the project.
- **Milestone:** A key event selected for its importance in the project. Commonly used in relation to progress, a milestone is often used to signify a key date.

If the first set of definitions, for the term Event, seemed to suggest that an Event is somehow different than a Milestone, the above seven definitions for the term Milestone clearly reverse that understanding!

Five out of the seven definitions define a Milestone as a type of Event! The problem this creates is that, if an Event and a Milestone are synonymous, then the above seven definitions paint a different picture of what constitutes an Event (Milestone).

Here we are being told that it is a “key or important intermediate goal,” or a “major Event,” or “an Activity with zero Duration,” or a “significant Event or point in the Project.”

So, based on the above 13 definitions, can you answer these questions?

- What is the difference between an Event and a Milestone? Is there any difference?
- Must an Event or Milestone be zero-Duration, or can it consume Time?
- Must an Event be “significant,” or is it just “something that happens?”
- Is a Milestone or Event an Activity?

2C1b: **Cognitive Project Management’s View of Events and Milestones**

Don’t feel bad if you aren’t sure how to answer some (or any) of these questions. We
weren’t sure either – as long as we tried to reconcile our answers with the prevailing definitions. And that is why Cognitive Project Management decided to ignore all of the popular definitions, and craft its own terms and meanings.

We must warn you: The terms we are about to introduce are ones that were conceived by Cognitive Project Management for use throughout the ICS-Compendium. In other words, the meanings that we have ascribed to these two terms – that you are about to learn – are not what you may encounter in the real world. In addition, we have invented additional new terms, to supplement the two over-worked (and under-clarified) terms, Event and Milestone.

As you read on, please refer to Figure F0232. Beginning at the second row from the top, notice Events and Milestones. Cognitive Project Management distinguishes these two
terms on the basis of whether they consume Time, or not. You will recall from the above thirteen definitions that there was no consistency on this point. Some said that Events (or Milestones) consume Time, while others said that they did not. From this point forward, we will use Cognitive Project Management’s interpretation, as follows:

- Events consume Time.
- Milestones do not consume Time.

2C1b-i: Introducing the Project Execution Commitment

Cognitive Project Management regards both Events and Milestones as categories of Project Execution Commitments, as explained in this ICS-Dictionary definition:

- **Project Execution Commitment:** As the words suggest, a Project Execution Commitment refers to a promise made by members of the Project Team with respect to how, when, and where Project Execution will be performed. Whether initially required by Contract or the result of negotiated, mutual Execution Strategy — what begins as a Commitment, once captured in the Project Schedule, transforms into a sacred promise. In the Project Schedule, Project Execution Commitments come in two forms: Events and Milestones.

2C1b-ii: Events and Milestones: Per Cognitive Project Management

Project Execution Commitments can be divided into Events and Milestones, as show in Figure F0232. The ICS-Dictionary distinguishes between these two, seemingly-synonymous terms, as follows:

- **Event:** In a Project Schedule, an Event is a Project Execution Commitment that consumes Time. Events are further categorized as either Anchored or Floating. An Anchored Event is inflexibly fixed in Time by influences external to the Project. A Floating Event is one in which the exact timing of the Event varies, depending on performance progress on the Project; an Anchored Event is fixed in Time.

- **Milestone:** In a Project Schedule, a Milestone is a Project Execution Commitment that consumes no Time, but is instead a Moment in Time. Milestones are further categorized as either Anchored or Floating. A Floating Milestone is one that occurs as a result of the Project’s natural performance progression and, thus, whose Timing cannot be known ahead of time with absolute certainty. An Anchored Milestone is one that is set ahead of time, with certainty, such as a Contractual Completion Deadline or a Contract Award Date.
2 Static Project Time Management Tools

2C1b-iii: Floating and Anchored Events

Each type of Project Execution Commitment can be further subdivided. According to Cognitive Project Management, Events are categorized as either Floating or Anchored. Per the ICS-Dictionary:

- **Floating Event**: As a category of Project Execution Commitments, Events are Moments in Time that are significant to the performance of a Work Scope Element; Events consume Time. Floating Events are ones about which the exact Timing of the Event varies, depending on forward progress on the Project. An example of a Floating Event is a Project’s Switchover to Permanent Power.

Let's further consider the example of a switchover of a Project from **Temporary Power** to **Permanent Power**. Until this switch over happens, generators will have been used to provide electricity to the Project in order to power construction operations. Meanwhile, the local power company has been working to bring permanent power to the site, by laying utility lines that connect from the main service at a nearby substation or transformer, up to and connected with the building itself.

The **Switchover to Permanent Power** process takes many hours and so it cannot be captured in the Schedule as a zero-duration Activity, as this would not be accurate. Because it therefore consumes Time, it cannot be classified as a Milestone, but is instead an Event. Further, it is a Floating Event because the exact timing of the Event varies, depending on the performance and progress leading up to the Event.

- **Anchored Event**: As a category of Project Execution Commitments, Events are Moments in Time that are significant to the performance of a Work Scope Element; Events consume Time. Anchored Events are ones which are fixed in Time as to when they will occur. An example of an Anchored Event is a Groundbreaking Ceremony.

A **Grand Opening Ceremony** often happens near the end of a Project, but still a few weeks (or months) before its actual end. When a **Grand Opening Ceremony** is happening, quite often all work grinds to a halt. We would want to include the Event in the Schedule because it is sandwiched between real work Activities. The Event is Anchored in Time by influences external to the Project, as happens when the planners of the Event establish a firm date and send out invitations.

You may be wondering why an Event is not considered an Activity, since it consumes...
Time, and most likely also Resources. The answer, according to the ICS-Compendium, is that the Event must be “significant.” One could think of an Event as an Activity of “major significance” to the Project. We would not object to that characterization.

2C1b-iv: Floating and Anchored Milestones

As for the term Milestone, we are also given two categories: Floating Milestones and Anchored Milestones. Here are the ICS-Dictionary definitions:

- **Floating Milestone:** Milestones establish temporal boundaries of when a Project's Work Elements can or must be performed; Milestones do not consume Time. Floating Milestones are significant Moments in Time that occur as a result of the Project's natural performance progression. An example of a Floating Milestone is a Zero-Duration Activity added to the end of a Subnetwork of Logic, called 2nd Floor Complete. This Milestone “happens when it happens.”

- **Anchored Milestone:** Milestones establish temporal boundaries of when a Project’s work elements can or must be performed; Milestones do not consume Time. Anchored Milestones are fixed in Time. Anchored Milestones come in two types: Release Milestones and Deadline Milestones. A Release Milestone, found at the start of a Work Scope Element, defers the commencement of Work performance on one or more Work Elements until some specific future date. Contrast this with a Deadline Milestone which, found at the end of a Work Scope Element, reflects a formal commitment to complete one or more Work elements by a specific future date.

Let us further expound on the two types of Anchored Milestones, to be found in virtually all Project Schedules. These definitions come from the ICS-Dictionary:

- **Release Milestone:** A Release Milestone is an Anchored Milestone that defers the commencement of work performance on one or more Work Elements until some specific future date. A Notice to Proceed document is an example of a Release (Fixed) Milestone.

- **Deadline Milestone:** A Deadline Milestone is an Anchored Finish Milestone that reflects a formal Project Execution Commitment to complete one or more Work Elements by a specific future date. A Substantial Completion date is an example of a Deadline Milestone.
Now that you know what a Project Execution Commitment is, we can at last introduce its casual-conversation nickname: Points of Navigational Significance ... or, PINS, for short. Here is its definition, from the ICS-Dictionary:

**Point of Navigational Significance (PIN):** This is a nickname for a Project Execution Commitment and its acronym conjures the image of a push pin on a map. PINS are used in just this way in Momentology – where, within the Project Schedule, they represent short-term Project Execution Commitments that the Project Team agrees to pursue and achieve in a timely manner.

Why do we call them PINS? Well, a Project Schedule can be thought of as a map, marking the route that one intends to take to get from **PROJECT START** to **PROJECT COMPLETION**. Suppose you are a long haul truck driver and are taking a load from New York City to Los Angeles. The dispatcher calls you on the night before you leave New York, and gives you a second pick-up, this one in Dallas. You have a large map on the wall in your home office. See Figure F0234.

You press a push pin into New York and another into Los Angeles. Then you press a third pin into Dallas. Now you stretch two rubber bands onto the PINS, one from New York to Dallas, and the other from Dallas to Los Angeles. These PINS mark what might be called Points of...
Navigational Significance. They establish the parameters for the upcoming trip (which, by the way, satisfies the definition of a Project).

Suppose that the dispatcher calls back in the morning, just as you are about to pull out, and tells you that the midway drop has been moved from Dallas to St. Louis. You reposition the middle push pin — and the rubber bands stretch accordingly.

Well, just as maps reflect distance, Schedules reflect Time. And you can think of the opening Milestone of a Schedule, say **START PROJECT**, as a Point of Origin. And you can consider the final Activity in the Schedule, say **COMPLETE PROJECT**, as a Point of Termination. And if, along the way, the Owner suddenly requires that the Project achieve **PERMANENT POWER** by a specific Calculated Date, then you would create an **ACHIEVE PERMANENT POWER** Deadline Milestone.
CHAPTER THREE

3A  Introducing Dynamic Project Time Management Tools
3A1  Characterized by Energy or Effective Action
3A2  Affected by the Passage of Time
3A3  Affected by the Presence of Absence of Power
3A4  Motivates or Affects Development

3B  Activity Relationships: Two VERY Different Perspectives
3B1  Relationship, Beyond Dependence
3B2  Cognitive’s Three Relationship Categories
   3B2a  Communal Relationships
   3B2b  Symbiotic Relationships
   3B2c  Progressive Relationships
3B2d  Introducing the Term, Restriction
   3B2d-i  Understanding the Four Restriction Objectives
   3B2d-ii  Comparing Dominant and Cognitive Relationship Terms

3C  From Activities to Logic Diagrams
3C1  The Terminology of Performance Restrictions
3C2  Activity Path: Through Immediately-Restricting Ties

3D  CPM’s Four Restriction Linkages
3D1  Four Performance Restriction Linkages
3D2  Performance Restriction’s Three Key Attributes
3D3  CPM Dependency Types: Traditional Clarifications
   3D3a  Popular Definitions for Finish-to-Start Dependency Type
   3D3b  Popular Definitions for Start-to-Start Dependency Type
   3D3c  Popular Definitions for Finish-to-Finish Dependency Type
   3D3d  Popular Definitions for Start-to-Finish Dependency Type
   3D3d-i  Meaning of Restriction Delay in a Start-to-Finish Type
   3D3d-ii  When is a Predecessor Actually a Successor?
3D4  Performance Restrictions: Per Cognitive Project Management
   3D4a  Restriction Delay: Work Performance or Time Passage?
   3D4b  Cognitive Project Management Restriction Stipulations
       3D4b-i  Stipulation #1: Restriction Delay in Start & Finish Restrictions
       3D4b-ii  Stipulation #2: Restriction Delays are Never Negative
       3D4b-iii  Stipulation #3: Restriction Delay in Default Restriction
Dynamic Project Time Management Tools

3A: Introducing Dynamic Project Time Management Tools

As the word Dynamic suggests, Dynamic Project Time Management Tools are ones that seem to be alive.

To complete an earlier line of thought, under the Dynamic Project Time Management heading we find the fourth and last in the evolutionary lineage of Schedule Forms: the Network Schedule. That is: Listings, Timetables, Bar Chart, and now — Networks. And of all types of Network Scheduling techniques, the Critical Path Method (CPM) is far and away the most popular and most used in the Construction Industry.

3A1: Characterized by Energy or Effective Action

Dictionary.com provides three different definitions for the word Dynamic which, taken together, help us to develop a sense of how Cognitive Project Management applies the term in the context of Construction Schedules.

Dynamic: Pertaining to or characterized by energy or effective action; vigorously active or forceful.

This first definition begins to paint a picture. A Dynamic Project Time Management Tool has a discernible energy to it. It has the ability to react to conditions. That is to say, it doesn’t just lie there. It is the opposite of Static which, according to the same source, is “a fixed or stationary condition.” We reason that, if a Schedule can react, then it must also exhibit “effective action.”

A Dynamic Schedule is one that interacts with the Project Team, and contributes directly to Project Execution efforts. It presents information that changes with circumstances, and such information makes it possible for Project Participants to intelligently modify their behavior, if they so choose. In this sense, it is a force, which means that it has a “power to influence, affect, or control,” also per Dictionary.com.
3A2: Affected by the Passage of Time

Now consider a second definition of the word, Dynamic, this time as relates to the computer industry.

**Dynamic:** Affected by the passage of time or the presence or absence of power: For example, dynamic memory must be constantly refreshed to avoid losing data.

With this definition we add to our understanding of the term, Dynamic, as relates to Project Time Management. First, the Project Schedule is “affected by the passage of time.” As you will soon learn, a CPM Schedule, unlike its more primitive Static Schedule cousins (Listings, Timetables, and Bar Charts), must be maintained. That is, in order to remain relevant, current, accurate – and dynamic, the CPM Schedule as a simulation of Project Execution Strategy must constantly be calibrated to reflect the most current and realistic awareness (cognizance) of changes in any conditions that might affect downstream Project Execution. And just what are those conditions?

- The most immediate and obvious of all Project conditions is *prior progress*: progress in the achievement of the Project's Work Scope, to be more precise. In other words, the Schedule needs to be updated to show what Work has been accomplished and, by omission, what Work has not.

- Another condition we would want to consider (in order to keep the Schedule relevant and reflective of reality) might be the latest news on expected delivery dates and times for critical material and equipment. Likewise, Owners notoriously add to or take away from the Contractor’s Scope of Work, and these modifications to the builder’s Contract constitute changes, as well. You get the point.

- But the condition that you might not have thought of as being an all-important one and thus quite often falls off the radar screen altogether … is the mere Passage of Time itself. That is, if no Work is performed for an entire reporting period – yet Time has nevertheless moved along (as it forever does) – then the Project falls further behind Schedule nonetheless. We all remember our days back in school when, with each passing day that we ignored some major assignment, we fell farther and farther behind.

3A3: Affected by the Presence or Absence of Power

The second definition drops more colors onto the canvas of our understanding as to what a Dynamic Project Time Management Tool really means: “affected by ... the presence or absence of power.” Boy, does that say a lot!
If we interpret power as energy, then surely the Schedule (as a reflection of the status of the Project upon which it reports) is affected by the presence or absence of effort. And the effect is not a simple on-or-off measure; it’s always a matter of degree. Determining how much effort was (or needs to be) applied on the Project is as important to discern as whether or not any effort was required or actually exerted at all.

Of course there is also another way to interpret the word “power,” such as in that which motivates. As such, Management has power. The Owner has power. The Contract has power. Money has power. You’ve heard the expression: “when the cat’s away, the mice play.” Well, that same effect can be spotted (and even measured) when a Superintendent is known (through the grapevine) to be “off site.” Just watch the Work effort become more relaxed!

And whatever happens or doesn’t happen on the Project is reflected in the Schedule — that is, if the Schedule is a Dynamic one and if it is being maintained. If it is Static, as in a Performance Listing or a Timetable or a Bar Chart, it just lies there. But if it is a Dynamic Schedule, such as a CPM Schedule, it has the ability to react to changed conditions as long as they are made known to it.

You may be thinking that a Performance Listing can be updated too, as can a Bar Chart. One need only slide the Activity Bars to the left or right. Dates on a Timetable need only be changed. So, “what’s the difference,” you may be wondering, “between updating a Static Schedule and a Dynamic Schedule?”

The answer has to do with the functionality of the various Project Time Management Tools. Think of the difference between numbers that are typed into word-processing software (e.g., MS Word) as text, and the same characters typed into a number-crunching spreadsheet program (e.g., MS Excel) as numbers. Numeric characters are arithmetically operable; they can have math functions applied to them, whereas text characters cannot.

That is how a Dynamic Project Time Management Tool differs from its Static Project Time Management Tool cousin. In a Dynamic Project Time Management Tool, Activity Durations can be adjusted according to certain algorithms and, as a result, Activity Start and Finish Dates can be automatically calculated. As well, Total Float is the result of other Arithmetic Formulas that subtract Earliest Dates from Latest Dates. We will discuss all of this in far more detail in Chapter Five, Calculating Primary CPM Dates.

Motivates or Affects Development

Now, let’s look at a third definition of the word, Dynamic, from Dictionary.com.
Dynamic: A basic or dynamic force, especially one that motivates or affects development or stability, etc.

Man, do we love this one! We especially love the word *motivates*, since ICS-Global is the creator of Momentology, which deals with the Management of Momentum on Projects. Plus, it speaks to a belief that is central to Cognitive Project Management, that the most effective way to manage a Project is by *motivating* Project Participants rather than attempting to *control* them. This belief argues that, in a very literal sense, there is no such thing as Project Control, any more than there is Cost Control or Schedule Control or, for that matter, even Momentum Control.

We say this because, when all is said and done, it is up to the individual worker whether or not he or she does a great job or a sloppy job, whether he or she works aggressively or lazily, or whether or not the collective efforts of a group of individuals work together as a team or competitively as opponents. We believe that the most important role of every Project Manager (and why we refer to them as Project Coordinators) is to influence worker attitude and behavior.

Said differently, the most effective Project Managers *motivate* – they don’t command. Command-and-Control as a management philosophy is out; motivation and influence are in. Now look at the latter half of that third definition of the word, Dynamic:

Dynamic: … affects development or stability.

Wow! Doesn’t that get right to the heart of what Project Time Management is all about? The ultimate goal and central point of Project Time Management is to “affect development” of the Project Scope! The Schedule, as a Project Time Management Tool, begins as a reflection of the collective wisdom and best intentions of the Project’s leadership. It quickly converts into a statement of an agreed-upon Project Execution Strategy. This is why Cognitive Project Management considers a Project Schedule to be a Project Execution Model.

The Project Schedule is used on a daily basis to orchestrate the efforts of countless Project Participants. The process of developing a Project – that is, the conversion of a designer’s sketched drawings into the ever-so-real bricks and carpets of the final product – plays out through adherence to a meticulously conceived and mutually agreed-upon Sequencing of Activities as depicted in the Project Schedule.

Such orchestration of coordinated effort requires more than a mere Series of Actions; it requires that the Project Participants understand how their Actions build upon one another in a Progressive manner. Awareness of this interconnection among Project Participants is crucial for effective project management.
Actions (and, therefore, between a Schedule's Activities) is essential to effective Project Time Management.

3B: Activity Relationships: Two VERY Different Perspectives

At the end of the previous chapter, we mentioned that Static Project Time Management Tools rarely convey the interconnections of Activities with any real clarity or usefulness, and that the depiction of how Actions on a Project relate to one another is best left for Dynamic Project Time Management Tools.

We now find ourselves at a critical crossroads in our discussion of Project Time Management Tools. We are about to delve deeply into something called Relationships. Relationships are what give human endeavors, such as Project Execution efforts (and the Schedules that seek to model them), their Dynamic quality.

So, what exactly are Relationships?

Our answer to this question will not only clarify terminology; it will constitute one of the more glaring examples of how Cognitive Project Management differs from Dominant Project Management with respect to Project Time Management. As we introduce these new ideas, we must also contrast them with the status quo, thus making for a rather challenging writing assignment. For, you see, we have two simultaneous goals with this book:

☐ We want to equip you, the reader, with the ability to immediately function (i.e., “fit right in”) with Dominant Project Management Ideology, Methodology, and Technology; so that you will be completely productive right out of the chute.

☐ Yet we also want you to be an agent for change; someone who will advance Project Time Management and the broader world of Project Management by introducing better ways of achieving effective Project Time Management.

Nowhere is our writing challenge greater than when it comes to the term, Relationship. This is because the viewpoints of the two Project Management models could not be more different when it comes to how Activities in a Schedule, or their corresponding Actions on the Project, actually relate to one another.

The next paragraph offers a Nutshell Summary of Cognitive Project Management's perspective on Relationships. Most of the strange terms in the next paragraph have not yet been introduced, so don't fret if the concepts in the paragraph don't make complete sense; we will spend many pages explaining them soon enough. But we offer this Nutshell Summary anyway, so that you may begin to appreciate just how differently Dominant Project Management and Cognitive Project Management understand the meaning of the term, Relationship.
In a nutshell, and as Figure F0302 illustrates, Cognitive Project Management asserts that there are three categories of Relationships between Activities in a Project Schedule (not just one): Communal, Symbiotic, and Progressive. Additionally, Progressive Relationships can be further distinguished by the immediacy of their influence on their downstream counterparts: that is, whether they are Immediately-Restricting or Eventually-Impacting.

By contrast, Dominant Project Management only recognizes the Dependent Relationship, the term they use as equivalent to Cognitive Project Management's Immediately-Restricting Progressive Relationship.

3B1: Relationship, Beyond Dependence

The following discussion, which seeks to briefly explain Cognitive Project Management’s take on the different Relationships between Activities in a Schedule is intentionally summary. The three Relationship Categories identified by Cognitive Project Management constitute a very complex set of topics, requiring a correspondingly in-depth treatment in order to be fully understood.
However, such a treatment would exceed the scope of this book, which is limited to the mechanics of the Critical Path Method. For this reason, and anticipating that the following cursory discussion may not answer all of your questions, we encourage you to read ICS-WHITE PAPER, WPA-KL-13, COGNITIVE'S THREE RELATIONSHIP CATEGORIES.

Now let's explain that Nutshell Summary. We begin by pondering whether the term, Dependency, is synonymous with the term Relationship. We say, “No!” While we agree that a Dependency, as Dominant Project Management uses the term, is one category of Relationships – we contend that there are other Relationship Categories that have nothing to do with Dependence. For example, I can be related to my brother (through common parents) without either of us necessarily being dependent on the other.

Dependence has to do with “being conditional or contingent on something,” according to Dictionary.com. But can Activities in a Schedule be related to one another besides through explicit Logic Ties? For instance, is there a possible Relationship between a Flooring Contractor who lays carpet and sets floor tile and a Landscaping Contractor who plants trees and lays sod in the Project's final weeks — even though it is quite unlikely that either is Logically-tied to the other? That is the question we are asking.

3B2: Cognitive's Three Relationship Categories

Cognitive Project Management makes the bold assertion that all Actions on a Project, and thus all Activities in a Project Schedule are related to one another, even if they are not dependent on one another. [1]

Cognitive Project Management has identified three Relationship Categories, as shown in Figure F0302. While the following definitions, from the ICS-Dictionary, may not make a whole lot of sense (unless you read the referenced ICS-WHITE PAPER, WPA-KL-13), they nonetheless underscore a central point — that the Dependent Relationship of Dominant Project Management dogma correlates with only a discrete subset of Cognitive Project Management's Progressive Relationship. Here are the definitions:

- **Communal Relationship**: A Communal Relationship is formed between two Activities if they share one or more common, Project Performance Determinants. A Progressive Relationship is one of three Relationship Categories found in a CPM Schedule.

- **Symbiotic Relationship**: A Symbiotic Relationship between two Activities is one in which the performance of each Activity may (or may not) have an impact/effect on the conditions under which the other Activity must perform. The potential to impact or affect the other Activity's operational conditions exists in both directions.

1 This means that Activities can be related to one another even if they are not connected to each other by way of Logic Ties. All of this is fully explained in ICS-White Paper, WPA-KL-13.
Progressive Relationship: A Progressive Relationship exists between two Activities that are linked together by way of Performance Restrictions (and possibly additional intervening Activities). In a Progressive Relationship, the timing of a downstream Activity is or will be affected by the timely performance of one or more upstream Activities.

There are two subcategories of Progressive Relationships: Eventually-Impacting and Immediately-Restricting. In an Immediately-Restricting Progressive Relationship, the Activity-Pair is separated at most by two Performance Linkages. In an Eventually-Impacting Progressive Relationship, the Activity-Pair is separated by both one or more Activities, and by two or more Restriction Linkages that string all Activities into a Progressive series. All of this will be explained in much greater detail, below.

Dominant Project Management uses the interchangeable terms Relationship or Dependency to refer to what Cognitive Project Management calls an Immediately-Restricting Progressive Relationship.

The four varieties of Performance Restrictions (called Restriction Linkages) are Default Restriction (Finish-to-Start), Start Restriction (Start-to-Start), Finish Restriction (Finish-to-Finish), and Holdback Restriction (Start-to-Finish).

3B2a: Communal Relationships

The central feature of a Communal Relationship between two Actions on a Project (and, thus, between their corresponding Activities in the Schedule) is that they share a common Project Performance Determinant that is in limited supply. It is the limitedness of the Performance Determinant that creates a predictable and ever-present tension in Communal Relationships. Such tension need not be a bad thing but, that said, tension is tension.

Performance Determinant: A Performance Determinant is a Project Performance Variable that is shared by Activities in a Communal Relationship. Examples of Project Determinants include information (e.g., timely answer to an RFI), participants (e.g., access to Owner, PM, etc), work space (e.g., access to job site or limited work area), and resources (e.g., labor, materials, equipment).
On a typical Construction Project, there are some Performance Determinants that are always in limited supply. Take, for instance, the Owner; there is only one Owner. There is only one Architect. There is only one General Contractor, and it has only one Project Manager. There is only one job site. There is only one neighborhood surrounding the job site.

And of all Performance Determinants in short supply, there is only one window of opportunity in which to perform the Project. All members of the Project Team must work together to use this limited amount of Time as wisely and harmoniously as possible. What we have just described is a Zero Sum Game situation.

In Gaming Theory, there is this concept called the Zero Sum Game which, simply stated, says that one participant's gain or loss is exactly balanced by the losses or gains of the other participants. In order for the General Contractor to make money, the Owner must let go of money. Likewise, the financial gains of the Subcontractor come at the expense of the General Contractor.

Zero Sum Game also relates to the wisest use of Time (aka, Project Time Management) where, even if a set of Actions as a whole have more than enough Time to make a downstream Deadline Milestone, if any upstream Action in the series is too slow to get going, then all of the downstream Actions will be stripped of that squandered Time, and might even be considered “behind Schedule!”

It should be obvious that every Activity in the Schedule, just as every Action on the Project, has a Communal Relationship with every other Activity or Project Action. The significance of Communal Relationships will be left for the end of Chapter Eight as well as other ICS-Compendium volumes. For now, though, simply know that this category of Relationships exists between Activities, even between Activities that are not linked to one another by way of Logic Ties.

3B2b: Symbiotic Relationships

A second Relationship Category among Activities in a Schedule that has been identified and formalized by Cognitive Project Management is called the Symbiotic Relationship. A Symbiotic Relationship is one in which, as a consequence of downstream Milestones that are common to certain Activities, behavior by one Activity influences the behavior of other Activities sharing the Symbiotic Relationship. Symbiotic Relationships help us to understand the role of human judgment and human reaction in how things do (or do not) get done on a Construction Project.

Symbiotic Relationships are extremely common on Construction Projects. Keep in mind that most Project Actions are part of a larger Project Execution Strategy, agreed to both collectively and individually by the Project Team. This Project Execution Strategy is
intended to ensure achievement of mutually agreed-upon goals. When one party sees that the other party will be late in arriving at a goal, the observing party quite often adjusts its own effort accordingly.

Two frequently observed examples of Symbiotic Relationships in action on Construction Projects are Cognitive Balance and Concurrent Delay. Under Cognitive Balance, one party anticipates the behavior of the other party (rightly, or wrongly) and ‘reacts’ – preemptively. In a Concurrent Delay situation, each party interprets the other party's Actions as justification for adjustments to its own Actions.

Are you clear on how a Communal Relationship differs from a Symbiotic Relationship? A Communal Relationship refers to one in which two parties are competing for access to a common, limited Project Performance Determinant. Because of their Zero Sum Game nature, the identifying characteristic of a Communal Relationship is that it often encumbers the other Relationship member’s ability to perform. By contrast, a Symbiotic Relationship is one in which one Relationship member’s behavior ultimately influences the other Relationship willingness to perform.

To Recap: In a Communal Relationship one member’s behavior affects the other member’s ability to perform. In a Symbiotic Relationship, one member’s behavior affects the other member’s willingness to perform. Likewise, Communal Relationships are ones that unequivocally exist on every Project, and evidence of their existence can be found in every Project Schedule. Contrast this with Symbiotic Relationships which may or may not exist on a Project, or be apparent in the Project Schedule.

3B2c: Progressive Relationships

This brings us to the third Relationship Category, the Progressive Relationship. In Progressive Relationships the performance of a downstream Activity is necessarily restricted by the prior performance of one or more upstream Activities to which it is connected. Progressive Relationships are expressed through the use of Performance Restrictions.

Here is the ICS-Dictionary definition of the term Performance Restriction:

Performance Restriction: Performance Restriction is the name that Cognitive Project Management has given to the four possible ways that Activities can be dependently linked to one another in Progressive Relationships within the Critical Path...
Method. Each distinct configuration is called a Restriction Linkage. The four Restriction Linkages of Cognitive Project Management correspond to the four Dependency Types (or Relationship Types) of Dominant Project Management: Default Restriction (Finish-to-Start Dependency), Start Restriction (Start-to-Start Dependency), Finish Restriction (Finish-to-Finish Dependency), and Holdback Restriction (Start-to-Finish Dependency).

Now let us take a moment to get clear on how Dominant Project Management uses the term, Dependent, and why Cognitive Project Management has elected to substitute it with a new term, Performance Restriction (or, Restriction, for short).

**Understanding the Term, Dependent**

According to Dominant Project Management the term, Dependent, only applies to two Activities that are *directly* connected to one another by way of a Logic Tie. For instance, consider a string of three sequential Activities, such that Activity A precedes Activity B, and Activity B precedes Activity C. Pursuant to Dominant Project Management, Activity C would be considered as being dependent on Activity B ... but *not* dependent on Activity A. By contrast, Cognitive Project Management contends that any Activity in a Network Diagram is dependent upon *every* preceding Activity to which it is Logically tied, *directly* or *indirectly*.

**Understanding the Term, Progressive**

Given these two very different understandings of what constitutes a Dependency between Activities linked by Logic Ties in a Schedule, the ICS-Compendium Development Team decided to coin the expression Progressive Relationship. The word Progressive is defined by Dictionary.com as follows:

**Progressive:** Proceeding or progressing by steps or degrees.

We feel that the word Progressive well describes the nature of Project Executions; that the Actions of a Project (and, thus, the Activities of the Schedule) build upon one another; that the Work of the Project *progresses*, if you will. The manner in which the discrete Actions of the Project build upon one another is reflected in the Schedule by way of Logic Ties, which define and clarify the Dependencies of Activities, one to the other.

**3B2d: Introducing the Term, Restriction**

The other semantic innovation was to select the term Restriction in substitution for the decades-old Dependency. We admit that Dominant Project Management is surely not wrong to describe the interconnections of Activities with the word Dependency, the
Successor Activity being *dependent* upon the Predecessor Activity.

But Cognitive Project Management prefers another term that goes beyond mere Dependency, one that highlights the real-life consequence of mutually-made Commitments and promises between Project Team members: Restriction.

Here is the ICS-Dictionary definition:

**Restriction:** According to Cognitive Project Management, any Immediately-Restricting Progressive Relationship connects two Activities such that the performance of the downstream Activity is to some measurable extent delayed by the prior performance of the upstream Activity. This delaying effect is called a Restriction, and the extent of the Restriction is numerically expressed as a Restriction Delay.

### 3B2d-i: Understanding the Four Restriction Objectives

You may be wondering why we felt the need to substitute the word Dependency with Restriction; after all, aren't we just splitting hairs over meaning? Our answer to this question begins by noting a growing trend in Dominant Project Time Management circles to more fully categorize and codify the various reasons why Relationship Linkages are drawn between Activities of a Schedule.

Until only a few years ago, Logic Ties were thought of as belonging to at most two broad categories: Hard Logic and Soft Logic. Hard Logic referred to Linkages between Activities that were *absolutely necessary*, whereas Soft Logic was always to some degree subjective or preferential.

To explain Hard Logic, in each of the following examples the downstream Activity is necessarily Restricted by some amount of performance of the upstream Activity.

- **Example #1:** Hang Drywall ► Tape/Spackle/Sand Drywall ► Paint Wall
- **Example #2:** Place Concrete Floor ► Erect Concrete Block Walls
- **Example #3:** Pave Roadway ► Paint/Stripe Roadway

Soft Logic, by contrast, has a subjective aspect to it; that is, there may be more than one way to perform the Work. The Project Team settles on an agreed-upon approach, and it is this approach that is formalized in the Schedule. Depending on the choice made by the Project Team, the string of Logic could go more than one way. For instance:

- **Example #4:** Install Flooring, 2nd Floor West ► Install Flooring, 2nd Floor East
- **Example #5:** Install Flooring, 2nd Floor East ► Install Flooring, 2nd Floor West
Cognitive Project Management contends that the range of reasons for inserting Logic Ties between Activities in a Schedule can be categorized under four Restriction Objectives. According to the ICS-Dictionary:

- **Mandatory Restriction Objective**: One of four Performance Restriction Objectives, a Mandatory Restriction is inserted into a Schedule to reflect a mandatory sequencing requirement. For example, a contract may require the completion of Phase II before Phase III commences.

- **Natural Restriction Objective**: One of four Performance Restriction Objectives, a Natural Restriction is inserted into a Schedule to reflect a particular sequencing of Activities as required by the very nature of the Work being performed. For instance, it is impossible to paint a wall before the wall is first built.

- **Practical Restriction Objective**: One of four Performance Restriction Objectives, a Practical Restriction is inserted into a Schedule to reflect what is deemed by the Project Team (those whose input establishes the content of the Schedule) to be the most practical approach to the Work. An example of a Practical Restriction is the decision to complete floor finishes from the top down in a multi-story structure. Finishing bottom-up would create a situation where dirt and debris of upper floors would be "swept" onto lower, already-completed floors (an impractical scenario).

- **Logistical Restriction Objective**: One of four Performance Restriction Objectives, a Logistical Restriction is inserted into a Schedule to reflect logistical considerations of Project Execution, including: the procurement, supply, and maintenance of equipment; the acquisition, deployment, and movement of personnel; the availability of and access to work spaces, and so forth.

To be sure, while there may be four Performance Restriction Objectives, only one of them actually depicts a truly Dependent Relationship: the Natural Restriction. Who would disagree with the assertion that one cannot paint a wall before the wall is constructed? In this sense, the PAINT WALL Activity is dependent on the CONSTRUCTION WALL Activity.

But when we look at the other three Restriction Objectives, the downstream Activity is not actually prevented from performing its Work. Instead, it is simply restricted from proceeding, in deference to an arbitrary agreement among the parties to the Project (and contributors to the Schedule).

When you take the word “Dependency” literally, except for Natural Restrictions, are any of the other “downstream” Activities absolutely dependent on the upstream Activities in order to perform their Work? Are they truly being barred from commencing? We think not.
And yet all four Restriction Objectives reflect self-imposed commitments made by the members of the Project Team to one another. Once incorporated into the Schedule, these Commitments constitute binding Performance Restrictions.

3B2d-ii: Comparing Dominant and Cognitive Relationship Terminology

So now you know why Cognitive Project Management decided to retire the word Dependency, and replace it with the more descriptive word, Restriction. As a helpful guide, here are substitution terms we will use throughout the ICS-Compendium:

<table>
<thead>
<tr>
<th>Dominant Project Management</th>
<th>Cognitive Project Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationship</td>
<td>Performance Restriction</td>
</tr>
<tr>
<td>Dependency</td>
<td>Immediate Restriction</td>
</tr>
<tr>
<td>Dependent Upon</td>
<td>Restricted by</td>
</tr>
<tr>
<td>Dependent Activity</td>
<td>Restricted Activity</td>
</tr>
<tr>
<td>Successor Activity</td>
<td>Restricted Activity</td>
</tr>
<tr>
<td>Predecessor Activity</td>
<td>Restricting Activity</td>
</tr>
<tr>
<td>Relationship Type</td>
<td>Restriction Linkage</td>
</tr>
<tr>
<td>Leads and Lags</td>
<td>Restriction Delays</td>
</tr>
<tr>
<td>Finish-to-Start Dependency</td>
<td>Default Restriction</td>
</tr>
<tr>
<td>Start-to-Start Dependency</td>
<td>Start Restriction</td>
</tr>
<tr>
<td>Finish-to-Finish Dependency</td>
<td>Finish Restriction</td>
</tr>
<tr>
<td>Start-to-Finish Dependency</td>
<td>Holdback Restriction</td>
</tr>
</tbody>
</table>

Consistent and encompassing use of the word Restriction reduces confusion. Notice how Restriction replaces Dependency, Leads, Lags, Predecessor, Successor, and Relationship. The term, Restriction, also suggests to us a different perspective — a different way for us to understand how Activities in a Network-Based Schedule really interrelate to one another. Where the word Dependency describes the nature of the Relationship, the word Restriction describes the effect of the Relationship.

This substitution of terms also fits well with our category, Progressive Relationship. For if we allow it to, the term Restriction Type can be used to differentiate between Immediately-Restricting and Eventually-Impacting; to distinguish between Activities that are immediately prior to the Restricted Activity, and those that are more distantly linked to the Restricted Activity.
Referring back to Figure F0302 on page 74, do you see the second bullet within the Progressive box, Immediately-Restricting? Well, that is what Dominant Project Management refers to as a Dependent Relationship. The Dependent Relationship per Dominant Project Management is the same as an Immediately-Restricting Relationship according to Cognitive Project Management.

We hope this clears up any confusion. Again, as stated at the outset of this section, if you still have questions about the three Relationship Categories, we strongly encourage you to set this book down, and take a few minutes to read **ICS-WHITE PAPER, WPA-KL-13, COGNITIVE’S THREE RELATIONSHIP CATEGORIES**. We are confident that it will answer all of your questions.

Given this lengthy clarification, we now state rather unequivocally that throughout this book, as well as throughout the entire ICS-Compendium, whenever we use the term Restriction by itself, unless clarified otherwise, we mean it to refer to an Immediately-Restricting Relationship within the Progressive Relationship category.

**3C: From Activities to Logic Diagrams**

Throughout the balance of this chapter we will focus on Immediately-Restricting Progressive Relationships which, to repeat, are referred to as Dependent Relationships in Dominant Project Management circles.

Elsewhere in this book, and within the ICS-Compendium, we will discuss Eventually-Impacting Restrictions.

**3C1: The Terminology of Performance Restrictions**

At last, we are in a position to introduce an important new term: Restriction Linkage, which refers to the interconnection between two Progressively-Related Activities in a Schedule. We trust you have not forgotten that a Schedule’s Activities, and the Performance Restrictions between them, are never more than artificial models of real-life Actions on the Project and the Interdependencies between them, respectively. Thus, pre-planned choreography among Activities on the Project are replicated by directly-tied Restriction Linkages in the Schedule.

The defining feature of a Dynamic Schedule, one that lives and breathes, is the interconnections between the Schedule’s Activities by way of traceable Restriction Linkages that bind the Activities together. In so doing, these Restriction Linkages precisely
define the Performance Restrictions that give the Schedule its Dynamic functionality.

As Figure F0304 shows, when a Schedule is depicted graphically using the Precedence Diagramming Method (PDM) the Logic of a Restriction Linkage is shown as an Arrow, while the two Activities that share a Performance Restriction are represented by simple boxes filled with meaningful text. This Arrow, that Logically connects Activities, is known in Scheduling circles as a Logic Tie.

The precise nature of an Immediately-Restricting Progressive Relationship between two Activities is reflected in what Cognitive Project Management refers to as a Restriction Statement. Restriction Statements have five essential components, as follows:

- **Restricted Activity**: In an Immediately-Restricting Progressive Relationship between two Activities, performance by the Restricting Activity inhibits performance of the Restricted Activity. Dominant Project Management refers to a Restricted Activity as a Successor Activity. The Restricted Activity is one of five components of a Restriction Statement.

- **Restriction Linkage**: What Cognitive Project Management refers to as a Restriction Linkage, Dominant Project Management calls a Relationship/Dependency Type. In an Immediately-Restricting Progressive Relationship where a Performance Restriction exists between two Activities, there are four possible Restriction Linkages: Default Restriction (Finish-to-Start Dependency), Start Restriction (Start-to-Start Dependency), Finish Restriction (Finish-to-Finish Dependency), and Holdback Restriction (Start-to-Finish) Dependency. Restriction Linkage is one of five components of a Restriction Statement.

- **Restriction Delay**: What Cognitive Project Management refers to as a Restriction Delay, Dominant Project Management somewhat interchangeably calls Leads and Lags. A Restriction Delay is a numeric value that expresses the delaying effect of an Immediately-Restricting Progressive Relationship. Restriction Delay is one of five components of a Restriction Statement.

---

2 PDM, and its ancestral counterpart, ADM (Arrow Diagramming Method), offer two different ways to graphically depict Schedule Logic. ADM is rarely used anymore; so, this book employs PDM.
**3 Dynamic Project Time Management Tools**

**Restriction Flow:** In an Immediately-Restricting Progressive Relationship where a Performance Restriction exists between two Activities, Restriction Flow refers to the direction of the Performance Restriction. Graphically, it is typically depicted by an Arrow (Logic Tie) that begins at the Restricting Activity and ends at the Restricted Activity. In tabular format, Restriction Flow is indicated by cross-referencing the Restricting Activity with the Restricted Activity. Restriction Flow is one of five components of a Restriction Statement.

A Restriction Statement may be written out in narrative form as a carefully worded sentence, or it may be stated as an arithmetic expression.

Here is the Restriction Statement written as an arithmetic expression that explains the Performance Restriction shown in Figure F0306:

**Activity A** ➸ **Activity B, SS:2**

Here is a same Restriction Statement, this time written in narrative format:

**Activity B** (Restricted Activity) cannot start (Restriction Flow) until two days (Restriction Delay) after **Activity A** (Restricting Activity) has started (Restriction Flow).

Notice that the Logic Code (e.g., SS:2) follows an alphanumeric convention whereby the alpha portion identifies which particular Restriction Linkage is involved, while the numeric portion describes the Restriction Delay.

**3C2: Activity Path: Through Immediately-Restricting Ties**

It is time to pull together all that we have discussed so far about Immediately-Restricting Progressive Relationships. Let’s introduce some terms related to the bundling of Activity clusters linked together by way of Performance Restrictions.

At its most basic, when a Series of Activities are strung together using Logic Ties between them, the resultant set of Sequentially-Related[^3] Activities is called an Activity Path

[^3]: In using the word “sequentially,” we do not mean to imply only Restrictions that are entirely linear, one after
(Path, for short) — as Figure F0308 illustrates. Later we will discuss Activity Paths in much more detail (and even introduce a new concept, invented by Cognitive Project Management, called Activity Path Segments).

Again, Activities that are sewn together with Logic Ties (that depict Restriction Linkages) create Activity Paths. When you group a number of Activity Paths together, based on some aspects of the Project that they have in common, this cluster of Activity Paths forms a major portion (but not all) of a Network Diagram. We call this grouping a Subnetwork (Subnet, for short). A typical Subnet is illustrated in Figure F0310.

Finally, we call the entire interconnected set of all Activities, which is comprised of one or more Subnetworks, a Network Diagram (Network for short). This is shown in Figure F0312. So, working backwards:

- At its most encompassing, we have a Network Diagram, or Network (Figure F0312).
- The Network Diagram is divided into major groupings, one of which we would

the other (as with the Default Restriction Linkage; the Finish-to-Start Dependency). The word, Sequential, also includes Activities that may be overlapped -- that is, following one another, but in more of a cascading, overlapped fashion -- thus, only partially Restricted.
call a Subnetwork, or Subnet\(^4\) (Figure F0310).

- A Subnet is divided into multiple Activity Paths (Figure F0308).
- An Activity Path is comprised of Activities that are linked in an Immediately-Restricting manner, with the details of the Performance Restriction expressed graphically through Logic Codes and Logic Ties (Figure F0306), and verbally through a Restriction Statement.

As we close this topic, let’s be clear about something: a Network Diagram and a Schedule are not the same thing. We say this for two reasons:

- **Other Schedule Formats:** A Schedule may be communicated in many different forms, such as a Listing, a Timetable, a Bar Chart, or – you guessed it – a Logic Diagram (a Network Diagram).\(^5\) That’s the distinction understood by the general Project Management community.

- **Dates Not Mandatory:** But among Project Facilitators, there is a more technical differentiation. A Logic Diagram does not (necessarily) have Calculated Dates associated with the Activities, whereas a Schedule always does. In other words, a Logic Diagram could be as simple as an elaborate process map, or flowchart, that shows the Project Execution Strategy as a series of Activities to be performed;

\(^4\) Subnets are also called Fragnets.

\(^5\) The terms, Network Diagram and Logic Diagram, are not entirely synonymous although they tend to be used interchangeably throughout the Project Time Management community. If we take them literally, the former is a diagram that depicts a Network of some interconnected things. In the world of Scheduling, those “things” are called Activities, and they are connected by Logic Ties. The other term, taken literally, emphasizes the rationale behind the diagram, it being a logical one. That is, the diagram reveals a Logical thinking process. But again, in common practice the two terms are used interchangeably.
inclusion of Calculated Dates would be optional. By contrast, a Schedule would be essentially useless without Calculated Dates, since it is a Project Time Management Tool.

3D: CPM’s Four Restriction Linkages

So far in this chapter we have held three major discussions:

☐ **Relationships:** We introduced this thing called the Relationship, and explained the contention of Cognitive Project Management that all the Activities in a Schedule are related to one another by way of one or more of three Relationship Categories: Communal, Symbiotic, and/or Progressive. We further noted that Dominant Project Management seems unaware of the first two, and instead applies the terms Relationship or Dependency exclusively to a subset of the third category, the Progressive Relationship.

☐ **Immediately-Restricting:** We noted that, while any Activity is restricted, to some degree, by the timely performance of all preceding Activities to which it is tied Logically in a Progressive Relationship, Dominant Project Management literature and teachings apply the term Dependency only to the limited condition where two Activities are directly connected to one another (with no other Activity intervening). Hence, the Immediately-Restricting Progressive Relationship of the Cognitive Project Management model is the same as the “dependent” Relationship of Dominant Project Management.

☐ **Depiction of Logic:** As we examined the nuances of a single Immediately-Restricting Progressive Relationship, we saw (a) how Logic is depicted as Activities sewn together with the thread of Logic Ties, (b) how Logically-strung Activities become Activity Paths, (c) how Activity Paths comprise Subnetworks, and (d) how Subnetworks collectively constitute a Network Diagram.

We are now about to introduce the four Performance Restriction Linkages – this time with the intent of dissecting and learning how each one actually works — in the Critical Path Method of Modeling a Project Execution Strategy. We must warn you now that this final section of **CHAPTER THREE** will likely be long. It most certainly will be a bit confusing at first, so put on your thinking caps and ... let’s go!

3D1: Four Performance Restriction Linkages

As **Figure F0314** shows, there are four Performance Restriction Linkages[^6] used by the

[^6]: As explained earlier, what Cognitive Project Management refers to as Performance Restriction Linkages is known as Relationship/Dependency Types in the Dominant Project Management community.
Critical Path Method.\(^7\) For each Performance Restriction Linkage let us now provide a brief description, taken from the ICS-Dictionary:

<table>
<thead>
<tr>
<th>&quot;DEPENDENCY TYPE&quot; *</th>
<th>Finish-to-Start</th>
<th>Start-to-Start</th>
<th>Finish-to-Finish</th>
<th>Start-to-Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTRICTION NAME</td>
<td>Default Restriction</td>
<td>Start Restriction</td>
<td>Finish Restriction</td>
<td>Holdback Restriction</td>
</tr>
<tr>
<td>RESTRICTION ABBREVIATION</td>
<td>FS</td>
<td>SS</td>
<td>FF</td>
<td>SF</td>
</tr>
<tr>
<td>RESTRICTION IMPACT RANGE</td>
<td>Entire</td>
<td>Entire</td>
<td>Partial</td>
<td>Partial</td>
</tr>
</tbody>
</table>

* Per Dominant Project Management

\(^7\) More specifically, used by the Precedence Diagramming Method (PDM) variant of the Critical Path Method — and by both Dominant and Cognitive Project Management. The other CPM variant, Arrow Diagramming Method (ADM) only utilizes one Restriction Type: Finish-to-Start.
the Restricted Activity) — as opposed to the mere passage of time. The Finish Restriction corresponds to the Finish-to-Finish Relationship Type of Dominant Project Management. Its Restriction Abbreviation is "FF."

**Holdback Restriction:** The Holdback Restriction corresponds to the Start-to-Finish Relationship Type of Dominant Project Management. Its Restriction Abbreviation is "SF." Throughout Dominant Project Management circles there is no consistent or universal understanding as to how this Performance Restriction works. As best as we can tell, a Holdback Restriction seems to be a hybrid combination of both a Start Restriction and a Finish Restriction. That is, in a Holdback Restriction, a final portion of the Restricted Activity cannot be performed until the Restricting Activity has both commenced and progressed to a certain extent. While Cognitive Project Management does not encourage the use of the Holdback Restriction, it provides a name for it (Holdback Restriction) in the interest of thoroughness.

Unless you know how these four Performance Restrictions actually work, it is impossible to get the most out of CPM Schedules. The problem, as you shall soon see, is that Dominant Project Management is internally conflicted as to how these four Performance Restrictions should, or do, work. That is why this section will be a bit long and possibly confusing. We will do our best to keep the confusion to a minimum.

Since we have only briefly described the four Performance Restrictions used in the Critical Path Method of Modeling, we must now do two things — in order clarify how they are differently understood between Dominant Project Management and Cognitive Project Management schools of thought.

☐ First, we will try our best to bring you the popular understanding of these concepts. You will come to realize, as we have, that the lack of consistency in definitions, as well as in Scheduling Software formulas, has created a royal mess. Worse, though, it has provided the fodder for various ‘solutions’ to the problem -- fixes that, in our opinion, have all too often only made matters worse.

☐ We will present Cognitive Project Management's view of the four Performance Restrictions and offer our definitions for each. We will also include mention of certain practical policies that ICS-Global imposes on all Project Time Management consultants working under the “ICS” banner, as these may be useful to you and your constructor organization.

**3D2: Performance Restriction's Three Key Attributes**

Cognitive Project Management states that a Performance Restriction has three key attributes: Restriction Linkage, Restriction Delay, and Restriction Flow. The first two are expressed through a single alphanumeric Restriction Code.
The Restriction Code itself does not indicate Restriction Flow. Instead, Restriction Flow is implicitly communicated by virtue of where the Logic Code is physically placed.

- **In Graphics**: In a Logic Diagram, it is typically located adjacent to the Logic Tie itself.
- **In a Database**: In a computer database of Schedule Logic, it is stored as a data field within an Activity record.
- **In Schedule Output**: In an output from the Schedule (printout, on screen, tablet, etc.) it is on the same line as the Restricted Activity.

Let us consider each of the three key Performance Restriction attributes separately:

- **Restriction Linkage**: The Restriction Linkage denotes precisely how a Restricting Activity connects to its Restricted Activity counterpart (in an Immediately-Restricting Progressive Relationship). Restriction Linkage is expressed as a Restriction Abbreviation that appears as the *alpha portion* of the Restriction Code. For example, if the Restriction Linkage is a Default Restriction, then the Restriction Abbreviation would be “FS.” See Figure F0314.

- **Restriction Name**: Cognitive Project Management has further facilitated verbal communication by introducing intuitive (and much shorter) names for the four Restriction Linkages. Accordingly, throughout the ICS-Compendium, including this text, we will use the Restriction Names shown in Figure F0314.

- **Restriction Delay**: The numeric portion of the Restriction Code expresses the delaying effect of the Performance Restriction. Example: “3” in “FF:3” Just where this delaying effect takes place depends on the particular Restriction Linkage. Cognitive Project Management's interpretation of the Restriction Delay value differs with each Restriction Linkage:
  - In a Start Restriction it represents the amount of Time required for the Restricting Activity to perform some early portion of its Work. The delaying effect is to the start of the Restricted Activity, as it waits for the Restricting Activity to progress to a degree of completion indicated by the Restriction Delay value.
  - In a Finish Restriction it represents the amount of Time required for the Restricted Activity to perform the final portion of its Work. The delaying effect is to the final portion of the Restricted Activity, the final portion of...
which cannot be performed until the Restricting Activity has completed entirely.

- In an Default Restriction, the Restricted Activity commences upon completion of the Restricting Activity. The delaying effect is to the start of Restricted Activity.

> In the case of either the Start Restriction or Finish Restriction, the delaying period (which is indicated by the Restriction Delay value) represents a Work Performance period within the Restricting or Restricted Duration, respectively. By contrast, in the Default Restriction, the delaying period refers to a mere Passage of Time (on a clock or calendar), and it does not represent any portion of the Activity Duration of either the Restricting Activity or the Restricted Activity.

**Restriction Flow:** Not part of the Logic Code, but surely a part of the Restriction Statement, Restriction Flow indicates the direction in which the Restriction plays out between two Activities in a Progressive Relationship. Like any statement of direction, there must be a Point of Origin and a Point of Termination. In the Critical Path Method of Modeling a Project Execution Strategy, the Point of Origin of a Performance Restriction is at the Restricting Activity, and the point of Termination is at the Restricted Activity. Since Restriction Flow is not expressed within the Restriction Code itself, it is instead indicated by where the Restriction Code is placed/written in the diagram. See *Figure F0316.*

- In a graphical representation of a Restriction Statement, Restriction Flow is depicted by an Arrow that begins at the Restricting Activity and ends at the Restricted Activity. The Arrow is commonly called a Logic Tie.
- In a tabular format, such as a computer printout, the Restriction Flow is indicated by locating the Restriction Code on the Restricted Activity record, and then referencing any Restricting Activities by their Activity Identifiers.

### 3D3: CPM Dependency Types: Traditional Clarifications

Let us begin by looking at some examples of how these terms are often defined or explained in Dominant Project Management literature. We found these definitions by performing a *Google* search of the respective terms. Let's take them one at a time.

The terms Dependency, Dependency Type, Predecessor and Successor belong to Dominant Project Management. Cognitive Project Management's counterpart terms are Performance Restriction, Performance Linkage, Restricting Activity, and Restricted Activity, respectively.
3D3a: Popular Definitions for Finish-to-Start Dependency Type

- **Definition 1:** B doesn't start before A is finished. [8]
- **Definition 2:** The predecessor must finish before the successor can start. [9]
- **Definition 3:** The dependent task (B) cannot begin until the task that it depends on (A) is complete. [10]
- **Definition 4:** The second task in the relationship can't begin until the first task finishes. [11]
- **Definition 5:** When a task can't begin until another task ends. [12]

9 Dummies.com at http://www.Dummies.com/how-to/content/how-to-determine-precedence-in-your-Projects-netwo.html
Definition 6: The “from” activity must finish before the “to” activity can start.\[13\]

Definition 7: The initiation of work on a succeeding schedule work activity is entirely dependent on the formal conclusion of work on a particular activity that directly precedes it.\[14\]

Our Commentary: For the most part, the above definitions tend to say the same thing: that the Dependent Activity cannot start before or until the predecessor Activity has finished. That’s it in a nutshell. And, to be sure, of the four Dependency Types, the Finish-to-Start Dependency (Cognitive’s Default Restriction) is the simplest to understand, the most common in use, and the most consistently defined.

Yet, for the sake of accuracy, we feel compelled to point out a technical inadequacy in all of the above definitions, except Definition 7. As we will discuss in greater detail in Chapter Five, Dominant Project Management doctrine does not explicitly prohibit the use of Leads or Lags\[15\] having negative numeric values. A negative Lead/Lag reflects a condition whereby the start of the Restricted Activity is allowed to commence at some Point in Time before completion of the Restricting Activity.

Cognitive Project Management joins many other leading Scheduling authorities in discouraging the use of negative Restriction Delays (Leads or Lags). We contend that it is difficult for the average person to get their head wrapped around the idea of a negative delay. We think this is about as intuitive as would be a “negative profit” in accounting! Isn’t that just a loss?

We acknowledge that the term Restriction Delay is unique to Cognitive Project Management, and not (yet, anyway) used commonly. Rather, the Dominant Project Management world still embraces either (or both) of the terms Lead and Lag, neither of which enjoys universal (or even majority) consensus as to what it actually stands for.

Some experts see Leads and Lags as being the same thing, while others have different definitions for each. To complicate matters further, some consider the Lead or Lag to include both the Restriction Linkage and the Restriction Delay (to use Cognitive Project Management terminology). Others, however, use Lead and Lag to refer just to the Restriction Linkage ... and Lead Value or Lag Value to refer to what we would call a Restriction Delay. In this book, we will use the term Restriction Delay.

---

13 University of California, through SAS Institute http://www.uc.edu/sashtml/orpm/chapa/index.htm
15 Leads and Lags are two terms that Dominant Project Management interchangeable uses to describe what Cognitive Project Management refers to as Restriction Delay.
In any event, it is not uncommon in Dominant Project Management practice to apply a Lead Value or Lag Value that is a negative number. Our point is that (except for Definition 7) the above definitions fail to account for negative Lead/Lag conditions.

**Definition 7** smartly describes the Dependency Type with the somewhat ambiguous words “dependent on,” thereby not insisting that the start of the Dependent Activity requires the prior completion of the Restricting Activity.

### 3D3b: Popular Definitions for Start-to-Start Dependency Type

- **Definition 8:** The predecessor must start before the successor can start. [16]
- **Definition 9:** The dependent task (B) cannot begin until the task that it depends on (A) begins. The dependent task can start at any time after the task that it depends on begins. The SS link type does not require that both tasks begin at the same time. [17]
- **Definition 10:** B can't start before A starts. [18]
- **Definition 11:** Used when the second task in the relationship can't begin until after the first task in the relationship begins. Start-to-Start dependencies don't require that both tasks start at the same time. They simply require that the first task has begun, in order for the second task to begin. [19]
- **Definition 12:** The start of the dependent task lags behind the start of the predecessor. [20]
- **Definition 13:** The “from” activity must start before the “to” activity can start. [21]
- **Definition 14:** For one project to begin, another must also be underway. [22]

**Our Commentary:** Now we are beginning to see more of a fracturing of opinions by recognized authorities on the topic of Project Time Management. As you read on, please appreciate the basic question we are asking, which is: as to the nature of the Relationship between the start of the Restricted Activity (Activity B) and the Restricting Activity (Activity A) … which is the correct interpretation of a Start-to-Start Dependency (Start Restriction)?

---

16 Dummies.com at http://www.Dummies.com/how-to/content/how-to-determine-precedence-in-your-Projects-network.html
20 The Complete Idiot's Guide to Project Management with Microsoft Project 2003
21 University of California, through SAS Institute http://www.uc.edu/sashtml/orpm/chapa/index.htm
22 Project Mgmt Knowledge http://Project-management-knowledge.com/definitions/f/Start-to-Start/
Interpretation A: The two starts must be at the same time.

Interpretation B: The two starts must have a gap of Time between them.

Interpretation C: The two starts may have a gap of Time between them.

Let’s take a look. Here is how the seven definitions weigh in on the question:

Supporting Interpretation A: None make this assertion.

Supporting Interpretation B: Definitions 8, 11, 12, and 13.

Supporting Interpretation C: Definition 9

Problematic Interpretations: The other two definitions remain unclear as to what they believe

- **Definition 10** doesn’t speak to whether a gap must or may occur. It only notes that the Dependent Activity cannot commence before the Restricting Activity commences.
- **Definition 14** is completely off track with its understanding that the Start-to-Start Relationship is between “Projects,” rather than Activities within a Project.

Of the three possible interpretations, Cognitive Project Management's view most closely aligns with Interpretation C. A Start-to-Start Dependency Type (Start Restriction) simply tells us that there is some Restriction between the start of the Restricting Activity and the start of the Restricted Activity. Only one of the seven definitions chooses this interpretation. Additionally, and like the Finish-to-Start definitions, these definitions fail to allow for negative Restriction Delays, which are how the Start-to-Start Dependency is often applied in Dominant Project Management.

Cognitive Project Management strongly discourages use of negative Restriction Delays.

### 3D3c: Popular Definitions for Finish-to-Finish Dependency Type

- **Definition 15**: The predecessor must finish before the successor can finish.  

- **Definition 16**: The dependent task (B) cannot be completed until the task that it depends on (A) is completed. The dependent task can be completed at any time after the task that it depends on is completed. The Finish-to-Finish link type does

23 Dummies.com at http://www.Dummies.com/how-to/content/how-to-determine-precedence-in-your-Projects-netwo.html
not require that both tasks be completed at the same time.\[^{24}\]

\* Definition 17: B can't finish before A is finished.\[^{25}\]

\* Definition 18: Don't require that both tasks be completed simultaneously. They simply require that the first task be finished, in order for the second task to finish. The second task can finish any time after the first task finishes.\[^{26}\]

\* Definition 19: The dependent task can’t finish until the predecessor task finishes.\[^{27}\]

\* Definition 20: The “from” activity must finish before the “to” activity can finish.\[^{28}\]

\* Definition 21: The succeeding activity cannot finish until the preceding activity has ultimately been concluded.\[^{29}\]

Our Commentary: All of what we said under about the seven Start-to-Start Relationship (Start Restriction) definitions applies likewise with respect to the above Finish-to-Finish Relationship (Finish Restriction) definitions.

3D3d: Popular Definitions for Start-to-Finish Dependency Type

\* Definition 22: B can't finish before A starts.\[^{30}\]

\* Definition 23: The predecessor must start before the successor can finish.\[^{31}\]

\* Definition 24: The dependent task (B) cannot be completed until the task that it depends on (A) begins. The dependent task can be completed at any time after the task that it depends on begins. The Start-to-Finish link type does not require that the dependent task be completed concurrent with the beginning of the task on which it depends.\[^{32}\]

---


\[^{27}\] The Complete Idiot's Guide to Project Management with Microsoft Project 2003

\[^{28}\] University of California, through SAS Institute http://www.uc.edu/sashml/orpm/chapa/index.htm

\[^{29}\] Project Management Knowledge at http://Project-management-knowledge.com/definitions/f/Finish-to-Finish/


\[^{31}\] Dummies.com at http://www.Dummies.com/how-to/content/how-to/determine-precedence-in-your-Projects-network.html

Definition 25: The second task in the relationship can't finish until the first task starts. However, the second task can finish any time after the first task starts. [33]

Definition 26: The dependent task can't finish until predecessor least started. [34]

Definition 27: The “from” activity must start before the “to” activity can finish. [35]

Definition 28: Successor does not begin until after the predecessor is completed. [36]

Our Commentary: This set of definitions, too, suffers the same inconsistencies that we found with the previously discussed two terms and their sets of definitions. However, all seven of the Start-to-Finish (Holdback Restriction) definitions have two additional problems of interpretation. We’ll take the easier of the two first.

3D3d-i: Meaning of Restriction Delay in a Start-to-Finish Dependency

Let us begin this explanatory side discussion by remembering that a Restriction Delay gives us a numeric measure of the extent to which a Restricted Activity is delayed either partially or entirely by a Restricting Activity. Let’s take a moment to examine what this means in practical terms.

Before explaining why all of the above Start-to-Finish definitions are inadequate, we need to return to the other three Dependency Types (Restriction Linkages) and better understand what we mean when we speak of a Restriction Delay.

We’ll start with the one Dependency Type that typically, and thus by default, has no delay: the Finish-to-Start Dependency (Default Restriction). Referencing Figure F0318, we see that, for the FS:0 default condition, there is no gap in Time (delay) in between

![F0318: Finish-to-Start Dependency (Default Restriction)](image-url)


34 The Complete Idiot's Guide to Project Management with Microsoft Project 2003

35 University of California, through SAS Institute http://www.uc.edu/sashtml/orpm/chapa/index.htm

the Restricted Activity (Activity B) and the Restricting Activity (Activity A).

Because the default Restriction Delay is zero in a Finish-to-Start Dependency, there is no need, and thus it is uncommon, to post the FS:0 Restriction Code.

Now let’s look at a typical Start-to-Start Dependency Type (Start Restriction) which, as we learned earlier, typically describes a gap in Time between the start of a Restricting Activity (the Predecessor Activity) and the start of the Restricted Activity (Successor Activity).

For instance, if we have an SS:3 Restriction Linkage between Activity A and Activity B, then we are saying that the Restricted Activity cannot start any earlier than three days after theRestricting Activity starts.

Let’s apply this to the example in Figure F0320. Here we see two Activities, each with an Activity Duration of ten days. Now this is an important thing for you to know: the Restriction Delay value of “3” is actually a reference to the first three days of Activity A's Duration.

According to Cognitive Project Management, an SS:3 is saying: “Activity B can begin no earlier than a Point in Time at which Activity A has completed at least three days of its Work Scope.” In Figure F0320 we have drawn this Restriction Statement graphically, both as Logic and as a Bar Chart. What we want you to see is that the Restriction Delay of “3” inside Restriction Code SS:3 actually refers to the first three of the ten days of Activity A’s Duration. Notice the shading in Activity A.

Next, let’s look at a Finish-to-Finish Relationship (Finish Restriction) and examine the
same values and points. Check out Figure F0322, where we see the same basic concept, except that this time the Restriction Delay is actually a part of the Activity Duration of the Restricted Duration. The Restriction Delay value of “3” inside Restriction Code FF:3 is actually a reference to the last three days of the Restricted Duration! The first seven days of Activity B are not restricted by the Finish-to-Finish Logic Tie, and so Activity B is free to start on Day 1, as visibly evident in the Bar Chart portion of the illustration.

Okay, fine! Now we can get back to our evaluation of the above definitions for the Start-to-Finish Relationship (Holdback Restriction). From all seven definitions we get a consistent message: there is some kind of Performance Restriction between the finish of Activity A and the start of Activity B.

Let's try to envision it. Activity A may be rolling along at a decent clip but somewhere near its end it must slow to a crawl (and perhaps even come to a halt) until Activity B starts up. Only after Activity A “sees” that Activity B has indeed commenced, can Activity A go on and finish! In a nutshell, that's the concept behind the Start-to-Finish Dependency, according to most Dominant Project Management authorities.

So here’s our question: to which Activity does the Restriction Delay apply? Easing into this question, we can agree that there are four possible ways to interpret an SF:3 Restriction Statement.

- **Possibility 1:** SF:3 refers to the last three days of Activity A.

  Figure F0324 shows the first possibility, where the “3” in SF:3 refers to the final three days of Activity A. Stated in words, the last three days of Activity A are held up until the very instant that Activity B starts. Activity B doesn’t actually have to perform any work, mind you; it just has to show the first instant of its work!
Possibility 2: SF:3 refers to the first three days of Activity B.

Now let’s consider a second possible interpretation of SF:3. Figure F0326 suggests that the “3” in Restriction Code SF:3 refers to the first three days of Activity B. In other words, Activity A can come right up to the very edge of being complete, like maybe 99.999% complete, but there it stops … and waits – for Activity A not only to start, but also for it to complete three days of its ten-day Duration. Is this what is meant by an SF:3 Restriction Linkage?

Possibility 3: SF:3 means that the last three days of Activity A happen at the same time as the first three days of Activity B.
The third possibility is one that we are led to, out of some sense of disappointment (or confusion) in the first two interpretations. After all, did either of those first two really make a lot of sense? Think about it: Try to appreciate the spirit behind this odd Restriction Linkage.

If the start of Activity B actually does have some kind of restraining effect on Activity A’s completion, then shouldn’t Activity A’s completion be held up by more than 0.0001% -- as is the interpretation of Possibility #2? In the inverse, if the last three days of Activity A are being restrained until Activity A sees that Activity B is underway, shouldn’t it wait for more than 0.00001% of Activity B’s opening portion-- as is the interpretation of Possibility #1?

Figure F0328 expresses our confusion graphically. Not knowing where to apply the “3” in SF:3, we apply it to both Activities. Is this the right interpretation? More importantly, is this how the formula actually works in most Scheduling Software, that the numeric value of a Start-to-Finish Dependency Type is counted twice? — once to represent the number of days that Activity B must progress, and again to represent the same number of final days that Activity A is being kept from completing?

We would not find this Restriction Linkage interpretation especially useful, since the odds are maybe 50/50 at best that the delaying period of Activity B just happens be the same as the delayed period of Activity A!

Possibility 4: SF:3 means that the final days of Activity A follow the first few days of Activity B, such that the combined length of Activity A’s final days and Activity B’s first few days equals three days. It is our opinion that for this
Restriction Linkage to be practical we would want something like what is shown in Figure F0330. This configuration allows for two different Restriction Delay values. In the example shown, we need 2 days of Activity B to happen before we can complete the final 1 day of Activity A.

Unfortunately, we know of no Scheduling Software that asks for two different Restriction Delay values, so we are fairly certain that POSSIBILITY #4 is not how the Start-to-Finish Restriction Type is conventionally understood or handled in Dominant Project Management circles.

Returning to the seven definitions, you will notice that none of them speak to the numerical aspect of this Restriction Linkage. They simply say something along the lines of, “You can finish the successor task only after the predecessor task has started.” Does that really tell us enough?

**3D3d-ii: When is a Predecessor Actually a Successor?**

Earlier we said that the above definitions posed “two additional problems of interpretation,” and that we would “take the easier of the two first.” Well now we can turn to discussing the more complicated of the two problems. The problem has to do with whether, in a Start-to-Finish Dependency (Cognitive Project Management's Holdback Restriction), the upstream Activity is a Predecessor or a Successor.

Surfing the Internet one day, we came across a fascinating debate about the Start-to-Finish Dependency Type. We especially like where the discussion ends up – asking whether, in a Start-to-Finish Dependency, the terms Predecessor and Successor really make any sense?
In this exchange Norris Goff is writing to Max Wideman, asking him to comment on an opinion that Norris had expressed to the Project Management Institute concerning a definition contained in its PMBOK Guide. The definition deals with the Start-to-Finish Relationship. Norris raises a question about the Catch-22 nature of a Start-to-Finish Relationship.

The debate may be a little confusing to follow, given that some of the concepts have not yet been introduced at this point in our book. But, stick with it and don’t give up.

You may find it helpful to refer to Activity A and Activity B from Figure F0324 through Figure F0330. The central question of their debate is this: Is Activity A the predecessor of Activity B ... or is it the other way around? As you read, while the debate is fairly detailed, please allow yourself to consider the spirit of what the Start-to-Finish appears to be all about.

Now, STOP and read the discussion that starts on the next page.

Well, we hope you enjoyed the debate. We included it to reinforce a repeating theme extensively made in this book and, indeed elsewhere across the ICS-Compendium: that for almost all of the most important terms used in Project Time Management, there is still way too much disagreement and inconsistency throughout the Dominant Project Management world.

3D4: Performance Restrictions: Per Cognitive Project Management

It should be apparent why Cognitive Project Management felt the need to craft its own definitions for these (and other) terms that are so central to the Critical Path Method of Modeling. Before we close this chapter, we need to admit that we glossed over a significant technical point in our earlier discussion, one that we must now confront.

3D4a: Restriction Delay: Work Performance or Time Passage?

The question at the center of the chapter's final discussion is this: Does a Restriction Delay represent a minimum amount of work that must be performed ... or does it simply reflect a mere Passage of Time? As innocent as this question may seem, it has been the subject of one of the most hotly-contested debates among Schedulers for decades!

Make sure you understand the question before you try answering it. Refer back to

37 Project Management Body of Knowledge, 4th Edition
38 It should be obvious that we are talking only about the Restriction Delays imposed on Start-to-Start or Finish-to-Finish Dependencies, since the Finish-to-Start Dependency clearly occurs between Restricting and Restricted Activities.
From time to time I receive emails that raise intriguing issues of general interest. Here is such a question (edited for web page clarity):

January 14, 2009
Subject: Re: PMBOK Guide 4th edition

Max, please give me your view on the following that I recently reported to the Project Management Institute. In “The Guide to the Project Management Body of Knowledge, 4th Edition [2008],” page 138, Start-to-Finish states: "The completion of the successor activity depends upon the initiation of the predecessor activity."

I believe that statement should read: "The completion of the predecessor activity depends upon the initiation of the successor." A most common example is that of a relay race. Racer #1 (the predecessor) cannot complete until he has passed the baton to Racer #2 (the successor).

The response I received from John Zlockie [PMI]: "I believe that the current descriptions of PDM dependencies in the PMBOK Guide - Fourth Edition are accurate."

If you agree with my position, perhaps you can persuade PMI.

Regards,
Norris S. Goff, PMP, University Learning Institute

Max responded:

Hello Norris. Thank you for your email. I think the relationships should read more logically according to the 1996 PMBOK Guide where the four possible types of logical relationships were expressed as follows:

FS: the 'from' Activity must finish before the 'to' Activity can start.
FF: the 'from' Activity must finish before the 'to' Activity can finish.
SS: the 'from' Activity must start before the 'to' Activity can start.
SF: the 'from' Activity must start before the 'to' Activity can finish.

I agree that the 4th Edition doesn't seem to make sense, but it is the same as the 2004 (3rd) Edition (page 132) and 2000 Edition (page 69). If the "successor" activity is nearing completion and the "predecessor" activity hasn't even started, then it can hardly be a "predecessor" activity! I think your wording would be consistent with the 1996 version. Not bad - only took nine years for someone to spot a serious flaw.
My friend Chris Quaife then weighed in:

Well, Max and Norris, I enjoy items of Scheduling theory even those without much practical use! The relay handover involves a finish-to-start dependency with a negative lag. That is so because the latter activity is the "dependent" activity. I think it is artificial to say the finish of the Runner #1 has to wait for the start of Runner #2. There is a handover for which the timing is determined primarily by the arrival of Runner #1 and secondarily by the skill of handover (unless it's botched).

An S-F relationship can arise when the finish timing of an Activity A is dependent on an independent start timing of an Activity B (which will likely finish later than Activity A). The term "successor" activity does not describe the logic well, although the applicable start and finish have a successor sequence in both logic and timing. "Dependencies" are what a CPM Schedule sets out to model.

PMI has fallen into a terminology trap of over-standardization, in which a "predecessor" is in a logic relationship that doesn't have to mean timing sequence, which is counter-intuitive!

Example: Supervisor A is working at a shift-based job and must have an overlap to give a report to replacement Supervisor B, who is held up in a snowstorm. The actual start of the Supervisor B's shift controls the finish of Supervisor A's shift; a start-to-finish dependency. This is not equivalent to a relay race. Shift A (that started sooner) has the dependent finish, which cannot be until the independent next Shift B has started, i.e. an S-F logic sequence with a few minutes of positive lag. Right?

However, I don't recall ever using an S-F link in a real or hypothetical Schedule! It could be avoided in the above [snow storm] example by an F-F dependency between substitute "Supervisor B Skids and Digs to Work" and 'Supervisor A Leaves Work." To be silly, I could zoom in on the relay race handover and find a sequence of F-S dependencies, such as:

- Runner #1 nears the transfer box
- Runner #2 (already in the box) starts running
- Runner #1 gets to within two-arms length of Runner #2
- Runner #1 places baton in hand of Runner #2
- Runner #2 takes hold of baton
- Runner #1 releases baton and is finished (in perhaps more ways than one! And so am I).

So there you have it. Anyone disagree?

Best regards,
Max Wideman
Figure F0320 on page 99. In the case of a Start Restriction we’re talking about the meaning of “3” in SS:3[^39]

Figure F0332 sets up this discussion with a scenario and an accompanying Bar Chart. We see that ACTIVITY A is scheduled to start at the beginning of Day 1. *When can ACTIVITY B start at the earliest?*

The general interpretation of SS:3 would be that ACTIVITY B can start *no earlier than three days* after ACTIVITY A starts? Well, what happens if ACTIVITY A starts at 8 a.m. on Day 1, but then comes to a grinding halt by 10 a.m. – and then does not resume until Day 5? Would we still want to be showing ACTIVITY B as able to commence at the start of Day 4 even though only *two hours* of ACTIVITY A’s first *three days* of work were actually performed?

We hope you appreciate the question. When we talk about delaying ACTIVITY B, are we waiting for a certain *amount of work* to be performed (in the Restricting Activity) or are we simply waiting for the *Passage of (a certain amount of) Time*, with the clock beginning once ACTIVITY A commences … even if ACTIVITY A suddenly stops only minutes after it gets started?

[^39]: Of course, the question (and the answer) for the Start-to-Start discussion will also apply to the Finish-to-Finish Tie as well. As for the Start-to-Finish Tie, you will soon read that Cognitive Project Management does not recommend using this Logic Tie at all, and in the ICS-Protocols (the guiding standards for ICS-Network Scheduling consultants) it is strictly prohibited.
By now, you will not be surprised to read that there is no universal consensus among Project Time Management authorities on this key point. That is why Cognitive Project Management has taken a stand with respect to how Restriction Delays will be treated — at least throughout the ICS-Compendium, regardless of which Performance Restriction is at play.

3D4b: Cognitive Project Management Restriction Stipulations

Cognitive Project Management has established several Restriction Delay Stipulations related to the three most used Restriction Linkages which, you will recall from the beginning of this chapter, include Default Restriction, Start Restriction, and Finish Restriction.

These Restriction Delay Stipulations are not hard-and-fast rules, like the ones so often mandated throughout Dominant Project Management. Rather, they are merely recommendations (albeit strong ones) for those who choose to adopt the ICS-Compendium as a foundation for their own Project Time Management program.

We present them here because they also provide an excellent educational reference for you, the student of the Critical Path Method of Modeling. Plus, since they are also a foundation upon which subsequent chapters are built, it is important that we all share the same understanding as we move forward together through this text.

Here are the three ICS-Compendium Restriction Delay Stipulations that are germane to our discussion of how Restriction Linkages actually work:

3D4b-i: Stipulation #1: Restriction Delay in Start and Finish Restrictions

Restriction Delay Stipulation #1 applies only to Start Restrictions and Finish Restrictions and says that whenever a Restriction Delay is expressed it represents a Work Performance period, and not the mere Passage of Time. Back to our example in Figure F0332, this would mean that until the work of Activity A not only resumes but also completes the equivalent of three days of its ten-day Duration, Activity B cannot commence.

3D4b-ii: Stipulation #2: Restriction Delays are Never Negative

Restriction Delay Stipulation #2 applies to all three Performance Restrictions and follows logically from Restriction Delay Stipulation #1 (at least with respect to Start Restrictions and Finish Restrictions). If a Restriction Delay only refers to Work Performance, how can Work Performance ever take less than zero amount of Time?

We realize that we haven’t given much attention to the use of negative Restriction Delays in either Start Restrictions or Finish Restrictions but, believe it or not, this is a widespread practice in Dominant Project Management. Moreover,
there is quite a bit of passion on the subject when Time Management practitioners get together over a brew. We would say that most practitioners, however, tend to regard the use of negative Restriction Delays as something less than desirable. [40] Here is why it is not looked upon favorably.

If the goal is to shorten a particular Activity Path, one unsavory technique is to reduce the Restriction Delays of downstream Logic. Obviously you can reduce a large Restriction Delay, say SS:8, to something smaller, say SS:4. And you can later reduce the SS:4 to SS:1. But what do you do when you have squeezed the Relationship to an SS:0? You guessed it! Go negative!

Returning to the matter at hand, in Figure F0334, how can Restricted Activity, Activity R, start before the commencement of the Restricting Activity, Activity Q? Yet, proponents of negative Restriction Delays are undaunted in their defense.

They explain it this way: “We want Activity R to start just ahead of Activity Q, so we use the negative Restriction Delay to tie the two activities together in such a way that, should Activity Q slide back or forth in time, Activity R will move with it ... as if tied at the hip.”

We don't think that a negative Start Restriction is necessary; a positive Start Restriction would be a more intuitive and forward-flowing alternative. In the rare event that you encounter a situation where you wish to link the start of a Restricted Activity to just before the start of the Restricting Activity, rather than using a negative Start Restriction, we recommend that you acknowledge what it is that is actually restraining Activity Q

---

40 Throughout Dominant Project Management the term Restriction Delay is not known or used. Instead you will encounter the terms Lead and Lag and their corresponding numerical expressions, Lead Value and Lag Value.
from starting, and then apply that same restraint to \textbf{Activity R}.

In Figure F0336, we see that some \textbf{Previous Activity}, bearing a ten-day Duration, restrains the start of \textbf{Activity Q}, with a default FS:0 Tie (Logic Code not shown since the Default Restriction Delay is zero).

In this example, if the completion of \textbf{Previous Activity} holds up the start of \textbf{Activity Q}, then we will require that seven of the ten days of \textbf{Previous Activity} be completed before starting \textbf{Activity R}. Using this technique, the same restraint affects both Activities equally which, you will recall, was the rationale for the negative Start Restriction according to its proponents. So, we have accomplished the same modeling, except through positive rather than negative Restriction Delays.

By the way, use of a positive Start Restriction is also a great substitute for a negative
Default Restriction as well. It is sad to say but negative Default Restriction usage is at an all-time high and, for the same reasons as earlier explained, it is considered by many to be an “unacceptable practice.”

Figure F0338 shows how a negative Default Restriction works, and Figure F0340 shows that substitution with a simple Start Restriction achieves the same end, but does so positively and reasonably.

3D4b-iii: Stipulation #3: Restriction Delay in Default Restriction

 Whereas Restriction Delay values used in conjunction with Start Restrictions and Finish Restrictions are always reflections of Work Performance (per Cognitive Project Management), Restriction Delays are always interpreted to reflect the mere Passage of Time in Default Restrictions. The reason for Restriction Delay Stipulation #3 ought to be obvious to you. Just look at Figure F0342, and tell us what work the FS:5 refers to?

A Default Restriction says that the Restricted Activity cannot start any earlier that X days after the Restricting Activity finishes, where X is a zero or positive value. Since the definition includes “after the Restricting Activity finishes,” there can be no work for the Restricted Activity to wait on, since the work of the Restricting Activity has “finished!” Which begs the question: Just what, then, is the Restricted Activity waiting on, when there is a Default Restriction with a positive value?

The Logic shown in Figure F0342 is interpreted as telling us that Activity R can start no earlier than five (5) days after the completion of Activity Q. But why? To know for certain, we would have to ask the Scheduler who overrode the default FS:0 by injecting the FS:5 in its place. Most Scheduling textbooks cite the curing of concrete as a legitimate example of when a Default Restriction with a positive “Lag Value” might reasonably be used.
As they explain it, after the concrete is placed it has to harden and this curing takes time. Curing is something that happens automatically, almost like evaporation. We get it! If you wash your dishes and leave them in the dish drain to air dry, then the same situation would exist.

A question to ask is whether we should create an Activity in the Schedule called, **Air Dry Dishes**? Most good Schedulers are concerned with keeping their Schedules as lean and mean as possible, and this translates into not including extraneous Activities (or Relationships) that add no real value to the Schedule. For this reason, they would rather use a Default Restriction with a positive Restriction Delay in lieu of creating yet another Activity.

But the downside to using a Default Restriction with a positive Restriction Delay is that the Restriction Delay does not come with any explanation. The Logic Code (e.g., **FS:5**) just sits there, leaving those who read the Schedule to wonder: *Why are we making Activity R wait five days after Activity Q finishes before it can start?* As we said earlier, we would have to ask the Scheduler. And therein lies the downside to using a Default Restriction with a positive Restriction Delay.

For this reason, Cognitive Project Management strongly discourages the use of positive Default Restrictions. However, it leaves the option open, since there is nothing patently wrong with its use. But whenever a positive Restriction Delay *is* used, Cognitive Project Management recommends that complete documentation be maintained.\(^{[41]}\) Since the Arrow will merely have a positive numeric value sitting above it, with no explanation for what this Restriction Delay is intended to represent, somewhere in the Schedule Development records a full explanation should be entered.

\(^{[41]}\) The reader should be aware that recent improvements in Scheduling Software are beginning to offer ways for Performance Restrictions to be notated as to the reasons why Restriction Delays have been assigned. ICS-Global credits Dr. Fred Plotnick with leading the charge in pursuit of this helpful software innovation.
In the ICS-PROTOCOLS, which govern the practices of Project Time Management consultants who work for ICS-Global, a Finish-to-Start with a positive Restriction Delay is strictly prohibited.
CHAPTER FOUR

4A Introducing the Network-Based Project Schedule
4A1 Using the Word “Date” a Bit Loosely
4A2 Gregorian Dates and Ordinal Dates
4A3 Calculating Ordinal Dates during Logic Development Sessions
4A4 Ordinal Dates are Much Faster and Simpler
4A5 Dates Comes Last; Not First

4B Key Elements of a CPM Schedule
4B1 Provided versus Calculated Schedule Elements
4B2 Mandatory versus Optional Schedule Elements
4B3 Key Elements are Mandatory; Whether Provided or Calculated

4C CPM Method: Algorithms or Graphics?
4C1 Distinguishing State of Being and Functional Process
4C2 Dissecting the “Activity Box”
4C3 The Connecting Arrow; the Logic Tie
4C4 How to Draw a Default Restriction
4C5 How to Draw a Start Restriction
4C6 How to Draw a Finish Restriction

4D The Point-of-Day Perspective
The first two chapters were fairly conceptual in nature; not too technical. In Chapter Two we learned about Static Project Time Management Tools, such as Performance Listings, Timetables, and Bar Charts. We discovered that these are not especially powerful tools, certainly not versatile or potent enough to single-handedly manage the primary and essential temporal objectives and processes on large Construction Projects.

Chapter Three explained that the Critical Path Method of Modeling a Project Execution Strategy lies at a higher-level in the hierarchy of Project Management aids and is by far the most popular within the Dynamic Project Time Management class of tools. We explored the main feature of the Critical Path Method of Modeling: the Relationship. In particular we noted that, at least according to Cognitive Project Management, there are three primary categories of Relationships between Activities in a Schedule: Communal, Symbiotic, and Progressive.

We then learned that, within the Progressive Relationship category, there are four Performance Restriction Linkages.\[1\] We further discovered, much to our surprise that, despite their widespread popularity and use, existing definitions mostly fail to explain how they actually work!

Working our way from the bottom, we started with the basic building block of every Schedule, the Activity, and noted that when Activities are connected by way of Logic Ties, they form Activity Paths. We learned that clusters of related Activity Paths constitute a Subnetwork, which collectively bond together to form a Network Diagram.

Altogether, we have slowly and deliberately assembled the pieces that we will need, in this chapter, to draw our very first string of CPM Logic. In Chapter Five we will create our first Network-Based CPM Schedule.

4A: Introducing the Network-Based Project Schedule

So what must be added to a CPM Logic Diagram in order to transform it into a CPM Schedule? What does a Network-based CPM Schedule have that a CPM Logic Diagram does not? Or are the two terms synonymous?

---

1 Relationship Types and Dependency Types are the terms that Dominant Project Management uses to describe what Cognitive calls Restriction Linkages.
The short answer is that the two terms are not the same. The very short answer – and this may be a bit over-simplified – is that CPM Schedules are just CPM Network (or Logic) Diagrams with Calculated Dates added. As you will soon learn, that answer is a bit inadequate, if not also somewhat inaccurate. But the main difference between a Logic Diagram and a CPM Schedule is the presence of dates assigned to the Schedule’s Activities. This, of course, is consistent with what we observed in Chapter Two, where we read that four out of six definitions of the word Schedule make reference to the element of Time.

Said differently, one can develop a Logic Diagram that does not have any Calculated Dates (Earliest Dates or Latest Dates), but instead only the pure Logic of the Activities themselves. Such a diagram would be akin to a process map or flowchart, where the calculation of Calculated Dates is not essential for the Logic Diagram to still have great value.

4A1: Using the Word “Date” a Bit Loosely

That brings up one small clarification that we need to get out of the way. Throughout this book we refer to Calculated Dates, but that reference is not entirely appropriate. A more accurate term than “dates” might be “Temporal End Points,” but this is awkward as well. The unspoken variable is the temporal unit of measure being used in the Schedule.

For instance, it is logical that when Activity Durations are being estimated in days, each Temporal End Point (when the Activity will start or finish) would be expressed in days as well. But if the Activity Durations are expressed in hours, then the Temporal End Points would most often be reported in hours as well – and not in days.

Of course there is no hard-and-fast rule on this. One could estimate Activity Durations in hours and still express Temporal End Points in days. Likewise, one could estimate Activity Durations in weeks or months and still post Temporal End Points as days.

The word most often used to describe the two key Temporal End Points that bracket every Activity is what Cognitive Project Management called the Calculated Date.\(^2\) We wish to make it clear that Cognitive Project Management uses the term, Date, in lieu of the multi-syllabic term Temporal End Point. Thus, every Activity will have a Start Date and a Finish Date, even if those Dates are not measured in days, but instead in some other unit of measure, such as hours or weeks or months.

\(^2\) The shortened name (nickname) for Calculated Date is, simply, Date.
4A2: Gregorian Dates and Ordinal Dates

For the most part, Calculated Dates in CPM Schedules are written as Gregorian Dates; that is, dates from a calendar. However, you should keep in mind that Ordinal Dates can be used as well. Ordinal Dates are just sequential numbers. For examples, see most any of the Logic Diagrams in this book; for instance, Figure F0318.

Obviously, since Construction Schedules are used in conjunction with the management of Projects, Schedules would be rather weak and inconvenient if they only displayed Ordinal Dates, thereby leaving it up to the reader of the Schedule to make the conversion to Gregorian Dates. As a default, Scheduling Software display Gregorian Dates.

Many Scheduling Software programs provide a toggle switch that alternates the format of temporal outputs between Ordinal Dates or Gregorian Dates.

The reason we are taking time on such a simple point is to provide a convenient segue into a related topic, but one that is not as straightforward to understand as it may seem. To the point, in the Critical Path Method of Modeling, dates are calculated … not given. And when performing those Arithmetic Calculations manually it is much easier to use Ordinal Dates than Gregorian Dates.

You may be thinking that with the widespread availability and use of computers you will not be doing manual Arithmetic Calculations; but you would be mistaken. The most seasoned Project Facilitators will tell you that one of the shortcomings they see in the current generation of Project Schedulers is a lack of awareness of when or how to use manual Arithmetic Calculations.

This chapter will teach you how to manually calculate the primary CPM Dates (Earliest Start, Earliest Finish, Latest Start, and Latest Finish). As for when you would perform manual Date Calculations, the most likely time is during Logic Development Sessions.

4A3: Calculating Ordinal Dates during Logic Development Sessions

At the heart of Schedule Development is what are known as Logic Development Sessions, which are meetings in which the intended Project Execution Strategy of the Project Team is converted into Logic Diagrams that will later be transformed into the Project Schedule.

The main reason for holding Logic Development Sessions is to transfer knowledge ... in both directions. You may be wondering why we said, “in both directions.” After all, those in attendance at the Logic Development Sessions from the Project Team (e.g.,

---

3 Throughout the book, the term Scheduler refers to the individual responsible for the development and maintenance of the Schedule.
Project Manager, Superintendent, key Subcontractors) surely would have much good information to transfer to you, the Scheduler. But what knowledge would you, the Project Facilitator, possess that would warrant communication in their direction?

The answer is that while they are deciding and explaining their Project Execution Strategy to you, in turn you will be busy teaching them how the Critical Path Method of Modeling works. And a significant first step in the agenda of the Logic Development Session is to teach all participants how the Critical Path Method of Modeling derives Calculated Dates.

As to how you will accomplish this, pay close attention to this and the next chapter and notice how we teach it to you! To let the cat out of the bag, the great trade secret is to – simply perform the Date Calculations in front of them! They will easily follow your process, as you speak it verbally. Within a few short minutes they will be calculating ahead of you. And don’t be surprised the first time that they correct your math errors. This happens with us all the time.

So, yes, we expect that you will be performing manual Date Calculations, and that is why Chapter Five is so important to you. Of course, it is up to you whether you perform manual Date Calculations as part of your Logic Development Sessions, but we would strongly recommend it.

4A4: Ordinal Dates Are Much Faster and Simpler

The reason we use Ordinal Dates when performing manual Date Calculations is that they are so much faster and simpler than Gregorian Dates to add and subtract mentally. To prove our point, use Figure F0402 to complete the following simple exercise. You will need a stop watch (or a wall clock with a second hand), because you are going to time yourself.

**EXPERIMENT A** involves adding up Gregorian Dates, one at a time. We have already done the first calculation for you; just pick up where we left off. Add 12 days to June 17 and you get ____? The assignment is to calculate the COMPLETION DATE. Be sure to take into account the different numbers of days in each month.
Time yourself; It took us 32 seconds.

Next, time yourself doing **Experiment B**, which adds up the Ordinal Dates in Figure F0404. This one took us only 15 seconds; half the time!

**Experiment A** took you longer because you had to mentally take into account the different number of days in each month. Now imagine how much longer it would have taken if you had to also discount for (jump over) recognized holidays and weekends! And what if the CPM Schedule had 1,000 Activities in it, and consider that each Activity has four Calculated Dates associated with it!

4A5: **Dates Come Last, Not First**

But there is another, more important, reason why you want to use Ordinal Dates during Logic Development Sessions. It has to do with what we were saying about Bar Charts back in **Chapter Three**, believe it or not. Do you remember us saying that field people, when they create Bar Charts, already tend to have specific Dates in mind, and that they simply draw an Activity Bar under a Timeline to correspond to preconceived Start Dates and Finish Dates?

Well, this is as good a time as any for us to make an important point about why CPM Schedules are so superior to Bar Charts for modeling a Project Execution Strategy. In a nutshell, Network-Based CPM Schedules are built *from the bottom up*. That is, the Calculated Dates are the last thing to be revealed, not the first as with Bar Charts.

The process for developing a Project Schedule using the Critical Path Method of Modeling requires that contributors think in terms of *what* needs to be accomplished (rather than *when* it will be accomplished). And they do this mental exercise at the most detailed level.

You will recall that the core building block of the Schedule is ….? That’s right, the Activity. So, Logic Development Sessions are all about figuring out what Activities to put into the Schedule, assigning Durations to them, and then stringing them together using Logic Ties. During this process, the last thing you would want is for contributors to try forcing a Subnetwork of Logic to “come out” on some preconceived calendar date that they might have in mind. Rather, you want the chips to fall where they may.

Do you follow us? You want the contributors to think about each and every Activity on its own merits. You want them to offer Activity Duration estimates that reflect the
Crew Configuration that they envision being used on that Activity.[4] Finally, you want them sewing the Logic together, based on credible and sound Restriction Objectives. We talked briefly about Restriction Objectives in Chapter Three and we will discuss them in greater detail elsewhere in the ICS-Compendium.

Every Performance Restriction has its own justification or rationale for existence. Restriction Objectives help us to understand and categorize those differences. Can you name and explain the four Restriction Objectives? Please flip back to page 81 and refresh your memory of what they are all about.

We are reminding you about the Restriction Objectives in order to reinforce the point that during Logic Development Sessions you want your contributors to think about how the Activities will relate to one another, for the most part regardless of when those Activities may occur beneath some graphical Timeline or on a wall calendar.

But whatever the reasons, you want your contributors' Logic and their Activity Durations to reflect their best assessment of a future reality ... and not for them to be influenced by what final Date emerges at the end of a given Subnetwork, or even for the Project as a whole.

"Why not," you may ask, "simply not show them any dates at all during the Logic Development Sessions? That way," you continue, "they won’t see where any Subnetwork ends up on a calendar?" Good question!

The answer comes back to that pesky word, “reality.” The fact is that this Schedule that you are constructing with their input is meant to simulate a future reality. And the overall timing of the Project, when measured or expressed on a somewhat macro level (specifically, Subnetwork level), is useful information that they need to have in mind as the Schedule is being developed, even during the early Logic Development Sessions.

For example, suppose that it is now late August and the Project in question has a Notice to Proceed date of September 8. Suppose further that the group in the Logic Development Session jointly thinks that December 15 is a safe date to use for “the beginning of winter conditions.” Basically, as the Project Team calculates mentally, they have a little over three months to “get out of the ground and somewhat enclosed” before winter freezes consume the Project. If you figure 5 days to a week, and 13 weeks to a quarter (three months), then there are about 65 Workdays available during this critical Time.

During the Logic Development Session, as the Activities begin to emerge on paper[5] and the Logic between the Activities are tied together, the experienced Scheduler will

---

4 Do you remember the concept of Continuous Crew Days?
5 Or on the white board or computer screen, depending on the technology you employ in your Logic Development Sessions
choose to perform ongoing manual Date Calculations, as the Session progresses, so that the group can keep tabs on that 65-day window they have to work within.

By using Ordinal Dates (from 1 to 65) the Project Team is able to get a sense of where they stand within this window of opportunity, but without getting distracted (or unduly influenced) as they would be by the presence of specific Gregorian Dates. Again, we are trying to fight a tendency prevalent among field personnel to think in terms of preconceived calendar dates. We don’t want them to give us bogus Activity Durations or convoluted Logic just to “make” certain calendar dates that they might have in mind.

We will add, somewhat parenthetically, that about 25% of the time, when using the above approach the Project Team discovers that the arbitrary deadline that they had had in mind was simply not all that reasonable after all. This is a great thing to discover now, while they are building the Project on paper in the warmth and comfort of a conference room, than three months later when the first snowstorm hits and they are caught with their pants down!

4B: Key Elements of a CPM Schedule

Let us now return to this chapter’s opening question: What does a Schedule have that a Network Diagram does not have? We already gave you the short answer: Dates. Now we will present a more thorough answer. But this more involved answer requires that we make a few distinctions up front.

4B1: Provided versus Calculated Schedule Elements

Some Schedule Elements have their genesis from outside of the Schedule itself. In other words, they are provided by the contributors to the Scheduling process. We might include in this group things like: the Activity List itself, Activity Identifiers, Activity Descriptions, Activity Durations, Restriction Statements, and more.

Other elements of the Schedule are determined by one or more Scheduling processes that operate on details contained in Provided Schedule Elements. We call these Calculated Schedule Elements. Items under this heading would include: Earliest Dates and Latest Dates, Free Float and Total Float, Critical Path determination, and such.

4B2: Mandatory versus Optional Schedule Elements

Another way to categorize elements of a CPM Schedule is with respect to whether the element is mandatory or optional. Mandatory Schedule Elements are those that must exist in a Schedule in order for it to function as a Dynamic Project Time Management
Tool. Examples would be Activity Identifiers, Activity Descriptions, Activity Durations, and so forth.

By contrast, Optional Schedule Elements are ones that may be omitted from the Schedule without limiting its ability to function as a meaningful Dynamic Project Time Management Tool. For instance, a Schedule may or may not contain Date Constraints or multiple Work Performance Calendars (both discussed at length in Chapter Nine).

<table>
<thead>
<tr>
<th>MANDATORY</th>
<th>OPTIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Identifier</td>
<td>Activity Code Definitions</td>
</tr>
<tr>
<td>Activity Description</td>
<td>Activity Code Assignments</td>
</tr>
<tr>
<td>Original Duration</td>
<td>Resource Code Definitions</td>
</tr>
<tr>
<td>Performance Restrictions (dependencies)</td>
<td>Resource Code Assignments</td>
</tr>
<tr>
<td>Activity Percent Complete*</td>
<td>Cost Code Definitions</td>
</tr>
<tr>
<td>Etc…</td>
<td>Cost Code Assignments</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CALCULATED</th>
<th>PROVIDED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remaining Duration*</td>
<td>Activity Code Assignments</td>
</tr>
<tr>
<td>Activity Earliest Start</td>
<td>Resource Code Assignments</td>
</tr>
<tr>
<td>Activity Earliest Finish</td>
<td>Resource Code Assignments</td>
</tr>
<tr>
<td>Activity Latest Start</td>
<td>Cost Code Definitions</td>
</tr>
<tr>
<td>Activity Latest Finish</td>
<td>Cost Code Assignments</td>
</tr>
<tr>
<td>Activity Total Float</td>
<td>Automated Calendar Non-WorkDays</td>
</tr>
<tr>
<td>Critical Path Identification**</td>
<td>Date Constraints</td>
</tr>
<tr>
<td>Etc…</td>
<td>Identification of Milestone Activities/Events</td>
</tr>
</tbody>
</table>

* These two values can swap places. If Percent Complete is provided, then Remaining Duration can be calculated ... and vice versa.

** Whether these are mandatory or optional depends on how you define schedule “functionality.” We are assuming that Critical Path identification is an expected ability of the scheduling tool, and that Free Float identification is not. Your expectations may differ.

F0406: Key Schedule Elements in CPM Network-Based Schedule

4B3: Key Elements are Mandatory; Whether Provided or Calculated

In Chapter Six we will immerse ourselves in a discussion of the Key Elements of a Project Schedule which, using the above-defined terms, we can clarify as being a discrete set of Mandatory Schedule Elements, some of which are Provided Schedule Elements that come from outside the Scheduling processes, while others are Calculated Schedule Elements that evolve during performance of various Scheduling processes.
Figure F0406 begins a chart that we would encourage you to continue developing on your own. Each Scheduling organization has its own ideas as to what are Mandatory Schedule Elements or Optional Elements. For purposes of the following discussion, however, we will refer to the items highlighted in dark green as being the Key Elements of a CPM Schedule that is being used to model Project Execution Strategy.

4C: CPM Method: Algorithms or Graphics?

We realize that you are as eager as we are to get into the nuts and bolts of CPM MECHANICS, in particular the derivation of the four most important calculated values in the entire Critical Path Method of Modeling: they being Earliest Dates and Latest Dates. But before we do so, we must first tend to a couple of technical points which will ensure that we have the same underlying understanding as we move forward together.

For one, we need to come to agreement on just what this Critical Path Method of Modeling is — at its most basic. Here is the question: Is the Critical Path Method of Modeling essentially a graphical symbolic language, or is it a set of formulas and arithmetic steps – or is it both?

Believe it or not, many CPM authorities, including a few professors that we personally know quite well, believe that CPM is first and foremost a graphical model of a Project Execution Strategy – and only secondarily a functional set of Arithmetic Formulas and processes.

As an historical side note, you may find it interesting to learn that the originators of the Critical Path Method considered it primarily as a computer algorithm, and they only invented the graphical symbolic language as a way to “explain the algorithms” to lay persons!

We hold to the opinion that it is both. In fact, we will go further and say that the two manifestations of CPM strongly reinforce and validate one another.

As you will read numerous times throughout this text, we believe that good Schedules start with Logic Diagrams which, by definition, are graphical. On the other hand, the point of Logic Diagrams is to grow them into Project Schedules, and Schedules must contain dates. And all of those dates are Calculated Schedule Elements, not Provided Schedule Elements.

For a few pages then we need to learn some of the basic symbols of the graphical form of the Critical Path Method of Modeling. Think of CPM as a symbolic language, a sort of shorthand for depicting more complex statements of cause and effect in graphical form.
4C1: Distinguishing State of Being and Functional Process

We will begin our exploration of the graphical side of the Critical Path Method of Modeling by establishing two important concepts: State of Being and Functional Process. What is the difference between a State of Being and a Functional Process?

Let’s imagine that you are trying to put your five-year-old to bed. You sing her a song, read her a story, and then you tuck her in. At last, her eyes have closed and she seems to be smiling happily as she drifts into Dreamland. Ever so quietly, you rise from the bed and tiptoe to the door. Just as you are about to step into the hallway you here a soft voice call out gently, “Leave the door open, daddy.”

You smile to yourself, as you silently step out into the hall and turn back to the door. Then, you just as softly touch your hand to the doorknob, trying not to make too much noise as you slowly pull the door closed. You stop short of completely closing the door, thus honoring your daughter’s request to “leave the door open.”

That familiar, sweet voice now penetrates the one-inch space between the door and the door frame. “No, daddy! Leave it OPEN!” So you push the door all the way open, as she gives you a sleepy smile. She lowers her head back down on the pillow and this time drifts off for the night.

Figure F0408 introduces the concept of State of Being. In this illustration, what is the State of Being of the door? Answer: open. Likewise, what is the door’s state in Figure F0410? Answer: closed.

But now look at Figure F0412, which shows the Functional Process of closing a door. Back to your daughter, your process for closing the door began the second you touched the doorknob, lasted as long as you were pulling the door toward you, and ended the second you let go of the doorknob.
Functional Processes require Action, and Actions consume Time. By contrast, a State of Being just is. It is a State of Being, not a State of Acting.

- A winter night is cold.
- A dog is furry.
- A machine is loud.
- A glass is half-full … or half-empty.
- An Activity is completed.
- An Activity has not yet started.

Those last two items describe a State of Being that exists either before and after some Functional Process takes place in between those two States of Being.

**4C2: Dissecting the Activity “Box”**

Let’s take these two concepts, State of Being and Functional Process, and apply them to the symbol for an Activity as depicted in a CPM Logic Diagram. In Figure F0414 we see an Activity Box, taken from an earlier set of figures. It looks like a box, right?

Well … look closer. It is actually four lines connected together, to look like a box. Two lines are vertical, and two are horizontal. But at the microscopic level, those four lines do not actually touch one another. See Figure F0416.

We are asking you to be willing to set aside your lifelong view of a box as a box, and instead perceive an Activity Box (at least ones found in a Logic Diagram) as being a combination of graphical symbols.

Specifically, try to envision the perimeter of an Activity Box not as one continuous line, but instead as four lines joined at the corners: two vertical lines that depict States of Being and two horizontal lines that each represent the same, single Functional Process. Let’s work with this admittedly strange perspective, shall we?

Figure F0418 presents to us the symbolic significance of what might otherwise appear to our naked eye as just a box with words in it. The first thing we notice is that the two vertical lines represent what the diagram calls Moments in Time. The diagram also reminds us that these Moments in Time are States of Being.
By differently coloring the two vertical lines we wish to reinforce in your mind that each vertical line represents a different Moment in Time, a different State of Being. Notice, though, that the two horizontal lines have the same color, black. This is because the upper and lower horizontal bars are identical; they both stand for the same thing: the Work Progress associated with the Activity. If the Activity Description were to read, **CLOSE DOOR**, then either horizontal bar would graphically represent the Action required to close the door: Reach out and grab door knob; pull the door to you until it latches; and then release door knob.

The vertical green line on the left side of the Activity Box represents the Conditional State **before** the work has begun. In our example, it would represent the door completely open, with the Functional Process of closing the door not yet begun.

The vertical line on the right side, shown in red, represents another Conditional State, the state of the Activity **after** the work of the Activity has completed. In our example, this Moment in Time corresponds to the physical Conditional State of the door being fully closed, and our hand no longer touching the doorknob.

We are spending a lot of time on this very simple concept because it is so important to your understanding of how the symbolic language of CPM graphics actually works. We need for you to begin seeing an Activity Box as four separate lines, each of which symbolizes a specific variable that, collectively with all others within the Schedule, contributes to the effective use of Time on your Projects.

You need to see that the left side of the Activity Box represents a Condition State: a Moment in Time **before** the work has begun. Please drill that into your mind. The left side represents a statement about the Conditional State just prior to the start of the work. The vertical line on the left side actually says,

- I warrant that all conditions necessary for the commencement of the Work of this Activity have been met, and that the Work of this Activity, depicted by either of the horizontal bars, is now **able** to commence -- but has not yet commenced!
The right-side vertical line tells another story. It says,

 Citadel: “I warrant that the Work of this Activity has been completed in full and that the Conditional State, known as 100% Complete, has been achieved.”

As for the horizontal bars, each represents the same Functional Process, and not a Conditional State. In a CPM Schedule, they represent the performance of Project Work Scope. They are not time-scaled so there is no implication that a dot halfway across a horizontal bar necessarily represents 50% Complete. The horizontal bar merely symbolizes Work Scope performance; it symbolically spans from a Conditional State before the Work has begun to a Conditional State after the Work is finished.

We are now ready to discuss the various points where a Restriction Flow Arrow (also called a Connecting Arrow, Arrow, or Logic Tie) touches an Activity Box and what each configuration means. Remember, the graphics of the Critical Path Method of Modeling are a symbolic language and, as such, you should be able to interpret a Restriction Statement without the need to read any written words (other than the Activity Descriptions or Restriction Code)! That is the true test of an effective symbolic language.

4C3: The Connecting Arrow; the Logic Tie

We begin with the Connecting Arrow itself. As you already know, the Arrow is what connects Activities together and therefore illustrates the Logic Ties between Activities. We see that a Connecting Arrow has an arrowhead and it has a tail. The arrowhead indicates the direction of the Restriction Flow.

Since work moves forward through Time, there is a cardinal rule, when drawing Schedule Logic, that the Arrow must always flow to the right on the horizontal axis. An Arrow pointing to the left would be pointing backwards in Time and that would be illogical.

Under certain circumstances, an Arrow need not be straight, but can have an elbow shape. In such a configuration, the vertical portion of the Arrow may point up or down, but it always connects to one of a Restricted Activity on the horizontal axis.

So let’s see if, without resorting to the use of words, we can employ these visual symbols to state something meaningful about the Dependency Relationships of Activities in a Network Diagram.
We will begin with the Arrow itself; check out Figure F0420. Each of these is a legitimate Arrow. An Arrow can point up, down, or to the right (but never to the left). Notice, also, that it is quite acceptable to combine or join Arrows, if they are coming from and/or leading into a common Activity.

4C4: How to Draw a Default Restriction

We can now look at how to use our newly-learned symbolism to make a partial Restriction Statement. We will do this through a series of baby steps. This may seem a bit elementary, but please indulge us. We would rather be purposely simplistic than unintentionally obscure.

We begin with Figure F0422 which shows a single Activity, Activity R. The red vertical line draws our focus to the right side of the Activity, of which Figure F0424 gives us a better view.

We recall from our earlier discussion that the right-side vertical line represents a Moment in Time that occurs after the Functional Process of the Activity has fully completed.

We are now ready to link a Connecting Arrow to an Activity Box and thereby form our first Restriction Statement, albeit written in the graphical symbolism of the Critical Path Method of Modeling Projects. Perhaps you would like to interpret Figure F0426?
We see that the Arrow originates at the right side of the Restricting Activity. So this means that whatever Activity this Arrow points to must occur after the Functional Process of the Restricting Activity has completely finished. Figure F0428 highlights the left side of an Activity Box which we recall represents a Moment in Time before the Functional Process of an Activity has begun.

As we see in Figure F0430, an Arrow going into the left side of an Activity Box signifies that the Restricting Activity (on the tail end of the Arrow) must be partially or fully completed prior to the start of the Restricted Activity.

It is now time to put the pieces together. Take a good look at Figure F0432, which shows two Activities connected by an Arrow—and your very first string of Logic. (Congratulations!)

You are looking at a standard Default Restriction (Finish-to-Start Dependency), which you will see a million times in your career as a member of the Construction Project Time Management team.

While it is likely that even before you ever picked up this book you might have been able to guess that Activity R precedes Activity S, you can now construct a properly-worded Restriction Statement that describes the true Performance Restriction between the two Activities. Let us give it a try:

The Conditional State required by Activity S for it to be able to commence its Work (Functional Process) is restricted by the prior Conditional State for Activity R, in which Activity R’s Work (Functional Process) has fully completed.

Now we admit that this Restriction Statement sounds like a lawyer wrote it, but we meticulously worded it to convey the various concepts we have been discussing thus far. The Arrow connects the right side of Activity R to the left side of Activity S. The
right side of Activity R represents a Conditional State in which the Work of Activity R is completely finished. The left side of Activity S reflects a Conditional State in which the Work of Activity S is able to begin, but has not yet begun.

The Arrow not only tells us that this is a Default Restriction, it tells us which two Activities share the Progressive Relationship; Activity R and Activity S. We do not need to add a Logic Code (i.e., FS:0) adjacent to the Arrow to know that this is a Default Restriction, do we? We can tell just from the way the Arrow (Logic Tie) is drawn that it is a Default Restriction. Specifically, the Arrow begins at a “finish” Moment in Time, and terminates at a “start” Moment in Time; hence, the Finish-to-Start Dependency type.

That said, in the rare event that you are defining a Default Restriction with a Restriction Delay other than zero, you would want to add a Logic Code to your Logic Diagram and place it adjacent to the Arrow. For example, suppose that we want to create a four-day Restriction Delay between the finish of Activity R and the start of Activity S. We would annotate the Arrow with an FS:4 Logic Code as in Figure F0434.

So, to recap:

- If the Default Restriction has a zero Restriction Delay, an accompanying Logic Code is not needed.
- If the Default Restriction has a positive Restriction Delay, an accompanying Logic Code is needed.
- If the Default Restriction has a negative Restriction Delay — well, you haven’t been listening. Go back and re-read this chapter. Cognitive Project Management strongly discourages the use of negative Default Restrictions. But, yes, if you do use them, then you will need to include a Logic Code.

4C5: How to Draw a Start Restriction

The Start Restriction and Finish Restriction are equally more complicated to draw than is the Default Restriction. For one thing, a Default Restriction is quite clear-cut: two Activities connected by an Arrow that, at the point it connects to the Restricted Activity, is always horizontal.

But Start Restriction Logic Ties differ in two significant ways:

- The two Activities are never linked end-to-end.
- At the point it connects to the Restricted Activity, the Arrow is always vertical.
Let’s get into this. We will start this discussion back at the Restricting Activity, where the first question is this: *where should the tail end of the Start Restriction's Logic Tie touch the Restricting Activity?* Consider Figure F0436, in which we are presented with five options.

We hope you realize by now that whatever we say about where an Arrow would connect with the upper bar applies equally to the bottom bar, since both horizontal bars represent the same thing: the Functional Process of performing the Work Scope of the Activity.

So, let’s take the five options one at a time. As illustrated in Figure F0438, we can rule out **Connection Point A**. We hope you made this determination on your own, but in case you did not, here is the reason why:

**Connection Point A** aligns with the left-vertical side of the Activity Box, and you will recall that the vertical lines represent Moments in Time either before or after the Work Performance itself (which is represented by the horizontal bars). If we were to start a Logic Tie from either left corner (top-left or bottom-left), we would be relating the Restricted Activity (to which the Arrow’s head would be connected) to a Moment in Time *before* the Restricting Activity's Work had ever begun!

Let’s go back to the meaning of a Start Restriction, which says that "the commencement of a Restricted Activity is constrained by both (a) the commencement and (b) partial performance of a Restricting..."
Activity.” Well, wouldn’t that point be somewhere along the horizontal bars (upper or lower), which are the only sides of the Activity Box that depict Work Scope being functionally processed?

The same argument applies to rejecting **Connection Point E** as enhanced in **Figure F0440**. The right-side vertical line represents a Moment in Time *after* the work has completed. Again, let us return to the meaning of a Start Restriction, which says further, “... constrained by both (a) the commencement and (b) partial performance...”. The words “partial performance” mandate that the Work of the Restricting Activity has not yet finished, yet the meaning of the Conditional State represented by the right-side vertical line of the Activity Box is that the Work of the Restricting Activity *has* finished.

So now we are down to **Connection Point B**, **Connection Point C**, and **Connection Point D**. *Which of these is an acceptable representation of a Start Restriction?* The correct answer is: *all three*. They are all correct, unlike **Connection Point A** and **Connection Point E**. But now we introduce an aesthetic consideration. (Yes, there is an artistic aspect to Scheduling!)

Cognitive Project Management recommends that we commence our Start Restriction Logic Ties somewhere in the *first third* of the horizontal bar but, as already discussed, not too close to the left corner. Why, the first third? The answer has to do with the *spirit* of the word, Start, in the label, Start Restriction.

You may be thinking that if we had commenced a Start Restriction Logic Tie at **Connection Point D** we would be consistent with the words “commencement and partial performance.” Again, technically you would be correct. But we think it seems a bit weird to show a Logic Tie for a *Start* Restriction commencing so close to the *end* of the Functional Process horizontal bar. Don’t you? For the same reason we think that the middle connection point also fails to imply the spirit of the word, *Start*.

That is why, as shown in **Figure F0442**, Cognitive Project Management prefers to commence Start Restriction Logic Ties in the first third of the horizontal bar (either of them). At the same time, we also shy away from getting too close to the left corners, for fear of it looking like we are commencing from the left vertical line which represents a Moment in Time *before* the work has actually begun. We hope all of this makes sense.

Recalling that a Start Restriction reflects a Start-to-Start Dependency, *where should the Arrowhead of the Logic Tie contact the Restricted Activity?* This should be easy
for you – we hope! Here, we are talking about the second Start in the Start-to-Start Dependency. The second Start refers to the Restricted Activity's start.

Let us focus on the meaning of a Start Restriction, with respect to the Restricted Activity: “The commencement of a Restricted Activity is constrained by...”. So, if we are going to connect the Arrow to a Moment in Time before the work of the Restricted Activity has even begun, where would the Arrow touch the Restricted Activity Box? On the left vertical, of course! Ah … but where on that vertical?

Referring to Figure F0444, our two choices are Connection Point A at the corners, or Connection Point B anywhere else along the left vertical. We would discourage using the corners, even though they are technically part of the vertical line and not part of the horizontal lines.

The problem with the corners is that they align with the horizontal bars and this might imply some association with the Functional Process of the work depicted by the horizontal bars. Our preference is to connect anywhere in the Connection Point B area.

So let’s put all of this together. Figure F0446 shows the two acceptable [6] ways to draw a Start Restriction Logic Tie. We hope you have also realized that, with a properly drawn Start Restriction Logic Tie you really don’t need a corresponding Logic Code to tell you that this is a Start Restriction. Simply by interpreting the symbolic language of how the Arrow is drawn we are able to tell that Activity L must be underway (“partial performance”) before either Activity M or Activity N can start!

Of course, unlike a Default Restriction, which requires no Logic Code (unless the Restriction Delay is other than zero), we must always provide a Logic Code with a Start Restriction, in order to know the magnitude of the Restriction Delay. How else would we know how much “partial performance” the Restricted Activity must wait for? And so, we will add the Logic Code; as in

---

6 Acceptable, per Cognitive Project Management.
Figure F0448. As a matter of style, Cognitive Project Management's Logic Diagramming Style Guide prefers that we place the Logic Code near the elbow of the Arrow, whenever possible.

4C6: How to Draw a Finish Restriction

This next discussion ought to go rather quickly, because the same reasoning that guided the placement of Logic Ties in Start Restrictions applies to Finish Restrictions as well.

Do you recall the meaning of a Finish Restriction, which we presented on page 89? It states that “the completion of a Restricted Activity is constrained by the prior completion of its Restricting Activity.” From this, it should be obvious that a Finish Restriction Logic Tie commences from the right side of the Restricting Activity Box, since the right-side vertical represents a Moment in Time after the Functional Process of the Restricting Activity has fully completed.

We also know, from this definition, that the arrowhead of the Logic Tie should contact the Restricted Activity somewhere toward the end of the Activity, but not at the right-most corners (and surely not at the left-most corners). Abiding by the same one-third rule we used with the Start Restriction Logic Tie, we prefer to terminate Finish Restriction Arrows somewhere within the right-most third of a horizontal bar, but not at the corners.

We also need to add a Logic Code in order to denote how much of the Restricted Activity is being constrained by the prior completion of the Restricting Activity. In Figure F0450, the Restriction Code of FF:2 tells us that the final two (2) days of Activity C must wait on the prior completion of Activity B.

As we close out this discussion about the symbolic language of the Critical Path Method of Modeling, we hope you appreciate the importance of how you draw your Logic, especially where you start and end Connecting Arrows. If you abide by the recommendations in this book, it will be quite easy for those who read your Schedules to translate your drawn Logic into quality Restriction Statements.
To test this, can you identify the Restriction Types based solely on the configurations of the Connecting Arrows in Figure F0452? Were you able to discern the following?

- Arrows A, C, and G depict Default Restrictions
- Arrows B and E depict Start Restrictions
- Arrows D and F depict Finish Restrictions

4D: The Point-of-Day Perspective

We are so very close to being able to start performing Date Calculations which we will do in the next chapter. However, before we can begin Date Calculations we need to return to two concepts briefly mentioned in Chapter Two: Start-of-Day and End-of-Day.

To quickly clarify the title of this subsection, the Point-of-Day Perspective refers to the choice that a Project Facilitator makes before performing manual Date Calculations (either Forward Pass or Backward Pass).[^7] This decision affects whether the Earliest Dates and Latest Dates are 100% correct, or only 50% correct. If this makes no sense just yet, it will by the end of this last subsection of the chapter.

The three choices of perspective are Start-of-Day, End-of-Day, or Point-of-Day. Whenever you perform manual Date Calculations, you need to decide ahead of time whether you will be using a Start-of-Day or End-of-Day Perspective. To be clear, you could use Point-of-Day, which is a mixture of both, but it will surely slow you down.[^8]

We now refer you to an ICS-White Paper that fully discusses the Point-of-Day Perspective. Please set this book down at this point while you read ICS-White Paper WPB-KJ-28, “Understanding the Point-of-Day Perspectives,” before going on to Chapter Five.

---

[^7]: The operative word, here, is "manual."
[^8]: Scheduling Software, which enjoy extremely fast computing speeds, uses Point-of-Day by default.
CHAPTER FIVE

5A    Introducing the Concept of Earliest Dates
5A1   Defining Forward Pass and Backward Pass Processes
5A2   Each Date Calculation Process has Four Elements
5A3   Date Calculation Arithmetic Formula Identifiers
5A4   Forward Pass Calculations through Default Restrictions
5A4a  Finding Earliest Dates thru One Default Restriction
5A4b  Finding Earliest Dates thru Multiple Default Restrictions
5A5   Forward Pass Calculations through Start Restrictions
5A5a  Finding Earliest Dates thru a Single Start Restriction
5A5b  Finding Earliest Dates thru Multiple Start Restrictions
5A6   Forward Pass Calculations through Finish Restrictions
5A7   Forward Passthrough Combined Start/Finish Restrictions
5A8   Forward Pass: Putting It All Together

5B    Introducing the Concept of Latest Dates
5B1   Calculating Latest Dates thru Default Restrictions
5B1   Calculating Latest Dates thru Start Restrictions
5B1   Calculating Latest Dates thru Finish Restrictions
5B1   Backward Pass: Putting It All Together
5A: **Introducing the Concept of Earliest Dates**

Congratulations! You are now adequately prepared to start learning how to calculate the **FOUR BASIC CALCULATED DATES** that lie at the heart of the Critical Path Method of Modeling. We will begin with a citation from the ICS-Dictionary:

**Four Basic Calculated Dates:** Within the Critical Path Method, as a method of modeling Project Execution Strategy, every Activity in the Schedule has Four Basic Calculated Dates associated with it: Earliest Start, Earliest Finish, Latest Start, and Latest Finish.

There are two Date Calculation Processes associated with these Four Calculated Dates:

- The Date Calculation Process that generates Earliest Dates is called a Forward Pass Process and is the subject of the first half of this chapter.
- The latter half of the chapter will explore the Date Calculation Process that generates Latest Dates, called a Backward Pass Process.

Each of the **FOUR BASIC CALCULATED DATES** has its own set of Arithmetic Formulas and Computational Rules. These Arithmetic Formulas and Computational Rules, even for just one Calculated Date, differ depending on certain Date Calculation Variables that dictate which Date Calculation Procedures are to be followed.

5A1: **Defining Forward Pass and Backward Pass Processes**

We want to be crystal clear about something: both the Forward Pass and the Backward Pass are *processes*, and each Date Calculation Process is comprised of Date Calculation Procedures, Arithmetic Formulas, Computational Rules, and Date Calculation Variables, as clarified by the following ICS-Compendium definitions:

**Forward Pass Process:** Used to determine the Earliest Start and Earliest Finish Dates of a series of Progressively-related Activities in a CPM Network Diagram, the Forward Pass Process is a combination of (a) Computational Procedures that utilize
(b) Arithmetic Formulas in accordance with (c) condition-specific Computational Rules that are mandated by particular (d) Date Calculation Variables.

- **Backward Pass Process:** Used to determine the Latest Start and Latest Finish Dates of a series of Progressively-related Activities in a CPM Network Diagram, the Backward Pass Process is a combination of (a) Computational Procedures that utilize (b) Arithmetic Formulas in accordance with (c) condition-specific Computational Rules that are mandated by particular (d) Date Calculation Variables.

5A2: **Each Date Calculation Process has Four Elements**

As these definitions tell us, each of the **Four Basic Calculated Dates** has its own set of Arithmetic Formulas and Computational Rules. These Arithmetic Formulas and Computational Rules vary, based on the Date Calculation Variables at play and which Date Calculation Procedures they instruct us to follow. Let us cite ICS-Dictionary's four elements of either Date Calculation Process:

- **Date Calculation Arithmetic Formulas:** Arithmetic Formulas are used to determine the Four Basic Calculated Dates associated with every (non-zero duration) Activity in a Critical Path Method (CPM) schedule. The appropriate Arithmetic Formula to be used for any particular Calculated Date is determined by Date Calculation Variables and Computational Rules.

- **Date Calculation Computation Rules:** Specific Date Calculation Computation Rules that apply to each different Date Calculation Computational Procedure used in Forward Pass and Backward Pass Processes. The Forward Pass and Backward Pass Processes are comprised of the Arithmetic Formulas, Computational Rules, and Date Calculation Variables that are integrated through Date Computation Procedures.

- **Date Calculation Computation Procedures:** Date Computation Procedures refers to a series of actions performed in a precise manner, which culminate in the determination of the Calculated Dates. The Forward Pass and Backward Pass Processes are comprised of the Arithmetic Formulas, Computational Rules, and Date Calculation Variables that are integrated through these Date Computation Procedures.

- **Date Calculation Variables:** Inherent in the raw content of the CPM Network Diagram are Date Calculation Variables that determine the Arithmetic Formulas and Computational Rules to be used during performance of Forward Pass and Backward Pass Procedures. The two Date Calculation Variables are Performance Restrictions and Activity Durations.
We realize that these technical definitions may seem way too cryptic to make a whole lot of sense at this point in your learning. The main message to comprehend, before we get into the nuts and bolts of this chapter, is that there is more to calculating CPM's **Four Basic Calculated Dates** than just solving Arithmetic Formulas.

This chapter will talk you through the above four elements of the Forward Pass and Backward Pass Processes. It will show you, step by step, how to work your way through the Date Calculation Procedures, how to apply the Computational Rules, and how to apply the correct Date Calculation Formulas, in each different situation.

Since this will be a rather lengthy chapter and, in the future (and perhaps even now, as you read *this* chapter), you may wish to have a more concise reference that explains the Forward Pass and Backward Pass Processes, we wrote **ICS-White Paper WPB-KI-13**, entitled, "**How to Calculate Dates in the Critical Path Method.**" \[1\]

We encourage you to STOP reading this book at this point, and redirect your attention to the above ICS-White Paper. Please read through the ICS-White Paper one time through, even if some parts of it do not make complete sense. Then return to this book and proceed through the end of this chapter. Once you reach the end of the chapter, please re-read the ICS-White Paper. It will make much more sense by the end of this chapter.

---

**5A3: Date Calculation Arithmetic Formula Identifiers**

To ease communications, each Arithmetic Formula has its own Arithmetic Formula Identifier. An Arithmetic Formula Identifier has two parts, separated by a vertical bar. For example, consider, **EF|du**.

- The **leftmost two characters** tells us which Calculated Date the Arithmetic Formula solves for. For example, the **EF** value tells is we are solving for the Earliest Finish.

- The **rightmost two characters** tell us about the controlling Date Calculation Variable. In the above example, **du** value tells that the Earliest Finish is determined based on the **Subject Activity's** Duration.

---

1. To receive the five White Papers referenced in this book, follow instructions at the bottom of the Copyright Page, on the backside of the Title Page.
Again, we encourage you to interrupt your progress in this book, and take a few minutes to read the referenced ICS-White Paper. If not now, before we commence our discussion of Forward Pass and Backward Pass Date Calculations, then be certain to read the ICS-White Paper at the end of this chapter. You'll be glad you did!

5A4: Forward Pass Calculations through Default Restrictions

As a segue into our discussion of the actual Forward Pass Process for calculating Earliest Dates, it might be helpful to restate the meaning of “Earliest,” in Earliest Dates. Let's start with the two relevant definitions, found in the ICS-Dictionary:

- **Earliest Start:** An estimate of the earliest possible Calculated Date by which an Activity can be reasonably expected to start, assuming the flawless performance of all prior Activities to which the Activity is Logically connected in a Progressive Relationship.

- **Earliest Finish:** An estimate of the earliest possible Calculated Date by which an Activity can be reasonably expected to finish, assuming the flawless performance of all prior Activities to which the Activity is Logically connected in a Progressive Relationship.

Every Activity has a start and it has a finish. This is because every Activity in a Schedule represents a corresponding Action on the Project – and Actions consume Time. Therefore, every Action must have a starting Moment in Time and a finishing Moment in Time.

As we have discussed at length, the start of a Functional Process is depicted graphically by the left-most end of either (top or bottom) horizontal line of the Activity Box. Likewise, the end of an Functional Process is represented by the right-most end of either horizontal line of an Activity Box.

One of the questions most frequently asked by Schedule stakeholders, that a Project Facilitator must be able to answer is, “When is the earliest _____?” The question may come from a Subcontractor, as in, “When is the earliest you will need me on site?” Or it may come from a manufacturer or vendor asking, “When is the earliest you will need this material (or this equipment) delivered to site?” Or the Owner may ask, “When is the earliest I can expect to see the building dried-in?”

---

2 In the above context, the term, Activity, does not include Milestones, which many software programs consider to be zero-duration Activities. In Cognitive Project Management's lexicon, Project Execution Commitments are either Events or Milestones, the former consuming Time, and the latter not.
Knowing the *earliest* that an Activity can be reasonably expected to start or finish is important knowledge. This should lead you to an even more basic question: “What determines the earliest that an Activity can start or finish?” The answer, in a few short words, is “prior Logic and Durations.”

Any Earliest Start is determined by what comes before the Activity commences its own Functional Process. What “comes before” are other Activities and Logic that collectively constitute the Interdependencies between and among them. The Forward Pass Process, working from left to right across the Network Diagram, calculates the *earliest possible* Start Date and the *earliest possible* Finish Date for each Activity.

The operative word here is “possible.” It is good for you to remember that the Forward Pass Process concentrates on what is *possible* to occur which, in real life, is quite often different than what is probable ... or even improbable!

The Forward Pass Process employs simple addition. If you can add in your head, then you can perform a Forward Pass. Let’s begin with the three Activities shown in Figure F0502.

We see that:

- **Activity A** has an Activity Duration of 6 Workdays.
- **Activity B** has an Activity Duration of 9 Workdays.
- **Activity C** has an Activity Duration of 7 Workdays.

Imagine your boss asking you when **Activity C** can be expected to finish, “at the earliest.”

**Finding the Earliest Start for Activity A**: Let us assume that yesterday was Day 100 of the Project – and so, **Activity A** is actually commencing today, Day 101. Because we are using the End-of-Day Perspective[^3] for our Date Calculations, we establish the Earliest Start of

[^3]: If you need a refresher on the three Date Calculation perspectives (Start-of-Day, End-of-Day, and Point-of-Day), then read ICS-White Paper WPB-KJ-28, "Understanding the Point-of-Day Perspectives."

Activity A to be Day 100. We start by saying, aloud (or in our head), “The earliest that Activity A can start is … Day 100.” See Figure F0504.

We will place the Earliest Start at the top-left corner of the Activity Box.

5A4a: Finding Earliest Dates thru One Default Restriction

Before we go any further, this is a good time to introduce our formatting conventions for all standardized⁴ Arithmetic Formulas. We will use the following abbreviations within formulas:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES</td>
<td>Earliest Start</td>
</tr>
<tr>
<td>EF</td>
<td>Earliest Finish</td>
</tr>
<tr>
<td>LS</td>
<td>Latest Start</td>
</tr>
<tr>
<td>LF</td>
<td>Latest Finish</td>
</tr>
<tr>
<td>TF</td>
<td>Total Float</td>
</tr>
<tr>
<td>STF</td>
<td>Start Total Float</td>
</tr>
<tr>
<td>FTF</td>
<td>Finish Total Float</td>
</tr>
<tr>
<td>DU</td>
<td>Activity Duration</td>
</tr>
<tr>
<td>X</td>
<td>Restricting Activity</td>
</tr>
<tr>
<td>Y</td>
<td>Subject Activity</td>
</tr>
<tr>
<td>Z</td>
<td>Restricted Activity</td>
</tr>
</tbody>
</table>

When multiple Activities are involved in a single Arithmetic Formula, we will use subscripts to indicate which Activity we are talking about. Thus, the Earliest Start of Restricting Activity A would be written, $ES_A$.

We will use these letter combinations to refer to the different Performance Restrictions:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$dr$</td>
<td>Default Restriction (Finish-to-Start)</td>
</tr>
<tr>
<td>$sr$</td>
<td>Start Restriction (Start-to-Start)</td>
</tr>
<tr>
<td>$fr$</td>
<td>Finish Restriction (Finish-to-Finish)</td>
</tr>
</tbody>
</table>

⁴ This is how they appear in the ICS-White Paper, WPB-KI-13, How to Calculate Dates in the Critical Path Method.
We will use unique letter combinations to describe different Restriction Delays:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL(dr)_A</td>
<td>Restriction Delay through Default Restriction from Activity A</td>
</tr>
<tr>
<td>DL(sr)_A</td>
<td>Restriction Delay through Start Restriction from Activity A</td>
</tr>
<tr>
<td>DL(fr)_A</td>
<td>Restriction Delay through Finish Restriction from Activity A</td>
</tr>
</tbody>
</table>

Find the Earliest Finish for Activity A: Using these formula notations, we can now construct our first **DATE CALCULATION FORMULA**. The Earliest Finish for Activity A (**Figure F0504**) is found by adding the Activity Duration to the previously-determined Earliest Start. This explains the Simple Arithmetic Formula \( EF|du, \) for Subject Activity Y.

**SIMPLE FORMULA: \( EF|du \)**

\[
EF_Y = ES_Y + DU_Y
\]

In **Figure F0506** we apply Arithmetic Formula \( EF|du \) to **Figure F0504** and easily determine that the Earliest Finish of Activity A is Day 106. Here is our arithmetic work-up, wherein we substituted "Y" for "A":

**SIMPLE FORMULA: \( EF|du \)**

\[
\begin{align*}
EF_A &= ES_A + DU_A \\
ES_A &= \text{Day 100} \\
DU_A &= 6 \\
EF_A &= \text{Day 100} + 6 = \text{Day 106}
\end{align*}
\]

Find Earliest Start for Activity B: We are now ready to move on to Activity B, which gives us the first opportunity to see how "what comes before" factors into the Forward Pass Process. Look at **Figure F0508**. From the Logic Tie we understand that Activity B cannot start any earlier than the date on which Activity A finishes. From our previous Date Calculation we have determined that the Earliest Finish of Activity A is Day 106. This means that the
Earliest Start of Activity B is Day 106, likewise. [5]

Again, the earliest that Activity A can finish is six days after its Earliest Start (the end of Day 100), making the Earliest Finish of Activity A the end of Day 106. Since Activity B’s only restraint in terms of prior Logic is Activity A, we conclude that Activity B can commence the very second that Activity A finishes (that’s what the Logic Tie tells us). And, therefore, the earliest that Activity B can possibly start corresponds to the earliest that Activity A can possibly finish, which is (the end of) Day 106.

Let’s see if we can construct a Simple[6] Arithmetic Formula for an Earliest Start through a Default Restriction:

**SIMPLE FORMULA: ES\[dr\]**

\[ ES_Y = EF_X + DL_{dr_X} \]

where \( Y \) is the Subject Activity, \( X \) is its Restricting Activity, and \( DL_{dr} \) refers to the Restriction Delay of the Default Restriction Type. Now let’s substitute letters to reflect the problem in Figure F0508:

**SIMPLE FORMULA: ES\[dr\]**

\[
ES_B = EF_A + DL_{dr_A} \\
EF_A = Day 106 \\
DL_{dr_A} = 0 \\
ES_B = Day 106 + 0 = Day 106
\]

We just considered a condition where the Subject Activity is preceded by a single Restricting Activity, linked by way of a Default Restriction. For this simple condition we have presented you with a Simple Arithmetic Formula \( ES\[dr\] \). In the next subsection, however, we will present the Complex Arithmetic Formula for \( ES\[dr\] \), which you will use when you encounter multiple Default Restrictions coming into a single Subject Activity.

**Find Earliest Finish for Activity B:** Once again we get to apply Simple Arithmetic Formula \( EF\[du\] \). Since Activity B has a 9-day Duration, then the earliest that Activity B can possibly finish is nine days after its Earliest Start. Therefore, as shown in Figure F0510, the Earliest

---

5 Remember we are using the End-of-Day Perspective. If you are still fuzzy about Points-of-Day, reread ICS-White Paper WPB-KJ-28, “Understanding Points-of-Day Perspectives.”

6 Throughout this chapter we will present Arithmetic Formulas as either Simple or Complex. Simple Formulas apply to the limited condition of one Restricting Activity and one Restricted Activity. Complex Formulas apply to situations where there are either multiple Restricting Activities or multiple Restricted Activities.
Finish of Activity B is Day 115.

Applying the Simple Arithmetic Formula \( EF_{du} \), through the Activity Duration, we get:

\[
\text{SIMPLE FORMULA: } EF_{du} \\
EF_B = ES_B + DU_B \\
ES_B = \text{Day 106} \\
DU_B = 9 \\
EF_B = \text{Day 106} + 9 = \text{Day 115}
\]

Okay, now it is your turn to go solo! Use Figure F0512 to work out answers to these two questions:

- What is the Earliest Start of Activity C?
- What is Activity C’s Earliest Finish?

Find Earliest Start for Activity C: Here is how we determined that the Earliest Start of Activity C is Day 115. Since Activity B’s Earliest Finish equals Day 115, and since the only Performance Restriction to the start of Activity C is the prior completion of Activity B, then the Earliest Start of Activity C must be the same as the Earliest Finish of Activity B, or Day 115.

We will use Simple Arithmetic Formulas for the Earliest Start.

\[
\text{SIMPLE FORMULA: } ES_{dr} \\
ES_C = EF_B + DL_{dr_B} \\
EF_B = \text{Day 115} \\
DL_{dr_B} = 0 \\
ES_C = \text{Day 115} + 0 = \text{Day 115}
\]
Find Earliest Finish for Activity C: The arithmetic solution for the Earliest Finish of Activity C is Day 122, as shown in Figure F0514. We know that Activity C cannot begin any earlier than Day 115, and that Activity C has a 7-day Duration. Therefore, the earliest possible date that Activity C can complete is Day 122.

Now let’s use the Simple Arithmetic Formula for Earliest Finish through a Duration to prove our conclusion:

\[
\text{SIMPLE FORMULA: } EF_{du} = ES_C + DU_C
\]

\[
ES_C = \text{Day 115} \\
DU_C = 7 \\
EF_C = \text{Day 115} + 7 = \text{Day 122}
\]

5A4b: Finding Earliest Dates thru Multiple Default Restrictions

The previous section gave you your first exposure to the Forward Pass Process. It began by showing you how to calculate Earliest Dates through a series of three, sequential Activities (each Restricted Activity immediately preceded by a single Restricting Activity by way of a Default Restriction). Now we want to consider a situation where a single Restricted Activity has multiple Restricting Activities preceding it, and where each Performance Restriction is a Default Restriction.

Consider Figure F0516. What do you think is the Earliest Start for Activity M? Well, what do the Logic Ties tell us? For starters, we see that there are two Arrows, not one, coming into Activity M.\(^7\) From this we conclude that there are two separate Performance Restrictions for us to deal with.

We recall that the left-side vertical line of Activity M’s Activity Box represents the first

---

\(^7\) Don’t be confused by our style of drawing, where we join the two Arrows, since they are terminating at the same Restricted Activity. These are, nonetheless, two separate Logic Ties -- just simply overlapping out of convenience.
possible Moment in Time when Activity M is no longer restricted from starting by any prior Logic. In order to reach this Conditional State, both Activity K and Activity L must have reached their Conditional States of completeness. The upper Arrow comes from Activity K, so from this we understand that Activity M can start no earlier than when Activity K completes (at its earliest), which is given as Day 106.

But we also determine, from the lower Arrow, that Activity M can start no earlier than the Earliest Finish of Activity L, which is given as Day 111. This problem presents us with a different circumstance than we have dealt with thus far: two concurrent Performance Restrictions affecting the start of a common Restricted Activity.

It is time to introduce the **FORWARD PASS MULTIPLE RESTRICTING RULE** (Figure F0518) which, according to the ICS-Dictionary, states that:

**Forward Pass Multiple Restricting Rule:** When a Subject Activity is immediately preceded by multiple Restricting Activities, the Earliest Date being solved for is determined by the latest of the Computed Choices.
Let’s try to apply the **FORWARD PASS MULTIPLE RESTRICTING RULE** to our problem. First we need to determine our Computed Choices. Second, we need to select the “latest” of the Computed Choices.

**Determining our Computed Choices:** We start by applying the Simple Arithmetic Formula *separately* to each of the Performance Restrictions of Activity M.

For the Default Restriction between Activity K and Activity M, we compute:

**SIMPLE FORMULA:** \( ES_{dr} \)

\[
ES_M = EF_K + DL_{dr_K} \\
EF_K = \text{Day 106} \\
DL_{dr_K} = 0 \\
ES_M = \text{Day 106} + 0 = \text{Day 106}
\]

Now let’s solve the formula for the Default Restriction between Activity L and Activity M:

**SIMPLE FORMULA:** \( ES_{dr} \)

\[
ES_M = EF_L + DL_{dr_L} \\
EF_L = \text{Day 111} \\
DL_{dr_L} = 0 \\
ES_M = \text{Day 111} + 0 = \text{Day 111}
\]

**Selecting the “Latest” of our Computed Choices:** The second step is quite simple. We merely pick whichever of our Computed Choices is the *latest*, chronologically.

- Through Activity K, the Earliest Start of Activity M is **Day 106**.
- Through Activity L, the Earliest Start of Activity M is **Day 111**.
- The later of the two Calculated Dates is **Day 111**.
- Thus, the Earliest Start of Activity M is **Day 111**.

Do you understand why we use the *latest* Earliest Start value, and not the *earliest*? Imagine that manager Lionel (L) wants to be briefed by employee Kathy (K) on how a sales presentation went with a prospective client. Lionel schedules a short Meeting (M) to take place just as soon as the two of them can clear their calendars. Obviously it would make little sense for Kathy to begin her briefing before Lionel arrives. Nor could the meeting be of much value to Lionel if Kathy is not there. So, the *earliest* that the Meeting can start is when the *later* of the two attendees arrives in the conference room.

Back to our example, the earliest that Activity K can possibly finish is **Day 106**. The earliest that Activity L can finish is **Day 111**. It should now make sense to you that Activity M can start no earlier than **Day 111**, which is when the *later* of the two Restricting Activities
Calculating Primary CPM Dates

have reached its Conditional State of "finished."

Before we leave this topic, we want to introduce you to the Complex Arithmetic Formula to be used when you have multiple Default Restrictions coming into a single Restricted Activity. Don't let its length scare you; it is just a combination of several simple ES|dr Formulas strung together.

Here it is:

**COMPLEX FORMULA: ES|dr**

\[ ES_Y = \max \{ (EF_{X_1} + DL_{dr_{X_1}}), (EF_{X_2} + DL_{dr_{X_2}}), (EF_{X_n} + DL_{dr_{X_n}}) \} \]

This Complex Arithmetic Formula states that the Earliest Start for Subject Activity Y is equal to the maximum value derived from among multiple Computed Choices by way of Default Restrictions coming from Restricting Activity X₁, Restricting Activity X₂ ... all the way to Restricting Activity Xₙ.

**Figure F0520** presents a twelve-activity Subnetwork that starts on Day 112. Take a few minutes to determine the Earliest Finish for Activity M. You will have to perform a Forward Pass through the entire Subnetwork. See if you come up with an Earliest Finish of Day 157, as we did. Our solution is shown at **Figure F0522**.
5A5: Forward Pass Calculations through Start Restrictions

By now, you should be comfortable with the general concept of using simple addition to perform Forward Pass Date Calculations, when the Activities are linked solely by Default Restrictions. It is time for us to crank things up a notch. Now we will show how to calculate Earliest Dates across Start Restrictions. The basic approach is the same as with Default Restrictions, but the mental process is slightly harder (but not too much harder).

5A5a: Calculating Earliest Dates thru a Single Start Restriction

Look at Figure F0524. Here we see Activity B being restrained by Activity A by way of a Start Restriction bearing a three-day Restriction Delay. A Project Time Management practitioner would read this symbolic notation as, “Activity B is preceded by Activity A with an SS:3.”

From the Logic we see that the Earliest Start of Activity A is Day 100. Since Activity B can
5 Calculating Primary CPM Dates

start no earlier than three days after the start of Activity A (the meaning of an SS:3), then the Earliest Start for Activity B is Day 103. That’s easy, right? If you think it is, then try doing the problem in Figure F0526; we’ll talk you through it. What is the Earliest Finish (not Earliest Start) of Activity F?

First, let's introduce a new Arithmetic Formula, this one for determining the Earliest Start through a single Start Restriction.

\[
\text{SIMPLE FORMULA: } \text{ES}_{sr} \quad \text{ES}_Y = \text{ES}_X + \text{DL}_{sr}X
\]

where \( Y \) is the Subject Activity, \( X \) is its Restricting Activity, and \( \text{DL}_{sr} \) refers to the Restriction Delay of the Start Restriction Type.

Now let's apply this Simple Arithmetic Formula to the problem in Figure F0526, and solve for the Earliest Start of Activity F through a Start Restriction:

\[
\text{SIMPLE FORMULA: } \text{ES}_{sr} \\
\text{ES}_F = \text{ES}_E + \text{DL}_{sr}E \\
\text{ES}_E = \text{Day 100} \\
\text{DL}_{sr}E = 4 \\
\text{ES}_F = \text{Day 100} + 4 = \text{Day 104}
\]

To answer the question, we must first determine Activity F's Earliest Start! We are given that the Earliest Start of Activity E is Day 100, and that Activity F ties back to Activity E with a Restriction Linkage of SS:4. Therefore, the Earliest Start for Activity F must be Day 104 (100+4=104).

Now we can solve for the Earliest Finish of Activity F. Since Activity F has a 6-day Duration, the Earliest Finish for Activity F must be Day 110 (104+6). Of course we will use Simple Arithmetic Formula EF\(du\) through a Duration:

\[
\text{SIMPLE FORMULA: } \text{EF}_{du} \\
\text{EF}_F = \text{ES}_F + \text{DU}_F \\
\text{ES}_F = \text{Day 104} \\
\text{DU}_F = 6 \\
\text{EF}_F = \text{Day 104} + 6 = \text{Day 110}
\]

And that’s how you perform a Forward Pass through a single Start Restriction.
5A5b: Finding Earliest Dates thru Multiple Start Restrictions

Now let’s bump it up a little bit. Let’s mix together multiple Start Restrictions coming into the same Restricted Activity. Just remember the FORWARD PASS MULTIPLE RESTRICTING RULE. Working through the Logic in Figure F0528, can you calculate the Earliest Finish for ACTIVITY S?

Our solution to this problem appears in Figure F0530. Here is a narrative explanation of our approach. As you read, please pay particular attention to the order in which we calculate the various dates because the order matters.

Of course we start at the leftmost side of the Network, because all Arrows flow left to right, and so do all Forward Pass Date Calculations.

Solving for Activity P’s Earliest Finish: We are given that ACTIVITY P has an Earliest Start of Day 100. We can immediately calculate the Earliest Finish for ACTIVITY P which, based on its Duration of seven-days, yields an Earliest Finish of Day 107.

Solving for Activity Q’s Earliest Start: Now we turn our attention to Activity Q. We recognize a Start Restriction between ACTIVITY P and Activity Q, and that the Restriction Delay is four-days (SS:4).

We understand this to mean that ACTIVITY Q can start no earlier than four days after ACTIVITY P’s Earliest Start, which we already know to be Day 100.

---

8 We could have gone to either Activity Q or Activity R; it didn’t matter which one we attacked first. We just happened to start at the top and work down; but, no other real reason why.
Therefore, the Earliest Start for Activity Q is Day 104.

**Solving for Activity R's Earliest Start:** Using this same reasoning, we conclude that the Earliest Start for Activity R is Day 105.

**Solving for Activity S's Earliest Start:** Next we consider the Earliest Start for Activity S. The first thing we notice is that there are multiple Arrows terminating at the start of Activity S. We quickly recall the **Forward Pass Multiple Restricting Rule** and realize that we must select the latest of the Computed Choices.

Each Logic Tie (Arrow) represents a different Restriction Linkage. Before we can determine the latest of these three Performance Restrictions, we need to be clear on what each Restriction Linkage would independently yield in the way of an Earliest Start for Activity S. Let’s take them one at a time.

- **Based on Activity Q Only:** The Earliest Start of Activity S would be Day 106, because:
  
  SIMPLE FORMULA: ES\(_{sr}\)
  
  \[
  ES_S = ES_Q + DLsr_Q \\
  ES_Q = Day 104 \\
  DLsr_Q = 2 \\
  ES_S = Day 104 + 2 = Day 106
  \]

- **Based on Activity R Only:** The Earliest Start of Activity S would be Day 108, because:
  
  SIMPLE FORMULA: ES\(_{sr}\)
  
  \[
  ES_S = ES_R + DLsr_R \\
  ES_R = Day 105 \\
  DLsr_R = 3 \\
  ES_S = Day 105 + 3 = Day 108
  \]

- **Based on Activity P Only:** The Earliest Start of Activity S would be Day 107, because:
  
  SIMPLE FORMULA: ES\(_{dr}\)
  
  \[
  ES_S = EF_P + DLdr_P \\
  EF_P = Day 107 \\
  DLdr_P = 0 \\
  ES_S = Day 107 + 0 = Day 107
  \]

We will now introduce you to the Complex Arithmetic Formula that is applicable in situations like the one that Activity S faces in Figure F0530 — where the Subject Activity is restricted by multiple Start Restrictions.
This Complex Arithmetic Formula asserts that the Earliest Start for Subject Activity Y is equal to the maximum value derived from among multiple Computed Choices by way of Start Restrictions coming from Restricting Activity X_1, Restricting Activity X_2 ... all the way to Restricting Activity X_N.

Now we are able to determine that the latest of the two possible Earliest Start date Computed Choices coming through Start Restrictions is Day 108. They are:

**COMPLEX FORMULA: ES|sr**

\[ \text{ES}_Y = \text{max} \{ (\text{ES}_{X_1} + \text{DL}_{sr_{X_1}}), (\text{ES}_{X_2} + \text{DL}_{sr_{X_2}}), (\text{ES}_{X_n} + \text{DL}_{sr_{X_n}}) \} \]

**MIXTURE FORMULA: ES|mx**

\[ \text{ES}_{mx} = \text{max} (\text{ES|dr}, \text{ES|sr}) \]

But we must not forget that there is also a Default Restriction route to the Earliest Start of Activity S! In fact, Figure F0530 gives us our first encounter with a Subject Activity restricted by a combination of Restriction Linkage types. Activity S is restricted by a combination of different Restriction Linkages (Default Restriction and Start Restriction).

You are now ready to learn about what we call the Mixture Arithmetic Formula. There is one Mixture Arithmetic Formula for computation of the Earliest Start, and another for the Latest Finish. Let's discuss the former here. We'll cover the latter when we get to the Backward Pass Process.

**Multiple Arrows Reminder**

Whenever you encounter multiple Arrows coming into a single Restricted Activity, be sure to take into account ALL Restricting Linkages when calculating an Earliest Date.

F0532: Multiple Arrows Reminder

Let's apply the Mixture Arithmetic Formula \( \text{ES|mx} \) to find the Earliest Start of Activity S in Figure F0530.
5 Calculating Primary CPM Dates

MIXTURE FORMULA: ES|dr, ES|sr

$$ES_S = \max (ES|dr, ES|sr)$$

$$ES|dr_S = Day\ 107$$

$$ES|sr_S = Day\ 108$$

$$ES_S = Day\ 108$$

We hope you noticed that we could not determine the Earliest Start for Activity S until we had worked the Forward Pass through all of the Restriction Linkages that connected to the start of Activity S.

This is what we call the Multiple Arrows Reminder (see Figure F0532) which, according to the ICS-Dictionary, says:

**Multiple Arrows Reminder:** Whenever you encounter multiple Arrows coming into a single Restricted Activity, be sure to take into account ALL Restricting Linkages when calculating an Earliest Start.

The Multiple Arrows Reminder applies to all Forward Pass Procedures, whether you are solving for Earliest Starts or Earliest Finishes.

5A6: Forward Pass Calculations through Finish Restrictions

At last we can address the third Performance Restriction you will likely encounter as a Project Time Management practitioner. The arithmetic behind the Finish Restriction is quite similar to what we just did with the Start Restriction. The big difference, though, is what the Restriction Delay refers to. You may remember that we talked about this in Chapter Three. See if Figure F0534 jogs your memory.

With a Start Restriction the Restriction Delay actually refers to the first portion of the Restricting Duration. By contrast, with a Finish Restriction the Restriction Delay refers to the last portion of the Restricted Duration.

There is a Simple Arithmetic Formula for the computation of an Earliest Finish through a single Finish Restriction. It states that:

**SIMPLE FORMULA: EF|fr**

$$EF_Y = EF_X + DL_{fr_X}$$

where Y is the Subject Activity, X is its Restricting Activity, and DL_{fr} refers to the Restriction Delay of the Finish Restriction Type. If we apply this Simple Arithmetic Formula to the Activity-Pair on the right side of Figure F0534, we get the following workup:
SIMPLE FORMULA: EF\_fr

\[ EF_B = EF_A + DL_{fr_A} \]

\[ EF_A = Day 10 \]

\[ DL_{fr_A} = 3 \]

\[ EF_B = Day 10 + 3 = Day 13 \]

5A7: Forward Pass thru Combined Start/Finish Restrictions

Now let’s move to Figure F0536, but we had better take it a little slow because this Logic Diagram combines both a Start Restriction and a Finish Restriction. Let’s calculate the Start Restriction first. Based on the SS: 2 we understand that Activity B’s Earliest Start can be no earlier than two-days after Activity A’s Earliest Start – which is given as Day 100. Therefore, the Earliest Start for Activity B is Day 102. Next, we will calculate the Earliest Finish
Calculating Primary CPM Dates

for Activity A. Since the Activity Duration for Activity A is seven-days, and the Earliest Start is given as Day 100, then the Earliest Finish for Activity A is Day 107.

We are left with one date to calculate, Activity B’s Earliest Finish. We want you to read the Earliest Finish Arrow Count Rule (Figure F0538) and burn it into your mind, because it is very important!

It states that the number of Date Calculation Variables affecting an Earliest Finish is always one number greater than the number of Finish Restriction Linkages (Arrows) coming into the Activity. That one extra variable is the Activity Duration itself (added to the Earliest Start).

So in the example of Figure F0536 there is one Finish Restriction Linkage coming into Activity B (from Activity A), so that means that there must be two Date Calculation Variables influencing Activity B’s Earliest Finish. The ICS-Dictionary provides the Earliest Finish Arrow Count Rule, which states:

- Earliest Finish Arrow Count Rule: The number of Date Calculation Variables affecting the Subject Activity’s Earliest Finish is always one number greater than the number of Finish Restrictions coming into the Subject Activity. Note: The Activity Duration accounts for the additional Date Calculation Variable.

According to our Forward Pass Multiple Restricting Rule, we use the latest Computed Choice in calculating the Earliest Finish. What are the two Computed Choices? Well, one route is through the Activity Duration, and the other is through the Finish Restriction. Let’s solve each – using the appropriate Simple Arithmetic Formula.

Through the Finish Restriction: The Earliest Finish for Activity B is Day 109. How do we know this? Because we have already determined that the Earliest Finish of Activity A is Day 107 (see above), and the FF:2 tells us that Activity B cannot finish any earlier than two-days after Activity A finishes. So, through the FF:2 Finish Restriction, the Earliest Finish of Activity B is Day 109. (See Figure F0540.)
**SIMPLE FORMULA: EF|fr**

\[
EF_B = EF_A + DL_{fr_A}
\]

\[
EF_B = \text{Day 107}
\]

\[
DL_{fr_Q} = 2
\]

\[
EF_B = \text{Day 107} + 2 = \text{Day 109}
\]

**Through the Activity Duration:** Now let’s look at the other Date Calculation Variable: through the Activity Duration. We know that the Earliest Start for Activity B is Day 102 (see above). The Duration for Activity B is six-days so the Earliest Finish for Activity B is Day 108.

**SIMPLE FORMULA: EF|du**

\[
EF_B = ES_B + DU_B
\]

\[
ES_B = \text{Day 102}
\]

\[
DU_B = 6
\]

\[
EF_B = \text{Day 102} + 6 = \text{Day 108}
\]

Which one of the two influences is latest, Day 108 or Day 109? The answer, then, is that the Earliest Finish for Activity B is Day 109. And this introduces us to the Mixture Arithmetic Formula for an Earliest Finish, \( \text{EF|mx} \).

**MIXTURE FORMULA: EF|mx**

\[
\text{EF|mx} = \text{max (EF|du, EF|fr)}
\]

Are you ready to try one on your own? Great!

Okay, take a crack at Figure F0542, and tell us when Activity F can finish at the earliest. You will want to start with the Earliest Start for Activity F. After that, don’t forget to count the number of Finish Restrictions and add one more to come up with the number of Date Calculation Variables affecting the Earliest Finish Date for Activity F. With a sheet
of paper, cover up our solution, which we provide in Figure F0544. So what answer did you come up with?

If you said **Day 120**, you would be right! Way to go! Here’s the thinking process you should have gone through, and now you can follow along in Figure F0544.

**Solve for Activity F’s Earliest Start:** Activity F is preceded by Activity E with a Start Restriction SS:4, which means that Activity F can start no earlier than **four days** after Activity E’s Earliest Start, which is given as **Day 100**. Therefore, Activity F’s Earliest Start is **Day 104**.

**Solve for Activity E’s Earliest Finish:** The Earliest Finish for Activity E is determined by adding the Activity Duration of **12** to the Earliest Start of **Day 100**. Thus, the Earliest Finish for Activity E is **Day 112**.

**Solve for Activity F’s Earliest Finish:** We recall the **Earliest Finish Arrow Count Rule** and count the number of Finish Restrictions coming into Activity F. There is only one. Therefore, we know that there will be two Date Calculation Variables influencing Activity F’s Earliest Finish.

One Date Calculation Variable will transmit through the Finish Restriction and the other through the Activity Duration. Starting with the Finish Restriction, the FF:3 tells us that Activity F’s Earliest Finish is no less than **three days** after the Earliest Finish of Activity E, which we have calculated to be **Day 112**. Therefore, through the Finish Restriction, the Earliest Finish of Activity F is **Day 115** (Day 112 + 3 = Day 115).

Now we look at the other possible route to an Earliest Finish calculation through the Activity Duration. Activity F has a Duration of **16-days**, and an Earliest Start of **Day 104** (see above).

Thus, the Earliest Finish for Activity F (through the Activity Duration) is **Day 120** (Day 104 + 16 = Day 120). Abiding by the **Forward Pass Multiple Restricting Rule**, between the two Computed Choices, **Day 120** is later. Therefore, the Earliest Finish for Activity F is **Day 120**.

**5A8: Forward Pass: Putting It All Together**

It is time to test your understanding of the Forward Pass Process. Figure F0546 will challenge your brain for sure. It may require a half hour or more. But PLEASE ... take your time: all it takes is one little mistake and your entire effort will result in failure.
There is a great story about Englishman William Shanks, who:

- calculated [the value of] pi [out] to 707 digits. The accomplishment was hailed throughout the civilized world as the unveiling of a great mathematical truth, according to Joy of Pi by David Blatner. (Shanks had made a mistake after the 527th place, making all subsequent numbers wrong -- an error that was not discovered until 72 years later!)

Shanks wasted the better part of his lifetime on a calculation that was flawed halfway through. Take your time. You don’t want to repeat Shank’s mistake!

Here’s the problem: Working through the Subnetwork in Figure F0546 (page 160), find the Earliest Finish for the entire Subnetwork (which means, through Activity X). When you are done you can check your answer with ours. Our solution is provided in Figure F0548 (page 162). But be sure to try it yourself ... before looking at our solution.

5B: Introducing the Concept of Latest Dates

The calculation process that generates Latest Dates is called a Backward Pass. Frankly, the process is not much different than the Forward Pass except, (a) you start at the end of the Logic Diagram and work backwards, and (b) the basic arithmetic is subtraction, rather than addition. The results of the Backward Pass are the Latest Start and the Latest Finish Dates.

We think it would be worth taking a few minutes to discuss the meaning of “earliest” and “latest,” when compared to one another. As was explained previously in this chapter, when we speak of the “earliest” that an Activity can start or finish, we are associating an earliest performance to the Logic and Durations that precede the Activity. It is as if to say, “Given everything that must come ahead of this Activity, the earliest that this Activity can start (or finish) is ______.” In this sense, the word “earliest” speaks of possibility.

Latest Dates, on the other hand, do not speak of possibility; they speak of necessity. You see, working backwards from the end of the Schedule, where the Project Execution Commitment is carved in granite (pardon the construction pun), every “latest” date correlates the start or finish of an Activity to their necessary effect on one or more downstream Project Execution Commitments (a quite common Commitment being PROJECT COMPLETE). Let’s take a look at the ICS-Dictionary definition for Latest Start and Latest Finish:

- Latest Start: An estimate of the latest plausible date to which an start may be postponed without rendering as unachievable the required completion of any downstream Deadline Milestones to which the Activity is Logically connected in a Progressive Relationship.
F0548: Complicated Forward Pass of Subnetwork (Solution)
Notice the word “plausible.” We chose this word especially because it so precisely characterized the very nature of Latest Dates. According to Dictionary.com:

- **Plausible:** Having an appearance of truth or reason; seemingly worthy of approval or acceptance; credible; believable.

- **Plausible:** Describe that which has the appearance of truth but might be deceptive. The person or thing that is plausible strikes the superficial judgment favorably; yet it may or may not be true.

These definitions allow us, perhaps even invite us, to be a little skeptical and to treat Latest Dates with a certain amount of incredulity. And why shouldn't we?

The Backward Pass is no different than the Forward Pass in making the incredible assumption that all will go precisely as Planned! That is, every Duration will be achieved, every Logic Tie will be honored, every Start Restriction will happen just as planned, and every Finish Restriction will happen just as planned. Do you believe that? — that everything will go off perfectly ... with no exceptions?

Well, that's the assumption that Latest Dates are based upon. And so, if you are anything like us, then you will view Latest Dates with a fair amount of skepticism. And that is why our definition includes the word “plausible.” For, while Latest Dates have “an appearance of truth or reason,” they “may or may not be true.”

Getting back to the meaning of a Latest Start, we might word it this way: “If you want to maintain your ability to achieve downstream Commitment X, then the latest that you can start this Activity is Date ______. If you start later than Date ______, you will almost certainly not achieve that deadline.” This conclusion is based on all of the Logic and Durations that stand between this Activity and that downstream Project Execution Commitment.
5B1: Calculating Latest Dates thru Default Restrictions

We start with a simple string of two Activities, Activity A and Activity B, shown in Figure F0550. As we said before, the Backward Pass starts at the end of the Schedule and works backwards. From the diagram we note that Activity B must finish no later than Day 200.

Now look at Figure F0552 and follow along. As we just said, starting with Activity B, we are given that it must be finished by Day 200. This is one of those downstream Commitments that we mentioned at the top of the page.

Solve for Activity B’s Latest Start: Activity B’s Latest Finish is given as Day 200. Since Activity B has a nine-day Duration, then its Latest start must be Day 191.

Now is a good time to introduce another Arithmetic Formula, this one for a Latest Start based on a Default Restriction.

The Latest Start for a Subject Activity is found by subtracting the Activity Duration from the previously-determined Latest Finish. This calculation is expressed in Simple Arithmetic Formula LS|du, for Subject Activity Y through a Duration.

Let's apply the Simple Arithmetic Formula LS|du to Activity B in Figure F0552.

Simple Formula: \( LS|du \)

\[
LS_B = LF_B - DU_B
\]

\[
LF_B = \text{Day 200}
\]

\[
DU_B = 9
\]

\[
LS_B = \text{Day 200} - 9 = \text{Day 191}
\]

Solve for Activity A’s Latest Finish: Now let's move to Activity A. What do you think the Latest Finish is for Activity A? Well, if Activity B cannot start later than Day 191 (in order to
Calculating Primary CPM Dates

maintain a downstream Commitment – i.e., Day 200), then Activity A cannot finish any later than Day 191. It is time for another Arithmetic Formula ... for the Latest Finish of an Activity followed by only one Default Restriction.

**SIMPLE FORMULA: LF|dr**

\[ LF_Y = LS_Z - DL_{dr_Z} \]

where \( Y \) is the Subject Activity, \( Z \) is its Restricted Activity, and \( DL_{dr} \) refers to the Restriction Delay of the Default Restriction Type. Now let's substitute letters to reflect the problem in Figure F0552:

**SIMPLE FORMULA: LF|dr**

\[
\begin{align*}
LF_A &= LS_B - DL_{dr_B} \\
LS_B &= Day 191 \\
DL_{dr_B} &= 0 \\
LF_A &= Day 191 - 0 = Day 191
\end{align*}
\]

**Solve for Activity A's Latest Start:** Now we can move on to the Latest Start for Activity A. Applying the LS|du Arithmetic Formula, we get:

**SIMPLE FORMULA: LS|du**

\[
\begin{align*}
LS_A &= LF_A - DU_A \\
LF_A &= Day 191 \\
DU_A &= 6 \\
LS_A &= Day 191 - 6 = Day 185
\end{align*}
\]

Do you see how we are using subtraction and working backwards to perform a Backward Pass? Now let us introduce you to the Backward Pass Multiple Restricted Rule (Figure F0554) -- which is just the opposite of the Forward Pass Multiple Restricting Rule.
This is just the opposite of the Forward Pass Multiple Restricting Rule, where you use the latest of the Computed Choices.

Solve for Activity A's Latest Finish: Take a look at Figure F0556. The question we are asking you is, “What is the Latest Finish for Activity A?” Assume that Activity C’s Latest Finish is Day 112, and Activity E’s Latest Finish is Day 105. Since Activity C has a Duration of eight-days, then its Latest Start must be Day 104 (112-8=104). The same reasoning gives us a Latest Start of Day 99 for Activity E.

Since Activity A is a Restricting Activity linked to both Activity C and Activity E, then it cannot finish any later than either of its Restricted Activities (Activity C and Activity E) Latest Start Dates, right?

Can Activity A take until Day 104 to finish? No, because Activity E needs to start no later than Day 99. So, Activity A’s Latest Finish is Day 99. Hence the Backward Pass Multiple Restricted Rule, which tells us to take the earliest of the Computed Choices when calculating Latest Dates. Now let’s see if you understand how to do a Backward Pass through Default Restrictions. Take a stab at Figure F0558 and calculate the Latest Start for Activity A? Check your answer with ours; our calculations are shown in Figure F0562.
Now that you are more familiar and proficient with the concept of a Backward Pass, in this and the next subsection we will limit our comments to what is unique about Date Calculations across Start Restrictions and Finish Restrictions, respectively.

First, let’s look at what is most confusing about calculating Latest Dates through Start Restrictions. For some students new to the Critical Path Method of Modeling, it is hard enough when thinking forward (Forward Pass) to get their heads around the idea of what a Start Restriction is saying! Thinking about what it means in reverse is, for many, even harder to comprehend.

Consider the diagram in Figure F0560. We are being asked to determine the Latest Start for Activity A, right?

Do you recall the Earliest Finish Arrow Count Rule from the Forward Pass Process, which says that the number of Date Calculation Variables affecting an Earliest Finish is one greater than the number of Finish Restrictions coming into the Activity? Well, the reverse of that rule applies to Latest Start dates (which is, by no coincidence, diagonally opposite corner of the Earliest Finish Date in an Activity Box).

So how many Start Restrictions are coming out of Activity A in Figure F0560? Just one; to Activity B, right? That means that there must be two Date Calculation Variables influencing the Latest Start of Activity A. One travels through the Start Restriction. The other passes through the Duration, and subtracts from the Latest Finish.

During the Backward Pass Process, the number of Date Calculation Variables affecting a Latest Start is always one number greater than the number of Start Restrictions coming out of the Activity? We call this the Latest Start Arrow Count Rule (Figure F0564).

Solve for Activity C’s Latest Start: Back to Figure F0560; let’s do the math. We’ll start at Activity C, where the Latest Finish is Day 125. Since the Activity Duration for Activity C is nine-days, then the Latest Start for Activity C is nine-days earlier than its Latest Finish, or Day 116.
Solve for Latest Finish of Both Activity A and Activity B: We see that the performance of Activity C is restricted by both Activity A and Activity B, so that means that each of these Activities must be finished (Latest Finish) no later than Day 116, which is the Latest Start date for Activity C. We place Day 116 at the bottom-right corner of the Activity Boxes for both Activity A and Activity B. Here are the two related formulas:

**SIMPLE FORMULA: LF|dr**

\[
LF_A = LS_C - DL_{drC}
\]

\[
LS_C = Day 116
\]

\[
DL_{drA} = 0
\]

\[
LF_A = Day 116 - 0 = Day 116
\]
5 Calculating Primary CPM Dates

SIMPLE FORMULA: LF|dr
LF<sub>B</sub> = LS<sub>C</sub> - DL<sub>C</sub>dr
LS<sub>B</sub> = Day 116
DL<sub>A</sub> = 0
LF<sub>B</sub> = Day 116 - 0 = Day 116

Solve for Latest Start of Both Activity A and Activity B:
Now, from years of experience, we happen to know that we must determine Activity B’s Latest Start before we can attempt calculation of Activity A’s Latest Start. Why is this so? Let’s say you cannot explain why we would solve for Activity B’s Latest Start first. Just for the heck of it, let’s try to calculate Activity A’s Latest Start first (leaving Activity B’s Latest Start for later).

The first thing we do is count the Start Restrictions coming out of the Activity and apply the Latest Start Arrow Count Rule. We determine that there are two Date Calculation Variables affecting the Latest Start of Activity A, and only one of those travels through the Activity Duration. The other must come through that Start Restriction! Ah-ha! Now we of course know that we need to calculate the Latest Start of Activity B in order to determine the effect of the Start Restriction leading back to Activity A.

Solve for Activity B’s Latest Start: What is the Latest Start of Activity B? Well, its Latest Finish is Day 116, and its Activity Duration is 6. So we can conclude that the Latest Start for Activity B is Day 110.

SIMPLE FORMULA: LS|du
LS<sub>B</sub> = LF<sub>B</sub> - DU<sub>B</sub>
LF<sub>B</sub> = Day 116
DU<sub>B</sub> = 6
LS<sub>B</sub> = Day 116 - 6 = Day 110

Solve for Activity A’s Latest Start, through Start Restriction: Now what does the Latest Start of Activity B tell us about the Latest Start of Activity A? Let’s sit back in our chairs for a moment, and remember what a Start Restriction is all about.

An SS:2 Restriction Linkage tells us that two-days of work of the Restricting Activity must occur before the Restricted Activity can commence! So, before Activity B can commence, two days of Activity A must be performed. Well, if we must not start Activity B any later than Day 110, and Activity A needs to work for two days before Activity B can start, then the Latest Start date for Activity A is two days earlier than the Latest Start for Activity B, or Day 108!. At least … that’s the Latest Start as computed through the Start Restriction.
It is time to introduce another Simple Arithmetic Formula: $LS|sr$, to derive the Latest Start through a trailing Start Restriction:

\[
LS_Y = LS_Z - DL_{sr}
\]

where $Y$ is the Subject Activity, $Z$ is its Restricted Activity, and $DL_{sr}$ refers to the Restriction Delay of the Start Restriction Type. Now let's substitute letters to reflect the problem in Figure F0560 and solve for Activity A's Latest Start:

**SIMPLE FORMULA: $LS|sr$**

$LS_A = LS_B - DL_{sr}$

$LS_B = Day 110$

$DL_{sr} = 2$

$LS_A = Day 110 - 2 = Day 108$

**Solve for Activity A's Latest Start, through Duration:** Now we need to determine what the Latest Start would be through Activity A’s Duration. We see that the Latest Finish for Activity A is Day 116. The Duration for Activity A is five-days. So, calculated through the Activity Duration, the Latest Start for Activity A is Day 111 ($116-5=111$). Here's the formulaic solution:

**SIMPLE FORMULA: $LS|du$**

$LS_A = LF_A - DU_A$

$LF_A = Day 116$

$DU_A = 5$

$LS_A = Day 116 - 5 = Day 111$

**Solve for Activity A's Latest Start (minimum value):** The final step in finding the Latest Start of Activity A is to apply the Backward Pass Multiple Restricted Rule. This rule tells us to use the earliest of multiple Computed Choices. Between Day 111 and Day 108, we choose the earlier of them, and we record Day 108 as the Latest Start for Activity A.

**MIXTURE FORMULA: $LS|mx$**

\[
LS_{mx} = \min (LS|du, LS|sr)
\]

$LS|du = Day 111$

$LS|sr = Day 108$

$LS_A = Day 111$

$LS_A = Day 108$

Let us now test your understanding of how to calculate Latest Dates through Start Restrictions. Calculate all of the Latest Dates in Figure F0566. Our solution to the problem is presented in Figure F0568.
At last we are ready to bring this long chapter to a close. We have yet to discuss the calculation of Latest Dates through Finish Restrictions. Actually, their calculation is much the same as what you just learned about calculating Latest Dates through Start Restrictions.

The major difference is something we have already covered at length; that the Restriction Delay of a Finish Restriction refers to the last portion of performance of the Restricted Activity, whereas the Restriction Delay of a Start Restriction refers to the first portion of performance of the Restricting Activity.

Look at Figure F0570. We start at the end, since we are performing a Backward Pass. We are given that Activity L’s Latest Finish is Day 200. Since its Duration is 16-days, we deduce that its Latest Start is Day 184. That was the easy part.

**Solve for Activity K’s Latest Finish, through Finish Restriction:** Now … in order to figure out the Latest Finish for Activity K, we take into account the FF:5, which tells us that the last five-days of Activity L must wait on the prior completion of Activity K. Since Activity L has a
Latest Finish of Day 200, then Activity K must have its Latest Finish five days earlier, or Day 195. Alas, we have another Simple Arithmetic Formula, LF|fr, for calculating a Latest Finish through a trailing Finish Restriction. It states:

**SIMPLE FORMULA: LF|fr**

\[ LF_Y = LF_Z - DL_{fr_Z} \]

where \( Y \) is the Subject Activity, \( Z \) is its Restricted Activity, and \( DL_{fr} \) refers to the Restriction Delay of the Finish Restriction Type. Now let's substitute letters to reflect the problem in Figure F0570 and solve for Activity K's Latest Finish:

**SIMPLE FORMULA: LF|fr**

\[ LF_K = LF_L - DL_{fr_L} \]

\[ LF_L = \text{Day 200} \]

\[ DL_{fr_L} = 5 \]

\[ LF_K = \text{Day 200} - 5 = \text{Day 195} \]
Solve for Activity K’s Latest Start, through the Duration:
Now we apply the Latest Start Arrow Count Rule. We notice that there is only one Start Restriction coming out of Activity K, so there must be two Date Calculation Variables affecting the Latest Start of Activity K. Calculating through the Activity Duration, we subtract the Activity Duration of ten-days from the Latest Finish of Day 195 and thereby determine that, through the Activity Duration, the Latest Start for Activity K would be Day 185.

SIMPLE FORMULA: LS|_{du}
\[ LS_K = LF_K - DU_K \]
\[ LF_K = \text{Day 195} \]
\[ DU_K = 10 \]
\[ LS_K = \text{Day 195} - 10 = \text{Day 185} \]

Solve for Activity K’s Latest Start, through Start Restriction:
Now we calculate through the Start Restriction. Activity L’s Latest Start is Day 184, and there is an SS:3 Restriction Linkage tied back to Activity K. This tells us that Activity K must start no less than three days before Day 184, or Day 181.

SIMPLE FORMULA: LS|_{sr}
\[ LS_K = LS_L - DLsr_L \]
\[ LS_L = \text{Day 184} \]
\[ DLsr_L = 3 \]
\[ LS_K = \text{Day 184} - 3 = \text{Day 181} \]

Finally, recalling the Backward Pass Multiple Restricted Rule, choose the earliest Computed Choice, and thus establish Day 181 as the Latest Start for Activity K.
5B4: Backward Pass: Putting It All Together

And now it is time to call upon everything you have learned in this chapter. We have covered a lot of material, and your head may be swimming. Don’t let it get to you. We’ve hit you with the hardest material right up front. From here on through to the end of the book, you will find things getting much easier.

The amount of new material in a given chapter will be less, the complexity of the new material will be easier, and the repeated use of things previously covered will serve to reinforce what you have been exposed to. While you will be doing many more Forward Pass and Backward Pass Date Calculations in upcoming chapters, each time you do so, the concepts will become clearer and clearer.

For now, though, we would encourage you to do your best to develop a fairly solid understanding of what we have discussed in this chapter before proceeding to the next chapter.

We have found that those new to the Critical Path Method of Modeling make a common error when calculating the Latest Start of a Restricting Activity through a Start Restriction. Instead of subtracting the Restriction Delay from the Restricted Latest Start, they subtract it from the Restricting Latest Finish.

Referring back to Figure F0574, we have a Restriction Delay of two-days (per SS:2). The Restricting Activity is Activity A, and the Restricted Activity is Activity B. In trying to calculate the Latest Start for the Restricting Activity (Activity A) through the Start Restriction Linkage, instead of subtracting the Restriction Delay from the Restricted (Activity B’s) Latest Start, they would mistakenly subtract the Restriction Delay from Restricting (Activity A’s) Latest Finish. We hope you won’t fall into this trap.

So, now you are ready to apply everything you have learned about Forward Passes and Backward Passes. Figure F0576 is admittedly complicated, but it includes everything...
we have covered in this chapter. When you are finished, check your work against our solution in **Figure F0578**. And don’t just check the first and last Activities to see that you got the Latest Start and Earliest Finish Dates (respectively) correct. Check each Date Calculation in the problem. You may find mistakes along the way that, while not affecting the ultimate dates, still reveal a less than full understanding of the processes. Good luck!

**For** One last suggestion? We encourage you to set this book down ... again ... and reread the **ICS-WHITE PAPER WPB-KI-13, “HOW TO CALCULATE DATES IN THE CRITICAL PATH METHOD.”** It should make a whole lot more sense now than it did when you read it the first time, at the start of this chapter.
5  Calculating Primary CPM Dates

Activity A
SS: 1  
FF: 2
100 107
Activity B
SS: 2  
FF: 3
154 164
Activity C
SS: 2  
FF: 4
157 166
Activity D
SS: 3  
FF: 5
166 179
Activity E
SS: 2  
FF: 3
167 173
Activity F
SS: 3  
FF: 2
166 173
Activity G
SS: 2  
FF: 2
167 173
Activity H
SS: 2  
FF: 3
161 181
Activity I
SS: 3  
FF: 2
169 187
Activity J
SS: 2  
FF: 2
170 181
Activity K
SS: 3  
FF: 2
180 195
Activity L
SS: 3  
FF: 3
189 199
Activity M
SS: 3  
FF: 3
195 201
Activity N
SS: 3  
FF: 3
201 11212
Activity O
SS: 1  
FF: 2
156 162
Activity P
SS: 1  
FF: 2
156 162
Activity Q
SS: 1  
FF: 2
157 167
Activity R
SS: 1  
FF: 2
161 187
Activity U
SS: 1  
FF: 2
161 187
Activity V
SS: 1  
FF: 2
161 187
Activity W
SS: 1  
FF: 2
161 187
Activity X
SS: 1  
FF: 2
161 187
Activity Y
SS: 1  
FF: 2
161 187
Activity Z
SS: 1  
FF: 2
161 187
Activity AA
SS: 1  
FF: 2
161 187
Activity BB
SS: 1  
FF: 2
161 187
Activity CC
SS: 1  
FF: 2
161 187
Activity DD
SS: 1  
FF: 2
161 187
Activity EE
SS: 1  
FF: 2
161 187
Activity FF
SS: 1  
FF: 2
161 187
Activity GG
SS: 1  
FF: 2
161 187
Activity HH
SS: 1  
FF: 2
161 187
Activity II
SS: 1  
FF: 2
161 187
Activity JJ
SS: 1  
FF: 2
161 187
Activity KK
SS: 1  
FF: 2
161 187
Activity LL
SS: 1  
FF: 2
161 187
Activity MM
SS: 1  
FF: 2
161 187
Activity NN
SS: 1  
FF: 2
161 187
Activity OO
SS: 1  
FF: 2
161 187
Activity PP
SS: 1  
FF: 2
161 187
Activity QQ
SS: 1  
FF: 2
161 187
Activity RR
SS: 1  
FF: 2
161 187
Activity UU
SS: 1  
FF: 2
161 187
Activity VV
SS: 1  
FF: 2
161 187
Activity WW
SS: 1  
FF: 2
161 187
Activity XX
SS: 1  
FF: 2
161 187
Activity YY
SS: 1  
FF: 2
161 187
Activity ZZ
SS: 1  
FF: 2
161 187
Activity AAA
SS: 1  
FF: 2
161 187
Activity BBB
SS: 1  
FF: 2
161 187
Activity CCC
SS: 1  
FF: 2
161 187
Activity DDD
SS: 1  
FF: 2
161 187
Activity EEE
SS: 1  
FF: 2
161 187
Activity FFF
SS: 1  
FF: 2
161 187
Activity GGG
SS: 1  
FF: 2
161 187
Activity HHH
SS: 1  
FF: 2
161 187
Activity III
SS: 1  
FF: 2
161 187
Activity JJJ
SS: 1  
FF: 2
161 187
Activity KKK
SS: 1  
FF: 2
161 187
Activity LLL
SS: 1  
FF: 2
161 187
Activity MRR
SS: 1  
FF: 2
161 187
Activity NNN
SS: 1  
FF: 2
161 187
Activity OOO
SS: 1  
FF: 2
161 187
Activity PPP
SS: 1  
FF: 2
161 187
Activity QQ
SS: 1  
FF: 2
161 187
Activity RRR
SS: 1  
FF: 2
161 187
Activity UUU
SS: 1  
FF: 2
161 187
Activity VVV
SS: 1  
FF: 2
161 187
Activity WWW
SS: 1  
FF: 2
161 187
Activity XXX
SS: 1  
FF: 2
161 187
Activity YYY
SS: 1  
FF: 2
161 187
Activity ZZZ
SS: 1  
FF: 2
161 187
Activity AAAA
SS: 1  
FF: 2
161 187
Activity BBBB
SS: 1  
FF: 2
161 187
Activity CCCC
SS: 1  
FF: 2
161 187
Activity DDDD
SS: 1  
FF: 2
161 187
Activity EEEE
SS: 1  
FF: 2
161 187
Activity FFFF
SS: 1  
FF: 2
161 187
Activity GGGG
SS: 1  
FF: 2
161 187
Activity HHHH
SS: 1  
FF: 2
161 187
Activity II
SS: 1  
FF: 2
161 187
Activity JJ
SS: 1  
FF: 2
161 187
Activity KK
SS: 1  
FF: 2
161 187
Activity LL
SS: 1  
FF: 2
161 187
Activity MM
SS: 1  
FF: 2
161 187
Activity NN
SS: 1  
FF: 2
161 187
Activity OO
SS: 1  
FF: 2
161 187
Activity PP
SS: 1  
FF: 2
161 187
Activity QQ
SS: 1  
FF: 2
161 187
Activity RR
SS: 1  
FF: 2
161 187
Activity UU
SS: 1  
FF: 2
161 187
Activity VV
SS: 1  
FF: 2
161 187
Activity WW
SS: 1  
FF: 2
161 187
Activity XX
SS: 1  
FF: 2
161 187
Activity YY
SS: 1  
FF: 2
161 187
Activity ZZ
SS: 1  
FF: 2
161 187
Activity AAAA
SS: 1  
FF: 2
161 187

F0578: Forward Pass/Backward Pass Comprehensive Exercise (Solution)
CHAPTER SIX

6A Five Key Elements to All CPM Schedules

6B Schedule Float: Two Most Prevalent Types

6B1 Total Float: A General Understanding
6B2 Free Float: A General Understanding
6B2a Vignette #1: A Day in the Life of Marianne and Jerry
6B2b Vignette #2: An Upset in the Life of Marianne and Jerry
6B3 Understanding Total Float
6B3a Total Float Calculations
6B3b Calculating Total Float across Sequentially-Tied Activities
6B3b-i Why There are Two Total Float Values per Activity
6B3b-ii The Basic Total Float Formulas
6B3b-iii Why Total Float is a Measure of Path Flexibility
6B3b-iv Two Ways for Total Float to Change
6B3b-v Determining Total Float across Default Restrictions
6B3c Calculating Total Float within Overlapped/Fragmented Activities
6B3c-i About Start Total Float
6B3c-ii About Finish Total Float
6B4 Total Float: Ideological Discussions
6B4a Total Float is an Activity Path Value, Not Activity Value
6B4b About Most Critical Total Float
6B4c Total Float Quantities are Unreasonably Optimistic
6B4c-i Total Float is an Inconsistent Quantifier
6B4c-ii Total Float Reflects Extreme Optimism
6B4c-iii Total Float is Unreliable as a Measure of Schedule Flexibility
6B4d Total Float Grossly Unstable
6B4e Total Float Even Misunderstood by the “Experts”
6B4e-i List of Most Common Mistakes in Definition of Total Float
All About Total Float

6A: Five Key Elements to All CPM Schedules

There are basically five Key Elements to a CPM Schedule used as a Project Execution model. They are:

- Activities (Activity Identifier, Activity Description, Activity Duration)
- Activity Relationships (Communal, Symbiotic, Progressive)
- Activity Dates (Earliest Dates; Latest Dates)
- Schedule Float (Total Float; Free Float)
- Schedule Critical Path(s)

Once we have addressed the final two bullets, you will have been introduced to all of the Key Elements of a good Network-Based Schedule. But there is so much more for you to understand about how the Critical Path Method of Modeling actually works in the real world. Keep in mind: these are just the basic five elements.

With each new release, popular Scheduling Software programs introduce additional features that increase one’s potential to inadvertently obscure or overwhelm the basic mechanics of the Critical Path Method, as it was originally conceived. Part of the discussions beyond this chapter (and even this book) will be about how such Scheduling Software options can be exploited to good end -- or misused and even abused (intentionally, or not).

To be sure, with proper training you can make your CPM Project Execution Model dance! But without all of the information needed to fully understand CPM MECHANICS at the core, you could unwittingly send your Project into a tailspin from which it might never recover.

Some credit the Critical Path Method with having given the general business world two new terms: Float and Critical Path. Across industries well outside of construction, one can hear business people referring to some issue (or some person) as being “on the Critical Path.” And while maybe less assimilated into the common culture but still having clearly escaped the exclusive grip of the Project Management world, the term Float, too, has made its way into the common vernacular.
As for where we are headed next,

- This chapter tells us that there are two primary types of Schedule Float: Total Float and Free Float. The predominant focus of this chapter is on Total Float, far and away the most popular of the two.

- Once we have Total Float under our belts, Chapter Seven will turn to the preeminent concept of Critical Path, the namesake of the most popular of all Dynamic Project Time Management Tools. Since Critical Path determination requires prior calculation of Total Float, it makes perfect sense for these first two chapters to be in the order that they are.

- Finally, Chapter Eight looks at the other popular Float value: Free Float. This Float measure may be less understood and less used, but it still makes frequent appearances in CPM documents, so it needs to be examined and understood. The chapter closes with a short, but important, continuation our contrasting discussion of Symbiotic and Progressive Relationship Types.

6B: Schedule Float: Two Most Prevalent Types

Without question the two most prevalent types of Schedule Float are Total Float and Free Float (and in that order of popularity and use). At the outset of the chapter, we simply want to provide some general thoughts about these two main types of Schedule Float. You will be able to follow subsequent, detailed discussion only if you first have a general understanding of what each Schedule Float type is all about.

6B1: Total Float: A General Understanding

We may sound like a broken record, but just as with virtually every other key term or concept, there is disagreement in the Dominant Project Management world as to how Total Float is defined, even though authorities seem to agree, at a general level, on what it is.

What seems to be in agreement is that Total Float reflects the numerical difference between a Schedule's Earliest Dates and its Latest Dates. At least this much few qualified authorities quibble about. For instance, if a one-day Activity has an Earliest Start (and Finish) of **July 14**, and a Latest Start (and Finish) of **July 25**, then it has Total Float of **TF +11** (assuming a seven-day Workweek). It’s that simple, really. Total Float is the numerical difference between Earliest Dates and Latest Dates.
Don’t be too concerned if you don’t fully comprehend the concept or readily appreciate the significance of this difference. You will by the end of the chapter. For now, just know that Total Float is a numeric value that measures the difference between (a) the *earliest* that an Activity can reasonably be expected to take place and (b) the *latest* that an Activity can be performed without threatening downstream Deadline Milestones that require its prior performance.

Here’s a practical example. Suppose that you have a school paper to write, and you estimate that it will take two-hours to research, draft, proof, and finalize. You look at the clock and it is 1:30 p.m. The paper has to be turned in this evening by 7:00 p.m. (You are taking an evening course at the local college).

You quickly think about the amount of time it will take to drive to the college, find a parking space, and walk to class: all together about 45 minutes. So the latest you can leave your house is 6:15 p.m. (By now, we hope that you recognize the previous two sentences as a narrative version of a Backward Pass, needed to derive a Latest Finish time for an Activity entitled, *Write Paper*, an Activity with a two-hour Activity Duration).

Since the paper will take two hours to write, you calculate that the latest you can start on that paper is 4:15 p.m. Of course this Latest Start assumes that the two-hour Activity Duration is a reasonable estimate. If it should take longer than two hours — well, you are in trouble.

You look up at the wall clock and see that it is now 1:45 p.m. (you’ve been daydreaming for fifteen minutes). Since you can’t make the clock roll back in time, you accept that 1:45 p.m. is the *earliest* that you can start the paper.

Finally you remember reading in that *CPM Mechanics* book that Total Float is the numerical difference between Earliest Dates and Latest Dates. So you subtract 1:45 p.m. from 4:15 p.m. and quickly conclude that you have 2½ hours of Total Float to perform this paper-writing Activity.

Based on this mental calculation, you decide to take a two-hour nap after making yourself a quick 15-minute lunch. Using the calculator in your mind you reason that even with these two additional Actions, you will still have 2½ hours to get the paper written ... and get to class on time! Look at Figure F0602. It looks like you have 15 minutes to spare! That 15 minutes is the Total Float in this string of Sequential Activities.

Before we leave our example, why don’t you try to sketch it out as a simple Logic Diagram? Good ahead – take out a sheet of paper and start drawing Activity Boxes. You should have four Activities, based on the items above. You should be able to assign Activity Durations and perform both a Forward Pass (starting at 1:45 p.m.) and a Backward...
Pass (starting at 7:00 p.m.). Before looking at our work-up, Figure F0604 below, try doing your own first.

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>DUR</th>
<th>ES</th>
<th>EF</th>
<th>LS</th>
<th>LF</th>
<th>TF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make/Eat Quick Lunch</td>
<td>15</td>
<td>1:45</td>
<td>2:00</td>
<td>2:00</td>
<td>2:15</td>
<td>15</td>
</tr>
<tr>
<td>Take a Short, Two-Hour Nap</td>
<td>120</td>
<td>2:00</td>
<td>4:00</td>
<td>2:15</td>
<td>4:15</td>
<td>15</td>
</tr>
<tr>
<td>Write Paper</td>
<td>120</td>
<td>4:00</td>
<td>6:00</td>
<td>4:15</td>
<td>6:15</td>
<td>15</td>
</tr>
<tr>
<td>Drive to School/Park/Walk</td>
<td>45</td>
<td>6:00</td>
<td>6:45</td>
<td>6:15</td>
<td>7:00</td>
<td>15</td>
</tr>
<tr>
<td>Arrive in Class</td>
<td>0</td>
<td>6:45</td>
<td>7:00</td>
<td>7:00</td>
<td>7:00</td>
<td>15</td>
</tr>
</tbody>
</table>

F0602: Schedule Tabular Report

This example makes it easy to understand that Total Float is an expression of how much “extra” time a string of Activities collectively enjoys, with respect to a specific Deadline Milestone toward which the Activities, jointly progress.

What all of this means, and how one can use Total Float to better manage the use of Time on a Project, is a subject we will discuss at length later in this book. For now, you know enough about what Total Float is, in order to consider the other popular Float measure: Free Float.

F0604: Getting to School

6B2: Free Float: A General Understanding

Let us begin with the ICS-Dictionary definition of Free Float:

- Free Float: Applying only to situations in which multiple Activity Path Segments feed into a common downstream Restricted Activity, Free Float quantifies how much shorter (measured in consecutive Duration Days) a given Activity Path Segment is when compared against the longest Activity Path Segment feeding into the common downstream Restricted Activity. Thus, Free Float is expressed as either a positive number, or zero; it cannot be negative.
This definition uses a term that will be unfamiliar to Dominant Project Management adherents: Activity Path Segment. While we do not wish to divert our primary discussion, since the term is used in the ICS-Dictionary definition of Free Float we feel that we owe it to the reader to at least explain what an Activity Path Segment is.

- **Activity Path Segment**: In a CPM Network Diagram, an Activity Path Segment is a portion of an Activity Path that starts at either a Path Start or a Path Convergence Point, and ends at either a Path Finish or a Path Divergence Point.

- **Activity Path**: In a CPM Network Diagram, an Activity Path is a unique Series of Activities that spans from Path Start to Path Finish.

- **Path Divergence Point**: A Path Divergence point is a Path Finish Condition whereby a Subject Activity shares an Immediately-Restricting Progressive Relationship with more than one Restricted Activity.

- **Path Convergence Point**: A Path Convergence Point is a Path Start Condition whereby a Subject Activity shares an Immediately-Restricting Progressive Relationship with more than one Restricting Activity.

**6B2a: Vignette 1; A Day in the Life of Marianne and Jerry**

Let’s return to our discussion of Free Float. To illustrate several important characteristics of Free Float, it might be helpful to envision a Scenario that we can modify in various ways:

Jerry’s wife, Marianne, is a science teacher and yesterday she stayed after school to grade papers and catch up on other paperwork that had backlogged on her desk. Jerry called her around 4:30 p.m. to see about plans for dinner. He was at home, working on getting tax information together for the accountant, as well as simultaneously doing some laundry.

While they were on the phone, Marianne remembered that it was Aunt Lisa’s 85th birthday and that they ought to call her. But Aunt Lisa is hard of hearing and she also lives alone. Plus she is a talker. Any phone call with Aunt Lisa would take at least 30 minutes! Understandably, neither Marianne nor Jerry wanted to interrupt the work they were in the middle of, so they agreed to call just as soon as both of them were finished with their current tasks.

Jerry had just transferred a load of wet clothes into the dryer, and he wanted to wait for it to finish (before placing the call) so he could hang up dress clothes and avoid them getting wrinkled. A dryer cycle is 35 minutes and he needed 10 minutes to hang up the dress clothes (you know – the ones labeled “wrinkle-free”).
For her part, Marianne needed another **30 minutes** to finish grading homework assignments. So Marianne and Jerry agreed that they would make a three-way call from their respective locations just as soon as Jerry was finished with hanging up the dried clothes. [1]

Based on the above information, what would be the Earliest Start time for that birthday call? Draw a simple Logic Diagram of the situation, and compare it to ours in Figure F0606. The answer to the question is **5:15 p.m**, the Earliest Start of **MAKE BIRTHDAY CALL**.

Now we are rather certain that you got the same answer as we did, since the problem could be solved in your head. The more important question is whether you were able to take a word problem and turn it into a Logic Diagram! For while we may be discussing the concept of Free Float, we are also still working on increasing your overall mastery of **CPM Mechanics** -- which includes the ability to draw Logic Diagrams from scratch and perform Forward Pass and Backward Pass calculations.

So let’s think this through together. What are the Restricting Activities in the above vignette? Well, Marianne’s grading papers is one, and Jerry wrapping up the laundry is the other. That leaves the birthday call as the common Restricted Activity.

Most definitions of Free Float that we have seen seem to get it right (at least, partially): something along the lines of, “the amount of time that an Activity can slip without affecting any subsequent Activity.” [2]

So, applying this paraphrased definition, we see that Marianne’s work is expected to wrap up sooner than Jerry’s – and so, she is the one with the extra “Time to spare,” right? She has 15 minutes of extra Time, to be specific. Thus, we would conclude as follows:

- **Grade Papers** has **15 minutes** of Free Float
- **Dry/Hang Up Clothes** has **0 minutes** of Free Float

---

1 Since Jerry needed another 40 minutes but Marianne only needed another 30 minutes it seemed that, between the two of them, Jerry would be the one “holding up the call.”

2 We are paraphrasing an amalgamation of definitions, of course, but it suffices for the moment. Later, we will examine various Free Float definitions in much greater depth.
Back to the general understanding (in the Dominant Project Management world) as to how Free Float works, let us ask a question: If Marianne takes 40 minutes to perform her work, will her delay affect “any other Activity?” The answer is “no,” because her Activity enjoys Free Float (15 minutes of it, to be exact).

Now look at Jerry’s Activity. If his work delays (even by one minute), will his delay affect any other Activity? The answer is “yes.” The Make Birthday Call Activity would be slipped by as many minutes as Jerry’s laundry work extends beyond 45 minutes.

6B2b: Vignette 2; An Upset in the Life of Marianne and Jerry

That’s how Free Float works – in a nutshell. But, back to our opening explanation, we said that the Restricted Activity is preceded by “one or more” Restricting Activities. Let’s expand our example to include more than two Restricting Activities, just to liven things up a bit.

Let's imagine that one of Marianne’s students comes into the room while Marianne and Jerry are on the phone, and asks if she can take her make-up exam now. That exam will require 60 minutes. Marianne checks with Jerry and they both agree that the student should take her exam.

Based on this new information, modify your Logic Diagram to depict this changed situation. What is the new Earliest Start for the **Make Birthday Call** Activity? Our answer, 5:30 p.m., as shown in Figure F0608.

Now let’s reexamine the Free Float condition with respect to each Restricting Activity. Which Activity is now determining the Restricted Activity's Earliest Start? Well, let's apply our definition and see which Activity cannot slip, by even one minute, without causing the Restricted Activity to slip correspondingly.

It is clear from the Logic Diagram that the **Take Makeup Exam** Activity now determines the Earliest Start of the Restricted Activity. This means that the other two Restricting Activities have Free Float. And just how much Free Float does each have?
The correct answer is:

- **Grade Papers** has **30 minutes** of Free Float
- **Dry/Hang Up Clothes** has **15 minutes** of Free Float
- **Take Makeup Exam** has **0 minutes** of Free Float

We want you to take special note of the comparative aspect to Free Float. Notice that the Free Float for both of the above Activities has changed from one vignette to the other, even though each one's Activity Duration and performance Times did not change at all. The only thing that changed was the introduction of an *additional* Restricting Activity, the Earliest Finish of which was *later* than the Earliest Finish of either of the other two Activities.

By the way, the comparative nature of Free Float is clear proof that Symbiotic Relationships do indeed exist between Activities, even when the Activities are not Progressively tied to one another.

There is certainly much more to say about Total Float and Free Float, and we will get into all of it soon enough. For now, though, rest assured that you have learned enough about each Schedule Float type to at least know how they differ from one another.

**6C: Understanding Total Float**

By the end of this chapter you will have been presented with all you need to know about Total Float in order to function effectively as a Project Facilitator. We will look at Total Float from three different points of view.

- We will consider its mathematical underpinning.
- We will discuss Total Float from a philosophical perspective.
- Through a semantic lens, we will examine an assortment of current definitions of Total Float with an eye toward spotting what is technically incorrect about each definition. The more adept you become at recognizing the fallacies in definitions, the better you will understand what Total Float is, and is not.

**6C1: Total Float Calculations**

The latter two subsections of this chapter would be of little meaning or value to you without a solid understanding of how Total Float actually works. So, straight away, we need to get into the mathematics of Total Float calculations.

We will begin this line of discussion by way of a series of increasingly complex Logic Diagrams. Not only will this expose you to the most intricate workings of Total Float,
it will give you still more practice working with Logic Diagrams, and Forward Pass and
Backward Pass calculations. Let’s begin.

6C2: Calculating Total Float across Sequentially-Tied Activities

Take a look at Figure F0610, in which we see two Activities in an Immediately-Restricting
Progressive Relationship, Activity A being the Restricting Activity and Activity B being
the Restricted Activity. Since Total Float is the numerical difference between Earliest
Dates and Latest Dates, we need to first calculate the Four Basic Calculated Dates.

Using the information provided in Figure F0610, please perform Forward Pass and
Backward Pass Date Calculations and determine the Earliest Start, Earliest Finish, Latest
Start, and Latest Finish for Activity A and Activity B. The correct answers
are shown in Figure F0612.

You will be surprised to discover just how easy it is to compute the Total
Float values for Activities, once the Earliest Dates and Latest Dates have
been calculated. Finding the Total Float values is a simple matter of
subtraction. In Figure F0610 you will notice that we have included placeholders for two different Total
Float values per Activity.

6C2a: Why There are Two Total Float Values per Activity

At this point in the chapter it will not be immediately clear to you just why there are two
different Total Float values per Activity. As you will discover in the next few pages, when
an Activity is preceded by a Start Restriction, or succeeded by a Finish Restriction, there
is a possibility for Start Total Float and Finish Total Float values to differ within a single
Activity. But when Activities do not overlap, such as when only Default Restrictions link Activity, Start Total Float and Finish Total Float values are necessarily the same.

Since the next few examples and figures deal strictly with Default Restrictions, we will not see any difference between Start Total Float and Finish Total Float values just yet. But be patient: they will be different once we get to the next subsection, where we analyze Start Restrictions and Finish Restrictions.

6C2b: The Basic Total Float Formulas

The basic formula for Start Total Float is this:

\[
\text{START TOTAL FLOAT FORMULA} \\
\text{STFA} = LSA - ESA \\
LSA = \text{Day 112} \\
ESA = \text{Day 100} \\
\text{STFA} = \text{Day 112} - \text{Day 100} = +12
\]

Notice that we prefixed the numerical value with a plus sign (+) to indicate positive Start Total Float. This is because Total Float can just as easily be negative as positive.

Okay – now why don’t you try to calculate the Finish Total Float of Activity A. The basic formula for Finish Total Float is this:

\[
\text{FINISH TOTAL FLOAT FORMULA} \\
\text{FTFA} = LFA - EFA \\
LFA = \text{Day 118} \\
EFA = \text{Day 106} \\
\text{FTFA} = \text{Day 118} - \text{Day 106} = +12
\]

Did you come up with +12 as your answer? Here's how we calculated the Finish Total Float for Activity A:

\[
\text{FINISH TOTAL FLOAT FORMULA} \\
\text{FTFA} = LFA - EFA \\
LFA = \text{Day 118} \\
EFA = \text{Day 106} \\
\text{FTFA} = \text{Day 118} - \text{Day 106} = +12
\]
Now go on and calculate the two Total Float values for Activity B. Here is our workup:

\[
\begin{align*}
\text{STF}_B &= \text{LS}_B - \text{ES}_B \\
\text{LS}_B &= \text{Day 118} \\
\text{ES}_B &= \text{Day 106} \\
\text{STF}_B &= \text{Day 118} - \text{Day 106} = +12
\end{align*}
\]

\[
\begin{align*}
\text{FTF}_B &= \text{LF}_B - \text{EF}_B \\
\text{LF}_B &= \text{Day 127} \\
\text{EF}_B &= \text{Day 115} \\
\text{FTF}_B &= \text{Day 127} - \text{Day 115} = +12
\end{align*}
\]

6C2c: Total Float Belongs to the Activity Path

We see that Activity B's two Total Float values are also TF $+12$. Why do you suppose that is? How is it that all four Total Float values are the same? The answer, which may seem more than a little obvious, is nonetheless extremely important to articulate because it leads us to a major observation about Total Float. We call it the Path Total Float Law (Figure F0614) and it says that Total Float is an Activity Path attribute. Said differently, Total Float belongs to the Activity Path, and not to any specific Activity. [3]

Let’s analyze Figure F0616 (which is a repeat of Figure F0612, but with our Total Float calculations showing) and see what we can conclude from it. For starters, we see that the length of the Activity Path (comprised of Activity A and Activity B, linked by way of one Default Restriction) is 15 days long. We know this because it starts on Day 100 and ends on Day 115. This, of course, corresponds to the sum of the Activity Durations of the Activity Path's two Activities: (Duration 6 + Duration 9 = Duration 15).

3 Unless, of course, the Activity Path is comprised of only one Activity.
The diagram in Figure F0616 uses a dark Arrowhead to indicate what is called, in the Scheduling world, a Date Constraint. We will discuss Date Constraints at length in a later chapter.

But for now, just understand that Date Constraints establish the outer temporal boundaries for performance. For instance, the right-pointing Arrowhead represents what is called a **START-NO-EARLIER-THAN** Date Constraint (abbreviated **SNET**). It tells us that **ACTIVITY A** can start *no earlier than Day 100*. Likewise, the left-pointing Arrowhead symbolizes a **FINISH-NO-LATER-THAN** Date Constraint (abbreviated **FNLT**). It tells us that **ACTIVITY B** must finish *no later than Day 127*.

From the information in the Network Diagram of Figure F0616, we can draw a few conclusions:

- We have **27 days** to complete the Network's Activities. We know this because the **SNET** tells us that the “Time Now” is assumed to be **Day 100**, and we are told by the **FNLT** that we have until **Day 127** to finish (a Project Execution Commitment [4]).
- We can reasonably expect the work of the Network Diagram to take **15 days**. We know this from the Forward Pass.
- By subtraction, we conclude that this Network of Activities (and, more specifically, the one and only Activity Path within it) has **12 days** of Time *in excess of* what is needed to get the work done. This is the real meaning of Total Float.

6C2d: **Why Total Float is a Measure of Path Flexibility**

Here is a question for you to consider: Does the Total Float belong to any one Activity, or is it in fact a shared value? For instance, if **ACTIVITY A** does not start until, say, **Day 112**, will **ACTIVITY A** have not just “eaten up” whatever extra Time either of the Activities previously had?

Let’s see? Look at Figure F0618. Once **ACTIVITY A** slips by **twelve days**, its Earliest Dates increase by **twelve days** … and so do the Earliest Dates of **ACTIVITY B**. So we see that Total Float is really this “first-come, first-served” kind of value. Any Time that an Activity in a Schedule consumes available Total Float, it necessarily reduces the Total Float for all downstream Activities that occupy the same Activity Path. This proves the **PATH TOTAL FLOAT LAW**, doesn't it?

6C2e: **Two Ways for Total Float to Change**

Another observation we can make about how Total Float works is that it can be affected by a change in either the Earliest Dates or Latest Dates. This should make perfect sense, since Total Float is a subtraction of those two values.

---

4 A Fixed Milestone (Deadline Milestone), to be specific.
In mathematics, if you change either term of a subtraction operation, either the subtrahend (value being subtracted) or the minuend (value being subtracted from), you will change the arithmetic difference between them.

In Figure F0618, we experimented with changing the Earliest Dates, when we changed the Earliest Start of Activity A from Day 100 (as it was in Figure F0616) to Day 112. This of course changed the Earliest Dates right down the length of the Activity Path, through the end of Activity B. Now let’s restore the Earliest Dates of Figure F0616, but this time let us change the Latest Dates based on new information from the Project's Owner. We are told that the Owner wants the Scope of Work (represented cumulatively by Activity A and Activity B) to be completed no later than Day 117. So we assign this FNLT value as the Latest Finish for Activity B and we perform another Backward Pass. See Figure F0620, which tells us that the two Activities in the Activity Path now enjoy only 2 days of Total Float between them.

What we see from this example is that Total Float can change as a result of adjustments to the Latest Dates, or to the Earliest Dates. This example also provides us with yet more evidence that Total Float does indeed belong to the Activity Path, and not to any
F0622: Tracing Least Float Path (Problem)
F0624: Tracing Least Float Path (Solution)
specific Activity along that Activity Path.

6C2f: Determining Total Float across Default Restrictions

Before we move on to the calculation of Total Float among Overlapped Activities, let’s work through a fairly large Logic Diagram to make sure you really understand how to calculate Start Total Float and Finish Total Float.

Using Figure F0622, perform the following five steps:

- Perform a Forward Pass and calculate the Earliest Start and Earliest Finish for all Activities in the Subnetwork.
- Perform a Backward Pass and calculate the Latest Start and Latest Finish for all Activities in the Subnetwork.
- Calculate the Start Total Float and Finish Total Float for each Activity.
- Determine the Subnetwork’s Least Total Float value.
- Using a highlighter, trace the Activity Path with the Least Total Float, from the beginning Moment in Time of Activity A and terminating at the ending Moment in Time of Activity M. Our solution appears in Figure F0624.

6C3: Calculating Total Float within Overlapped/Fragmented Activities

As we said a couple of pages back, when an Activity overlaps another Activity there is always a possibility that its Start Total Float will be different than its Finish Total Float.

5 Overlapped Activities are connected by way of Start Restrictions and/or Finish Restrictions.
This is true because a Fragmented Activity is one which potentially or actually *splits apart* due to being tied by one or more Start Restrictions or Finish Restrictions.

Cognitive Project Management refers to Overlapped Activities as Fragmented Activities. We chose the word *fragmented* because it helps us to understand what is actually happening when we link two Activities by way of Start Restriction or Finish Restriction Linkages. As Figure F0626 illustrates, in the case of a Restricted Activity into which a Finish Restriction Linkage terminates there is the *potential* during the Forward Pass for the ending (affected) portion of the Restricted Activity to be detached from the beginning (non-affected) portion of the Restricted Activity.

Studying Activity B we see that, by “detached,” we mean that the work of the Activity may come to a halt as it waits for the condition(s) imposed by the Finish Restriction to be met, in order for Activity B to resume and for its ending part to be accomplished.

In Figure F0628, our focus turns to the Restricting Activity, Activity A, which links to its Restricted Activity partner through a Start Restriction Linkage. As illustrated, we see that there is a potential during the Backward Pass for the beginning (affected) portion of the Restricting Activity to become detached from the ending (non-affected) portion of that same Restricting Activity.

Returning to our point, among Fragmented Activities there is always a possibility that the Start Total Float can be different than the Finish Total Float. Examine Figure F0630, which provides an example of a situation where the Start Total Float and Finish Total Float within a single Activity, Activity B, are different. We want to take a minute to make sure that you really understand what the different Total Float values are telling you.
Figure F0632 presents the same Logic Diagram as Figure F0630, but it highlights two different possible Activity Paths between Activity A’s starting Moment in Time and Activity B’s ending Moment in Time.

Notice that the Solid Red Path travels completely through Activity A (Activity Duration, 11 days) and through the ending portion of Activity B (Restriction Delay of two-days, from the FF:2). Because the final two-days of Activity B must await the completion of Activity A in its entirety, we compute that it will take 13 days to get from the start of Activity A to the end of Activity B, following the Solid Red Path.

Now look at the Dotted Blue Path, which travels through the beginning portion of Activity A (Restriction Delay of three-days, from the SS:3) and completely through Activity B (Activity Duration, six-days). This route reflects the need for the entire Scope of Activity B to await the performance of the first three-days of Activity A. Accordingly, we compute that it would take nine-days to get from the start of Activity A to the end of Activity B, following the Dotted Blue Path.

So, Solid Red Path is four days longer than Dotted Blue Path, which corresponds to the difference in Total Float values of TF +7 and TF +3. To further explain, had it not been for the Finish Restriction altogether, Activity B’s end could be reached four days sooner than with the FF:2 imposed, as it is. And that is why the Finish Total Float is four days less than the Start Total Float; because the ending portion of Activity B is simply more critical to the overall string of Logic (in this case, Activity A and Activity B) than is Activity B’s beginning portion.
Now you understand why Fragmented Activities can have (but not necessarily must have) different Start Total Float and Finish Total Float values. By contrast, Non-Fragmented Activities\(^6\) will always have the same Start Total Float and Finish Total Float values --- because the Activity is not being interrupted once it starts.

Equipped with a fairly good comprehension of how Total Float is calculated, it is time to step things up a notch. In the previous subsection we considered Progressive Relationships where the Activities were linked one after the other, with no overlaps. Now we want to consider the Fragmented Activity condition, which is brought on by the interference of Start Restrictions or Finish Restrictions.

6C3a: About Start Total Float

At this point you should fully grasp the concept behind a Start Restriction Linkage. It says that “some beginning portion of a Restricting Activity must be accomplished before the Restricted Activity can even commence.” To repeat from earlier, the formula for Start Total Float is:

\[
\text{START TOTAL FLOAT FORMULA} \quad \text{STF}_Y = \text{LS}_Y - \text{ES}_Y
\]

As was discussed in Chapter Three, in the Scheduling world there is active debate over the question of what a Start Restriction Linkage really means. The two possible interpretations are: a Passage of Time, or Work Performance. Cognitive Project Management subscribes to the latter interpretation, and fully rejects the idea that a Passage of Time should ever be portrayed through either Start Restrictions or Finish Restrictions.

Moving on, some Scheduling authorities have pointed to the two Total Float values phenomenon as justification to return to the days of Arrow Diagramming (which we have not yet discussed in this book). Arrow Diagramming Method (ADM) was the format of the original Critical Path Method, used throughout its first few decades.

While a more complete treatment of ADM is found in Chapter Fourteen, we wish to note here one of its dominant features: that it only utilized one Restriction Linkage Type: Default Restriction (Finish-to-Start). As a consequence, this meant that, in practice, if two Actions out on the Project were expected to occur in an overlapped manner, four Activities would have to be created in the Schedule to represent that expectation.

To clarify the previous sentence, let’s consider the situation illustrated in Figure F0634. In the Upper Panel we see two overlapped, Immediately-Restricting Activities, not unlike

---

\(^6\) Activities linked only by way of Default Restrictions.
many others we have been working with ever since we introduced Fragmented Activities. In this example, the Contractor needs to lay a storm pipe that will run from a connection point (out on a city street) up to the junction at the base of the new building. Obviously this pipe will reside underground, and so the Restricting Action is to excavate a trench into which the pipe will be placed, and later covered over.

F0634: Analyzing Activity Fragmentation
It is estimated that the **Excavate Trench** Activity will take **11 days**, whereas the **Lay Storm Water Pipe** Activity will only take **6 days**. The Contractor wants to get about **25%** of the trench cut in before he brings the pipe-laying crew into the picture. At the other end of the pair of Fragmented Activities, it makes complete sense that pipe-laying cannot complete any sooner than the completion of trench excavation.

In fact, for the same reason that there was a bit of an offset at the start of the two Activities, the Contractor doesn’t want the pipe-laying crew to be “right on the excavator’s butts,” as he told us. So, we use a Finish Restriction Linkage to simulate a condition whereby the pipe-laying crew wraps up their work **two days** after the trench excavators have completed their work.

From the Earliest Dates and Latest Dates we calculate Start Total Float and Finish Total Float, respectively, for each Activity. What we find is similar to what we saw back in **Figure F0630**, where the controlling Activity Path went through the *entire length* of the Restricting Activity (in this case, **Excavate Trench**) and through the ending *portion* of the Restricted Activity (**Lay Storm Water Pipe**).

Now let us convert the Fragmented Activities into sequential, non-Fragmented equivalents. In the **Lower Panel** you see that we have split each of the **Upper Panel** Activities into two distinct Activities. And so, **Excavate Trench** is now represented by **Start Excavating Trench** and **Finish Excavating Trench**. Likewise, **Lay Storm Water Pipe** is replaced by **Start Laying Storm Water Pipe** and **Finish Laying Storm Water Pipe**. We have also allocated the Activity Durations of the original Activities to the new Activities, guided by the Restriction Delays as provided in the **Upper Panel**.

The **Lower Panel** makes it much easier to see why the beginning portion of the Restricted Activity in the upper panel (**Lay Storm Water Pipe**) has a Start Total Float that is larger than its Finish Total Float. The **Lower Panel** clearly illustrates that the beginning portion of pipe-laying will have a little *extra time* than will the ending **two-day** portion of the same Activity.

Some Scheduling authorities (typically old-timers whose first forays into Network-Based Scheduling took place in the 1970s, or earlier) lament over the demise of the Arrow Diagramming Method and pine for a return to the “good ole days” when ADM was the only way to model a Project. Expression of this yearning ranges from the sentimental to the passionate, and even to the defiant.

At their most extreme, there are those who argue that the current modeling method (Precedence Diagramming Method, PDM) – the form strongly endorsed by Cognitive Project Management and widely adopted throughout the world -- is *not* a legitimate way to depict Schedule Logic because it allows Overlapped Activities.

---

7 As would have been necessary had we been diagramming using the Arrow Diagramming Method (ADM).
Only slightly less extreme, are those who are willing to accept that PDM is a legitimate Project Modeling method, but insist that the use of Start Restrictions or Finish Restrictions are “bad practice” and should be outlawed. The vast majority of ADM loyalists insist that the use of Start Restrictions and Finish Restrictions constitutes Schedule manipulation and enables fraudulent Scheduling (and Project Management) practices.

It is the official position of Cognitive Project Management that the Critical Path Method is, after all is said and done, a method of modeling Project Execution Strategy that comes to us complete with specially-devised symbolism. As such, any shortcuts that make it easier or more effective for the Project Time Management practitioner to model the intentions of the Project Team can only be construed as helpful.

Consider a court reporter, who is taught a very specialized set of abbreviations and symbols that allow the reporter to capture the spoken words of others in rapid manner. Or think, perhaps, of an Architect who is taught a full set of symbols and abbreviations used to graphically depict spatial and aesthetic concepts as they are verbally described by a hopeful Project Owner, the Architect’s client.

How are these shortcuts any different than the symbolism of the Restriction Codes or Logic Ties (Arrows), Date Constraints, Work Performance Calendars, various Software Settings, and even Activity Durations themselves? Each of these will be discussed in much greater length elsewhere in this book. Aren’t they all just different ways for a Project Facilitator to model a Project Execution Strategy that starts out in the heads of the Project Executors?

We think of the Start Restriction or Finish Restriction Linkages as contractions: just like the word “can’t” means “cannot;” like “won’t” means “will not;” like “doesn’t” means “does not.” There are some old-Timers (probably back in England) who think that contractions are an atrocity. But, across the planet, folks are able to understand contractions just as well as the fully-worded counterparts that they effectively shorten.

The argument that a Start Restriction is not a legitimate representation of Project Execution Strategy is, to our minds, a foolish one. As long as all parties who use it understand what it signifies or depicts, how can this Restriction Linkage be anything other than legitimate? During Schedule Logic Development Sessions, when the Schedule’s underlying Logic is being built for the first time, it is crucial that each participant understands what a Start Restriction means and how its computations work. That is why it is an ICS-Protocols requirement that every new Logic Development Session commence with a short fifteen-minute primer on the three dominant Restriction Linkages found in Progressive Relationships.
Recognizing Fragmented Activities (Problem)

Total Activities: 16
Non-Fragmented Activities: 13
Fragmented Activities: 3
Potentially: 3
Actually: 2

Activities:
- Activity F
- Activity G
- Activity H
- Activity I
- Activity J
- Activity K
- Activity L
- Activity M
- Activity N
- Activity P
- Activity R
- Activity T
- Activity V

Early Start (SS):
- Activity F: 200
- Activity G: 11
- Activity H: 8
- Activity I: 12
- Activity J: 9
- Activity K: 7
- Activity L: 12
- Activity M: 7
- Activity N: 5
- Activity P: 10
- Activity R: 13
- Activity T: 12
- Activity V: 5

Free Float (FF):
- Activity F: 3
- Activity G: 5
- Activity H: 6
- Activity I: 6
- Activity J: 4
- Activity K: 4
- Activity L: 6
- Activity M: 8
- Activity N: 7
- Activity P: 8
- Activity R: 8
- Activity T: 4
- Activity V: 250

Copyright, 2012 International Center for Scheduling, Inc. (ICS-Global) All rights reserved.
Let us close this subsection with a reminder that Start Total Float, like all Total Float, is a comparative value. In the case of Start Total Float, the numeric value gives us a sense of the amount of wiggle room that the beginning portion of an Activity has. In practice, Contractors use Start Total Float to get a sense of how urgently they need to start a given Activity and whether anyone (downstream) might be waiting on that opening portion of Activity work to be completed. Contractors also compare Start Total Float and Finish Total Float values to get a comparative sense of what is the more critical to overall Project Execution Strategy -- the start of the Action or its full completion. Said differently, Start Total Float helps Contractors prioritize their efforts.

6C3b: About Finish Total Float

The same can be said for Finish Total Float – just in reverse. It tells us that some ending portion of a Restricted Activity must await the prior completion of a Restricting Activity in its entirety. And just like Start Total Float, Finish Total Float tells a Contractor where to put his emphasis and what is most important to timely Project Execution.

The basic formula for Finish Total Float is this:

\[ \text{FINISH TOTAL FLOAT FORMULA} \]

\[ \text{FTF}_Y = \text{LF}_Y - \text{EF}_Y \]

As we wrap up this section on the calculation of Start Total Float and Finish Total Float, we want to make sure that you (a) know how to determine it, no matter how complex the Schedule Logic may be and (b) that you know how to interpret what Total Float tells you.

Toward this end, please perform all calculations for the Logic Diagram shown in Figure F0636. Calculate Earliest Start, Earliest Finish, Latest Start, Latest Finish, Start Total Float, and Finish Total Float for each Activity. When you are done, identify the Path of Logic that collectively determines the minimum length of Time that this entire Subnetwork of Activities will most likely consume. When you are done, you may wish to compare your results with ours, shown in Figure F0638.

Here are two final questions meant to drive home the significance of Fragmented Activities:

- How many potentially Fragmented Activities did you find?
- Of these, how many were actually Fragmented Activities?

6D: Total Float: Ideological Discussions

Schedule Float, in particular Total Float, is a hugely popular and essential Project Time Management informational value. Elsewhere in the ICS-Compendium, we discuss the
F0638: Recognizing Fragmented Activities (Solution)
Critical Path Method of Managing a Project where we note that Dominant Project Management extends great reverence to the concept of Total Float. Consider that:

- Total Float is the basis for determination of the Critical Path.
- Total Float is the primary basis for prioritizing Project Actions, where Timely performance is a major Project Management concern.
- Total Float, along with the Critical Path concept, constitutes the ultimate proof that a Time Extension of one or more Deadline Milestones is contractually warranted.

With so much depending on a correct determination of Total Float, it never ceases to amaze us just how much confusion and misunderstanding there is about the topic. Even well-respected Project Time Management experts are often misguided, misinformed or, worse, misinforming others about Total Float. While one could write a whole book just on this one point, in the interest of providing this text’s readers with the straight scoop, we will briefly touch on five ways in which Total Float is all too commonly misunderstood.

### 6D1: Total Float is an Activity Path Value, Not an Activity Value

Perhaps the greatest misconception is the idea that Total Float is an Activity-level unit of measure. We understand why it is easy to reach this erroneous conclusion. After all, didn’t we just finish a subsection in which we repeatedly defined Total Float as the difference between Earliest Dates and Latest Dates? And aren’t Earliest Dates and Latest Dates Activity-level pieces of data?

But, as we shall now clarify, Total Float is not an Activity-level value. Rather, Total Float belongs to an Activity Path, to a string of Activities that (a) are related to one another by way of Immediately-Restricting Progressive Relationships, and (b) collectively determine the earliest possible...
completion of a single, common, downstream Deadline Milestone.

In Figure F0640 we see five Activities, strung together through Finish Restrictions, which means that their individual Start Total Float and Finish Total Float values will be the same. There is one Activity Path, with a combined length of 25 days, with each Activity requiring an equal amount of Time to get its individual work done: five days. From the Latest Finish of Activity E, we see that the overall Project grants 30 days to get the Schedule’s work accomplished. Hence, the Activity Path, as a whole, enjoys five days of Total Float.

So, the question is this: Can Activity A consume all five days of Total Float as it performs its work? Will doing so, in and of itself, cause Activity E to finish late? Said more bluntly, can Activity A eat up all of the available Total Float?

The technically correct answer is, “Yes, Activity A can.” But then, we asked the wrong question, didn’t we? We should have asked, “Is it fair to the other four Activities that Activity A should eat up all of the Total Float?” But an even better question to ask is this: “If Activity A does eat up all of the Total Float, do the other Activities still have their five days of Total Float?”

You’re probably re-reading that last question, to see if it is a trick question, or if you misread it in the first place. Well, Surprise! Surprise! It was a trick question! No downstream Activity loses “their” Total Float, because it never was theirs to begin with! The set of five Activities collectively has a five-day surplus of Time … before the work actually begins. That is what the Logic tells us: that these five Activities are collectively expected to take 25 days, and the Deadline to which these five Activities jointly lead is 30 days away. Any one or more of the Activities can bite into that Total Float. So, to be specific, no matter which Activity is first to eat up one day of Total Float, the remaining Total Float now available to all downstream Activities is (by subtraction) four days.

The reason we are making such a big deal about this is because the overwhelming understanding in Dominant Project Management is that Total Float is an Activity-level value. In a few pages, we will examine popular definitions of the term Total Float, and you will see how many of them wrongly associate Total Float with an Activity, and how few with the Activity Path.

Informed by this discussion, you will likely share our puzzlement as to why it is standard Project Management practice to report Total Float alongside each Activity in a Standard Schedule Report. While the format and content of Schedule Reports is beyond the Scope of this book on CPM Mechanics, we can inform you that the traditional CPM tabular report format contains the following column headings -- and usually in this particular order.
Look at **Figure F0642** which is an image taken from a real Project Schedule report. Just to prove our point, we took a world-leading Scheduling Software program, \(^9\) and quickly created a five-Activity “Schedule” to correspond with the Logic in **Figure F0640**. It is clear from the Earliest Dates and Latest Dates that the five Activities are sequentially tied. You can see that **Activity E** is Date-Constrained, as indicated by an asterisk, indicating that this final Activity must be finished no later than the 30\(^{th}\) day. The Total Float is calculated (correctly) at five days.

<table>
<thead>
<tr>
<th>ACT ID</th>
<th>ACTIVITY DESCRIPTION</th>
<th>ORIG DUR</th>
<th>REM DUR</th>
<th>EARLIEST START</th>
<th>EARLIEST FINISH</th>
<th>LATEST START</th>
<th>LATEST FINISH</th>
<th>% COMP</th>
<th>TOTAL FLOAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010</td>
<td>Activity A</td>
<td>5</td>
<td>5</td>
<td>01JAN11</td>
<td>05JAN11</td>
<td>06JAN11</td>
<td>10JAN11</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0020</td>
<td>Activity B</td>
<td>5</td>
<td>5</td>
<td>06JAN11</td>
<td>10JAN11</td>
<td>11JAN11</td>
<td>15JAN11</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0030</td>
<td>Activity C</td>
<td>5</td>
<td>5</td>
<td>11JAN11</td>
<td>15JAN11</td>
<td>16JAN11</td>
<td>20JAN11</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0040</td>
<td>Activity D</td>
<td>5</td>
<td>5</td>
<td>16JAN11</td>
<td>20JAN11</td>
<td>21JAN11</td>
<td>25JAN11</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0050</td>
<td>Activity E</td>
<td>5</td>
<td>5</td>
<td>21JAN11</td>
<td>25JAN11</td>
<td>26JAN11</td>
<td>30JAN11*</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

**F0642: Total Float is Redundantly Stated in the Typical CPM Printout**

However, notice that the Total Float is repeatedly stated alongside each Activity! To us, this is a gross misrepresentation of Total Float. Imagine an estate attorney sending the following letter to one of the four children of a wealthy businessman who has just passed away.

Dear Janet,

While we share your sadness at the loss of your dear father, it may come as some small consolation that he left a rather sizable estate, roughly $1,000,000. Pursuant to the terms of his Will, his four children are to be the sole beneficiaries of that estate. Accordingly, you are entitled to $1,000,000.”

---

\(^9\) Primavera Project Planner, better known as P3, by Oracle-Primavera. (Printouts reproduced for clarity.)
We are sure you would agree that, at the very least, the above letter would be misleading. But it also could be the basis for fighting among the children if Janet was first to the bank and withdrew the entire $1,000,000, citing the letter as authorization for the withdrawal.

Now you may be thinking that anyone with half a brain would understand from the first sentence that the $1,000,000 is to be split among the four children. That’s because most people understand how estate distributions (and fairness) work. But far fewer understand how Total Float works. In fact, a common request we receive from Project Managers to whom we are consulting, is that we mask the Total Float column because the Subcontractor sees it as “his to use or not.”

Our message to you, then, is that you never forget that Total Float belongs to all of the Activities that (a) are related to one another by way of Immediately-Restricting Linkages, and (b) collectively determine the earliest possible completion of a single, common Deadline Milestone. Hence, the Path Total Float Law (see Figure F0614, page 189).

6D2: About Most Critical Total Float

Another common error among even the many seasoned Project Facilitators is to speak of Total Float as a single value and forget or overlook the existence of its two values, Start Total Float and Finish Total Float.

Look again at Figure F0642, which we were examining in the previous discussion. Do you notice that there is only one data field concerning Schedule Float, and it is called Total Float? Well, what if an Activity has different values as between Start Total Float and Finish Total Float? With only one column available to report Total Float, which one would be reported and which one would not be mentioned?

To appreciate this question, we need to digress for a few paragraphs and remind ourselves of how Total Float is actually used by Project Management. We begin by recalling that Total Float measures the difference between Earliest Dates and Latest Dates and provides a numeric measure of Time abundance (if the Total Float is a positive number) or deficiency (if the Total Float is a negative number).

As we have also noted, this Total Float measure suggests a degree of flexibility available to the entire Series of Activities that comprise the Activity Path and lead to a given Deadline Milestone. In practice, Project Management uses the Total Float value to prioritize work, deploy and allocate Resources, evaluate the impact of changes to Project Scope or performance conditions, and so forth.

To the point, Total Float is used by Project Management to make decisions. And in so doing, it trusts that the Total Float value will always have the same orientation (positive
or negative) to Time abundance or deficiency, respectively. That is:

- If Total Float is negative, then the associated Activities collectively must have insufficient Time to meet downstream Deadlines.
- If Total Float is positive, then the associated Activities collectively must have an excess of available Time.

Given this, would it ever make sense not to report the lesser of the two Total Float values, any time the Start Total Float and Finish Total Float values differ? Most Scheduling Software programs offer the choice of using Start Total Float, Finish Total Float or – our preference – Most Critical Total Float (which is determined as the lesser of the other two). But would you believe it if we told you that fewSchedulers are even aware that there are two different Total Float measures (STF and FTF) per Activity, or that these two Total Float measures can often reflect different numeric values?

Let’s take a look at how the three choices would report different information in the **TOTAL FLOAT** column of a standard Schedule Report. We went back to **Figure F0636**, which is the Logic Diagram for which we asked you to calculate Earliest Dates, Latest Dates, and Total Float.

You will recall that we had automated the correct answers, which we had shown you in **Figure F0638**, on page 203. You will want to pay particular attention to the three Activities where Start Total Float and Finish Total Float were different. For ease of comparison, **Figure F0644** lists those three sets of Total Float values.

What we are going to do now is generate the same report, but each time we will change the Software Setting that controls which Total Float value to report. First let’s look at a standard Schedule Report, where the Total Float is set as Most Critical Total Float. See **Figure F0646**.

Notice that Total Float is reported as follows:

<table>
<thead>
<tr>
<th>Activity</th>
<th>STF</th>
<th>FTF</th>
<th>MCTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity F</td>
<td>+5</td>
<td>+13</td>
<td>+5</td>
</tr>
<tr>
<td>Activity J</td>
<td>+17</td>
<td>+13</td>
<td>+13</td>
</tr>
<tr>
<td>Activity R</td>
<td>+5</td>
<td>+9</td>
<td>+5</td>
</tr>
</tbody>
</table>

**F0644: Comparing Total Float Options**

Notice that Total Float is reported as follows:

- **Activity F** = **TF +5**, **Activity J** = **TF +13**, and **Activity R** = **TF +5**

Now, in **Figure F0648**, we change the setting to Start Total Float, and we get these result:

- **Activity F** = **TF +5**, **Activity J** = **TF +17**, and **Activity R** = **TF +5**

Finally, in **Figure F0650**, we switch the setting to where Total Float equals Finish Total Float. The results are:

- **Activity F** = **TF +15**, **Activity J** = **TF +13**, and **Activity R** = **TF +9**
We hope you enjoyed this short discussion about the three different ways to report Total Float. But be sure to appreciate that this is not just an automation question as to which setting to choose when automating a Schedule.

Rather – and the point of including this discussion altogether in a book on the mechanics of the Critical Path Method of Project Modeling – you should always keep in mind that multiple Total Float values exist. One of your responsibilities as a Project Facilitator is to insure that the Total Float value being reported corresponds to the intended uses of the Scheduling information.

Earlier in this subsection, we asked, somewhat facetiously, “Would it ever make sense not to report the lesser of the two Total Float values, if ever the Start Total Float and Finish Total Float values differ?” The answer to this question depends on how the Scheduling information is to be used.

F0646: Most Critical Total Float

F0648: Start Total Float
For instance, perhaps over the first three weeks of a Project quite a number of Activities are scheduled to start all at once. The Project Manager might want to know which Activities are most critical (to downstream Deadline Milestones) and, thus, should get started right away, versus which Activities can wait a little longer. To answer this, you would want to generate a report where Total Float is set equal to Start Total Float.

Likewise, suppose that the Architect is besieged by submittals from the Contractor and wishes to know which submittals are the Most Critical (to downstream Deadline Milestones) for him to review and get returned to the Contractor. You would want to generate a report where Total Float is set equal to Finish Total Float.

As a default setting, we would strongly recommend using Most Critical Total Float because this speaks to the Criticality of every Activity without splitting hairs as to whether the Criticality is at the start, end, or across the entire Activity.

In closing, the underlying point of this subsection is that a surprisingly large number of Project Management professionals are completely unaware of the three different ways that Total Float can be reported. Making matters worse, few textbooks on CPM Scheduling even mention these three options. And so, the confusion lingers.

6D3: Total Float Quantities are Unreasonably Optimistic

This subtitle, intentionally provocative, embodies three separate assertions:

- Total Float is an inconsistent quantifier.
- Total Float reflects extreme optimism.
- Total Float is mostly unreliable as a measure of Schedule Flexibility.

If these three assertions are indeed true, then this book would have been remiss if it hadn’t brought them to your attention. Let’s consider each one separately.
**6D3a: Total Float is an Inconsistent Quantifier**

As is indisputable, Total Float measures the difference between Earliest Dates and Latest Dates. Anything that measures provides a quantity; thus, anything that provides a quantity thereby quantifies. But that’s not all that needs to be said about Total Float being a quantifier.
A quantity is a measurement, and each measurement is expressed in units of that measurement. So that begs the question: what is the unit of measure of Total Float? A correct (though general) answer would be – Time. Since Total Float measures a gap in Calculated Dates, it expresses that difference in Time units. But what are the Time units?

The more complete answer to the above question is that Total Float is measured in Time Units determined by either the Scheduling Software (if the computer is doing the calculating) or by the Project Facilitator (if the calculations are being performed manually). This topic will be discussed in greater detail in Chapter Nine when we get into the effects that Work Performance Calendars can have on various CPM Date Calculations.

Without stealing the thunder from our subsequent discussion about Work Performance Calendars, as a general rule or common practice the unit of measure of Total Float is based on the unit of measure of the Activity to which the Total Float is attached. For instance, consider Figure F0652, which shows a simple Activity Path with three Activities, whose cumulative Duration is 15 Workdays, comprised as follows:

- **Activity A**: 6 Workdays
- **Activity B**: 3 Workdays
- **Activity C**: 6 Workdays

This example uses the Point-of-Day Perspective (rather than Start-of-Day or End-of-Day), and Gregorian Dates rather than Ordinal Dates.

Suppose that the contract requires the Contractor to complete these three Activities no later than January 20th. In the Upper Panel, we see the three Activities scheduled to be performed using a five-day Workweek. Under this arrangement, the Work is expected to finish on January 19th. January 20th is a weekend, and so the Work must be finished on January 19th – which happens to be the Earliest Finish Date for the final Activity, Activity C. As a result, the Total Float for the string of Activities is $TF + 0$. There is no Time to spare if the Contractor wishes to meet the contractual Commitment.

Worrying that this is too tight for his comfort level, the Contractor decides to work six days per week. The Lower Panel shows that, using a six-day Workweek, the Earliest Finish of Activity C is now January 17th. Moreover, because only Sunday is considered a Non-Workday (according to the Schedule), the Latest Finish Date for Activity C can now be established as January 20th. As a result, this Schedule (using a six-day Workweek) reports that the Schedule has Total Float of $TF + 3$.

What we see from Figure F0652 is that the same Activities, assuming the same Durations per Activity – can have entirely different Total Float values depending on the underlying Workweek assumption. We could have left this point for the later discussion of Work
Performance Calendars, but we introduced it here to drive home the point that Total Float, as a quantifier is very subjective, and thus not necessarily consistent.

6D3b: Total Float Reflects Extreme Optimism

If you were a teacher with 30 students in your class, what probability would you assign to the prospect that all students will turn in their homework on time? 100%? How about on time and completely done? 100%? How about on time, completely done, and entirely correct? 100%?

Okay – too tough of a question? How about this one? Your best friend tells you that she has laid out a list of New Year's Resolutions that she plans to achieve over the course of the next twelve months. What probability would you place on her accomplishing all of them?

What is the probability that you will go through an entire lifetime of driving without being involved in even one fender-bender? What is the probability that any time you go to a grocery store (for the next 18 months), anything you may want to buy will be in stock?

Our point is that the only thing in Life that is 100% certain is that nothing is 100% certain! As the saying goes, stuff happens!

Let’s set that observation on a mental shelf for a few minutes and turn to Figure F0654. This illustration may take a little concentration to make sense, but hang in there with us. For starters, the illustration has three panels, all of which depict a simple string of Logic, from the perspective of Activity B (highlighted in blue).

☐ The Top Panel diagrams the Current Plan, which is for Activity A to take two days, Activity B to follow Activity A and consume one day, and Activity C to complete the Scope of Work of this Logic Diagram, while consuming four days. Combined, the Logic String should take seven days.

As viewed from its particular perspective, Activity B can expect to perform its one day of work on Day 3. This, of course, assumes that Activity A starts on Time (Day 1) and wraps up its work in two days, as planned. Likewise, Activity B trusts that Activity C will perform its work in four days, and will commence immediately after Activity B completes.

If everyone does everything that they have promised to do, then the entire set of Activities should finish by the end of Day 7. And since the work is not required, by Contract, until the end of Day 20, the Activity Path shares Total Float of TF +13. The Latest Dates for the three Activities are shown with green bars.
In the **Middle Panel** we consider what happens to the string’s Total Float if the Restricting Activity is reestimated to take longer to complete. We see that if the Restricting Activity takes longer, the Earliest Dates of **Activity B** and **Activity C** slip accordingly.

**F0654: How Float Works**
Notice that the Latest Dates for Activity B and Activity C do not shift, because their Activity Durations have not changed – and because the effect of changes in performance of Restricting Activities only affect the Latest Dates to the “left” (upstream) of the Subject Activity. We also note that the Total Float for the entire string has now shrunk to TF +9; the loss of Time due to a Restricting Activity that finished four days later than expected.

The Bottom Panel depicts the inverse situation. Here we see the results of news just received that a future Restricted Activity, Activity C, is now expected to take almost twice as long as previously thought. Instead of taking four days, it is now expected that it will require seven days. Notice that this new information does not have any effect on the Earliest Date estimates for our Subject Activity, Activity B. However, it does alter the Latest Dates for Activity B. And now, as a result of the Latest Dates “moving to the left” (as we say in the Scheduling world), the string of Logic now has a Total Float estimate of TF +10.

What these two alternative scenarios teach us is that Total Float reported against any Activity reflects the assumption that all will go precisely according to Plan. And if it does not (either before-the-fact, in Activities that precede the Subject Activity; or after-the-fact in Activities that follow the Subject Activity) the Activity Path’s Total Float (and hence the Total Float reported against each affected individual Activity) will change according.

What all of this means is that there is a direct correlation between the credibility of Total Float and the probability that everything will go precisely according to Plan. Now we can retrieve that observation we put on our mental shelf a few minutes ago, which was: the only thing in Life that is 100% certain is that nothing in Life is 100% certain!

Of course, looking at a Logic Diagram with only three Activities make for a gross understatement of the true potential for Total Float to unintentionally distort the reality that besieges all Projects. Imagine a Construction Schedule with several thousand Activities. Just how probable is it, in your opinion, that for any given Activity, all preceding Activities will perform their work no later than their Earliest Dates?

That would mean that all Activity Durations prove to be 100% dead on, that all Restrictions Delays are exactly right, and – most of all – that the Project itself experiences no surprises. Or, if surprises are encountered, that each one is somehow neutralized by the Project Team. How reasonable is that set of assumptions? And we’ve only been asking about the Activities that precede the Subject Activity in the Schedule!

How about all of the Activities that succeed the Subject Activity? What is the probability that all Eventually-Impacted Activities will be performed precisely according to Schedule? This means that, for the next eighteen months not one Activity will miss its dates? Yeah, right!
And to drive the final nail in the coffin, recall that all Calculated Dates reflect Logic and Durations, and that Durations are all “best estimates” that have no real hard science behind them. Further consider that most Estimated Durations tend to have had the “padding” squeezed out of them. That is, across the spectrum from Most Optimistic to Most Pessimistic, the Activity Duration that ends up in the Schedule is usually skewed toward the former. For all of these reasons we think it fair to say that Total Float reflects extreme optimism.

6D3c: **Total Float is Unreliable as a Measure of Schedule Flexibility**

Given all of the above, it seems to us that Total Float, as a measure of Schedule Flexibility, is fairly unreliable. Between its tendency to be inconsistently reported and its wishful premise that all will proceed precisely as Planned, we find it quite hard to draw much comfort from single-digit positive Total Float, or terribly worried about single-digit negative Total Float. Of course, when Total Float turns double-digit in either direction, it may have something to tell us about the relative importance of a Logic String as relates to a given Project Execution Commitment.

6D4: **Total Float Grossly Unstable**

This fourth observation may at first seem like a restatement of the immediately previous paragraph that Total Float is unreliable. But the point to be made here has to do with the stability of Total Float even if it were reliable.

What we are referring to is discussed in much more detail elsewhere in the ICS-Compendium, where we consider the Critical Path Method used as a Project Management Methodology. There we discover that Dominant Project Management teaches that the effective management of Time on Projects comes down to focusing attention and Resources on those Activities that are most Critical, with “Critical” being defined by Total Float.

As the Critical Path Method of Managing (Project Execution Strategy) is currently practiced, when Management concentrates resources on an Activity Path with low or even negative Total Float, something ‘miraculous’ tends to happen (and we’re being intentionally sarcastic): The Activity Path that was longer than all others (hence, the one with the Least Total Float) suddenly ceases being the longest. (Imagine that!) Now, some other Activity Path of Logic is the one behind the eight ball.

This back and forth approach to Project Management continues for the life of the Project. As a result, and as any construction Claims Consultant will confirm, Total Float rarely stays in the same place for more than one or two updates. For this reason, we contend
that Total Float is an unstable value, in addition to it also being an unreliable measure of Schedule Flexibility. The bottom line to this section is simple: while Total Float, as a concept, has something meaningful to contribute to the Time Management of a Project, it is not all that it has been chalked up to be.

6D5: Total Float Even Misunderstood by the “Experts”

And that leads us to our final observation in support of the conclusion that Total Float is broadly misunderstood: even the Scheduling experts seem unsure as to what Total Float is. You need only Google on the term Total Float to have your computer screen flooded with differing definitions and explanations of Total Float. If the definitions were mainly in agreement, that would be one thing. But when they differ from one another as much as they do, it leaves the novice wondering which one to believe?

But worse, the definitions themselves – even the ones that seem to mostly agree with one another – contain inherent flaws. Here is what we have learned so far in this book about Total Float:

- Total Float is not an Activity-centric attribute; Total Float is Path-centric.
- Total Float is a simple Arithmetic Calculation: the difference between Earliest Dates and Latest Dates, when calculated for any single Activity.
- There are in fact two calculable Total Float values per Activity, not one: Start Total Float and Finish Total Float.
- Start Total Float is the difference derived by subtracting Earliest Start from Latest Start, while Finish Total Float results from subtracting Earliest Finish from Latest Finish.
- To be most useful and meaningful, one should always consider the lesser of the two Total Float values per Activity – commonly called Most Critical Total Float.
- Total Float derives from a subtraction of Latest Dates, and Latest Dates derive from a Backward Pass, and a Backward Pass recognizes one or more Finish Date Constraints (discussed at length in Chapter Nine). Therefore, Total Float measures the abundance or shortage of Time to meet one or more Project Execution Commitments. Said differently, definitions that associate Total Float only with Project Completion are only partially correct statements.
- Total Float can be a negative value, since it is possible for Latest Dates to be earlier than Earliest Dates. Therefore, Total Float definitions that contemplate only positive or Zero Total Float are erroneous by omission.
We bring our discussion of Total Float to a close by citing a dozen or more definitions of the term that we found on the Internet. Rather than offer tedious (and repetitive) commentary about the limitations of each one, we decided to create a coded list of the most common mistakes that we find most prevalent among them. We have assigned a Letter Code to each, and then tagged the erring portions of the definitions accordingly. This should adequately validate the assertion of this subsection’s title: **TOTAL FLOAT EVEN MISUNDERSTOOD BY THE EXPERTS**.

If our critique of these definitions convinces you, then it is no wonder why there is so much confusion throughout the Project Management world concerning the Critical Path Method of Modeling.

6D5a: **List of Most Common Mistakes in Definitions of Total Float**

Following is a list of what we consider to be significant “mistakes” (M) in the definitions offered for Total Float. Beneath this list are the definitions themselves, which were found through a quick search of the Internet.

- **Too General** (M1): Uses the term Float in place of either Total Float or Free Float. Total Float and Free Float are sufficiently different and one definition simply cannot make sense for both.

- **Ambiguous** (M2): Uses ambiguous, undefined, or unfamiliar terms that are not common or immediately understood in the world of Project Time Management.

- **No Arithmetic Reference** (M3): Failing to provide the arithmetic derivative for Total Float.

- **Relates Only to Project End** (M4): Mistakenly assumes that the only Deadline Milestone to which the Total Float relates is **PROJECT COMPLETION**. Yet a Schedule can have any number of other Deadline Milestones, not just **PROJECT COMPLETION**. Each such Deadline constitutes another factor for the Backward Pass to consider. The compounding effect on Total Float of multiple Finish Date Constraints is discussed at length in Chapter Nine.

- **Ignores Negative Total Float** (M5): The expression “without delaying” precludes the condition where Total Float is negative. When Total Float is negative, a delaying condition clearly exists; thus, “without delaying” is a train that has already left the station. This would be similar to defining your Bank Balance as the amount of money you can spend without going broke. If your bank balance is already in the red, then the definition no longer makes any sense.

- **Associated with Activity, Not Path** (M6): Implicitly or explicitly associates Total Float exclusively with “an Activity” rather than with a Series of Activities. Gives
the impression that Total Float is an Activity-centric attribute. Fails to clarify that Total Float belongs to the Activity Path, and is only reported (redundantly) alongside multiple Activities that are residents of the same Activity Path.

- **Ignores Start/Finish Total Float (M7):** Does not acknowledge that there are two Total Float values: Start Total Float and Finish Total Float.

- **Understates the Effect (M8):** Fails to acknowledge either or both of the ways that Total Float can be affected: altered start of the Activity – or altered Activity Duration.

- **Overlooks Acceleration (M9):** Fails to acknowledge that Total Float can be altered in two directions: Acceleration as well as Delay – and not just Delay.

- **Ignores Finish Total Float (M10):** Provides only a partial explanation of the arithmetic basis for Total Float computation; states it as difference between Start Dates only, or Finish Dates only ... but not both.

- **Ignores Two Bases for Changes (M11):** Associates Total Float with changes to either Earliest Dates or Latest Dates – but not both.

Here are the definitions for the term, Total Float, that we found through a quick search of the Internet. We have tagged them with what we consider to be significant errors.

- **Float** in project management is the amount of time that a terminal element in a project network can be delayed by, without causing a delay to subsequent terminal elements (free float) or project completion date (total float). [Overall]

- Total float is the maximum number of work periods by which an activity can be delayed, without delaying project completion or violating a target finish date. [Overall]

- Total float is the amount that an activity can be lengthened, without delaying the project completion, assuming that all other activities are done in their normal time. [Overall]

- Total float is the amount of time (in work units) that an activity may be delayed from its early start, without delaying the project finish date. [Overall]

- Total float is the excess time available for an activity to be expanded or delayed without affecting the rest of the project, assuming it begins at its earliest time. [Overall]

- Total float is a measure of scheduling flexibility available on an activity path. [Overall]
Total float is the amount of time that an activity may be delayed beyond its early start/early finish dates without delaying the contract completion date. [Overall]

Total float is the number of days an activity can be delayed without delaying the project completion date. [Overall]

Total float is the total amount of time that a schedule activity may be delayed from its early start date without delaying the project finish date, or violating a schedule constraint. Calculated using the critical path method technique and determining the difference between the early finish dates and late finish dates. [Overall]
CHAPTER SEVEN

7A Understanding the Concept of a Critical Path
7B Critical Path: In a Nutshell
7C Two Schools of Thought: (Maybe Three!)
  7C1 A Cursory Introduction to Date Constraints
  7C2 Longest Path Definition
  7C3 Least Total Float Path Definition
  7C4 Driving Path Definition
7D Critical Path Misunderstood by the “Experts”
  7D1 What We Have Learned about Critical Path
  7D2 Common Mistakes in Definitions of Critical Path
7E Critical Path: Ideological Discussions
  7E1 ICS Would Prefer Paramount Path, But„„
  7E2 One Paramount (Critical) Path per Finish Milestone
  7E3 What is an Activity Path?
  7E3a Defining Elements of an Activity Path
  7E3b An Activity Path’s Inherent Components
  7E3c An Activity Path’s Points of Origin and Termination
  7E3d A Finish Milestone Has Two Different Conditional States
7F Tracing the Critical/Paramount Path
  7F1 Critical/Paramount Path Across Default Restrictions
  7F2 Tracing a Critical Path Across Overlapped Activities
7A: **Understanding the Concept of a Critical Path**

Well, here we are – at the very essence of the Critical Path Method: about to talk about the Critical Path itself. But we have a problem. We, the authors of this book, don’t know how to tell you what a Critical Path is because — we just don’t know ourselves.

“How is that possible?” you may wonder. Well, it should come as no surprise when we write that (a) there is little consistent understanding within Dominant Project Management as to what constitutes a Critical Path and (b) among the various more common definitions, we take exception to virtually every one of them. Yet **CHAPTER SEVEN** calls upon us to teach you all about the Critical Path. This is our dilemma.

Our solution is to do our best to educate you on the prevailing schools of thought and then, later, to share Cognitive Project Management's position on Critical Path. So without further ado we invite you to examine the current thinking as to what a Critical Path is, how to spot one, and what it means in the context of the Critical Path Method of Modeling a Project Execution Strategy.

7B: **Critical Path: In a Nutshell**

Without knowing it, you have actually been working all around the concept of a Critical Path. Back in **Figure F0636** when you were asked to trace the Activity Path with the Least Total Float you were essentially being asked to locate the Critical Path – at least, according to one of the two prevailing definitions of the term, Critical Path. Our response to this request appeared in **Figure F0638**, on page 203.

You may be thinking that if this is so, that you have already been working with Critical Path, then **CHAPTER SEVEN** should be fairly short, right? Well, if Dominant Project Management authorities were in agreement as to what constitutes a Critical Path, then it would be a short chapter indeed. But, like most every other key term in the Project Time Management lexicon, there is widespread lack of consensus and, in the case of Critical Path, even passionate disagreement.
7C: Two Schools of Thought (Maybe Three!)

As just stated, there are two prevailing schools of thought regarding how to identify a Critical Path. One school insists that the Critical Path is the *Longest Path* through the Schedule. The other school of thought argues that it is the Activity Path bearing the *Least Total Float*. A third opinion, meant to serve as a hybrid compromise of sorts, is that the Critical Path is the *Driving Path*! Let’s take a closer look at each of these three popular interpretations.

7C1: A Cursory Introduction to Date Constraints

To understand why there are two different schools of thought about Critical Path, the reader needs to know a little more about the Critical Path Method of Modeling than has been covered thus far. In particular, one needs to know about Date Constraints. From the ICS-Dictionary:

**Date Constraints:** Date Constraints are computational criteria for Calculated Dates that are superimposed on a given Activity, such that any Activity Path Segments (to which the Activity belongs) may be artificially accelerated, delayed, or fixed in time.
By now you know that the Backward Pass Process is meant to determine the latest that Activities may start or finish without threatening the Timely achievement of downstream Project Execution Commitments. That said, every Finish Date Constraint has the potential to influence the Latest Start and Latest Finish dates of Restricting Activities feeding into it.

Take a look at Figure F0702. In the Upper Panel are three Activities arranged in a Logic String similar to what you have been seeing for quite some time now. According to the Upper Panel, Activity C must finish no later than Day 140, as evidenced by the Finish-No-Later-Than (FNLT) Date Constraint. We see from the Forward Pass that the entire string of three Activities is estimated to take 25 days, and will not begin before Day 100, the postponement due to a Start-No-Earlier-Than (SNET) Date Constraint. Thus, the string of Activities shares Total Float of 15 days.

Now look at the Lower Panel. Here, we have imposed a second Deadline Milestone, on the completion of Activity B, which reflects the Project Owner telling us that she wants Activity B finished no later than Day 120 but still wants the entire Project finished by Day 140. When we do our Backward Pass, we find ourselves with our pencil hovering over the bottom-right corner of Activity B, where its Latest Finish needs to be written.

According the Latest Start of Activity C, the latest that Activity B can finish is Day 133. But if we take into account the new Deadline Milestone, then the latest that Activity B can finish is the Owner's mandatory date, Day 120.

Proceeding on with our Backward Pass, if the latest that Activity B can finish is Day 120 and it has a 13-day Activity Duration, then the Latest Start for Activity B must be Day 107. Finally, Activity A has Latest Start and Latest Finish Dates of Day 102 and Day 107, respectively.

Our last step is to calculate the Start Total Float and Finish Total Float for all three Activities. What we discover is that while Activity C still carries the same Total Float value (TF +15) as it had in the Upper Panel, the other two Activities now have a different Total Float value, TF +2.

From this simple example we see that a Finish Date Constraint has the potential to influence the Latest Start and Latest Finish Dates of Restricting Activities that feed into it. You may be wondering why we stopped at the word “potential,” rather than stating that a Finish Date Constraint will influence upstream Latest Dates. The answer has to do with whether the Finish Date Constraint is earlier than what the Latest Finish would have been otherwise.

Consider Figure F0704, in which we have changed the Finish Date Constraint imposed on Activity B to Day 135. If we ignore this Date Constraint and determine what the Latest Finish for Activity B otherwise would be, we find that the controlling influence on it
comes from the Latest Start for Activity C, which is Day 133. Since Day 133 is earlier than the Finish Date Constraint of Day 135, the Finish Date Constraint remains inconsequential, having no overriding effect on the Backward Pass Date Calculations.

But don’t be fooled; it still has the potential to affect Latest Dates for Restricting Activities. For example, suppose the Owner later decides that it does not need this particular string of Logic completed by Day 140, but instead moves the Deadline for Activity C back ten days to Day 150. Let’s see what happens when we change Activity C’s Finish Date Constraint.

In Figure F0706 we keep the same Finish Date Constraint for Activity B but change Activity C’s Finish Date Constraint to Day 150. Not surprisingly, Activity C’s Total Float changes to TF +25. But notice that the two preceding Activities now show Total Float of TF +17, where in Figure F0704 they had shared Total Float of TF +15.

Why do they experience a change in Total Float, even though there were no changes in Logic, Durations, or Finish Date Constraints for these two Activities? The answer is that Activity C’s Latest Start was no longer the controlling influence on the Latest
F0708: Finding the Longest Path through the Subnet
Finish of Activity B; Activity B’s Finish Date Constraint was. And so we see that a Finish Date Constraint introduces the potential for changes to Latest Dates of a Subject Activity and its preceding Activities and, as a consequence, the potential for changes to corresponding Total Float values.

7C2: Longest Path Definition

Now we are able have a meaningful discussion about whether a Critical Path is the Longest Path through a Schedule as some contend, or is the Activity Path with the Least Total Float as others insist. We’ll begin with the Longest Path argument.

We will be working with iterations of a small Subnetwork of Logic, as initially shown in Figure F0708. In the Subnet we see 26 Activities, each bearing a Duration of 10 days. We quickly perform a Forward Pass and find that, if all goes according to Plan, the Project should take 110 days. We calculate Total Float for the Subnet and find that the longest Activity Path has Total Float of TF +0. We also notice the imposition of a Finish Date Constraint of Day 110 onto the end of the Subnet at Activity Z.

By the general meaning of the word “longest,” we observe that the Longest Path of Activities (shown with the red dashed line) winds through Activity A ► Activity J ► Activity K ► Activity N ► Activity O, Activity P ► Activity V ► Activity W ► Activity X ► Activity Y ► and Activity Z.

To the naked eye it would appear that the Longest Path is also the Least Total Float Path through the Schedule since all of the Activities on this Longest Path also have the lowest Total Float values that we can find anywhere in the Subnet, TF +0. Since this Activity Path, whether regarded as the Longest Path or the Least Total Float Path, is also the Activity Path that determines the earliest that the Subnet’s only downstream Deadline Milestone can be met, we conclude that surely we must have identified the Critical Path for the Subnet of Logic, right? Yes, you are right, but ...

But as we shall now see, any time that a Schedule has more than one Deadline Milestone or more than one Start Milestone, the Longest Path may not always be the one that determines the earliest that a Project can finish. To see what happens when a Schedule has more than one pair of Release/Deadline Milestones, let’s add two additional Finish Date Constraints, and one additional Start Date Constraint, such that the Schedule now has two Release Milestones and four Deadline Milestones, as follows. See Figure F0710:

- Activity A Release Milestone: Start No Earlier Than Day 0
- Activity N Release Milestone: Start No Earlier Than Day 34
- Activity M Deadline Milestone: Finish No Later Than Day 52
- Activity U Deadline Milestone: Finish No Later Than Day 73
- Activity X Deadline Milestone: Finish No Later Than Day 86
- Activity Z Deadline Milestone: Finish No Later Than Day 124
F0710: Paramount (Critical) Path Leading to Finish Milestone Z
Take a good long look at Figure F0710 in which we have saved you the trouble of performing Forward Pass and Backward Pass Date Calculations. The first thing we notice is that the Longest Path through the Network, the one that we traced in Figure F0708, now has different Total Float values at different points along its length (highlighted with different color ovals).

This of course means that there is no longer a direct correlation between the Longest Path and the Least Total Float Path. And that means that we now have a new question on the table: Which definition is right? They cannot both be right!

**7C3: Least Total Float Path Definition**

With respect to the Least Total Float Path, we make another startling observation: it doesn’t actually connect to the end of the Project; it stops at **Activity X** (see the blue oval in Figure F0710). There are six Activities that bear the Subnet's Least Total Float, of **TF -8**. These six Activities are among the eleven Activities that comprise the Subnet's Longest Path. Another leg of the Longest Path, containing two Activities that bear Total Float of **TF +10**, terminate at the Subnet's final Deadline Milestone. At the front end of the Longest Path are three Activities bearing Total Float of **TF -4**.

What we learn from all of this is that when a Schedule has more than one Finish Date Constraint, it is possible for Activities to be more critical to interim Deadline Milestones than to the final Deadline Milestone. And so, any definition of Critical Path should be careful not to limit the Critical Path only to the completion of the Project. [1]

There is one other observation that we can make, but we will reserve full commentary about it until the next subsection where we engage in a short ideological discussion of Critical Path. That observation is that we have been operating without a clear understanding as to what constitutes an Activity Path in the first place. Specifically, *Where does an Activity Path begin and where does it end? Are the two ends of an Activity Path determined by Date Constraints?*

- If so, then the Longest Path may not actually span the entire length of the Schedule?
- If not – that is, if we insist that an Activity Path is a continuous string of connected Activities that extends until the Activity Path runs out of Activities – then (a) the Longest Path may have multiple Total Float values and (b) it may lead to misleading reports that the Project is ahead of Schedule, when in fact numerous internal Deadline Milestones are actually seriously behind Schedule!

---

1 Yet, overwhelmingly, published definitions of the term, Critical Path, invariably link the Path to some variant of **Project Completion**. See our examination of popular Critical Path definitions starting on the next page.
7C4: Driving Path Definition

Because of the problems created by Date Constraints, Scheduling experts in recent years have proposed a “solution” of sorts. They call it the Driving Path, the idea being to trace the Schedule Logic that (according to Earliest Dates at least) seems to constitute a continuous, albeit interrupted, string of Activities that transcends any Date Constraints and spans from the beginning of the Project on through to its end.

In Figure F0710, this would correspond to the Activity Path we have traced in red –which, you will recall, is the same as the originally identified Longest Path before we added those pesky Date Constraints. You may be curious to know that when Scheduling Software performs Driving Path calculations, it essentially ignores the Date Constraints. [2]

The elephant in the living room is, of course, the fact that the Driving Path does not give us a single, consistent value that we can reliably use to either measure Schedule Criticality or guide effective Project Time Management. The fact that the Driving Path discounts the effects of Date Constraints blatantly and simultaneously ignores the very strategic or pragmatic realities that those Date Constraints were meant to represent and model with their inclusion. In the end, the Driving Path contains an assortment of Activities with a number of different Total Float values that transcend and ignore important mid-Project Deadline Milestones.

7D: Critical Path Misunderstood by the “Experts”

As the following ICS-Dictionary definition explains, there is no universal consensus on what the term, Critical Path, actually means:

- Critical Path: In principle, as Dominant Project Management explains it, the Critical Path is that string of logically-tied Activities that has the greatest potential or actual influence on the timely completion of a downstream Finish Milestone. In practice, however, definitions among the most respected Project Time Management authorities differ widely on what constitutes a Critical Path.

  First, they disagree on how to define a Critical Path (Longest Path or Least Total Float Path). Second, they disagree on whether a Project Schedule can have more than one Critical Path. Third, they seem unable to define the underlying term, Path, itself. Cognitive Project Management rejects the very notion that a Project must have only one single Critical Path.

  Instead, it recommends (a) replacement of the word, Critical, with an alternative

---

2 This is, of course, tantamount to us reverting back to Figure F0708 conditions, as a way to deal with the problems posed by Figure F0710’s Date Constraints.
set of non-comparative terms, (b) definition of the term Path, (c) application of a Monitored Path Ranking System, and (d) ranking of ALL Activity Paths leading to EACH Finish Milestone in the Project Schedule. Paramount Path is the term that Cognitive Project Management uses to represent the Activity Path that comes closest to resembling the same Least Total Float Path which Dominant Project Management calls a Critical Path.

Still worse, though, we will soon see that almost every definition is either incomplete or incorrect, as determined by the observations we have made on the previous pages. This may sound like a “He Said-She Said” situation, where respected authorities insist that their views are correct while Cognitive Project Management disagrees. We will leave it for you to decide where you weigh in on this topic. For our part, we are comfortable in our criticism of the following definitions, and for the reasons presented.

7D1: What We Have Learned about Critical Path

As we did earlier with Total Float definitions, we begin our critique of Critical Path definitions by stating what we have learned about the Critical Path.

If we accept the intent behind the designation of a Critical Path, per se — that it might somehow indicate to Project Executors which Logic String has the potential to most significantly influence the timely completion of one or more downstream Finish Milestones — then the Critical Path must be that string of Logic (Activities, Performance Restrictions, and Date Constraints) that determines the earliest that a given Deadline Milestone can be met. This leads us to a few conclusions:

- There can be, and most likely will be, a different Critical Path for each separate Deadline Milestone in the Schedule.
- The Longest Path through the Schedule may or may not terminate at the last Deadline Milestone (e.g., PROJECT COMPLETE). If there is more than one Deadline Milestone (as annotated by Finish Date Constraints) in the Schedule, then there cannot be just one Critical Path as determined by the Longest Path.
- The Activity Path with the Least Total Float may or may not have any bearing on the Earliest Date by which the Project may finish, overall. Moreover, if there is more than one Finish Date Constraint in the Schedule, then there cannot be only one Critical Path, as determined by Least Total Float.
- Because Total Float has two components (Start Total Float and Finish Total Float), and because Critical Path determination relies on Total Float determination (even using the Longest Path approach), a Critical Path can, and often does, pass partially (as opposed to fully) through Activities.
7D2: Common Mistakes in Definitions of Critical Path

As we did earlier with Total Float definitions, we have developed a list of what we consider to be significant “mistakes” (M) in the definitions offered for Critical Path. Beneath this list are the definitions themselves, which were found through a quick search of the Internet.

- **More than Activities Alone (M1):** A Critical Path is comprised of more than just Activities; it is also defined by its Performance Restrictions and Date Constraints.
- **Path Not Defined (M2):** A Critical Path references the term, “Path,” without defining what a Path is.
- **Ignores Positive Total Float (M3):** A Critical Path can have positive Total Float.
- **Assumes Only One Deadline Milestone (M4):** A Critical Path need not (and may not) terminate at a Project's end.
- **Ignores Longest Path (M5):** A Critical Path need not be the Path with the Least Total Float.
- **Ignores Least Total Float Path (M6):** A Critical Path need not be the Longest Path through the Schedule.

Here are the definitions for Critical Path that we found through a quick search of the Internet. We have tagged them with what we consider to be significant errors.

- The series of tasks $M_1$ that must finish on time for the entire project to finish on time.
- In a network diagram, the longest path $M_2$ from start to finish or the path without any slack $M_3$, and thus the path $M_2$ corresponding to the shortest time in which the project $M_4$ can be completed.
- In a network diagram, the path $M_2$ with the longest duration $M_6$.
- The line of project activities $M_1$ having the least float $M_5$, especially when float is close to, or below, zero. $M_3$
- The route $M_2$ through the network that has only critical Activities. $^{[3]}$
- The series of interdependent activities $M_1$ of a project connected end-to-end, which determines the shortest total length of the project. $M_4$ Note: This definition overlooks the three other dependency types beyond finish-to-start (which it refers to as “end-to-end”).

---

$^{[3]}$ This definition employs circular reasoning. Since the overwhelming understanding in the Project Management world is that an Activity is Critical if it lies on a Critical Path, it is completely circular to define the Critical Path as the Path containing Critical Activities.
The path \( M_2 \) (sequence) of activities \( M_1 \) which represent the longest total time \( M_6 \) required to complete the project \( M_4 \).

- The longest connected route \( M_2, M_6 \) through \( M_4 \) the CPM network.

- The sequence of activities \( M_1 \) through a project network from start to finish, the sum of whose durations \( M_6 \) determines the overall project duration \( M_4 \). There may be more than one critical path depending on workflow logic. A delay to progress of any activity on the critical path will, without acceleration or resequencing, cause the overall project duration \( M_4 \) to be extended, and is therefore referred to as a critical delay.

- The series of interrelated tasks \( M_1 \) that comprises the longest duration \( M_6 \) to complete a project \( M_4 \), where a delay in any task will extend the overall schedule \( M_4 \).

- An analysis technique used to predict project duration \( M_4 \). Analysis of activities, sequencing, and paths \( M_2 \) and determining which path has the least amount of schedule flexibility (float) \( M_5 \).

- Any delay of an activity on the critical path directly impacts the Planned project completion date \( M_4 \) (that is, there is no float on the critical path). A project can have several parallel critical paths.

7E: Critical Path: Ideological Discussions

Later in this chapter, we will experiment with finding a Critical Path. That is a fairly important skill for you to acquire since, after all, you are learning about the Critical Path Method. But before we can show you how to spot a Critical Path in a Logic Diagram, we first have to describe what a Critical Path actually is. And as we have so far established, there is not broad consensus within Dominant Project Management as to what constitutes a Critical Path. That, therefore, leaves Cognitive Project Management no choice but to establish its own definition of the term, Critical Path.

7E1: Cognitive Recommends the Paramount Path, But…

In a separate ICS-White Paper, [4] Cognitive Project Management argues that the very word, critical, is a poor choice as an adjective to describe an Activity Path. It insists that the term, path, has never been formally defined by Dominant Project Management. So if one does not know what an Activity Path is, and if one does not know when something becomes Critical, then how can one consistently or meaningfully use the compound term, Critical Path, to any real advantage?

---

4 See ICS-White Paper WPA-KK-14, "When is the Critical Path NOT the Most Critical Path?"
We think it would be extremely helpful for you to once again set down this book, and take a few minutes to read the referenced ICS-White Paper. Here, however, we wish to shed light on an important point from that ICS-White Paper, one that is fundamental to our establishing a litmus test for detecting a Critical Path within a Schedule.

The fundamental point to make is that the word, Critical, as currently used by Dominant Project Management, is a comparative one. That is, one Activity Path is deemed to be Critical simply because other Activity Paths are deemed to be less Critical, based on some subjective measure. Conversely, an Activity Path may not be deemed Critical, if one or more other Activity Paths appear to be more Critical.

As we noted previously, there are two popular measures of Activity Path Criticality: Total Float or Path Length. Under the Least Total Float doctrine, the Activities sharing the Least Total Float are viewed as constituting a Critical Path of Activities. Under the Longest Path doctrine the string of continuous Activities whose individual Activity Durations and intervening Performance Restrictions combine to create the Longest Path through the Schedule are deemed to constitute a Critical Path.

Under either doctrine however the determination is a comparative one. The Critical Path is not unlike a championship title, such as World Heavyweight Champion which a fighter holds until someone else defeats him. Or, like the world’s tallest building, which is a record held until some other structure is erected even taller. And so, the Longest Path in one Schedule Edition may become the second or third Longest Path in a subsequent Schedule Edition — but then might just as easily return to being the Longest Path in a fourth Schedule Edition. Likewise, the Path with the Least Total Float may be displaced by another Activity Path with even lower Total Float in a subsequent Schedule Edition.

The comparative nature of the word, Critical, reduces the real-world value of knowing which Activities reside on the Critical Path. This is because the Criticality of an Activity is linked to the Criticality of the Activity Path on which it resides. As the Activity Path moves in and out of Critical status, so does the Criticality rating of Activities resident on that Activity Path. If Project Management is using Activity Criticality (as they do when they report Total Float against individual Activities) to prioritize the Work, then the Project is likely to waste energy moving Resources to and fro in response to an unstable, constantly flickering Critical Path.

An obvious fix to this problem is to redefine the word, Critical, to where it is no longer comparative. In the field of medicine, a range of well-defined terms have been established to represent different degrees of well-being in a patient, that include Good, Fair, Serious, Critical, Terminal and so forth. Notice that these health condition rankings are not issued contingent on the ranking of other patients, but instead are reflective of conditions inherent
completely within the subject patient. The rankings are assigned after comparison of patient key health statistics to a fixed set of parameters.

In the referenced ICS-White Paper, Cognitive Project Management explains the reasoning behind its adoption of a Scale of Path Criticality which establishes degrees of Criticality for any Activity Path based on objective variables inherent in the Activity Path itself. Under the Monitored Path Ranking System, every Activity Path feeding into a given Deadline Milestone is evaluated and ranked against a fixed Scale of Path Criticality. Moreover, every Deadline Milestone is examined, not just the last Deadline Milestone in the Schedule. As a result, every Activity Path in the Schedule is being monitored.

Of course, for each Deadline Milestone there is value in knowing which Activity Path has the greatest influence on the Deadline Milestone's timely achievement. We call this the Paramount Path.\(^5\) We chose the word Paramount because, of all of the Activity Paths leading to a Deadline Milestone, this Activity Path is the one that is of paramount importance and has the greatest influence in determining the Earliest Date by which a Deadline Milestone will likely be accomplished.

The reason that we are injecting mention of the Monitored Path Ranking System and the Paramount Path into this chapter is in order to set these concepts aside for the balance of this book. In other words, the official position of Cognitive Project Management is that the adjective “critical” is a poor choice as long as it is comparatively determined.

That said, we also recognize that Dominant Project Management fully embraces the term Critical Path and until it is replaced it will remain the de facto standard for determining Activity Criticality. Since one of the two main goals of this book is to prepare the reader for immediate, productive employment in the field of Construction Project Time Management, we are left with no choice but to teach Critical Path theory as it is currently practiced – notwithstanding whether the current practice actually makes a whole lot of sense or even works all that well.

The point of this subsection is to say that in this book we will, from this point forward, accept and embrace the Critical Path concept, and do our best to teach it to you as it is being practiced today

**7E2: One Paramount (Critical) Path per Finish Milestone**

The previous paragraph notwithstanding, we will not shy away from pointing out

---

5 The Paramount Path is the closest counterpart to Dominant Project Management's Critical Path. Of course, the biggest difference between the two concepts is that there is a separate Paramount Path for each Deadline Milestone, whereas according to Dominant Project Management there is only one Critical Path for the entire Schedule, no matter how many Deadline Milestones.
errors in Dominant Project Management’s understanding of Critical Path theory, where such misunderstandings lead to avoidable Project chaos and inefficiency. One such misunderstanding is that a Schedule has only one Critical Path. We invite you to peruse readily-available literature on the Critical Path Method and observe just how frequently the words “the Critical Path” are used.

Does a Schedule have just one Critical Path? The correct answer is not simply “Yes” or “No,” but instead, “It depends.” Could you take out a city map and highlight the “Best Route” without having answers to some basic questions, such as:

- Where are we starting from?
- Where are we heading to?
- Are we in the middle of rush hour?
- What shall be the criteria for determining the word “best?” (Most scenic? Shortest? Fastest?)

A Project Schedule is not unlike a city map with countless possible routes that connect the start of the journey to its end. Is there really such a thing as one Best Route? Is there ever just the Critical Path?

That same ICS-White Paper mentioned above goes on to suggest that there is one particular Activity Path with supreme importance to each separate Deadline Milestone in the Schedule. Whether we call such a Path a Paramount Path as Cognitive Project Management does, or a Critical Path as Dominant Project Management does, there is one such Activity Path that is the most temporally influential for each Deadline Milestone!

To answer the earlier question, there may be only one Critical Path if there is only one Deadline Milestone in the Schedule. But that is not often the case, at least on most Construction Projects. In the vast majority of cases there are at least two Deadline Milestones, one called SUBSTANTIAL COMPLETION and the other called FINAL COMPLETION.

These two Deadline Milestones have contractual significance and are discussed at length in another volume of the ICS-Compendium.

In addition to these common Deadline Milestones, Construction Projects are often further constricted by other Owner-imposed Project Execution Commitments, such as BUILDING DRY-IN by a certain date, or PERMANENT POWER by a certain date. Similarly, the Owner might perhaps wish that Project Execution be sequenced to support tenant move-in priorities and so the Project is divided into Project Phases, and each Project Phase may have a specific Deadline Milestone.

Cognitive Project Management holds the view that each Deadline Milestone can have as many different Activity Paths leading to it as there are Start Milestones sharing an Eventually-Impacting Progressive Relationship with it. Adjusting for terminological
differences, this book concedes that the achievement date for each Deadline Milestone is ultimately determined by the most influential of these Activity Paths, no matter what it is called: Critical Path or Paramount Path.

The main point of this subsection is that for every Deadline Milestone there is at least one distinct Critical Path (or possibly more than one) that determine the Deadline Milestone’s earliest possible achievement, and not necessarily just one such Project-length Critical Path, as Dominant Project Management literature would have us believe.

For this reason, we strongly suggest that you adopt a practice of attaching a qualifier to any use of the term Critical Path. For instance, say DRY-IN CRITICAL PATH, or SUBSTANTIAL COMPLETION CRITICAL PATH, or PHASE II CRITICAL PATH. The point of the qualifier is to associate the Critical Path with the downstream Deadline Milestone to which it is linked.

7E3: What is an Activity Path?

Before we can propose a reliable approach to Critical Path identification that will lead to consistent results regardless of the Schedule and regardless of the criteria for criticality, we must come to terms with the yet undefined term, Activity Path.

7E3a: Defining Elements of an Activity Path

In the same ICS-WHITE PAPER, WPA-KK-14, WHEN IS THE CRITICAL PATH NOT THE MOST CRITICAL PATH?” we answer this question thoroughly. Here, let us simply provide the ICS-White Paper's final conclusion. To more fully understand the rationale behind our decision, we again encourage you to read the entire ICS-White Paper.

7E3b: An Activity Path's Inherent Components

Cognitive Project Management defines an Activity Path in terms of (a) its inherent components (b) and its Points of Origin and Termination. Let's reprint the ICS-Dictionary definition for Activity Path. Also note the definition for the term, Series of Activities:

- **Activity Path:** In a CPM Network Diagram, an Activity Path is a unique Series of Activities that spans from Path Start to Path Finish.

- **Series of Activities:** A Series of Activities is a unique arrangement of Activities in temporal progression, such that each Activity has one Restricting Activity before it and one Restricted Activity after it.

These two definitions collectively tell us that the components of an Activity Path include both the Activities themselves, as well as the Performance Restrictions that link those Activities.
7E3c: An Activity Path's Points of Origin and Termination

The Activity Path definition also makes mention of two Path Ends: Path Start and Path Finish. ICS-Dictionary defines these terms this way:

- **Path Ends:** The term, Path Ends, refers to the Point of Origin and Point of Termination of an Activity Path. The two Path Ends are Path Start and Path Finish.

- **Path Start:** In a Progressively-Related Network Diagram the term, Path Start, refers to an Activity that marks the beginning of an Activity Path. To qualify as a Path Start, an Activity must either be bound by a Start Date Constraint or suffer from an Open End Condition known as Unrestricted Start. In an Unrestricted Start Condition, the start of a Subject Activity is not restricted by any upstream Activities.

- **Path Finish:** In a Progressively-Related Network Diagram the term, Path Finish, refers to an Activity that marks the end of an Activity Path. To qualify as a Path Finish, an Activity must either be bound by a Finish Date Constraint or suffer an Open End Condition known as Unrestricting Finish. In an Unrestricting Finish Condition, the finish of a Subject Activity does not itself restrict any downstream Activities.

These definitions make mention of a condition widely referred to by Schedulers as Open-Ended Activities (also called Dangling Activities). Our examination of the few definitions we could find for these expressions left us wanting more precision. So, here is the ICS-Dictionary definition of an Open End Condition:

- **Open End Condition:** In a Progressively-related Network Diagram, an Open End Condition refers to a break in sequential Logic whereby a Subject Activity is either not restricted on its start by any upstream Activities or, conversely, its own finish is not restricting any downstream Activities.

It should be noted that, just as there is a correlation between the terms Action and Activity (the former being on the Project and the latter being in the Schedule), there is a similar correlation between the terms Milestone and Date Constraint. Out in the real world of the Project there are Start Milestones and Finish Milestones. Corresponding to these within the Schedule are Start Date Constraints and Finish Date Constraints, as follows:

- **Start Date Constraint:** A Start Date Constraint establishes the earliest that an Activity can start. The Start Date Constraint corresponds to a Start-No-Earlier-Than (SNET) Software Setting. A Start Date Constraint constitutes a Path Start and, as
a result, its effect is that if the Start Date Constraint delays the Path Start Activity, it delays the entire Activity Path, correspondingly. In the inverse, if the Start Date Constraint allows for an advanced start to the Path Start Activity, the probability of timely completion of the downstream Activity Path is correspondingly improved.

Finish Date Constraint: A Finish Date Constraint establishes the latest that an Activity can finish. The Finish Date Constraint corresponds to a Finish-No-Later-Than (FNLT) Software Setting. A Finish Date Constraint constitutes a Path Finish and, as a result, its effect is that if the Finish Date Constraint advances the Path Finish Activity (Deadline Milestone), it necessarily advances the entire Activity Path that it terminates. In the inverse, if the Finish Date Constraint recedes and allows the Path Finish Activity to finish later, the performance urgency for the entire upstream Activity Path is correspondingly eased.

7E3d: A Finish Milestone Has Two Different Conditional States

We are beginning to form a better understanding of what an Activity Path is all about. As to its inherent components, an Activity Path is a Series of Activities, which of course includes the Performance Restrictions that link those Activities. As to its Path Ends, an Activity Path spans from a Path Start to a Path Finish. A Path Finish terminates at a Deadline Milestone. [6]

Your understanding of Deadline Milestones may benefit from further explanation. Let us be clear about this: a Deadline Milestone is a Conditional State; as such, it represents a Moment in Time. Actually, since we are equipped with two CPM arithmetic processes, the Forward Pass and the Backward Pass, a Deadline Milestone actually has two Conditional States.

☐ It has an Earliest Date that it can conceivably take place.
☐ And, it has a Latest Date by which it must take place.

Since we know that Total Float is the numerical difference between Earliest Dates and Latest Dates, it follows that a Deadline Milestone has its own Total Float values. One could then say that the Total Float values that we so commonly attribute to Activities along an Activity Path are, more correctly stated, reflective of the Total Float of the Activity Path itself.

To complete the thought process we are now making, if each Activity Path (assuming more than one) leading to a Deadline Milestone has its own Total Float value, then the Activity Path with the Least Total Float determines the Total Float of the Deadline Milestone.

6 Unless it terminates into an Open End Condition.
Having just acknowledged that a Deadline Milestone may have more than one Activity Path feeding into it, we are now able to craft a definition for the all-important term, Critical Path. Since Cognitive Project Management uses the term Paramount Path to mean the same thing as what Dominant Project Management calls a Critical Path, we will present the ICS-Dictionary definition for a Paramount Path, which is:

**Paramount Path:** Within the Monitored Path Ranking System, the one Activity Path that most influences the timely achievement of a downstream corresponding Deadline Milestone is known as the Paramount Path. A Paramount Path can belong to any of the three Path Ranking Classes: Critical, Watch, or Free. Of all of the Activity Paths feeding into a given Deadline Milestone, the Paramount Path is the Activity Path bearing the Least Total Float. Since each Deadline Milestone in a Schedule has at least one Paramount Path, a Project Schedule will minimally have as many different Paramount Paths as there are Deadline Milestones. Paramount Path is the term that Cognitive Project Management uses to represent the same Least Total Float Path that Dominant Project Management calls a Critical Path. The biggest difference, of course, is that Dominant Project Management insists that a Schedule can have only one Critical Path, even if it contains multiple Deadline Milestones.

### 7F: Tracing the Critical/Paramount Path

We are now ready to apply what we have learned and can successfully trace a Critical/Paramount Path. From the above discussion, we know that Critical/Paramount Path detection involves these four simple steps:

- **Step 1:** Identify all of the Deadline Milestones that exist in the Schedule.
- **Step 2:** Identify all Activity Paths feeding into each separate Deadline Milestone.
- **Step 3:** Select the Activity Path with the Least Total Float; this qualifies as the Critical/Paramount Path for that Deadline Milestone.
- **Step 4:** Trace the Critical/Paramount Path back to its Path Start, and identify its Start Date Constraint or confirm that it has an Open Start Condition.

### 7F1: Critical/Paramount Path Across Default Restrictions

Let’s apply these steps to a simple Logic Diagram that only contains Default Restrictions.

---

With respect to a single Deadline Milestone, that is.
F0712: Paramount (Critical) Path Leading to Finish Milestone M from Start Milestone A
Using the same Subnet that we worked with earlier (Figure F0710), answer these questions:

☐ How many Finish Milestones are there?
☐ How are they labeled?
☐ What are their respective Total Float values?

Here is our answer. There are four Finish Milestones, as follows.

☐ Deadline M (FNLT 52): Total Float is TF +2
☐ Deadline U (FNLT 73): Total Float is TF +13
☐ Deadline X (FNLT 86): Total Float is TF -8
☐ Deadline Z (FNLT 124): Total Float is TF +10

Applying the above steps for tracing a Critical/Paramount Path, the first thing we determine is that there are four separate Finish Date Constraints and, per our definition, each must have its own Critical/Paramount Path.

☐ Critical/Paramount Path to Deadline M: Look at Figure F0712, in which we have traced the Critical/Paramount Path for Finish Milestone M, which we will now call Deadline M Critical Path (remember that we said to prefix every Critical/Paramount Path with an adjective describing the Deadline Milestone to which it is tied).

We see that Deadline M has Total Float of TF +2. We will trace the Logic backwards until we encounter whatever Start Date Constraint explains the succession of Earliest Dates that leads to the Earliest Finish for the last Activity in the string of Logic. Since Activity M is the last Activity in the string of Logic leading to Deadline M, and since Activity M’s Earliest Finish is Day 50, we trace backwards from Activity M following the Earliest Dates until we find the Path Start for the particular Activity Path that ultimately determines the Earliest Date that Deadline M can occur. We discover that Start A (SNET 0) is the Path Start. We have highlighted the Activity Path in Figure F0712.

Now you may be confused by the fact that the Activities within this Activity Path do not all share the same Total Float, as our definition for a Critical/Paramount Path clearly requires. You are likely noticing that Activity A, Activity J, and Activity K report Total Float of TF -4, whereas the remaining two Activities of this Critical Path, Activity L and Activity M, show Total Float of TF +2.

Our answer is an extremely important point for you to understand: that an Activity can simultaneously have different Total Float values, each with respect to a different downstream Deadline Milestone to which it is logically tied. Flip back
F0714: Paramount (Critical) Path Leading to Finish Milestone U from Start Milestone A
All About Critical Path

to Figure F0522 on page 150, which we last saw when we were discussing how to perform a Forward Pass through complex Logic where a Restricted Activity has multiple Restricting Activities. Didn’t we learn that one must consider every different Logic Tie leading into a Restricted Activity before deciding which one is ultimately determining the Earliest Start?

As we saw in Figure F0522, the Earliest Start for Activity J would be Day 128 if the only Restricting Activity was Activity D. Yet if the only Restricting Activity was Activity H, then the Earliest Start for Activity J would be Day 132. But, in fact, there are three Restricting Activities, and the one that ultimately determines the Earliest Start of Activity J is Activity F, and it establishes that the Earliest Start for Activity J is Day 133.

Just as we must consider all possibilities when performing a Forward Pass (and, of course, a Backward Pass as well), we must also consider all possibilities with respect to Activity Paths, too. Remember that an Activity Path's Total Float values can change at the drop of a hat, even while the Logic and Durations remain steadfast.

Returning to Figure F0712, the length of the Activity Path stretching from Activity A to Deadline M is 50 days long. That is a fact. The Commitment Date for Deadline M is given as Day 52. Therefore, the Activity Path that originates at Activity A and terminates at Deadline M has a Total Float value of TF +2. The fact that three of its Activities report lower Total Float (TF -4) only means that those Activities must also reside on another Activity Path, one whose Total Float is TF -4.

- Critical/Paramount Path to Deadline U: Look at Figure F0714. Here we have traced the Critical/Paramount Path for Deadline U, which carries Total Float of TF +13. By now you should be able to quickly trace the Logic backward to Activity A … and not be bothered by Activity A reporting Total Float of TF -4 (which it gets from being a resident of another, more critical, Activity Path).

- Critical/Paramount Path to Deadline X: Figure F0716 traces the Critical/Paramount Path for Deadline X which claims Total Float of TF -8. You may recall this Activity Path as being the one we had identified back at Figure F0710 as the Path in the Schedule with the Lowest Total Float. What is the Path Start for this Activity Path? If you answered, Start A, you would be wrong. The correct answer is Start N (SNET 34).

And why is this so? Notice the Earliest Finish of Activity K, which is Day 30. Now pay attention to the SNET Start Date Constraint imposed
F0716: Paramount (Critical) Path Leading to Finish Milestone X from Start Milestone N
on Activity N. Which of these two values is determining the Earliest Start of Activity N? It is the SNET Start Date Constraint, and therefore Activity K is not part of this Activity Path. The Activity Path commences with Activity N, and its Start Date Constraint.

- Critical/Paramount Path to Deadline Z: Figure F0718 presents our final Critical Path analysis for Deadline Z, which reports Total Float of TF +10. When we follow the Earliest Dates backwards from Deadline Z, we find that its Path Start is also Activity N. So even though Activity N commences an Activity Path that is eight days too long (Deadline X Critical Path), it also heads another Activity Path that has ten days of extra Time, Deadline Z Critical Path.

7F2: Tracing a Critical Path across Overlapped Activities

While the above examples are all through Default Restrictions only, you now know everything you need to in order to trace Critical/Paramount Paths through Logic that includes Start Restrictions and Finish Restrictions as well. Just remember to pay close attention to differing Start Total Float and Finish Total Float values for a single Activity. The Critical/Paramount Path can easily pass through a portion, rather than all, of an Activity.
F0718: Paramount (Critical) Path Leading to Finish Milestone Z from Start Milestone N
CHAPTER EIGHT

8A Understanding Free Float
8A1 Free Float Calculations
8A1a Calculating Free Float Across Default Restrictions
8A1b Calculating Free Float Across Overlapped Activities
8A1b-i Start Free Float
8A1b-ii Finish Free Float
8A1b-iii Most Critical Total Float
8A2 Free Float Ideological Discussion
8A2a Free Float is a Pistol with a Two-Foot Range
8A2b Free Float Belongs to Path, Not Activities
8A2c Free Float is Unstable
8A2d Free Float and Total Float Speak Two Different Languages

8B More about Symbiotic and Progressive Relationships
8B1 More about Symbiotic Relationships
8B2 More about Progressive Relationships
8B3 Phenomenon of Multi-Path Residency
Free Float and Symbiotic Networks

8A: Understanding Free Float

We have purposely left for last our discussion of the concept of Free Float.

- First, while by their similar-sounding names they may seem like allied concepts, Total Float and Free Float are really quite different.

- Second, we thought it was far more important to immediately follow our coverage of Total Float with the full discussion of Critical Path, which depends directly on Total Float.

- Third, in practice Free Float is not used hardly as much as Total Float.

What exactly is Free Float? The simplest description is that it is a numeric value that measures extra Time that an Activity Path Segment has available to perform its work. By “extra Time,” we mean that the Path Segment can delay its completion by as much as the Free Float value, without affecting any other Path Segment in the Schedule.

For those unfamiliar with Free Float, the above description may seem a little hard to grasp. To those who think that they do understand Free Float, their first reaction to our explanation may be to mumble, “Nonsense! And besides ... what the heck is an Activity Path Segment?” To our Scheduling colleagues we respectfully respond to your “Nonsense” with this retort: “The books have it wrong. Free Float does not belong to an Activity; just like Total Float, it belongs to an Activity Path.”[1]

And we can prove it!

8A1: Free Float Calculations

Before we begin to spar with our more seasoned Scheduling buddies, why don’t we bring those who are new to the Critical Path Method of Modeling up to speed? Check virtually any definition of the term and it will tell you that Free Float is the amount of Time that an Activity can delay without delaying any other Activity in the Schedule.[2] And while, later in this subsection, we will examine the most popular definitions of Free

1 More precisely, it belongs to the Activity Path Segment.
2 Samples of popular definitions begin on page 262.
Float (just as we did with Total Float and Critical Path), in order to have something to work with right now, here is a very typical definition of Free Float:

- The amount of time an activity can be delayed beyond its early start/finish dates without delaying the early start or early finish of any activity.

8A1a: Calculating Free Float across Default Restrictions

Let’s use some CPM Logic to understand what the above definition means, at least according to conventional wisdom. The next three figures are actually extracted from Figure F0522 used earlier.

In Figure F0802, we see two Restricting Activities feeding into a common Restricted Activity, Activity D, which has an Earliest Start of Day 123. Which of the Restricting Activities is the more restrictive to Activity D? Well, Activity B has an Earliest Finish of Day 123 while Activity C has an Earliest Finish of Day 119.

Obviously, Activity B is the more restrictive to Activity D. What we can easily understand is that Activity C could stall by four days and not affect “without delaying the early start or early finish of any activity.” We would say, then, that Activity C has Free Float of FF +4, whereas Activity B has Free Float of FF +0. As for Activity D, we cannot say anything about its Free Float, since its Restricted Activities are not shown.

Let’s try another example. Consider Figure F0804 where we see three Restricting Activities feeding into a common Restricted Activity, Activity J. What is the Free Float for each Restricting Activity? Answer:

- Activity D Free Float = FF +5
- Activity F Free Float = FF +0
- Activity H Free Float = FF +1
Now, one final example and we will assume that you clearly understand how Free Float is determined.

Look at Figure F0806. Activity K has Free Float of FF +0 while Activity L has Free Float of FF +3. Take another look at that definition of Free Float on the previous page. As we saw in all three examples, Free Float did measure the “amount of Time” that each Activity could be delayed without affecting immediately following Activities.

8A1b: Calculating Free Float across Overlapped Activities

Let’s see how this understanding works when the Logic involves Overlapped Activities. In Figure F0808 (along with the next two diagrams), Activity A and Activity B are overlapped, sharing both a Start Restriction and a Finish Restriction. What is the Free Float for Activity A, the Restricting Activity?

8A1b-i: Start Free Float

Hopefully you realize that there are two possible answers, depending on whether you consider the Start Restriction or the Finish Restriction. You would also be right if you assumed that we would call these two values Start Free Float and Finish Free Float, respectively.

What is Activity A’s Start Free Float value? To answer this, we need to determine whether the start of Activity A could be delayed by even one day without affecting any other Activity? The answer to this question is “No,” because Activity B’s Earliest Start, Day 102, is based entirely on the prior start of Activity A two days earlier (per the SS:2 Logic Tie). Hence the Start Free Float for Activity A is SFF +0.
What is Activity A’s Finish Free Float value? Can the finish of Activity A be delayed without affecting Activity B in any way? We see that Activity B’s Earliest Finish is Day 111, and it has a Finish Restriction tied back to Activity A with an FF:2 Logic Tie. So the latest that Activity A could finish without affecting the completion of Activity B would be Day 109. The Earliest Finish for Activity A is shown as Day 107. Therefore Activity A has a Finish Free Float value $\text{FFF +2}$. 

Before we leave Finish Free Float we should look at a situation where the Restricting Activity flows into more than one Restricted Activity. Consider the Logic in Figure F0810. Here we see Activity C tied into both Activity F and Activity Q, with Finish Restrictions of FF:5 and FF:3, respectively. Can you figure out what Activity C’s Finish Free Float value is? We will take each Logic Tie separately.

Into Activity F, Activity C flows with a FF:5 Logic Tie. Activity C has an Earliest Finish Date of Day 113. Add to this the five days required by the FF:5, and then compare this to the Earliest Finish of Activity F, which is Day 124. Applied, we get $(\text{Day 124} - (\text{Day 113} + 5))$ and the result is Finish Free Float of $\text{FFF +6}$. 

Now let’s look at the other Finish Restriction coming out of Activity C, this time linked to Activity Q. Using the same approach, we derive this result: $(\text{Day 116} - (\text{Day 113} + 3))$ which tells us that the Finish Free Float is $\text{FFF +0}$. 

So what we have figured out is that Activity C’s finish can slip six days before delaying the finish of Activity F, but cannot slip at all without delaying the finish of Activity Q. Taking the lesser of the two values, we conclude that the Finish Free Float of Activity C is $\text{FFF +0}$. 

One last question about Figure F0810: Does Activity C have a Start Free Float? You might think that it does not, since there are no Start Restrictions coming out of Activity C. But you would be wrong. Every Activity has both Start Free Float and Finish Free Float. These values answer the simply-worded question: “Can this Activity's start or finish be delayed without delaying any other Activity?”
Let’s consider what would happen if we were to delay the start of Activity C by even one day. Wouldn’t Activity C’s Earliest Start slide back one day? Wouldn’t this also slip the Earliest Finish by one day? And since we have already determined that the Finish Free Float for Activity C is FFF +0, wouldn’t a slip in the start of Activity C result in an impact to the finish of Activity Q? And so, wouldn’t we have to conclude that Activity C has Start Free Float of SFF +0?

8A1b-iii: Most Critical Free Float

You will recall that, in the case of Total Float, there are two separate values, Start Total Float and Finish Total Float. You will also recall that we decided that it would be of little benefit to Project Time Management goals to use only one of these two values in our management of a Project. Rather, we concluded, the most meaningful use of these two values is always to compare them and report the lower of the two as the composite Total Float value for the Activity.

The same holds true for Free Float. If the intent behind knowing what a Free Float value happens to be is to know how much leeway an Activity may have for delaying itself, then wouldn’t we want to report the lower/lesser of the two Free Float values?

Take a look at Figure F0812 and determine Activity G’s Most Critical Free Float value. Spend a few minutes and work through the two Free Float values. Here are the calculated values:

- **Activity G**, Start Free Float SFF +0
- **Activity G**, Finish Free Float FFF +9
- **Activity G**, Most Critical Free Float MCFF +0

8A2: Free Float: Ideological Discussions

As we said earlier, Free Float is not widely used. Still, we would be remiss if we did not make you aware of its existence. To that end, we want to explore some of the nuances of Free Float and in so doing perhaps we may discover why it enjoys only limited popularity and use.
8A2a: **Free Float is a Pistol with a Two-Foot Range**

We realize that this is a strange subtitle, but it hopefully conveys our opinion that Free Float just doesn’t protect very much. In this respect it is quite unlike Total Float which, you will recall, associates an Activity Path to a downstream Project Execution Commitment, which is represented in the Schedule by a Deadline Milestone. In a word, Total Float helps Project Management protect important Project Execution Commitments.

But just what does Free Float protect? Popular Answer: the “Activity immediately following” (the Activity Path Segment in which the Free Float is found)!

Is there anything special about this “immediately following Activity” other than the fact that it is “following?” Does it have any special significance? As we have seen, Free Float simply lets us know whether a string of Activities can, as a collective whole, delay their Work without affecting any other downstream Activities. How terribly useful is that information? Compare that to Total Float which relates the timing of an Activity Path to a Deadline Milestone!

8A2b: **Free Float Belongs to Path, Not Activities**

You may have picked up on how we have repeatedly associated Free Float with a String of Activities or Path of Activities or Activity Path Segment — but not with a single...
Activity, per se. As you shall soon discover when we examine current definitions of Free Float, there is virtually universal agreement among Scheduling experts that Free Float belongs to a single Activity, and not to a Path Segment as we contend. Through the course of the next five figures, we will completely disprove their postulate.

Let us start by considering the Logic Diagram in Figure F0814. We see eight Activities arranged such that two parallel Activity Path Segments have the same Restricting Activity, Activity A, and the same Restricted Activity, Activity Z.

From the Earliest Dates we notice that the Upper Path Segment (Activity F ► Activity G ► Activity H) has a span of 28 days, whereas the Lower Path Segment (Activity R ► Activity S ► Activity T) spans 14 days. In other words, the Upper Path Segment is expected to take twice as long as the Lower Path Segment. If you and I had never read about Free Float, we would simply be saying that the Lower Path Segment has 14 extra days to perform its collective work without affecting any other Activities in the Schedule.

This is precisely the position that Cognitive Project Management takes with respect to Free Float; it belongs to the entire Path Segment, not just to a single Activity in the Path Segment! But according to Dominant Project Management thinking, only Activity T would be credited with Free Float of FF +14.
To prove our point, look at Figure F0816, in which we record a delay to the start of the first Activity in the Lower Path Segment, Activity R. Notice that the Activity Durations of the Lower Path Segment Activities remain the same as before; all we have done is delay the start of Activity R. Now how much Free Float does Activity T have? According to conventional thinking, Activity T now has Free Float of FF +10.

Our point is that only the legal Owner of a bank account has the right to spend the money that is in it. And there is nothing in any Project Management journal or treatise that prevents any Activity along a given Path Segment from consuming the Free Float of that Path Segment. Hence, the Free Float must belong to the entire Path Segment, and not to whichever Activity happens to be at the end of that Path Segment!

Let’s pound the nail further into the wood. Consider Figure F0818. In this situation, we learn that Activity R will take three times longer than originally estimated. Instead of taking four days, it will take 12 days. Recalculating the Earliest Dates we find that the Path Segment’s Free Float reduces by eight days, and drops to FF +6 from FF +14.

And just to show that any Activity on a Path Segment can eat up Free Float, in Figure F0820 we return Activity R to its originally-planned Activity Duration, but this time it is Activity S’s Duration that we considerably extend. Where the Activity Duration for Activity S had been three days, we now pretend that it will take 13 days. Recalculating,
we see that Free Float at Activity T has dropped by 10 days, from FF +14 to FF +4.

We have one more illustration to share before we move on to other observations about Free Float. Take a close look at Figure F0822 and compare this to the original scenario shown in Figure F0814. The only difference between the two is the timing of Activity T. In this scenario, the Project Manager has just learned that the equipment to be installed via Activity T will not be arriving until the end of Day 27. In response to this, the Project Facilitator slides Activity T to the right, indicating when it is expected to now take place.

What has happened to the Free Float of FF +14 days? Did it go away, or has it merely slid to the left by one Activity? Our point is that the prevailing definitions only got it half right: the last Activity in a Path Segment may be the one to show the Free Float, but the Free Float belongs to the entire Activity Path Segment, not just that last Activity.

So, back to our question: Who cares whether a given Activity will impact the “immediately following Activity?” We think this information is of limited value, especially when one considers how unstable and fleeting Free Float turns out to be. Perhaps if more Project Time Management practitioners understood that Free Float reflects the temporal flexibility of an Activity Path Segment, such knowledge might be of greater value in the ongoing Planning and execution of the work in the short term.
8A2c: **Free Float is Unstable**

For all of the same reasons that we gave for why Total Float is an unstable numeric value, Free Float is no more stable than the combined Activity Durations and Performance Restrictions that together yield the Free Float values. When one reflects on the imprecise nature of Construction Schedules in general, it is easy to conclude that putting so much importance on Total Float or Free Float is probably not such a smart idea. Activity Durations are, at best, just estimates of how long future Activities will take. Performance Restrictions are notational indicators of intended future interactions among Project Participants. Date Constraints reflect best intentions before the Project begins. How stable are any of the Calculated Dates in the Schedule? In turn, how stable are any values derived by subtracting those Calculated Dates from one another?

8A2d: **Free Float and Total Float Speak Two Different Languages**

Finally, we want to point out something that ought to be obvious but is, sadly, a common misunderstanding even among seasoned Project Facilitators. Total Float and Free Float are entirely different animals. They have nothing to do with one another.
For one thing, have you noticed that Free Float does not involve Latest Dates at all? It is strictly based on a comparison of Earliest Dates. Thus, the most that Free Float can opine on is the forward flow of the work, as planned. That makes it fairly limited.

By comparison, Total Float measures the difference between the earliest that Activities are likely to occur, and the latest that those same Activities can occur without impacting downstream Project Execution Commitments. Free Float gives us no similar insights.

Notwithstanding the previous point, many Scheduling experts insist that there is a relationship between Free Float and Total Float, as if the former is somehow a subset of the latter. They have even invented new terms: Independent Float, to represent the portion of Free Float that does not impact Total Float, and Interfering Float to represent the portion that does impact Total Float.

While we will not get into a detailed rebuttal of their postulates, we will go on record to say that their association of Free Float with Total Float falls apart under certain circumstances (for instance, when Total Float is a negative value).

At the very least, such attempts to correlate Free Float with Total Float serve only to confuse the layperson. And as noted earlier, given that Free Float has limited value to Project Management altogether, such elaborate portrayals of Free Float as a subset of Total Float merely serves to give Free Float more attention and credence that we believe it rightly deserves.

**8A3: Free Float Defined**

It is now time to review and critique some of the more popular definitions of the term, Free Float.

**8A3a: What We Have Learned about Free Float**

As we did earlier with Total Float and Critical Path definitions, we begin our critique of Free Float definitions by stating what we have learned about Free Float.

- Free Float measures gaps between Earliest Dates, and has nothing to do with either Latest Dates or Deadline Milestones.
- Free Float has nothing to do with Total Float; they are entirely different animals.
- Free Float belongs to an Activity Path Segment that collectively leads to one or more other Path Segments.
- Free Float can never be a negative value.
- Free Float, just like Total Float, has two values: Start Free Float and Finish Free Float.
8A3b: List of Most Common Mistakes in Definitions of Free Float

And just as we did earlier with both Total Float and Critical Path definitions, we have developed a list of what we consider to be significant “mistakes” (M) in the definitions offered for Free Float. Beneath this list are the definitions themselves, which were found through a quick search of the Internet.

- **Associates with Activity (M1):** Free Float is portrayed as an Activity attribute rather than a Activity Path attribute.
- **Ignores 2nd Free Float Value (M2):** Given that Free Float has two parts (Start Free Float and Finish Free Float), it is a definitional error to associate Free Float only with either the start or the finish of the immediately following Activity. Likewise, the definition limits impact to the Restricted start (ignoring possible impact to its end, such as through a Finish-to-Finish Tie).
- **Ignores SFF and FFF (M3):** The Free Float definition does not mention Start Free Float and Finish Free Float.
- **Associates with Total Float (M4):** Free Float has nothing to do with Total Float, or with Critical Path.
- **Incomplete Definition (M5):** Incomplete definition; leaves out pertinent factors.
- **Factually Incorrect (M6):** Factually incorrect definition.

Here are the definitions for Free Float that we found through a quick search of the Internet. We have tagged them with what we consider to be significant errors.

- The amount of time an activity can be delayed beyond its early start/finish dates without delaying the early start or early finish of any activity. [Entire]
- The total float of a path is combined “free float” values of all activities in a path. [Entire]
- Free float is used to describe amount of time that spans from the completion of one previously scheduled activity and extends to the point at which the next scheduled activity is set to begin. Free float can be calculated by determining the amount of the time between the earliest start date of the initial activity and the earliest start date of the succeeding activity, and then subtracting from that total the amount of time that it is expected the first activity will take to complete. [Entire]
- Free float can only occur when two or more activities share a common successor. [Entire]

3 Ridiculous! Total Float is the difference between Earliest Dates and Latest Dates of one Activity, whereas Free Float is the difference between Earliest Dates only, and between two Activities.
Free float is the number of days an activity $M_1$ can be delayed without taking float away from the next activity. Another way of saying the same thing is that free float is the number of days an activity $M_1$ can be delayed without delaying the early start date $M_2$ of the next activity. [Entire $M_3$]

The length of time, expressed as work units, that a specific activity $M_1$ may be delayed without delaying the start of another activity scheduled to follow immediately after. [Entire $M_3$]

8B: More about a Symbiotic and Progressive Relationships

Leaving behind our discussion of Total Float and Free Float, we would like to close this chapter with a brief return to what Cognitive Project Management contends are the three Relationship Categories that can be found in virtually every Critical Path Network Schedule. We want to talk more about two of them.

Let's start by reprinting their definitions:

- **Symbiotic Relationship:** A Symbiotic Relationship between two Activities is one in which the performance of each Activity may (or may not) have an impact/effect on the conditions under which the other Activity must perform. The potential to impact or affect the other Activity's operational conditions exists in both directions – but need not necessarily ever occur. A Symbiotic Relationship is one of three Relationship Categories that may be found in a CPM Schedule.

- **Progressive Relationship:** A Progressive Relationship exists between two Activities that are linked together by way of Performance Restrictions (and possibly additional intervening Activities). In a Progressive Relationship, the timing of a downstream Activity is or will be affected by the timely performance of one or more upstream Activities.

It is only now, after having received an in-depth understanding of how Total Float and Free Float actually work, that you can better appreciate these two Relationship Categories. We’ll begin with Symbiotic Relationships.

8B1: More about Symbiotic Relationships

The examination of Free Float that we just completed in the last section provides an excellent example of what we mean by a Symbiotic Relationship. But in order for you to get the full gist of what we are trying to say you need to remind yourself that the Project Schedule is an actual tool that is being used by the Project Executors. The Project Schedule and the Project Executors interact with one another.
One could even say that the relationship between the Project Schedule and the Project Executors is a symbiotic one. We say this because the Project Executors draw information from the Schedule in order to direct the Actions on the Project, and then those Actions, once taken, become new information fed back into the Project Schedule. Each entity depends upon and informs the other.

Let’s take a look at how this might play out on a real Project. Imagine that Figure F0824 represents a small Subnetwork of Activities on a larger Project.

- This Subnetwork is comprised of two parallel Activity Path Segments, both feeding into a common downstream Path Segment (commencing with Activity Z).
  - **Upper Path Segment**: Includes Activity F, Activity G, and Activity H.
  - **Lower Path Segment**: Includes Activity R, Activity S, and Activity T.
  - **Downstream Path Segment**: Only 1st Activity (Activity Z) in Path Segment shown.
Figure F0824 is divided into four horizontal panels:

- **Schedule Edition 1**: Data Date is Day 0
- **Schedule Edition 2**: Data Date is Day 20
- **Schedule Edition 3**: Data Date is Day 40
- **Schedule Edition 4**: Data Date is Day 60

**Schedule Edition 1**: In **Schedule Edition 1**, the seven Activities have not yet been started. A quick analysis concludes that the **Upper Path Segment** dictates the Earliest Start and Earliest Finish of **Activity Z**. We can surmise that the **Lower Path Segment** enjoys a certain amount of Free Float, although without Activity Durations we cannot quantify the Free Float.

Now, let us imagine that the Project Executors also recognize the *comparative more important* influence of the **Upper Path Segment** on the Timely performance of **Activity Z**, and so they rightly decide to give the **Upper Path Segment** preferential attention during the first month of the work.

**Schedule Edition 2**: One month later **Schedule Edition 2** is produced. We see the results of that special focus. **Activity F** was completed in less time than originally Planned. But such increased performance did not come without a price. **Activity R**, which was Scheduled to have been finished, never finished at all. Not surprisingly, the **Lower Path Segment** is *now* the longer Path Segment of the two, and the **Upper Path Segment** now enjoys the Free Float; the **Lower Path Segment** has Zero Free Float.

**Schedule Edition 3**: By the end of the second month, the pendulum has swung back in the other direction. Management had put preferential attention on the **Lower Path Segment** after they read the **Scheduling Report** from **Schedule Edition 2**. And what happened as a result of the special attention?

As you would have predicted, **Activity S** was completed in less time than originally expected but, out of comparative neglect, **Activity G** dragged out. Now the Free Float is back on the **Lower Path Segment**, and the **Upper Path Segment** is one again “behind the eight ball.”

**Schedule Edition 4**: And in the bottom panel we see the situation at the end of three months.

The message to take away from this example is that the two Path Segments (and their resident Activities) are symbiotically related to one another. Re-read the ICS-Dictionary definition of the Symbiotic Relationship Category. “The performance of each Activity may (or may not) have an impact/effect on the conditions under which the other Activity must perform.”
Doesn’t this pretty much describe what we just saw? The performance of one Activity affects the conditions of the other. The *conditions* we are referring to, in blunt terms, is being “behind the eight ball.” In Scheduling jargon, the *condition* is not having any Free Float. An Activity with no Free Float is an Activity that has the “potential to impact or affect” other Activities. For sure, the Path Segment with no Free Float has the immediate likelihood of “affecting” downstream Path Segments. This would be the effect of a Progressive Relationship.

But it *also* has an effect on any other concurrent Path Segments (such as we saw between the **Upper Path Segment** and the **Lower Path Segment**). And this demonstrates the presence of a Symbiotic Relationship at play, between Activities that are *not* progressively linked!

### 8B2: More about Progressive Relationships

The essential element in a Progressive Relationship is that the “Restricting Activity will necessarily affect the performance of the Restricted Activity, under certain circumstances.” The best way to understand this requirement is to think of whether there is a continuous, unbroken, *forward-flowing* line of ink from one Activity to another, regardless of whether the dates are continuous and without Time gaps.

Take a look at **Figure F0826**. Can you trace ink from **Activity P** through to **Activity Z**? Yes, you can, and so there is a Progressive Relationship between **Activity P** and **Activity Z**. The fact that **Activity P** is on a Path Segment that enjoys Total Float of **TF +10** doesn’t matter. Under the right circumstances (e.g., a **15 day** delay in the start of **Activity P**), **Activity Z** would be delayed, right?

Now we can clarify the types of Progressive Relationships that exist, per Cognitive Project Management. There are two: Immediately-Restricting and Eventually-Impacting.
8 Free Float and Symbiotic Networks

- **Immediately-Restricting Progressive Relationships:** This Relationship Subcategory describes two Activities that are connected to one another with no other Activities between them. Hence, Activity P and Activity R share an Immediately-Restricting Progressive Relationship. Activity R and Activity T share an Immediately-Restricting Progressive Relationship. But Activity P and Activity T do not share an Immediately-Restricting Progressive Relationship.

- **Eventually-Impacting Progressive Relationship:** Any Progressive Relationship that is not Immediately-Restricting, by process of elimination, is Eventually-Impacting. Hence, Activity P and Activity T share an Eventually-Impacting Progressive Relationship.

We call this a Progressive Relationship because the Actions (on the Project) build upon one another in a progressive way. Correspondingly, each Activity in the Schedule builds upon the work of the Activities immediately before it, in such a way that the overall Project work (slowly but surely) progresses forward.

Back to Figure F0826. Do Activity G and Activity T share a Progressive Relationship, either Eventually-Impacting or Immediately-Restricting? They do not, because no amount of delay of one will change the Calculated Dates of the other. But they do share a Symbiotic Relationship, in that the delay of one may well cause reactive changes in pursuit of the other (as we observed in Figure F0824). You bet they do!

We hope this short discussion of Symbiotic and Progressive Relationships helps you understand them better. Elsewhere in the ICS-Compendium we learn how to use our understanding of Symbiotic and Progressive Relationships to effectively manage the use of Time on Construction Projects. [4]

8B3: Phenomenon of Multi-Path Residency

Before closing, we want to draw your attention to a phenomenon that Cognitive Project Management calls Multi-Path Residency. It stems from an observation that was made earlier, that an Activity can reside on more than one Activity Path at the same time.

For most Project Facilitators, this statement is either obvious, or debatable. We want to drive the point home that the vast majority of Activities in a Schedule (upwards of 90% or higher) coexist on multiple Activity Paths.

Why do we find this phenomenon so important to highlight? Understanding Multi-Path Residency goes hand-in-hand with understanding Progressive Relationships. It also goes a long way toward better understanding Symbiotic Relationships.

[4] As mentioned several times before, Dominant Project Management appears entirely oblivious to the existence and influence of Symbiotic Relationships.
Look again at Figure F0826. We see two Activity Path Segments, not unlike similar arrangements we have been staring at throughout this chapter. Each Path Segment is comprised of three Activities, and both Path Segments have the same common Impacting Activity, ACTIVITY A, and the same common Impacted Activity, ACTIVITY Z.

- **Upper Path Segment:** The **Upper Path Segment** members are Activity F, Activity G, and Activity H.

- **Lower Path Segment:** The **Lower Path Segment** is made up of Activity P, Activity R, and Activity T.

You will notice that we have been careful not to call either of the three-Activity strings an Activity Path, per se. This is because the three-Activity Path Segments do not commence, or terminate, with qualifying Path Ends. Go back and re-read the definitions of Path Start and Path Finish, on page 239. Based on these definitions, we recognize two Activity Paths.

- **The Upper Activity Path** proceeds through Activity A ► Activity F ► Activity G ► Activity H ► Activity Z. It commences with a Start Date Constraint and terminates with a Finish Date Constraint.

- **The Lower Activity Path** traces through Activity A ► Activity P ► Activity R ► Activity T ► Activity Z. It commences with a Start Date Constraint and terminates with a Finish Date Constraint.

It would seem, then, that both Activity A and Activity Z belong to two Activity Paths, whereas each of the other six Activities belong to only one Activity Path.

Notice that the two Activity Paths claim the same Start Date Constraint and Finish Date Constraint pairs! Yet, we consider them to be two distinctly different Activity Paths. Now you understand why the word “unique” appears in the definition for an Activity Path.

We want to take a brief moment to brag. Nowhere in all of Dominant Project Management literature will you find any discussion of Path Segments (no matter what term is it called by). And you will be hard-pressed to even find a definition of an Activity Path ... even though all of these sources amply discuss the concept of a Critical Path. We think that Cognitive Project Management's steadfast efforts to craft a coordinated set of definitions for Activity, Activity Path, Activity Path Segment, Path Ends, Path Start, Path Finish, Start Date Constraint, and Finish Date Constraint have significantly advanced the overall field of Project Time Management. We hope you agree.
Back to our discussion of Free Float.

So far, there shouldn’t be much objection to our observations. But now is when things get a little dicey. Take a look at the Total Float for the Lower Path Segment (Activity P, Activity R, and Activity T). Notice that this Path Segment has a Total Float value of TF +10. Its counterpart, the Upper Path Segment, bears Total Float of TF +3.

**Question:** Is there any correlation between the Lower Path Segment's Total Float of TF +10 and Upper Path Segment's Total Float of TF +3? After all, as we have just noted, these two Path Segments have nothing to do with one another, right? And yet, we will submit for your consideration that the Total Float of TF +10 (in the Lower Path Segment) is actually comprised of two portions; a TF + 3 portion and a TF + 7 portion.

![Figure F0828: In Overlapping Activity Paths, Total Float is a Cumulative Value](image)

Look at Figure F0828. Here we have drawn the two Activity Paths, using red and green pens. We immediately see that Activity A and Activity Z reside on two Activity Paths – but this was something we could visualize earlier, and to which you took no exception.

Now look at that oval on the far left, the one that suggests that the Start Total Float of STF +10 shown for Activity P is actually two different Total Float values, just added together. We can prove that this is so. What would happen if we were to change the Start-No-Earlier-Than (SNET) Date Constraint for Activity A from Day 0 to Day 3? Wouldn’t that reduce the Total Float for all Activities?

Figure F0830 confirms that it does just that. And doesn’t this also prove that the Total Float portion of TF +7 belongs to the Lower Path Segment exclusively?

So why does this matter to you as a Project Facilitator?

- First, this last examination reinforces why we call this type of Relationship a Progressive one; because the Activities are related in such a way that the performance of upstream Activities have the potential to, and quite often do, restrict or impact the Time performance of downstream Activities.
Second, the Symbiotic Relationship between the **Upper Path Segment** and **Lower Path Segment** is also better understood. For we see that all that separates the two three-Activity strings from being equally critical is the seven-day difference in their lengths. If any of the Activities in the Upper Path Segment string were to take less Time, or the ones in the Lower Path Segment were to take more Time, the difference between them would change immediately. And with those changes, which happen all of the time on Projects, attitudes toward rushing (or foot-dragging) change as well.

The main point we are trying to make is that virtually every Activity in a Schedule resides on more than one Activity Path. So when you see Total Float or Free Float values reported against a given Activity, just remember that what you are reading is the “worst case”[^5] among all of the values that are actually at play on that single Activity.

Don’t start turning off your mental light just yet as this chapter winds down. Stay with us a little longer! Look back to **Activity Z** (or **Activity A**) in [Figure F0828](#). Do you now see that each of these Activities has two Total Float and Free Float values – not just the one being reported?

You see, it was easier for you to appreciate that the Total Float of **TF +10** in **Activity P** was comprised of two portions. But we didn’t extend our thinking to what that really means.

- It means that **Activity P** has Total Float of **TF +10** because the **Lower Path** (**Activity A**) is on the **Lower Path** (**Activity A** ► **Activity P** ► **Activity R** ► **Activity T** ► **Activity Z**). This has Total Float of **TF +10**, and **Activity P** is only on the **Lower Path**.

- Likewise, **Activity F** shows Total Float of **TF +3** because it belongs to the **Upper Path** (**Activity A** ► **Activity F** ► **Activity G** ► **Activity H** ► **Activity Z**), which has Total Float of **TF +3**.

[^5]: Technically, you are reading the Least Total Float.
But Activity A and Activity Z belong to two Activity Paths, and so they actually have two Total Float values applied to them. We only report the most critical of these.

So the next time you see a Total Float or Free Float value reported alongside an Activity, remember that the odds are extremely high that the Activity most likely has other Total Float or Free Float values lying just below the surface.

Enough said.
CHAPTER NINE

9A  Essential, Supplemental, Exploitive Schedule Elements
9B  Elements that Provide Advanced Modeling Capabilities
    9B1 Understanding Date Constraints
        9B1a Date Constraint Types and the Calculation Parameters
            9B1a-i Minimum Condition Date Constraints
            9B1a-ii Maximum Condition Date Constraints
            9B1a-iii Fixed Condition Date Constraints
        9B1b Date Constraint Calculations and Concerns
            9B1b-i CPM Calculations Applying Two Main Date Constraint Types
            9B1b-ii Implications of Using Two Main Date Constraint Types
    9B2 Understanding Work Performance Calendars
    9B3 Understanding Software Settings
        9B3a Interruptible versus Contiguous Durations
        9B3b Zero Total Float
        9B3c Zero Free Float
    9C  Elements that Facilitate Schedule Data Management
        9C1 Understanding Activity Codes
        9C2 Understanding a Work Breakdown Structure (WBS)
            9C2a What a WBS Is
            9C2b How a WBS is Used
        9C3 Activity-Resident Notations and Logs
Supplemental Schedule Elements

9A: Essential, Supplemental, and Exploitive Schedule Elements

The elements that comprise a Project Schedule can be divided into those that are essential, those that are supplemental, and those that are exploitive. Up to this point in the book we have been discussing Essential Schedule Elements. In this chapter we will explore Supplemental Schedule Elements. In Chapter Ten we will consider Exploitive Schedule Elements.

Here is what the ICS-Dictionary says about the three different categories of Schedule Elements:

- **Schedule Elements Taxonomy:** The informational content of Project Schedules can be divided into Essential, Supplemental, or Exploitive Schedule Elements. Essential Schedule Elements are ones without which a Project Schedule cannot effectively perform as a Project Time Management Tool. While not absolutely required, Supplemental Schedule Elements are ones that nevertheless enhance the Project Schedule’s ability to model Project Execution Strategy. The ultimate objective of Exploitive Schedule Elements is to extract temporal information from the Schedule for non-temporal Project Management purposes. If not carefully controlled, the incorporation of Exploitive Schedule Elements can weaken, and even destroy, a Schedule’s ability to function as a reliable Project Time Management Tool.

As to what is contained under the heading of Essential Schedule Elements, consider this itemization:

- **Activities:** Activity Identifier, Activity Description, Activity Duration.
- **Relationships:** Relationship Categories (Communal, Symbiotic, and Progressive); Progressive Restrictions (Immediately-Restricting, Eventually-Impacting); Restriction Linkages (Default, Start, Finish, Holdback); Performance Restriction Attributes (Restriction Type, Restriction Delay, Restriction Flow).
Calculated Dates: Earliest Dates (Earliest Start, Earliest Finish); Latest Dates (Latest Start, Latest Finish).

Derived Values: Activity Path Designations (Activity Path, Path Segments, Subnetworks, Networks); Float (Total Float, Free Float).

Hopefully the terms and concepts in the above itemization seem pleasantly familiar to you. They should, because without knowing it you have reached a major milestone in your study of Project Time Management. You have learned everything that you need to know to properly and effectively create basic Critical Path Method Logic Diagrams that are capable of meaningfully and usefully modeling Project Execution Strategy.

Now it is time to move on to a discussion of various Supplemental Schedule Elements, a term that the ICS-Dictionary defines as follows:

Supplemental Schedule Elements: The informational content of Project Schedules can be divided into Essential, Supplemental, or Exploitive Schedule Elements. For Project Time Management purposes, Supplemental Schedule Elements are non-essential, but can nevertheless enhance the Project Schedule’s ability to model Project Execution Strategy. The two primary benefits of Supplemental Schedule Elements are (a) providing advanced modeling of Project Execution Strategy (Date Constraints, Work Performance Calendars, Software Settings) and (b) facilitating the management of inherent schedule data (Activity Codes, Activity Notations and Logs).

Consistent with this definition, we have divided this chapter into two general discussions:

- Schedule Elements that Provide Advanced Modeling Capabilities
- Schedule Elements that Facilitate Schedule Data Management

9B: Elements that Provide Advanced Modeling Capabilities

The Progressive Relationships of the Critical Path Method — with their four Performance Restriction Linkages, robust symbolic language, and powerful set of Arithmetic Formulas — have given us the ability to very effectively model Project Execution Strategy. But despite their collective versatility there are still nuances of Project Execution Strategy that these tools, in and of themselves, fail to address.

For these nuances, we need additional mechanisms. Specifically, this subsection will now discuss:

- Date Constraints
- Work Performance Calendars
- Scheduling Software Settings
If some, or all, of these headings sound somewhat familiar it is because we have actually begun to explore each of them to a limited extent in earlier chapters. Now we will take an in-depth look at them in order to ensure that you really understand how these guys work and what they can do for – or to – your Project Schedules.

**9B1: Understanding Date Constraints**

Let’s reprint the definition for Date Constraints, compliments of the ICS-Dictionary:

**Date Constraints**: Date Constraints are computational criteria for Calculated Dates that are superimposed on a given Activity, such that any Activity Path Segments (to which the Activity belongs) may be artificially accelerated, delayed, or fixed in time.

The word “may” tells us that Date Constraints have the potential to influence Earliest Dates and Latest Dates, but need not actually have that effect in every situation.

Whether or not a Date Constraint actually causes a delay, acceleration, or anchoring of Calculated Dates depends on factors that will be fully discussed in this chapter.

**9B1a: Date Constraint Types and the Calculation Parameters**

Date Constraints are conditions that we impose on the underlying CPM Logic, which are taken into consideration when Forward Pass and Backward Pass Date Calculations are performed. Some Date Constraints create minimum conditions while others set maximum conditions. Still others (less frequently used) establish absolute conditions. Let’s take a look at each Date Constraint category separately.

**9B1a-i: Minimum Condition Date Constraints**

The following two Date Constraints are what Cognitive Project Management calls Minimum Condition Date Constraints in that they dictate the minimum Earliest Dates that an Activity can have.

Notice that this means that Minimum Condition Date Constraints only apply when performing a Forward Pass; they have no effect on Latest Dates.
We begin our study of Date Constraints with the Start-No-Earlier-Than Date Constraint, which establishes the earliest that an Activity can start. Normally, when we perform a Forward Pass, we calculate the Earliest Start for a given Activity as you have been taught in Chapter Five. But if a Start-No-Earlier-Than Date Constraint has been imposed on an Activity, we compare the calculated Earliest Start date with the Start-No-Earlier-Than (SNET) Date Constraint. If the Calculated Date is later in Time, then it prevails. If the Date Constraint is later in Time, then it prevails. Take a look at Figure F0902, which we hope is self-explanatory.

While the Finish-No-Earlier-Than Date Constraint works the same as the previous one, it is used far less frequently than the Start-No-Earlier-Than Date Constraint. Figure F0904 shows how this Date Constraint works. It determines the earliest that an Activity might finish.

Maximum Condition Date Constraints

Maximum Condition Date Constraints dictate the maximum Latest Dates that an Activity can have. This of course means that they apply during the Backward Pass Date Calculations and affect only Latest Dates.
9 Supplemental Schedule Elements

Start-No-Later-Than Date Constraint

The Start-No-Later-Than Date Constraint is not very commonly used, but there are rare occasions when it is the best model of intended Project Execution Strategy. This Date Constraint basically says that, Progressive Logic notwithstanding, an Activity cannot start any later than a given date. Study Figure F0906.

F0904: Finish-No-Earlier-Than Date Constraint

F0906: Start-No-Later-Than Date Constraint
Finish-No-Later-Than Date Constraint

The Finish-No-Later-Than (FNLT) Date Constraint is, by far, the most used Date Constraint in CPM Schedules, the one we use to depict a Project Execution Commitment condition. A Deadline Milestone can be “set in concrete” by imposition of a Finish-No-Later-Than Date Constraint. See Figure F0908.

Absolute Condition Date Constraints

The third category of Date Constraints as categorized by Cognitive Project Management are called Absolute Condition Date Constraints, in that they impose firm dates that cannot be overridden by Logic, during either a Forward Pass or Backward Pass.

Because they override Logic, thereby denying the Logic-driven Schedule the very flexibility that it needs to function as a dynamic model of Project Execution Strategy, their use is widely discouraged by seasoned Project Facilitators. We include a brief discussion of Absolute Condition Date Constraints so that you know that they exist and how they work.

Expected Finish Date Constraint

The Expected Finish Date Constraint provides a way for the Project Facilitator to force an Earliest Finish Date to fall on a particular Calendar Date. Not only does this Date Constraint override Logic, it also alters the Remaining Duration. Consider Figure F0910.
In the **UPPER PANEL** we see two Activities tied by way of a Default Restriction. In the **LOWER PANEL** we see the effects of an Expected Finish. Notice that it actually changed the Activity Duration of Activity B. [1]

Needless to say, this Date Constraint is extremely potent and, therefore, equally dangerous if not carefully employed. As with many of the Date Constraints discussed in this chapter, they should be used sparingly, for many of them have the power to override Performance Restriction Linkages and, in so doing, may weaken the Critical Path Method’s ability to dynamically model the Project Execution Strategy.

---

[1] The behavior of the Expected Finish, described above, reflects how this constraint works in P3. Other Scheduling Software may have a different name for the function, and it may behave differently as well.
Restriction Linkages, and which shared a straight-forward Total Float value of \( TF +19 \), is now something else entirely. The three Activities have three different Total Float values, none of which is the same as the one Total Float value that they shared before imposition of the Mandatory Start Date Constraint.

Keep in mind that the domino effect of a Mandatory Start Date Constraint is almost always far more extensive than what is shown in Figure F0912. Here we only see three Activities, but if these Activities were in the middle of a 2,000 Activity Schedule, depending on the Logic, the Mandatory Start Date Constraint could easily affect Latest Dates all the way back to the start of the Schedule, and Earliest Dates all the way to the end of the Schedule!

- **Mandatory Finish Date Constraint**

  The Mandatory Finish Date Constraint works the same was as its counterpart, the Mandatory Start Date Constraint. Figure F0914 shows how it, too, can send unwanted ripples throughout a Schedule if not very carefully administered.

- **Start On Date Constraint**

  Perhaps the most egregious of all Date Constraints is the Start On Date Constraint. It is identical to the Mandatory Start Date Constraint, with one major exception. *It ignores Logic!* That is, it can fix an Earliest Start that is earlier than the Restricting Activity's Earliest Finish. Likewise, it can impose
a Latest Start Date (or calculated Latest Finish Date) that is later than its Restricted Activity's Latest Start. For examples of these violations of drawn Logic, see Figure F0916.
Earliest Dates Overridden: Notice how Activity B has an Earliest Start of Day 36, as imposed by the Start On Date Constraint, even though Activity A, the Restricting Activity to Activity B, has an Earliest Finish 14 days later!

Latest Dates Overridden: Activity C has a Latest Start of Day 41, which should also constitute the Latest Finish of Activity B. Since Activity B has an Activity Duration of ten days, its Latest Start should be Day 31. However, thanks to the Start On Date Constraint, its Latest Start is shown as Day 36, five days later than the Restricted Activity’s Latest Start should have been.

All of the previous Date Constraint names and calculations are based on Primavera Project Planner (“P3”) version 3.1. While this software is no longer sold by Primavera Products (now part of Oracle), it remains in use all around the world. It is highly likely that you will run into Schedules built and maintained in P3; you need to be aware of Date Constraints that could override otherwise sound Progressive Relationship Logic.

9B1b: Date Constraint Calculations and Concerns

As previously stated, most of the Date Constraints discussed above are not that commonly used. Two, though, are extensively applied to Critical Path Method Schedules, so we need to examine them in much greater depth. In this subsection we will experiment with
the imposition of Start-No-Earlier-Than (SNET) and Finish-No-Later-Than (FNLT) Date Constraints. The former is a Minimum Condition Date Constraint; the latter is a Maximum Condition Date Constraint.

We will also take a closer look at the Expected Finish, an Absolute Condition Date Constraint. While this Date Constraint is less popular, it is used more than enough to justify us spending a little time explaining how it can affect basic Critical Path Method Date Calculations.

**9B1b-i:  CPM Calculations Applying the Two Main Date Constraint Types**

**More About the Start-No-Earlier-Than Date Constraint:** Start-No-Earlier-Than is the Date Constraint you would normally use to reflect a Release Milestone. For example, you could use it to establish a Start Date for an entire Schedule, or some major portion of the Schedule. Most Construction Projects have an official Start Date. That date may be noted in a NOTICE TO PROCEED (NTP) document, or perhaps in an AWARD LETTER sent to the Contractor by the Owner. This official “go-ahead” is reflected in the Project Schedule by way of a Start-No-Earlier-Than Date Constraint, with the official Project KICK-OFF DATE imposed on the very first Activity subject to the release of Work.

Another common use of the Start-No-Earlier-Than Date Constraint is to artificially postpone certain portion of the Project Work until after a given Calendar Date. Sometimes,
for practical reasons, an Owner doesn’t want the entire Project turned over to the Contractor all at once, and one way to control the completion of phases of Work is to control the start of those phases.

Start-No-Earlier-Than Date Constraints only affect Earliest Dates — of the Activity receiving the Date Constraint as well as of any Progressively-related downstream Activities. In Figure F0918, we have drawn a simple Logic Diagram that contains two Start-No-Earlier-Than Date Constraints. We want you to tell us how much Total Float Activity D lost as a result of the imposition of the Start-No-Earlier-Than Date Constraint on its Earliest Start.

To answer this question you will of course need to perform Forward Pass and Backward Pass calculations, and then calculate Start Total Float and Finish Total Float for each Activity. What is your answer? Try calculating all of these values on your own, before looking at our solution, which appears in Figure F0920. From this exercise, it is easy to see how a Start-No-Earlier-Than Date Constraint can squander valuable Total Float.

More About the Finish-No-Later-Than Date Constraint: Now let’s look at the Finish-No-Later-Than Date Constraint and how it interacts with a Start-No-Earlier-Than Date Constraint in the same Logic. First, try to work through the CPM values in Figure F0922 before proceeding to the next paragraph.
We hope you were able to perform all of the Forward Pass and Backward Pass calculations, as well as determine the Start Total Float and Finish Total Float values for each Activity. Did your results match ours in Figure F0924? If not, please don’t go any further until you identify what it is that still alludes you and then go back to acquire a solid mastery of it.

We will now proceed from this point forward with the assumption that you are completely proficient at performing the basic CPM calculations.

We want to ask you a few questions about Figure F0924. Try to answer each question on your own before reading our answers. If you need to, take a blank sheet of paper and cover up our answers, slowly sliding the paper down the page just enough to reveal the question – and then attempt your answer before reading ours.

☐ Question 1: What is the Project's Critical Path?

☐ Question 2: Based on the Cognitive Project Management definition of a Critical Path, with respect to PROJECT COMPLETION what is the Critical Path and what is its Total Float value?

☐ Question 3: What is the Critical Path for ACTIVITY D and what is its Total Float value?

☐ Question 4: How many different Activity Paths are there in this small Logic Diagram?

☐ Question 5: How many Critical Paths are there in the Logic Diagram?

☐ Question 6: What are the Total Float values for the other three Activity Paths leading
Question 7: What Activities are on the Activity Path that has Total Float of $TF +15$ (even though some of its Activities report lower Total Float as a result of being multi-path residents? How can you tell?

Question 8: Which Activities belong to only one Activity Path?

Question 9: How is the most number of Activity Paths than any Activity belongs to?

Question 10: If you could shorten one Activity (by doubling its workforce and thus cutting its Duration in half), which Activity would you choose to shorten the Project's overall length by the greatest amount of Time?

Answer 1: You should have answered this question with another question: “How do you define Critical Path?”

Answer 2: Well, the last Activity in the Schedule is Activity H and it is constrained by a Finish-No-Later-Than date of Day 56. Activity H shows Total Float of $TF +8$, and if we trace that Total Float backwards we determine the Start-No-Earlier-Than Date Constraint imposed on the start of Activity E having a value of Day 31. So, the correct answer to the question would be: Activity E $\Rightarrow$ Activity H. The Total Float value is $TF +8$.

Answer 3: Activity D reports Total Float of $TF -2$ and its Critical Path traces through Activity A and Activity D.

Answer 4: We count five (5) different Paths, as follows:

Path #1: Activity A $\Rightarrow$ Activity B $\Rightarrow$ Activity C $\Rightarrow$ Activity H
Path #2: Activity A $\Rightarrow$ Activity D $\Rightarrow$ Activity E $\Rightarrow$ Activity H
Path #3: Activity A $\Rightarrow$ Activity F $\Rightarrow$ Activity G $\Rightarrow$ Activity H
Path #4: Activity A $\Rightarrow$ Activity D
Path #5: Activity E $\Rightarrow$ Activity H

Answer 5: There are two, one for each Deadline Milestone. Of the four Paths leading to Activity H’s completion, the Least Total Float Path bears Total Float of $TF +8$, and travels through Activity E $\Rightarrow$ Activity H.

Answer 6: $TF +22$, $TF +16$, and $TF +15$.

Answer 7: Activity A $\Rightarrow$ Activity D $\Rightarrow$ Activity E $\Rightarrow$ Activity H. We can tell by resuming the Backward Pass from the Latest Start of Activity E, but without considering the Finish-No-Later-Than Date Constraint of Day 22 imposed onto Activity D. Without this Date Constraint, Activity D’s Latest Finish would be the same as Activity E’s Latest Start, or Day 39. Since Activity D’s Earliest Finish is Day 24, the difference ($TF +15$) is the Total Float of this Activity, in terms of
F0926: Finding Critical Path Through Date Constraints
its residency along the Activity A ► Activity D ► Activity E ► Activity H path.

Answer 8: Activity B, Activity C, Activity F, and Activity G.

Answer 9: Four (Activity H)

Answer 10: Activity H. Obviously Activity A and Activity H have the largest Durations, and therefore cutting either of their Durations in half would yield the largest reduction to their respect Critical Paths. But, as we learned from Question 7, were it not for the Finish-No-Later-Than Date Constraint on Activity D, Activity A would be reporting Total Float of TF +15.

What is causing the final Activity in the Schedule, Activity H, not to finish until Day 48 is the Start-No-Earlier-Than Date Constraint of Day 31, imposed on Activity E. Cutting in half the Activity Duration of either Activity E or Activity H would improve the Project's end date. But since Activity H has a larger Duration than Activity E, it is the better choice for overall Project Acceleration. If Activity H’s Duration is reduced to, say, six days, the Earliest Finish for Activity H would become Day 43. Cutting Activity A in half, a reduction in 5 days would do nothing to shorten the Schedule’s overall length.

9B1b-ii: Implications of Using Two Main Date Constraint Types

Try Googling on the impact of multiple Date Constraints on CPM calculations and you will invariably get a slew of responses that all say roughly the same thing: “It causes havoc!” You have now seen this consequence with your own eyes. But what you do not yet realize, because its coverage is located in another volume of the ICS-Compendium, is that many standard Project Time Management reports depend on the ability to sort Activities by their Total Float.

However, when a Schedule contains even just a handful of Date Constraints the ability to trace either a Longest Path or a Least Total Float Path seems to disintegrate. And if you don’t set your software for Most Critical Total Float (which we shall discuss in just a few pages), it may become difficult if not entirely impossible to trace a consecutive Critical Path from Path Start to Path Finish.

To understand this better, first perform the CPM calculations needed to fill in all of the missing information in Figure F0926. We’ll wait. Now check your answers with ours, shown in Figure F0928. The following discussion will be based on Figure F0928.

Date Constraints Consume Total Float

The most obvious implication of Date Constraints is that, when they are in effect, they arbitrarily consume (or squander) Total Float. For instance, the
<table>
<thead>
<tr>
<th>Activity</th>
<th>DUR</th>
<th>FTFSTF</th>
<th>LS LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity A</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity B</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity C</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity D</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity E</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity F</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity G</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity H</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity J</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity K</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity L</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity M</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity N</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity P</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity Q</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity R</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity T</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>Activity V</td>
<td>8</td>
<td>-6</td>
<td>10</td>
</tr>
</tbody>
</table>

F0928: Our Solution to Problem in F0926
Finish-No-Later-Than Date Constraint tacked on to Activity E eliminated 28 days of Total Float. Were it not for that Date Constraint, Activity E, Activity D, Activity C, Activity B, and the latter portion of Activity A would have enjoyed Total Float of TF +22. Instead, those same Activities are now reporting Total Float of TF -6.

Likewise, the Start-No-Earlier-Than Date Constraint pinned to the start of Activity P consumed eight days of Total Float to Activity P, Activity Q, Activity R, Activity V, the latter portion of Activity Q, and Activity X.

Date Constraints Make Path Tracing More Difficult

As we have already noticed, with each additional Date Constraint it becomes more and more difficult to trace an Activity Path from its Path Start to its Path Finish. Path Tracing becomes even more challenging if the Software Setting for Total Float is set for either Start Total Float or Finish Total Float. This is why Cognitive Project Management recommends using the Most Critical Total Float setting.

Date Constraints Defy the Three Popular Definitions of a Critical Path

Now we can return to a discussion we had back in Chapter Seven at a point when you had not yet been fully exposed to Date Constraints and the chaos that they can bring to a Network-Based Schedule. Do you remember that we were comparing the two most popular ways of defining or identifying a Critical Path: Longest Path or Least Total Float Path? Do you also remember that we acknowledged a third Activity Path designation, the Driving Path (although this is not actually a Critical Path, per se)?

Referring again to Figure F0928, let’s apply each of these definitions and see how well they hold up.

- Least Total Float Path: It is easy to find the Activities with the Least Total Float Path. They are:

Activity A ► Activity B ► Activity C ► Activity D ► Activity E.

These five Activities bear Total Float of TF -6, the lowest of any Activities in the Schedule. How comfortable do you feel citing these five Activities as the Critical Path of the Schedule? True, they do constitute the Least Total Float Path, but on the other hand – these five Activities do not actually contribute to why the final Activity in the Schedule, Activity X, has Total Float of TF -2. Do they?
• **Longest Path**: To find the Longest Path, we remove all Date Constraints and perform a pure Forward Pass. Let’s try. Look at Figure F0930. Based on the Forward Pass, it appears that the Longest Path through the Schedule is 86 days long, and cuts through the Activities shown by the red line. We used the Earliest Finish to set the Latest Finish for Activity X, and then did the Backward Pass. The Activity Path with Total Float of TF +0 is the Longest Path. The Activities are: Activity A ► Activity M ► Activity N ► Activity P ► Activity Q ► Activity R ► Activity V ► Activity W ► Activity X.

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>Start TF</th>
<th>Finish TF</th>
<th>Most Critical TF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity A</td>
<td>-6</td>
<td>-2</td>
<td>-6</td>
</tr>
<tr>
<td>Activity M</td>
<td>+6</td>
<td>+6</td>
<td>+6</td>
</tr>
<tr>
<td>Activity N</td>
<td>+6</td>
<td>+6</td>
<td>+6</td>
</tr>
<tr>
<td>Activity P</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>Activity Q</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>Activity R</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>Activity V</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>Activity W</td>
<td>+7</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>Activity X</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
</tbody>
</table>

**F0932: Three Different Total Float Values**

Now let’s flip back to Figure F0928, and pay attention to the Total Float values for these same Activities, the ones we have found to be residing on the Longest Path through the Schedule. Of course it depends on whether we use Start Total Float values, Finish Total Float values, or Most Critical Total Float values.

**Figure F0932** shows that if we were to have used Start Total Float, the Longest Path would contain four different Total Float values. Using Finish Total Float, then Total Float values would change twice. Using Most Critical Total Float, Total Float values would change three times! How can the Critical Path of a Schedule have different Total Float values along its length?

• **Driving Path**: That leaves us with the most recent “solution” to the elusive Critical Path “problem.” Using this approach, we work backwards from the end of the Schedule and, for each Activity, find the Restricting Activity with the latest Earliest Finish. Feel free to take a crack at it, but when we tried we ended up with the same Activities as listed above. And we are left with the same question: “How can a Project’s Critical Path go from negative Total Float to positive Total Float and then back to negative Total Float?”
We hope this discussion has helped you better appreciate three unfortunate realities.

- Date Constraints can play havoc on Total Float and Critical Path computations.
- The idea of a single Critical Path for a Schedule containing more than one Release/Deadline Milestone pair is often unrealistic.
- Finally, since most Schedules have three or more Date Constraints, the prevailing Critical Path definitions are simply inadequate.

9B2: Understanding Work Performance Calendars

Work Performance Calendars are basically tables that cross-reference the days that have been allocated for performing Project Work, called Workdays, with the standard Date Calendar that hangs on a wall. Each Calendar Day is designated as either a Workday or a Non-Workday, and then the Workdays are sequentially numbered.

- Figure F0934 shows a Work Performance Calendar for a seven-day Workweek. The Workday numbers appear in the center of each box, and the Calendar Day (date) numbers appear in the upper left corner of each box. Hence, to no one’s surprise, Workday 16 falls on Calendar Day 16. Figure F0934 is labeled, C1: 7-DAY WORK PERFORMANCE CALENDAR.

- Figure F0936 presents a Work Performance Calendar in which Sundays have been blocked out as Non-Workdays. It is entitled, C2: 6-DAY WORK PERFORMANCE CALENDAR. Notice that the Workday numbers skip the Sundays. Now, Workday 16 falls on Calendar Day 19.

- In Figure F0938, we find C3: 5-DAY WORK PERFORMANCE CALENDAR, in which both Saturday and Sunday have been designated as Non-Workdays. Not surprisingly, Workday 16 now falls on Calendar Day 22.
Our final example, Figure F0940, is of a Five-Day Work Performance Calendar where Federal holidays are recognized. This being a calendar for the month of January, January 1st is blocked out for New Year’s Day, and January 18th is marked as a Non-Workday in honor of Martin Luther King’s Birthday. In this calendar, labeled C4: 5-Day Work Performance Calendar with/Holidays, Workday 16 falls on Calendar Day 26.

Now that you know what Work Performance Calendars are and how they function, let’s start using them to convert Ordinal Dates to Gregorian Dates as a final step in our Forward Pass and Backward Pass calculations. Figure F0942 provides us with a small Logic Diagram that we will be working with for a few pages. There are three steps to the process of converting from Ordinal Dates to Gregorian Dates.

- Decide which Work Performance Calendar applies to each Activity.
- Convert the Start Dates from End-of-Day Perspective to Start-of-Day Perspective.
- Substitute the Ordinal Dates with the Gregorian Dates.
We will begin our date conversion process by assuming a **SEVEN-DAY WORK WEEK**, so we will use calendar **C1**. The results are shown in **Figure F0944**. This one was a no-brainer, so now let’s convert to Gregorian Dates using **C3**, the **FIVE-DAY Work Performance Calendar**. The results are in **Figure F0946**.

We wonder if you have noticed what appears to be an inconsistency. The Total Float values have not changed even though the spread between Earliest Dates and Latest Dates is greater when Gregorian Dates are used to calculate the difference. For instance, for **ACTIVITY H**, the difference between the Earliest Start (**13JAN**) and the Latest Start (**18JAN**) is **five** Calendar Days.

The reason that Total Float has not changed is that Total Float is always measured as Workdays, not Calendar Days. More specifically, **Total Float is measured based on the Work Performance Calendar applied to that Activity**. In other words, in the Scheduling Software, a reverse conversion process occurs, where the Calendar Days are converted
back to Workdays, and the difference is calculated. To reword ourselves, the Total Float values report the spread in Workdays, not Calendar Days.

At this point we want you to do a Gregorian Date conversion, this Time using **Calendar, C4**, which is the Five-Day Work Week plus Holidays. Use **Figure F0948** as your worksheet. Good luck. Our solution is shown at **Figure F0950**.

Now that you think you have mastered the art of converting Ordinal Dates to Gregorian Dates, we are about to make things very complicated. You see, on real Projects not all Actions are subject to the same Work Performance Calendars.

For instance, a Contractor may work his field crews on a six-day week, perhaps even working holidays. But the home office folks who handle submittals and paper work will typically work a five-day work week. For sure, the Owner will insist on a five-day work
This last exercise that we will be asking you to perform is intended to teach you just how much havoc can be cast upon an otherwise good Schedule when *multiple* Work Performance Calendars are applied.

Before you can do this last date conversion exercise, we want to show you how to integrate Activities that use different Work Performance Calendars. You will be working with Figure F0954.

Just for this exercise, assume that for each Activity in this Logic Diagram the Work Performance Calendar number will be equal to the Activity Duration minus one. So, for instance, *Activity A* has a three-day Duration, so it will use Work Performance Calendar C2. *Activity R* has a five-day Duration, so it will use Work Performance Calendar C4. And so forth – okay?

To get you started, let’s calculate the Gregorian Dates for *Activity A* and *Activity F*. Follow along in Figure F0952.

- **Step 1:** We start with *Activity A* and observe that it has a Duration of three, so we know to use Work Performance Calendar C2. The first available Workday in Work Performance Calendar C2 is 01JAN. We count out the Activity Duration, which is three, across the available Workdays, and conclude that the Earliest Finish for *Activity A* is 04JAN.

- **Step 2:** Next we note that *Activity F* has a Duration of two, so we will be using Work Performance Calendar C1. Since *Activity A*’s Earliest Finish is 04JAN, referring to Work Performance Calendar C1 we determine that the next available Workday is 05JAN. That means that the Earliest Finish for *Activity F* will be 06JAN.
Now we would like you to perform the Gregorian Date conversion for the entire Logic Diagram shown in Figure F0954. Follow the same process that we just showed you, all the way through the Forward Pass. Once you are done, switch directions and commence the Backward Pass. Remember to use every preceding Date Calculation as your starting point for the next Date Calculation. Give it your best effort. Use Figure F0954 as your worksheet.

If it gets too hard, then try to follow what we did in Figure F0956. When you are done, we want you to notice that Activity F has Total Float of TF +4, even though it is succeeded by Activity G with Total Float of TF +3, and preceded by Activity A with Total Float of TF +0. What you are seeing is the effect of different Work Performance Calendars on Total Float values. This is why the use of multiple Work Performance Calendars should be carefully balanced against the masking effect they will have on Total Float (and, thus, Critical Path) reporting.
9B3: Understanding Software Settings

Despite the previous section making specific references to Scheduling Software options found in PRIMAVERA PROJECT PLANNER (P3), this textbook is not meant to recommend, encourage or otherwise be tailored to any particular software brand. Now we are about to discuss certain Scheduling Software Settings that require our attention, either because they are especially common, or measurably potent, or both.

Software Settings represent choices that we can make to determine or affect the way that the Scheduling Software performs. Software Settings differ from attributes that are assigned to Schedule Elements. For example, a Date Constraint is an attribute that is assigned to an Activity. Designating July 4th as a Non-Workday is an attribute assigned to an Work Performance Calendar. By comparison, Software Settings tend not to be specific to a given component, although occasionally they are. (Every rule has its exceptions, we are told).

Think of Software Settings as Procedural Rules. If we were performing a particular process manually, we might call such a formalized process a WRITTEN PROCEDURE. When we instruct the software to abide by a formalized process, such instruction is accomplished by way of a Software Setting.

One Software Setting that we have already become quite familiar with is the one that tells the software to define a Total Float as the Most Critical Total Float. If you and I were performing manual Total Float calculations, we might agree ahead of time that we will use the lesser of the Start Total Float or Finish Total Float.

Another formalized process is to use Start-of-Day (or End-of-Day) when performing Forward Pass and Backward Pass Date Calculations. Some software programs allow you to dictate which Point-of-Day Perspective to use; others build that decision into the program’s hard coding, and there is no option available to the user.

In this third, and final, section of our discussion of ELEMENTS THAT PROVIDE ADVANCED MODELING CAPABILITIES, we will examine four Software Settings that are of extreme importance to you as a Project Facilitator. Each of them has the power to single-handedly destroy an otherwise great Schedule. Knowing that they exist and how they work is not only valuable to you just because you will be a user of Scheduling Software that will likely contain these options. Your understanding of the dynamic effect of these Software Settings will greatly advance your overall understanding of CPM MECHANICS, the main goal of this text.
Let’s start with the Logic Diagram shown in Figure F0958. We see that Activity A and Activity B are in an Immediately-Restricting Progressive Relationship, linked by a Start Restriction of SS:2 and a Finish Restriction of FF:2. Activity A has an Earliest Start of Day 100 and an eight day Duration, and so we calculate an Earliest Finish for Activity A of Day 108.

Now here is where we need for you to pay special attention. Activity B’s Earliest Start, based on the Start Restriction of SS:2, would be calculated as Day 102, right? And with a Duration of six days, if it were not for the Finish Restriction of FF:2, Activity B’s Earliest Finish would be Day 108 (Day 102 plus the Activity Duration of six), right? But thanks to that FF:2 the Earliest Finish for Activity B is calculated at Day 110.

Without ever saying so, we have been using what is called, at least by P3, an Interruptible Duration. To understand why, look at the Bar Chart in Figure F0958. Do you see how Activity B’s performance was interrupted, waiting for the completion of Activity A before it could complete the last two days of its own work?

Notice also that as a consequence of the interruption, the Start Total Float for Activity B is different (larger by two days) than its counterpart, the Finish Total Float. Why is the difference in Total Float values two days? Think about it: there are two ways to get from the starting Moment in Time of Activity A to the ending Moment in Time of Activity B.

☐ Route through All of Activity A: One route is through all of Activity A, and then through the last two days (FF:2) of Activity B. The cumulative length of this route would be 8 + 2, or ten days.
Route through All of Activity B: The other route is through the first two days (SS:2) of Activity A and then through all of Activity B. The cumulative length of this route would be 2 + 6, or eight days.

So, it would seem that the longer route is through all of Activity A and the end of Activity B. This explains why Activity B has to “hang around” for two days (as shown by the broken horizontal line in the Bar Chart) waiting for the chance to resume and complete its work.

And as just noted, this difference of two days is what is reflected in Activity B’s Start Total Float of TF +5 versus its Finish Total Float of TF +3 (which is the same as the Total Float for all of Activity A).

Driven by a concern for those workers having to stand around for two days while waiting for Activity A to finish, some kind souls have reasoned that it would actually be more efficient if Activity B was to be delayed from starting until such a time as it could then proceed right on through to the end of its work, uninterrupted by the Finish Restriction. They came up with a technique which Primavera calls a Contiguous Duration.

And here is how it works. Instead of calculating the Earliest Finish for Activity B in the traditional way (as Earliest Start of Activity A plus the Start Restriction Delay of SS:2, and then calculating the Earliest Finish for Activity A – you know, following a Forward Pass mentality) … they insist that Activity B’s Earliest Finish be calculated first – and then they subtract the Activity Duration to arrive at Activity B’s Earliest Start!
We realize that this is backwards, at least as far as you have been taught about the Forward Pass. But here is how the calculations would work. You’ll want to follow along in Figure F0960. Using the Contiguous Duration approach, the Earliest Start and Earliest Finish for Activity A would be determined in the normal manner. But when it comes to Activity B, this is where things start going backwards.

Since the Earliest Finish of Activity A is Day 108 and Activity B’s Earliest Finish is linked back to the completion of Activity A with an FF:2 Restriction Delay, the Earliest Finish of Activity B is Day 110. Now this is where things really go backwards: since Activity B’s Earliest Finish has just been calculated at Day 110 and Activity B’s Duration is given as six days, then the Earliest Start for Activity B would be derived by the formula:

CONTIGUOUS DURATION: EARLIEST START FORMULA

\[
ES_y = EF_y - DU_y
\]

\[
EF_y = \text{Day 110}
\]

\[
DU_y = 6
\]

\[
ES_y = \text{Day 110} - 6 = \text{Day 104}
\]

Notice that this changes the Start Total Float from TF +5 to now being TF +3. What this technique has done, then, is delay the start of Activity B by two days (thus giving up two days of Total Float on the front of the Activity), so that the Activity, once started, can work straight through continuously (or, per P3 lingo, contiguously).

We will leave for elsewhere in the ICS-Compendium to comment on the prudence of this technique. But you should be aware of its existence, and be on the lookout for it in Schedules that you encounter.
To give you an idea of how it might confront you, take a look at Figure F0962, which gives you just a glimpse of how innocuous this practice might be. In this figure, everything looks normal. Now compare this to Figure F0964, where the Contiguous Duration setting has been chosen. Notice all of the changes in Earliest Dates and reductions in Start Total Float values?

While the choice to set Activity Durations to Contiguous rather than Interruptible may force an elimination of stops and starts, it does so at a high price. It does so by arbitrarily squandering precious Total Float. If you add up the Total Float values (both Start Total Float and Finish Total Float) for the nine Activities, you will find that in the Interruptible scenario, the aggregate Total Float available to the Schedule is 85 days. The aggregate for the Contiguous scenario is 44 days. This means that, numerically speaking, virtually half of the available Total Float was simply wasted – as performers chose, one after the other, not to commence their work until the last available minute. Does this make sense to you?

9B3b: Zero Total Float

This, and the next discussed topic, Zero Free Float, are Software Settings offered by P3. Other software may provide similar functionality to P3’s Zero Total Float. As stated

2 We acknowledge that this is a technical observation, not one with much practical relevance. We say this because, as noted earlier in this book, when Total Float is associated with a Series of Activities along an Activity Path Segment, each separate mention of the Total Float refers to the same Total Float amount. In the above example, there are not actually 85 (or 44) distinct days of Total Float. Instead, those numbers represent the collective opportunities to postpone work without negatively impacting downstream Deadline Milestones. That is why we used the word "available."
earlier, discussion of Software Settings is included in this book to prepare a student for their likely encounter of them. But it is also included because, through the course of struggling to appreciate how certain Software Settings might affect the calculation of various CPM values, the student will surely gain a far deeper, more intuitive understanding of the raw mechanics of the Critical Path Method of Modeling.

In the case of Zero Total Float, this setting causes the Latest Dates for an Activity to be the same as the Earliest Dates. Since Total Float is the difference between Earliest Dates and Latest Dates, the software could have ‘gone either way.’ That is, it could have forced the Earliest Dates to match the Latest Dates – or, as P3 has in fact chosen to do, it could force the Latest Dates to match the Earliest Dates.

Of the two options, we think that the route P3 chose is better for the sake of the Project. Had they forced the Earliest Dates to match the Latest Dates, doing so would have caused Activities to start later than they were actually able to start. This would have defied the very meaning of the term Earliest Dates.

Note that the previously-discussed Contiguous Duration setting does just that: it causes a delay in the Earliest Dates of an Activity.

Figure F0966 shows two Activity Paths leading from the start of Activity A to the end of Activity H.

- The Upper Path travels through Activity A ➤ Activity B ➤ Activity C ➤ Activity D ➤ Activity H, and it reports Total Float of TF +27.
- The Lower Path travels through Activity A ➤ Activity E ➤ Activity F ➤ Activity G ➤ Activity H and reports Total Float of TF +13. Because Activity with the Least Total Float.
What can we say about this entire Logic Diagram? Well, we can certainly say that the entire lot of Activities, all eight of them, have a healthy amount of “extra Time” to get their respective Scopes of Work accomplished. In fact, by using the lesser Total Float value, we can conclude that the entire Logic Diagram could be slipped by two weeks (ten Workdays) and still have positive Total Float for all eight Activities, right? We would also conclude that the Critical Path for this Logic travels through the LOWER PATH.

Now take a look at Figure F0968. Our attention is drawn to Activity C, which has been subjected to a Zero Total Float Software Setting. True to its mechanism, this setting has changed the Latest Dates to match its Earliest Dates. This, of course, has changed the Total Float for Activity C to TF +0.

But notice what else it has done. Following the normal Backward Pass Process, the Latest Finish of Activity B – now changed to Day 13 from the previous Day 40 – forces the Latest Finish of Activity A to change as well. In fact, the Backward Pass Process forces a change in all Restricting Activities to Activity C, all the way back to the beginning of the Schedule.

Now where is this Logic Diagram’s Critical Path? It has moved from the LOWER PATH to the UPPER PATH. Plus, and surely more significantly, the previous Critical Path has significant positive Total Float, TF +13. Now the UPPER PATH forms the Critical Path, and it is has concerning Total Float of TF +0.

Here’s the question? Before the Zero Total Float was applied, this Schedule had a healthy amount of Time. Has the Zero Total Float actually changed the reality on the Project, that the eight-Activity Schedule has an abundance of Time to “complete in a timely manner?”

The answer comes down to what we consider a true Project Execution Commitment. In Figure F0966, we were given a Finish Date Constraint of Day 62 for Activity H. While we
were not told whether this reflects a contractual deadline, or some less rigid deadline, we accept that these eight Activities need to be finished by Day 62.

Has that need changed with the imposition of a Zero Total Float Software Setting? Does a Zero Total Float constitute an expression of a Project Execution Commitment? You might be thinking, “Well, it could....” Our retort is that Schedules that are meant to model reality must be blatant in their description of that reality. If there is a reason to disallow Activity C from finishing any later than Day 21, then we should impose a Finish-No-Later-Than Date Constraint on Activity C, so that the reasoning behind the requirement is obvious. Then we would know, emphatically, that finishing Activity C earlier than previously mandated is now a new requirement.

Instead, the Zero Total Float Software Setting merely expresses a management desire to put emphasis on the finish of Activity C, as well as all Activities leading up to Activity C, so that all three of them can complete as soon as possible. By using the Zero Total Float Software Setting, we are saying “your ‘as-soon-as-possible’ (aka Earliest Dates) have just become your ‘no-later-than’ (aka Latest Dates).”

Now, this may well be what the Project Executors have told you (the Project Facilitator) was their intention. And the Zero Total Float Software Setting would be a convenient way to represent that intention. We are not saying that Zero Total Float use is necessarily a bad practice. We only want to you to understand (a) how it works, (b) what ramifications there may be for its use, and (c) that the rationale behind its use is not readily apparent. Cognitive Project Management strongly discourages it use.

9B3c: Zero Free Float

The other Software Setting we want to discuss is called Zero Free Float. The point of Zero Free Float is to allow Activities to start later than their normally-calculated Earliest Dates. The mentality behind use of this Software Setting is fairly similar to what instigates a Contiguous Duration setting: a desire to keep a set of Project Actions continuous – ostensibly for greater workforce efficiency.

We are not saying that such a bonding of Actions is never appropriate. One could easily envision a situation where a meeting needs to be held immediately before a certain piece of work is performed. By setting the Activity to Zero Free Float, its Earliest Dates will be “shifted to the right” (so to speak), such that the Earliest Finish for the impacted Activity will be immediately before the Earliest Start of the successor (Restricted) Activity.

Look at Figure F0970, which has two panels. In the Left Panel we see three Activities as they are configured before imposing the Zero Free Float Software Setting. Activity A is shown with Free Float of FF +8, because its Earliest Finish (05JAN) is eight Calendar Days
sooner than the Earliest Start (14JAN) of its Restricted Activity, ACTIVITY C.\footnote{In this example, we are using Point-of-Day Perspective.} We also note that ACTIVITY A enjoys Total Float of $TF +10$, compared to the other two Activities which bear Total Float of $TF +2$.

In the RIGHT PANEL, we activate the Zero Free Float software option, and immediately details begin to change for ACTIVITY A. As promised, its Earliest Dates adjust such that all Free Float is stripped away. The Earliest Finish of ACTIVITY A is set to the day before the Earliest Start of ACTIVITY C, its successor. The Earliest Start for ACTIVITY A is adjusted accordingly.

With these changes, not only is the Free Float zeroed out for ACTIVITY A, its Total Float has also been reduced by eight days. Where ACTIVITY A had enjoyed ten days of Total Float, it now has only two days of leeway.

As with Zero Total Float, we are not saying that Zero Free Float is an inherently bad practice. There may be Times when its use is appropriate. But just remember that both of these setting “give away” valuable Float, with just the flip of a switch (Software Setting).

As a matter of management strategy, Cognitive Project Management recommends that these settings not be used; but instead let the Project Executors impose certain requirements onto real-Time Project Time Management tools, employed during Project Execution.

An example might be a THREE WEEK LOOK-AHEAD report developed for use in the WEEKLY COORDINATION MEETING, attended by the Contractor and Subcontractors. With the corresponding Project Actions only days away, this is a great time to talk about delaying certain work for strategic or tactical reasons. Contrast this with performing the same operation during development of the initial Baseline Schedule so many months or
years before the corresponding work – which merely creates Earliest Dates that are not, in truth, reflective of the 

earliest that work can be done based on prior Logic, Durations, Date Constraints, and such.

For a CPM Schedule to be a reliable Project Execution Model, it must provide the truest reflection of options. By overriding Logic with Software Settings – such as Zero Free Float, Zero Total Float, or Contiguous Durations – the Project Execution Model's ability to expand and contract freely is diminished. Date Constraints should be used with moderation out of equal concern.

9C: Elements that Facilitate Schedule Data Management

In this subsection, we want to consider Schedule Elements that can be added to a Project Schedule which make it easier to work with a database that can often include thousands of Activities. But it isn’t just because Schedules can get large that we may want ways to slice and dice the contents of a Schedule.

There is actually a much more fundamental and practical need for the ability to select, group, and sort specific subsets of Schedule Data: end user comprehension and application. For when all is said and done, we want the Project Executors to use the Project Schedule that the Project Facilitators dutifully develop and maintain!

And doesn’t that get to the very heart of what Project Time Management is all about? We hope that by now you fully understand that Project Time Management is a two-sided coin. On one side are those who develop and maintain Project Time Management products and services; ICS-Global calls them the Project Facilitators. On the other side of the coin are those who actually use the Project Schedule as the primary tool of overall Project Management; we call them the Project Executors.

What has been learned from decades of experience applying the Critical Path Method of Managing is that Project Executors are quite picky about the information that they either need or want (not always the same). In terms of quantity, they don’t want too much information … nor do they want too little. In terms of timing, they want it when they need it, and they don’t want to have to fish around to find it.

The obvious solution is to provide Project Executors with tailored information. This brings us back to the Schedule Reporting Mantra, which we introduced many chapters ago:

Schedule Reporting Mantra: The Project Schedule should provide each Project Team member with all the information needed to do his or her job — no more, no less. And the information should always be in a form that is relevant, meaningful, timely, and accurate.
Supplemental Schedule Elements

9C1: Understanding Activity Codes

So the question then becomes: How can you isolate specific pieces of Schedule information, in order to limit the amount of data being transmitted to a Schedule’s end user? The answer is a Supplemental Schedule Element known as Activity Codes. As the word suggests, the Activity is being coded to indicate one or more attributes about the Activity.

Let’s consider a few such attributes that are commonly used to manipulate Schedule data.

- **Location-Indicating Activity Code:** We might tag code Activities as to where the work of the Activity is to take place. On a Project with multiple buildings, for instance, we might create an Activity Code called **LOCATION** and then, for each Activity in the Schedule, assign a value to that Activity Code, such as Bldg A, Bldg B, Bldg C.

- **Responsibility-Indicating Activity Code:** We might want the ability to isolate certain Activities that are the responsibility of one particular Project performer. To do so, we would create an Activity Code called **RESPONSIBILITY** and then code each Activity in the Schedule with an appropriate value, such as Owner, Contractor, Architect, Engineer, etc.

- **Timing-Indicating Activity Code:** Maybe the Project has been divided into major phases, either pursuant to contract requirements, or by consensus of the Project Team. Either way, we might desire to select just those Activities for a particular Project Phase. We would create an Activity Code called **PHASE** and apply appropriate code values to the Activities, such as Phase 1, Phase 2, or Phase 3.

- **Work Type-Indicating Activity Code:** It is quite useful to be able to produce a Schedule Listing that focuses on a particular Project Work Category. For example, a procurement officer might appreciate an itemization of all outstanding materials and equipment for which the Project is waiting deliver. A Tabular Listing entitled **FABRICATION/DELIVERY** Activities, perhaps sorted by Total Float (thus, in order of priority with respect to the Timely achievement of downstream Project Execution Commitments) could be quite useful.

   To accomplish this, we would start by creating an Activity Code called **WORK TYPE** to which we would assign an appropriate code value for all Activities in the Schedule. For instance, code values might include: Submit; Review/Approve; Fab/Del; Install; or Inspect. Next, we would tailor a report that selects only Activities whose Work Category value equals Fab/Del.

Do you see how Activity Codes work and why they are so popular with seasoned Project Facilitators? We would go as far as to say that a Project Facilitator who does not use Activity Codes is, in our humble opinion, a Scheduling novice!
Activity Codes are essential to the effective use of Project Schedules. Since this is a book on the Critical Path Method of Modeling – and not the Critical Path Method of Managing – we really should not be getting into the design of Scheduling Reports. In another ICS-Compendium volume we discuss Scheduling Reports at great length. We have merely touched on reporting options, here, in order to explain one of the ways that Activity Codes can be used.

But make no mistake: there are other uses for Activity Codes, as well. And they build upon the compounding effect of three data manipulation functions: Data Selecting, Data Grouping, and Data Sorting. Let’s take each one separately. From the ICS-Dictionary:

- **Data Selecting**: Data Selecting involves identifying (and thereby selecting) Activities that share one or more common attributes. We might ask the software to identify all Activities whose Earliest Start date falls during a given month and year. We might ask for Activities that have a Duration greater than a certain value. Data Selecting picks certain records out of a larger database of records. Some software programs refer to the ability to select records as Data Filtering.

- **Data Sorting**: Data Sorting gives us the ability to arrange records in a certain order. We might want to list Activities in chronological order, perhaps as they are expected to be worked. In such a case, we would sort according to Earliest Start date.

- **Data Grouping**: Data Grouping is actually an extension of Data Selecting, for it also identifies a cluster of Activities that meet one or more selection criteria. The difference between Data Selecting and Data Grouping has to do with what becomes of the items that do not satisfy the criteria. In the case of Data Selecting, the non-qualifying records are dropped (dismissed, ignored, or excluded) from the resultant report. In the case of Data Grouping, all records are included (none are dropped), but the records are clustered according to the criteria value.

Let’s explain how these three data manipulation functions can work together, through an example. Think about a typical city phone book (and, yes, we realize that they are becoming an extinct commodity with so many Internet-based alternatives). We start with all of the phone numbers in the country.

- **Data Manipulation Step 1, Data Selecting**: Select just those phone numbers that exist within the geographic range of the phone book, such as the city’s “greater metropolitan area.” Through Data Selecting, all phone numbers that are not located in the designated area would be excluded from the master list of subject phone numbers.

- **Data Manipulation Step 2, Data Grouping**: Group all selected phone numbers by whether they belong to a business or a residence. We would use Data Grouping to accomplish
this. Notice that Data Grouping does not exclude any records; it just bundles them together.

- **Data Manipulation Step 3, Data Sorting:** Arrange the selected and grouped records in some particular order. An alphabetical order is commonly useful, and so we would apply Data Sorting to accomplish this, by sorting the phone Listings by last name (for residential Listings) or by company name (for business Listings).

From this simple example we see how the three data manipulation functions can work together to generate a very useful body of information. The same functions can be applied to the thousands of Activities in a typical Construction Schedule. The key, however, is that the Activity Records must be sufficiently coded to allow the kind of selecting, grouping, and sorting techniques we would want to perform.

Two important points to drive home, especially as we approach our discussion of a Schedule’s Work Breakdown Structure, are that:

- A Project Schedule can have, and usually does have, more than one defined Activity Code; and,
- Any single Activity within a Project Schedule can be, and usually is, assigned more than one Activity Code.

Think about what we have just said. A single Activity might be coded as to location, work type, responsible party, phase, Activity Type (Summary Activity, Hammock Activity, Milestone Activity, or Action Activity), and so forth. With multiple Activity Codes assigned to all of the Activities in a Schedule, the possible ways that a Schedule Listing can be filtered, grouped, and sorted is virtually unlimited.

### 9C2: Understanding a Work Breakdown Structure (WBS)

The previous paragraph provides the perfect segue into a short discussion of one of Dominant Project Management’s most treasured inventions: the Work Breakdown Structure, also known as WBS. So much has been written about WBS that one can hardly pick up a book on either Project Management or Project Time Management without being assaulted by this term, and all of the praise heaped upon it.

So what is a Work Breakdown Structure? In a nutshell, it is a unique method of Activity Coding used to subdivide and document a Project's Scope of Work. But that doesn't tell the whole story.

---

4 We are assuming that each phone number record has been coded as to whether it is a commercial or residential number.
Heavily Structured: In practice, a WBS tends to be heavily structured along hierarchical lines.

Highly Regulated: Countless documents contain hard-and-fast rules on how to develop a proper Work Breakdown Structure. White papers and “Best Practices” have been crafted so as to standardize how a WBS is created and maintained.

Required "Scheduling Best Practice": Some Dominant Project Management authorities insist that one simply cannot develop an acceptable Project Schedule if a WBS is not utilized.

To be perfectly candid, Cognitive Project Management dislikes the Work Breakdown Structure approach to Activity Coding. Our reasons for discouraging its use are briefly discussed in Chapter Ten, where it is included as one of three Exploitive Schedule Elements. But our objections are much more thoroughly discussed elsewhere in the ICS-Compendium. In this chapter, we will simply explain how it works, but not whether one should use a WBS.

9C2a: What a WBS Is

A Work Breakdown Structure is an outline of the entire Work Scope of a Project.

A WBS Provides a Detailed Breakdown: It is developed by systematically decomposing the Scope in terms of the components of the Project's ultimate product. Thus, a Project is comprised of a primary Component, and each Component is divided into Sub-Components and each Sub-Component is further divided into Sub-Sub-Components, and so forth.

A WBS is Organized Hierarchically: WBS utilizes a highly-regulated outline structure and corresponding numbering system. For instance, 1st Level Deliverables may be given a number, such as 1.0. A 2nd Level Deliverable might be given a number, such as 1.3. A 3rd Level Deliverable might be numbered, 1.3.a. And so forth.

A WBS is Product Oriented: Dominant Project Management literature overwhelmingly mandates that WBS elements should align all Activities in the Schedule with Project and Product deliverables. Unfortunately Activities in a Construction Schedule do not always culminate in a neatly-wrapped Deliverable. Quite often, Schedule Activities align with multiple Deliverables; sometimes they do not align with any specific Deliverable.

Only One WBS Code per Schedule Activity: According to many Dominant Project Management rules, a Schedule Activity can only be assigned to one WBS.
code. This rule poses a problem in Construction Project Schedules, because the Activities in a CPM Schedule may be oriented by Location, Phase, System, or Deliverable. It is very unlikely that all of the Activities in a Schedule will align with just one orientation. No matter which of the four orientations one chooses as the basis for the WBS, WBS coding will be awkward if not impossible for Activities aligned with the other three orientations.

9C2b: **How a WBS is Used**

Dominant Project Management literature typically offers two reasons why development of a Work Breakdown Structure is important (and frequently mandated): to support Schedule Development and to support other Project Controls, most notably Cost Management.

- **In Support of Schedule Development:** We can all agree that, in order to be effective and reliable, a Project Schedule must include the entire Project Scope to be modeled and managed by that Schedule. Proponents of the Work Breakdown Structure method insist that a well-developed WBS serves to insure that all Work Scope is accounted for. Cognitive Project Management certainly agrees that, used as a final checklist, a WBS is an excellent way to double-check to make sure that all Scope items are accounted for. But we note that such use does not require a formal WBS numbering system. We also note that there are various other ways to achieve the same assurance. Some even more effective than a WBS.

- **In Support of Other Project Controls:** The other use of a WBS is to force configuration of the Project Schedule Database so that it dovetails with the databases of other Project Control Systems. It is with this use that Cognitive Project Management has most of its heartburn concerning the requirement that a WBS system be used in Schedule Development. Our objections are briefly explained in Chapter Ten, more fully discussed in ICS-WHITE PAPER WPB-LA-11, **How the WBS Can Break a Schedule**, and completely covered in elsewhere in the ICS-Compendium.

Let us not forget that the ultimate goal of Project Time Management is to assist the Project Team in addressing its three primary Project Execution concerns: Coordination, Collaboration, and Communication.

A WBS is not essential to a Project Schedule’s development or use. And, as discussed earlier, creation of a comprehensive WBS that works across a Project having differing contextual schema within it is often very difficult to accomplish. Moreover, there is nothing that a WBS provides to data management that a well-conceived set of Activity Codes cannot accomplish just as well, if not better!
This discussion of the Work Breakdown Structure concept was kept particularly short because we would like you to set the book down – again – while you locate and read ICS-WHITE PAPER, WPB-LA-11, HOW THE WBS CAN BREAK A SCHEDULE. Only after you to read this White Paper should you move on to CHAPTER TEN.

9C3: Activity-Resident Notations and Logs

Beyond Activity Codes and WBS, Activity-resident Notations and Logs are another way that the tremendous amount of vital Project Execution information can be managed in a Project Schedule.

A Schedule is a database of Activities, each one of which correlates to one or more corresponding Actions out on the Project. As is repeatedly noted throughout Cognitive Project Management literature, the three prior concerns of the Project Team are: Coordination, Collaboration, and Communication.

It can be strongly argued that one cannot do either of the first two, without well doing the third. Indeed, Communication is the single most important skill of every Project Executor. For its part, Communication involves the transfer of knowledge. What is known, however, includes more than just cold, raw data. It also includes feelings, understandings, attitudes, intentions, expectations, requirements, and on and on.

Since the Project Schedule contains a complete list of all that must be accomplished in order for the Project to realize successful completion, it stands to reason that among the Activities in the Schedule one should always be able to find an appropriate place to record important Project Execution information.

In Project Time Management circles, such incorporated comments are known as Schedule Notations or Activity Logs. Either term refers to the same process; a written entry into the body of the Schedule Database of some random piece of information of special importance.

As you can see, a Schedule needn't contain any Schedule Notations or Activity Logs. In practice, the vast majority of Schedules do not. But you should be aware that Software programs provide functionality that allows sundry notes and logs to be captured, and that these entries can be related directly to individual Schedule Activities.
CHAPTER TEN

10A  Work Breakdown Structure (WBS)
   10A1  What a WBS Is
   10A2  How a WBS is Used

10B  Activity-Resident Project Cost Information
   10B1  What Cost Management Is
   10B2  How Cost Management Works
      10B2a  How Project Pricing Works
      10B2b  How Budget Development Works
      10B2c  How Cost Accounting Works
      10B2d  How Cost Control Works
   10B3  How Cost Management Affects Project Time Management
      10B3a  It Can Force Activities to Be Added
      10B3b  It Can Force Activities to Be Subdivided
      10B3c  It Can Mandate the Use of WBS
      10B3d  It is Often Used to Opine on Project Temporal Outcomes

10C  Activity-Resident Project Resource Information
   10C1  What Resource Management Is
   10C2  How Resource Management is Applied to Project Schedule
   10C3  How Resource Management Techniques Affect Project Model
      10C3a  Resource-Leveling is Unpopular in Construction Scheduling
      10C3b  Activity Resource-Loading Usually Requires Arm Twist
This chapter offers a short discussion of three specific Project Schedule Elements that are typically exploitive in how they interact with the CPM Schedule as a model of Project Execution Strategy. To be sure, the word *exploitation* need not connote something damaging, predatory, or negative — although, admittedly, that is how the word is most often construed.

Dictionary.com says this about the word, *exploit*:

- **Exploit**: To utilize, especially for profit; turn to practical account
- **Exploit**: To make the best use of
- **Exploit**: To advance or further through exploitation; promote
- **Exploit**: To use selfishly for one's own ends
- **Exploit**: To take advantage of (a person, situation, etc), esp unethically or unjustly for one's own ends

The first three definitions suggest that exploitation need not be sinister. One could simply be maximizing the value extracted from a process or product (in our situation, from the Project Schedule); as the second definition says, “to make the best use of.”

But the more common understanding of the term is reflected in the bottom two definitions, which suggest that the use can be self-centered, with little regard for how such use may negatively impact others.

This chapter discusses the use of a Work Breakdown Structure, as well as the Cost-Loading and Resource-Loading of CPM Schedules. The latter two processes are really quite similar in concept and approach. Each of these expansions of the Critical Path model have the potential to erode its integrity and *temporal* functionality, if not carefully controlled.

- **Cost-Loading**: In this process, a cost is estimated for the performance of the Work of an Activity and that monetary value is then assigned to each Activity within the CPM Schedule database. So attached, the dollar amount is available to be used
in conjunction with various non-temporal Cost Management processes, to derive valuable information about the achievement of fiscal objectives on the Project.

Resource-Loading: Likewise, estimates of the quantity and type of resources needed to perform the Work of an Activity are assigned to each Activity within the CPM Schedule database. So attached, these resource estimates (which may include estimates of the quantities of needed labor, equipment, materials or other non-financial resources) are valuable to various non-temporal Project Management processes for planning and implementation purposes. Included might be Human Resources Management, Procurement Management, Risk Management, Contract Management, and so forth.

An important distinction must be made between inclusion within the Project Schedule of the Schedule Elements themselves, and the practical application of those included Schedule Elements by the referenced non-temporal Project Management interests. To be fair, the mere inclusion of the Schedule Elements is usually quite innocuous. Rather, it is in the processes that underly and utilize these two Schedule Elements that the potential for harmful exploitation may occur.

This chapter discusses three categories of Exploitive Schedule Elements, commonly found in the Project Schedules of Dominant Project Management: Work Breakdown Structure, Cost-Loading, and Resource-Loading — the latter admittedly much less problematic than the former two.

It is worth noting that the ICS-Compendium Development Team engaged in a lively discussion about whether to include this chapter in a book on CPM Mechanics. After all, the subject of potential Schedule exploitation falls more directly under Schedule Usage, the topic of another ICS-Compendium volume. In the end, it was felt that to not discuss Cost-Loading or Resource-Loading at all would be remiss, since both are frequently found in construction Project Schedules, and to discuss either without making brief mention of their potentially-harmful potency would be a great omission.

As we said above, in both cases these managerial interests draw important information from the Schedule. They are not, in and of themselves, essential to the performance or integrity of the Project Schedule as a Project Time Management tool. This is not to say, however, that the Project Schedule may not benefit by their inclusion; only that their existence in the Schedule is not essential. That said, they can be exploitive in that they are introduced into the Schedule strictly for what they can take out of the Schedule.
The ICS-Dictionary provides this definition of Exploitive Schedule Elements:

**Exploitive Schedule Elements:** The informational content of Project Schedules can be divided into Essential, Supplemental, or Exploitive Schedule Elements. The ultimate objective of Exploitive Schedule Elements is to extract temporal information from the Schedule for non-temporal Project Management purposes. If not carefully controlled, the incorporation of Exploitive Schedule Elements can weaken, and even destroy, a Schedule’s ability to function as a reliable Project Time Management Tool. Under this heading would be Activity Cost Data, Activity Resource Data, and Work Breakdown Structure.

**10A: Work Breakdown Structure (WBS)**

We will begin our discussion by examining the concept of a Work Breakdown Structure. Much of the reasoning for why the ICS-Compendium Development Team regards a Work Breakdown Structure as an Exploitive Schedule Element can be found in ICS-WHITE PAPER, WPB-LA-11, HOW THE WBS CAN BREAK A SCHEDULE. If you have not done so already, we ask you to put the book down and read that paper NOW.

While our assumption, at this point, is that you have read the referenced White Paper, the following abbreviated discussion will summarize the substance of that White Paper, so that the book remains a coherent, self-contained work product. Accordingly, if you find this subsection too cryptic, please revisit the White Paper for a better understanding of why a Work Breakdown Structure can be regarded as an Exploitive Schedule Element.

Much like most every other term that comprises the Project Time Management lexicon, Dominant Project Management literature is far from in agreement as to what a Work Breakdown Structure is or how it should be used. We examined a number of diverse opinions from which we were able to identify the most salient attributes of a WBS, which can be subdivided as follows:

- **What a WBS Is**
  - Decomposition of Project Work Scope
  - Detailed Breakdown
  - Organized Hierarchically (“family tree” or outline fashion):
  - Product-Oriented

- **How a WBS Is Used**
  - To Create Project Schedule
  - To Support Other Project Control Systems
10A1: What a WBS Is

A Work Breakdown Structure is an outline of the entire Work Scope of a Project. It is developed by systematically decomposing the Scope in terms of the components of the final Project's main product, each such component viewed as a Deliverable.

A WBS Provides a Detailed Breakdown: From the referenced White Paper you should have a fairly sound understanding of how a WBS works and how it is created by further and further decomposing the initial Deliverable into smaller and smaller components of that Deliverable. The ultimate goal is to create an unbroken link between the whole and its most minute parts.

A WBS is Organized Hierarchically: You probably made your own observation that the numbering system can become quite unwieldy as you get further and further into the minutia of the WBS. But WBS numbering is not the main problem with the hierarchical approach of WBS.

Only One WBS Code per Schedule Activity: The biggest problem is that an Activity can only be assigned to one WBS code. In the typical Construction Project Schedule, the orientation of Activities can range from physical space, to phase, to system, to ultimate deliverable. It is very unlikely that all of the Activities in a Schedule will align with just one orientation schema. Whatever schema or context you use, it must be applied consistently throughout the entire WBS. So if a WBS is structured to reflect one schema, it at once becomes incompatible with Activities oriented to any other orientation schemas.

A WBS is Product-Oriented: The fourth observation we make is that a WBS is Product-Oriented. In certain industries Projects are definitely Product-Oriented; one might even say, extremely so. SOFTWARE DEVELOPMENT is one such industry. Here again we encounter a problem with respect to Construction Projects, where not all Activities in the Project Schedule necessarily lead to Deliverables — without, that is, greatly stretching the meaning of the term, “Deliverable.”

10A2: How a WBS is Used

The message to be taken from the previous discussion is that development of a WBS Structure that is compatible with the content and alignment of Activities in a Construction Schedule is often a quite daunting challenge. Unfortunately, way too frequently not enough thought is given to WBS creation — and the negative effects are not felt until well into the Project. By that point, changes to the underlying WBS could be extremely costly, time-consuming, disruptive, and even impractical.

If there is any hope for the Work Breakdown Structure to become a reasonably useful tool, Cognitive Project Management recommends that the WBS architecture be finalized
10 Exploitive Schedule Elements

by the Project Team during Schedule Development, pursuant to specifications evolved during the Schedule Design Summit.\footnote{According to Cognitive Project Management, Schedule Design is a set of processes that should precede Schedule Development. With few exceptions, Dominant Project Management fails to mention, let alone require, Schedule Design; instead, it declares Schedule Development as the first step in Project Time Management.}

This recommendation stands in contrast to current Dominant Project Management best practices, which have the WBS created (a) prior to Schedule Development — and not by Project Time Management personnel, but instead by Cost Management personnel.

With all of the technical challenges to creating one WBS coding schema that will work for all of the different types of Activities in a Schedule and different types of Deliverables on the Project, one has to wonder whether it is even worth the struggle. The answer depends on what benefits can be derived from an effective Work Breakdown Structure.

To know whether a WBS is effective, we must first understand what effect we are hoping for! And as we already noted, there are two main uses for a Work Breakdown Structure: to help insure that all Work Scope is captured in the Project Schedule, and to support various non-temporal Project Management processes.

It is our opinion that a WBS is mainly used to support other non-temporal Project Management processes and only secondarily to support Project Time Management objectives. In support of this assertion, consider these points:

- Can a competent Project Schedule be developed and maintained without inclusion of a formal WBS? Answer: Yes.

- While we agree that a WBS can be used as a quality control technique, we wholeheartedly disagree that it is the only way to make certain that the Project Scope is fully addressed in the Schedule. Numerous other methods are available to the Project Facilitator that are every bit as effective.

- Alternatively, can many of the other non-temporal Project Management systems be effectively accomplished without use of the Project Schedule? Answer: No.

Cognitive Project Management reminds us that the ultimate goal of Project Time Management is to assist the Project Team in addressing its four primary Project Execution concerns: Coordination, Collaboration, Cooperation, and Communication.

A WBS is not essential to a Project Schedule’s development or use. And, as discussed earlier, creation of a comprehensive WBS that works across a Project having differing contextual schema within it is often very difficult to accomplish. Moreover, there is
nothing that a WBS provides to data management that a well-conceived set of Activity Codes cannot accomplish just as well … if not better! One last time, for a full discussion of Work Breakdown Structure, please read ICS-White Paper, WPB-LA-11, *How the Work Breakdown Structure Can Break a Schedule.*

**10B: Activity-Resident Project Cost Information**

Who would argue that the top two areas of Construction Project Management concern are almost always *Schedule and Budget*?

**10B1: What Cost Management Is**

An Owner of a Construction Project retains the services of a competent construction Contractor to deliver the Project for a pre-established price and by a pre-established date. From the Contractor’s perspective this means that there is almost always at least one *Deadline Milestone* and a surely *Bottom Line*. To guard against missing the Deadline Milestone, the Contractor creates a Project Schedule. To protect the Bottom Line, the Contractor creates a Project Budget.

Cost Management is that aspect of Project Management concerned with delivering the Project within the pre-determined Project Budget. Cost Management can be divided into the following four primary processes. They may go by different names in different regions of the world, industries, or organizations, but the practical essence of each remains the same:

- **Project Pricing**: Determine what the Project will cost
- **Budget Development**: Based on Pricing, create an operating budget
- **Cost Accounting**: During operations (Project Performance), monitor costs
- **Cost Control**: Where budget overruns are spotted, take corrective action

**10B2: How Cost Management Works**

Let’s briefly consider how each of these four steps works.

**10B2a: How Project Pricing Works**

The first step of Cost Management is to determine how much the Project will cost to complete. The answer to this question of course depends on who is doing the asking. The Owner’s Project “cost” boils down to the quotes of the various Contractors, and of the various suppliers of Owner-furnished items, plus other things like the cost of money (financing costs), and administrative costs (e.g., taxes, permits, general operational overheads), and so forth.
The Contractor’s quote to the Owner includes the quotes of its Subcontractors and suppliers of Contractor-furnished items, plus its own direct and indirect costs. The Contractor’s quote to the Owner also includes a reasonable mark-up for profit. The Subcontractors and the suppliers also have their price which includes their costs, plus profit mark-up.

10B2b:  How Budget Development Works

Once a Contractor knows how much each individual item is expected to cost, he simply compiles what is called a Schedule of Values (SOV), which is a table that breaks the Project Scope into separate, measurable line items. When Not-to-Exceed dollar values are assigned to each SOV line item, what results is called a Project Budget.

As a matter of practicality and convenience, however, long ago it was noticed by Cost Management community that the Project Schedule already “breaks the Project Scope into separate, measurable line items.” As the thinking quickly went, why not simply assign a monetary value to each Activity in the Schedule, such that:

- Every Schedule Activity has been assigned a corresponding “cost;” and,
- The sum of all Activity costs equals the Project cost.

10B2c:  How Cost Accounting Works

With each Schedule Activity having been assigned an estimated cost, the Project Schedule becomes the ideal mechanism for keeping a constant eye on whether the Project Budget is in jeopardy of being overrun. This Ideology of course conveniently coincides with Sir Isaac Newton’s notion that to understand the whole, one need merely understand the parts.

Think about it:

- If a Project Budget is divided into, say, 100 Budget Line Items, and
- If the total of all Budget Line Items equals the total Project cost, and
- If we make certain not to exceed the monetary value of any of these 100 Budget Line Items,
- Then it would be impossible to exceed the Project Budget.

The secret to using the Project Schedule as an application of the Project Budget is in establishing a direct correlation between the Budget's Line Items and Schedule's Activities. As long as such a correlation can be worked out, the Schedule’s normal maintenance processes will yield a current evaluation of budgetary adherence. Here’s how it works:
As an Action is worked in the field, Project Executors assess the progress of that Work which they express as a percent of completion of the corresponding Activity in the Project Schedule.

For example, suppose the Action involves the first step in constructing the foundation for a new building. Let's say that the construction drawings call for a continuous footing to run beneath where the building’s outer four walls will ultimately stand. The field Action involves excavating a trench, along four sides, where the concrete footing is to be placed into the ground. Let us assume that the building footprint is completely square and that the required time to perform this excavation work is estimated at one Workday, per side. We might therefore create a Schedule Activity, entitled **Excavate Bldg A Continuous Footing**, and assign to it an Activity Duration of four days.

Suppose further that, at the end of the month, the Schedule is updated to reflect Work progress to date. Just as they would do with every other Activity in the Schedule, the Project Executors would be asked about this Activity: *How far along is Excavate Bldg A Continuous Footing?* If, for discussion's sake, only one side of the building had been excavated, we would assess this Activity at 25% complete.

Now, if this Activity had been Cost-Loaded with a monetary value of, say, $3,200, then a simple arithmetic exercise would conclude that this Activity had “earned” $800 ($3,200 x 25%). Repeating this same process on every Activity in the Schedule, would result in an estimate of the portion of the overall Project Budget that had been “earned” to date, right?

**10B2d: How Cost Control Works**

But another important value would be derived from this arithmetic process as well: a statistical reading of the health of each individual Budget Line Item. By adding up the “earned” portion of all Activities assigned to a single Budget Line Item, one could then compare the earned amount to the actual amount and quickly spot any Budget Line Item overrunning its individual budget. Let’s pursue this concept, using Figure F1002.

Suppose there are ten excavation Activities in the Project Schedule, with **Excavate Bldg A Continuous Footing** (the one we were just working with) being just one them. Suppose that the Project Budget has a single Budget Line Item called **Excavation Activities**, and it has a maximum Project Budget limit of $30,000.

Now if we were to estimate the Percent Complete of all ten Activities, as shown in Figure F1002, we would determine that this entire Budget Line Item has “earned” $22,285
to date. Suppose that we check our records and find that, as of today, we have collectively spent $23,760 on excavation Activities. What this tells us is that the actual cost incurred in order to perform the Work thus far accomplished is greater than what had been budgeted for performance of that Work. This awareness gives us the opportunity to focus on the remaining excavation Activities and see if we could find ways to reign in any remaining costs. This, at least in principle, is what is meant by Cost Control.

### 10B3: How Cost Management Affects Project Time Management

Since this book is limited to the mechanics of the Critical Path Method of Modeling, anything more than a superficial discussion of Cost Management exceeds the scope of this text. But we wanted you at least to understand why we would load cost information onto the Activities in a Project Schedule, and how information might be used as part of Cost Management.

In this final subsection, we want to point out a few of the ways that inclusion of Cost Management information might affect, and even encumber, its functionality of a CPM.
Schedule as a dynamic Project Time Management tool. We said “might,” because it all comes down to the extent to which the Cost Management influence imposes additional requirements on the Project Time Management tool (the Project Schedule).

It is this potential encroachment on the sanctity of the Project Schedule as a temporal tool that brought it into the Exploitive Schedule Element category. In this regard, we can think of a few commonly-imposed requirements that have historically had negative effects on the Project Schedule and its application as a Project Time Management Tool. We will briefly discuss four potential effects here.

### 10B3a: It Can Force Activities to Be Added

You will recall that an underlying premise behind a Project Budget is that the entire Project Cost can be subdivided into a set of Budget Line Items, the monetary total of which equals the Project cost. But what happens if the Project Schedule does not contain sufficient Activities to correlate with all Budget Line Items?

For instance, a common budgetary line item is called General Conditions Costs, which would include expenses like a performance bond, various forms of insurance, permits and fees, and so forth. There might not be any Activity in the Schedule that would be an appropriate holder of these costs. A common solution is to create a new Activity, the sole purpose being to act as a container for these costs.

As for the Activity Duration for such an Activity – well, that poses another problem for a CPM Schedule meant to be a model of Project Execution Strategy. It comes down to how the cost of this General Conditions Activity is to be “earned.” Suppose the Project is expected to last ten months, and so each month the Contractor would be allowed to invoice for 10% of the General Conditions Cost. In order to implement this approach, the Activity would be given a Duration of say 200 Workdays (at 20 Workdays per month). Each month, the Percent Complete would be increased by 10%.

Do you see a problem with this approach from a Project Time Management perspective? Didn’t we agree many chapters ago that Activity Durations should reflect Continuous Crew Days? Aren’t we now violating that rule? Wouldn’t the inclusion of numerous such Activities, ones with differing Activity Duration bases, make it impossible for us to perform any apples-to-apples Schedule Analyses based on Activity Duration values?

The "work around" that Project Facilitators have figured out is to take the extra step to exclude (using Data Filtering) Non-Work Activities from such analyses. But the point of this subheading is not to be missed: inclusion of these "extra" Activities provide no added value for Project Time Management purposes; they only benefit Cost Management recipients. Plus, their inclusion in the Schedule actually makes the Project Facilitator's job more difficult.
10B3b: It Can Force Activities to Be Subdivided

Sometimes the Project Schedule contains single Activities that have multiple performers. For instance, there might be an Activity called, **Install In-Wall Utilities**, with an Activity Duration of two days.

Imagine that the Work Scope involves a handful of walls into which must be inserted a plumbing pipe for a sink, a few runs of electrical conduit for power outlets and light switches, pneumatic tubing for a thermostat, and a narrow air duct for an air supply register at the top of the wall.

Again, assume that the four trades have agreed that they can do their respective work, collectively and without getting into one another’s way, with in a short, two-day period. It would make no practical sense for the Project Scheduler to create four separate Activities, each with a two-day Duration, since that would actually overstate both the amount of Time required and the underlying Resources to perform the work. And so, the Scheduler settles on a single Activity called **Install In-Wall Utilities**.

Along comes a Cost Engineer who complains that the single Activity does not align with the Budget Line Items, which are categorized by trade. In other words, there is a Budget Line Item for electrical conduit and wiring, another for plumbing piping, another for pneumatic tubing, and another for duct work. Each has its own Budget Line Item code. How can this one Activity be properly tagged with four different Budget Line Items codes?

There is only one solution: divide the single Activity into four Activities. Not only does the Scheduler now struggle with what to do about the Activity Durations, which will all be overstated, she also struggles with the Logic Ties. These four Activities are not actually concurrent, since they can’t all be worked at the same Time in the same small area. So now she must do two more things:

- She must sequence the four Activities, one after the other. But which should come first, second, third, and fourth?
- With the four Activities in sequence, the Activity Duration of each must be reduced to the smallest value possible: one day.

Can you see how the informational need for greater detail by the Cost Management team has created a deterioration of the Schedule as a Project Time Management Tool? Also, did you notice that the minimum Activity Duration for each Activity (you can't go lower than one day) now creates a collective Activity Path Duration of **four days**, which is twice what the trades said they needed?

If you are hoping we will tell you what the work-around is for this problem -- bad news! There is none. And that is just one of the reasons why seasoned Project Schedulers often
grit their teeth when told that the Project Schedule is expected to support comprehensive Cost Control processes. All too often, Project Time Management interests pay the price. This is why Cognitive Project Management strongly encourages Project Owners to carefully consider the ramifications before contractually requiring the Schedule to be used to support Cost Management functions.

**10B3c: It Can Mandate the Use of WBS**

As noted earlier, a Work Breakdown Structure is not an essential process in the development of a Project Schedule. To be sure, it is a very good way to confirm that all Project Scope has been accounted for in the Schedule but it is by no means the only way to do so. As was also pointed out, there is nothing that a WBS does that Activity Codes cannot do just as well, if not better.

That said, a WBS is far better than any combination of Activity Codes when it comes to supporting a Project Budget. This is because a WBS is essentially an outline of the Project Scope, configured to correlate with the Schedule of Values. A Project Budget also subdivides the Project along hierarchical lines. It is no wonder, then, that when the decision is taken to use the Project Schedule to support Cost Management objectives, a WBS suddenly becomes a required element of the Schedule.

The use of the Project Schedule to validate Contractor progress payment requests has become virtually commonplace. Most of the time this use of the Schedule is mandated by Contract. Even when it is not, it is entirely expected for the Schedule to be “statused” so that the “earned” monetary value can be computed “to date.” Totaling costs by WBS is easy to do, and if the WBS is already aligned with the Budget Line Items, then the Linkage between measured performance and progress payment and budget control is complete!

The only real objection we have to the inclusion of a WBS is either when it is accompanied by an edict to not use Activity Codes or when it subjugates the content and Logic of the Schedule in order to accommodate a restrictive, awkward, or impossible WBS configuration. For more on this point, see ICS-White Paper, WPB-LA-11, HOW THE WORK BREAKDOWN STRUCTURE CAN BREAK A SCHEDULE.

**10B3d: It is Often Used to Opine on Project Temporal Outcomes**

This last observation has to do with the Earned Value Management System (EVMS) which is discussed at length elsewhere in the ICS-Compendium. Since Earned Value, as a Project Management concept, is possibly unfamiliar to the reader let us describe it just enough to make the point of this topic’s heading.
If you look once again at Figure F1002, you will see two important values: the *amount budgeted*, and the *amount actually spent*. What is not shown is a third important value: the *amount that should have been spent*. We don’t see this value in our spreadsheet because the underlying basis for what should have been spent comes from the Schedule itself.

But if, based on the Earliest Dates in the Project Schedule, we were to calculate the Work that should have been performed and, hence, the amounts that *should have been spent*, the results might look like Figure F1004.

Notice, first, that we have changed some of the column headings from the previous table. We have introduced three EVMS terms:

- **Budgeted Cost of Work Scheduled (BCWS):** This is the cumulative budgeted cost of all Activities or portions of Activities that were *scheduled* per the Project Schedule's Earliest Dates to be performed during the reporting period.
- **Budgeted Cost of Work Performed (BCWP):** This is the cumulative budgeted cost of all Activities or portions of Activities that were actually performed, entirely or in part, during the reporting period. This is also known as the "Earned Value."

- **Actual Cost of Work Performed (ACWP):** This is the cumulative actual cost of all Activities or portions of Activities that were actually performed, entirely or in part, during the reporting period.

Whereas a comparison of BCWP and ACWP, as we did in Figure F1002, correlates what we spent with what we estimated we would spend, the comparison of BCWS and BCWP in Figure F1004 looks at whether the Work we scheduled to perform (BCWS) is more or less than the Work we actually performed (BCWP).

What has happened over the years is that, as Earned Value has become more and more of a mainstay of Cost Management (and, therefore, has had more and more of a voice in Project Schedule development and usage), it has come to be accepted as an accurate and legitimate measurement of Work Performance. In fact, it has actually acquired the nickname, **Performance Measurement.**

Our problem with Earned Value opining on the temporal status of the Project – as to whether it is behind or ahead – is that there are numerous flaws in the premises that underlie Earned Value as a true measure of Work Performance.

- All that it measures, or could ever measure, is the percent of cost that should have been earned versus the amount that was actually earned.

- Missing from Earned Value calculations is any consideration of Schedule Logic, of Total Float, the expenditure of costs for rework, misdirected work or extra work, or of the differing configuration of work crews, or composition of labor, material, and equipment costs from one Activity to the next.

- Earned Value interprets Activities that are not performed by their Earliest Dates as “behind schedule.”

Nevertheless, Earned Value has built such a strong following that its pronouncements as to Project temporal status quite often rival, and even override, the more credible and relevant indicators coming directly from the Critical Path Method of Modeling.

**10C: Activity-Resident Project Resource Information**

In this last subsection we simply want you to be aware that, much like the Cost Management folks, other Project Management entities also find great power and value in the Project Schedule. As just one example, those concerned with Human Resources Management, for instance, how found it very helpful to Resource-Load Activities in the Schedule with manpower requirements in order to develop and maintain more accurate staffing plans.
10C1: What Resource Management Is

In much the same way that Costing information could be summed, so can required head counts. By Activity Resource-Loading the Schedule with the required manpower levels for each Activity, a cumulative graph can be easily generated that profiles the Project's staffing demands beneath a Timeline than spans the Project Length.

Manpower is hardly the only “resource” that can be managed via the Schedule. Equipment requirements can be assigned to each Activity. So can needed materials. Procurement Management very much appreciates reports generated from the Schedule that indicate when various pieces of equipment will be needed. The equipment we are talking about is not what goes into the final product, such as passenger elevators or mechanical air handlers. Rather, we are referring to construction equipment, such as trucks, bulldozers, cranes, and so forth.

10C2: How Resource Management is Applied to the Project Schedule

Any Scheduling Software worth its salt provides functionality for the user to enter Resource information at the Activity level. In fact, the software has gotten so sophisticated that each Resource can be individually assigned to its own Work Performance Calendar. So, for instance, if a particular inspector is only available for six months of the year, a special calendar can be created that marks his availability. Then, any Activity requiring that particular inspector would be “Resource-Loaded” to indicate him as a Resource. The computer, in performing the Forward Pass and Backward Pass, would take into account the non-work periods for this single Resource.

Another popular function provided by most quality Scheduling Software programs is called Resource-Leveling. The idea here is to let the computer postpone the performance of certain Activities (based on available Total Float) in order to level the demand for given Resources. Keep in mind that the default calculation of Earliest Dates is just that – the earliest that they can be performed. So maybe three one-day Activities can all be started on Monday, and they all need to be finished by Friday. The Resource-Leveling function would spread out those three Activities so that they aren’t all performed at once.

10C3: How Resource Management Techniques Can Affect the Project Model

As a matter of practice, Resource-Leveling is not used very much in construction Scheduling. More common, the practice of Activity Resource-Loading still tends to happen mainly when required by Contract.²

² Do not confuse Resource-Leveling with Resource-Loading.
10C3a: Why Resource-Leveling is Unpopular in Construction Scheduling

It has to do with how Construction Projects are executed; through the Actions of a closely coordinated (thanks to the Schedule) band of independent Contractors. The Project Participants are obligated to one another through the terms and conditions of various Contract documents, the Schedule being one such contractual document.

An underlying principle of any Contract is that there is a promise by the provider and a Commitment by the consumer. The provider (Contractor) promises to deliver a product by a certain Calendar Date and for a certain price. The consumer (Owner) commits to (a) providing the working conditions (e.g., site access), (b) guaranteeing no interference by itself or its agents, and (c) assuring commensurate, timely, and periodic compensation for work put in place, and so forth.

Whether the Contractor is especially profitable or efficient in its efforts is just not a large blip on the Owner's radar. This may explain why it is uncommon for a Contract to require Resource-Leveling, the main point of which is to foster greater efficiency in the use of Resources.

10C3b: Why Activity Resource-Loading Usually Requires an Arm Twist

Under the terms of most contracts, the Owner is expected to let the Contractor “do his thing.” Telling the Contractor how many workers to use, how many pieces of equipment to have on site, or how much excess inventory (materials and supplies) to have on hand — all of this could be easily construed as Owner interference. This is not to say that the Owner cannot ask the Contractor, in advance of the Work, to explain its plans for staffing, equipping, or supplying the Project.

Indeed, some of the required pre-Project explanation can be found in the Baseline Schedule. For sure, Project Execution Strategy is conveyed by the Schedule and when it is Resource-Loaded then some of these other questions are answered as well.

So, to some extent, while Owners can ask about a Contractor’s Plans ahead of time, once the Project begins it is left to the Contractor to actually supply those Resources. The Owner may, and quite often does, compare Planned Resource levels with Actual Resource levels. When these two sets of data vary significantly, the Owner may ask the Contractor to explain or reconcile the differences.

The extent that an Owner may monitor Contractor staffing, equipping, and supplying efforts is typically greater than the Contractor’s similar oversight of its Subcontractors. In practice, the further down the food chain you look, the less the superior entity tends
to (or has the ability to) micro-manage the subordinate entity.

What this means, in practice, is that the percent of Contractor-Subcontractor agreements that require detailed pre-Project Planning information is considerably less than the percent of Owner-Contractor agreements requiring the same degree of Planning information. As a consequence, Subcontractors tend not to share any more information that they have to, and the Contractor is hard-pressed to squeeze this information out of them.

Finally, since the overwhelming majority of Schedule Activities are performed by these Subcontractors (the expression in the Construction Industry is that the work has been “brokered”) and only a small percent are self-performed by the General Contractor, acquiring details about Planned manpower, equipment, or material levels from a Subcontractor is quite often about as easy (and effective) as nailing Jello to a wall. And that is why few Construction Project Schedules are Resource-Loaded – unless required by the contract.

So how then, if at all, might an Activity Resource-Loading requirement negatively impact a Project Schedule in its development or usage? To be sure, any competent Schedule should base its Activity Durations on sound and agree-upon Resource assumptions. That said, how would loading Resource information into a Schedule potentially erode the Project Time Management effort?

It should not – unless, as a practical matter of putting that information into the Schedule, it causes the Schedule’s content or Logic to be altered from what best models the Project Execution Strategy. Most of the Time this is not a problem; but every once in a while, those charged with Resource-Loading get carried away and start mandating that certain Activities be split up, or others be further divided.
## CHAPTER ELEVEN

11A **The Project Schedule Must Model Reality**
11A1 Planning, Scheduling ... and Reality
11A2 Reflecting Reality Makes the Project Schedule Really Dynamic
11A3 “Reality” is Performance, Not Just Ingredients

11B **How Time is Understood in a Project Schedule**
11B1 About Schedule Editions and Schedule Iterations
11B2 About Schedule Tense
11B3 Introducing Data Date
11B4 Time Passage and Performance Achievement

11C **Temporal Context of Schedules: Schedule Segments**
11C1 Performance Periods: Their Dimensions and Boundaries
11C1a Current Period: Dimensions and Boundaries
11C1a-i Relationship of Current Period to Project Length
11C1a-ii Introducing Just Done Period and Imminent Period
11C1a-iii Calculation of Current Period Width and Start/End Dates
11C1b Calculation of Past Period Width and Start/End Dates
11C1c Calculation of Future Period Width and Start/End Dates
11C2 Schedule Segment Data: Contextual Perspective
11C2a Current Segment: Contextually Introspective
11C2b Past Segment: Contextually Retrospective
11C2c Future Segment: Contextually Prospective
11C3 Inherent Nature of Different Schedule Segment Data Types
11C3a Past Segment: Realized Data and Gleaned Data
11C3a-i Introducing Realized Data
11C3a-ii Introducing Gleaned Data
11C3b Current Segment: Apparent Data
11C3b-i Introducing Apparent Data
11C3c Future Segment: Strategic Data and Predictive Data
11C3c-i Introducing Strategic Data
11C3c-ii Introducing Predictive Data
11C3d A Helpful Mnemonic: the Acronym, G.R.A.S.P.
11A: The Project Schedule Must Model Reality

The Critical Path Method of Modeling would not be very potent if it did not offer the ability to model reality. In other words, there is this state called before-the-fact, and then there is this other state called after-the-fact. Which state does the Project Schedule cover? Let’s think about it for a moment.

11A1: Planning, Scheduling … and Reality

We must go back to an earlier discussion about the difference between Planning for a happening and Scheduling for that same undertaking. For instance, let’s imagine that the “happening” is the family’s summer vacation.

Perhaps while sitting around the Thanksgiving table, the question arises as to whether the family will take a vacation during the upcoming summer. And if so, it is asked, “Where should we go?” The discussion is lively, and one of the teenagers rushes for their iPad to look at possible destinations. After much deliberation a number of key decisions have been made.

- They definitely will take a vacation.
- Due to prior commitments by various family members, the vacation must take place during a limited window of opportunity, between June 18 and August 11.
- Due to financial limitations, the budget for the vacation is capped at $3,000 for a family of four.
- Given the financial limitations, it is agreed that driving is more economical, and therefore the destination should not be more than two-day drive in either direction.
- Further consideration to the age of the family members, as well as the time of year, it is decided that a week in Ontario might be a lot of fun. Niagara Falls is less than two hours from Toronto in one direction, and the 1,000 Islands is an hour away in the other direction. Plus, extended family lives in the area.

The above thought process is a simple example of Project Planning. Notice that the Planning process does not require getting buried in details, although some details are instrumental to reaching Planning decisions. Notice also that Planning occurs before-the-fact.
In early February the parents start working out the details. They check the websites for venues in and around Toronto, and they uncover other possible attractions that might be worth taking in. They build an MS Excel table of attractions, prices, hours, and so forth. They also look at hotel availability at their preferred facilities. It is decided that the current fare war allows the family to fly and rent a car for the cost of driving two days that would necessitate mid-trip hotels in each direction.

Armed with this information, and after one last look around the room for unanimous consent, airline tickets are bought, hotel reservations are made, and funds moved from General Savings to a Vacation Fund. With the primary details nailed down, different family members begin to “prepare” for the upcoming vacation. Arrangements are made to board the family pet, a disappointed Golden Retriever named Max, who doesn’t like the idea of being left behind. The daughter needs to make arrangements for someone to assume her duties as a tutor at the library. The father needs to file appropriate papers to secure vacation time at work. And on it goes.

The above Activities may just seem like more detailed Planning, but they also represent Scheduling. An often-cited difference between Planning and Scheduling is the inclusion of dates – Times when things will happen. But Cognitive Project Management notes that, while dates are the most frequently cited distinguishing characteristic of Scheduling, a number of other details are also characteristic of Scheduling. Indeed, the next most common distinction between Planning and Scheduling is the Level of Detail. Scheduling tends to get into the details, whereas Planning is normally more general in focus or consequence.

Once again, though, we draw your attention to the observation that Scheduling, just like its partner Planning, is a before-the-fact process. Once the family has arrived at the hotel and unpacked their bags, as they are loading into the elevator to head down to dinner, one would no longer describe them as Planning their vacation … or Scheduling their vacation. They are on their vacation!

Now, you and I can agree that after they return home, ten days later, the vacation will be history! And any comments or thoughts about it would be after-the-fact. But what would you say about changes in Plans during the vacation? Suppose, at dinner, they hear that the roads out to the 1000 Islands have been washed out by a catastrophic geological shift in tectonic plates? Suddenly they find it necessary to rearrange the Schedule so that other attractions, Planned for later in the week, can happen sooner … thus buying them time to find something else to do later in the week.

Is this change in Plans a Planning process – a Scheduling process – or both? Or neither? The answer comes down to what the happening is; back to that Level of Detail. If we
are still talking about the **Summer Vacation** as the happening, then these minor changes are not really Planning for, or Scheduling of, the vacation – since they are *on* vacation. And if the change is a minor one, like the change of one venue, then it wouldn’t even qualify as a re-Planning, would it? But what if they couldn’t find their passports, and so entering into **Ontario** altogether was now out of the question … wouldn’t that cause a re-Planning of the vacation?

But back to the hotel, what would we call this need to change venues? Is it re-Scheduling? If Planning and Scheduling are *before-the-fact* processes, then the subject of the re-Scheduling cannot be the vacation itself, but rather just one part of the vacation. As for the overall vacation, it is being neither re-Planned nor re-Scheduled with the change of one venue. Likewise, if one family member has a minor bout with a stomach bug and doesn’t get out of the hotel room in time for the Planned shuttle to an attraction, and the family must wait for the next shuttle … this change in how things happen does not constitute a re-Planning or a re-Scheduling.

### 11A2: Reflecting Reality Makes the Project Schedule Really Dynamic

In the world of Project Time Management, periodic regulated changes to Schedule Content – meant to keep the Schedule current and relevant – are called Schedule Updates. In a nutshell, a Schedule Update is a process whereby the Project Schedule is modified to reflect the best understanding of the current reality. This is accomplished by injecting the Schedule with better (more complete or current) information than existed during initial Schedule Development. Included in the Schedule Update process, then, is (at least the opportunity for, if not the actual performance of) a re-Planning of the remainder of the Project. We will talk more about this later in this chapter.

A Network-Based Project Schedule is *dynamic* as we have said many times, but not just because the Activities are linked to one another by way of Performance Restrictions. It is dynamic also – and, perhaps, especially – because the Critical Path Method of Modeling allows for the incorporation and influence of real time data as part of that model.

Take a moment to reread that last sentence. For a Project Execution Strategy model to work, it must reflect not just the initial intentions of the Project Executors, but also the unfolding reality. And doesn’t this get to the very heart of the word, Project? A Project is the performance of Actions, right? It is not just the Strategic Execution Strategy *before-the-fact*. It is the actual “unfolding” of that Plan as it manifests on the job site, and elsewhere. The Project is the *implementation* of the Project Execution Strategy. That is why Cognitive Project Management is careful to declare that “a Project Schedule models not just the Project Execution Strategy, but the Project Execution as well."
What good would a Project Schedule be if it did not model reality as it unfolds, but instead was only a lifeless, detailed embellishment of an original theoretical Plan? If it only did the latter, then it would not be a model of the Project Execution itself, but instead just a detailed depiction of the wishful thinking of its original contributors.

And so, the topic of this chapter is all about how the Project Execution Strategy model can be (and should be) periodically modified to capture information about the Project's unfolding – and how this new data can be melded with what remains of the original Project Execution Strategy.

11A3: “Reality” is Performance, Not Just Ingredients

There is a medieval saying that is still popular today:

“The proof of the pudding is in the eating.”

We herein adapt that saying and assert that “the proof of the Schedule is in the Project Execution.”

If we are to evaluate the merits of our Project Execution Model (the Schedule) on the basis of how well it held up during Project Execution, then it must model that execution: both before-the-fact (as our best reflection of cooperative intentions), during-the-fact (as a reliable depiction of the current Project state), and after-the-fact (as conclusive evidence of performance achievement).

The importance of the above point may not be all that obvious at first. We raise it however because we have been privy to one too many debates among ostensibly senior Project Time Management practitioners (who ought to know better), who forget that the Project Schedule is not the Project! Rather, the Project Schedule is a model of Project Execution – designed to both influence (i.e., Planned) and reflect (i.e., actual) Project Performance.

Let us be abundantly clear: The Project Schedule is a Project Performance model.

By the way, this understanding aligns with how we humans naturally think. What is a theatrical play? Well, certainly you can describe it in terms of its component parts and contributors. You could say that a stage play is the intelligent contribution of playwright, actor, director, producer, set designer, costume designer, and so forth. But when you say, “I saw a play last night,” one understands that you are referring to the performance of a play.
the play. An airplane flight involves an aircraft, the burning of fuel, a pilot, a route, and so forth. But when you say, “I was on a flight last night,” we understand that you are referring to a flight in the air!

This is how humans interpret words that have both static and dynamic states. Unless directed otherwise, we gravitate to the dynamic state, where the Action happens. We do so because the dynamic state constitutes the ends toward which the component parts are only the means. For instance, “I was confronted by his negative attitude.” By “attitude,” do we mean:

- **Attitude:** Manner, disposition, feeling, position, etc., with regard to a person or thing; tendency or orientation, especially of the mind.

As Dictionary.com defines the term? Or do we mean how it felt to us, being given the cold shoulder, having to put up with sarcastic remarks under his breath, hearing from others that he was bad-mouthing me, and so forth?

Likewise, a Construction Project is not just some combination of design drawings, contract documents, stacks of materials, assemblage of workers, or physical space enclosed by a chain-linked fence. Those may be the static elements of a Project. No, a Project is an endeavor, which in turn implies Action: be that Action eventual, imminent, or realized. Let us prove our position. Recall the popular definition of a Project:

- **Project:** A temporary endeavor undertaken to create a unique product or service.

Dictionary.com tells us that an “endeavor” is a “strenuous effort.” It further clarifies that “effort” is “exertion of physical or mental power.” Finally, it says that “exertion” is a “vigorous Action.” Therefore, as we just said, a Project implies “action.” And so, a Project Execution Model (which the Project Schedule is) must model that Action, and not just the component parts of that Project.

Just why this distinction matters, between the static nature of Project elements and the dynamic nature of unfolding Project Performance, is because it guides us as to how we use the Project Schedule and how we measure it effectiveness.

Let us return to that age-old adage about pudding. With acknowledgement and thanks given to [The Phrase Finder](http://www.phrases.org.uk/meanings/proof-of-the-pudding.html) website, we offer a short departure from our heady discussion, a detour that we think you will find both entertaining and instructional.
The Proof of the Pudding

Meaning: To fully test something you need to experience it yourself.

Origin: The “proof of the pudding” is just shorthand for “the proof of the pudding is in the eating.” That longer version makes sense at least, whereas the shortened version really doesn’t mean anything - nor does the often-quoted incorrect variation, “the proof is in the pudding.” The continued use of that meaningless version is no doubt bolstered by the fact that the correct version isn't at all easy to understand.

The meaning becomes clear when you know that “proof” here is a verb meaning “test.” The more common meaning of “proof” in our day and age is the noun meaning “the evidence that demonstrates a truth” - as in a mathematical or legal proof. The verb form, meaning “to test” is less often used these days, although it does survive in several commonly used phrases: “the exception that proves the rule,” “proof-read,” “proving-ground,” etc. When bakers “prove” yeast they are letting it stand in warm water for a time, to determine that it is active.

Clearly, the distinction between these two forms of the word was originally quite slight and the proof (in a “showing to be true” sense) is merely the successful outcome of a test of whether a proposition is correct or not.

“The proof of the pudding is in the eating” is a very old proverb. The Oxford Dictionary of Quotations dates it back to the early 14th century, albeit without offering any supporting evidence for that assertion. The phrase is widely attributed to Cervantes in The History of Don Quixote. This appears to be by virtue of an early 18th century translation by Peter Motteux, which has been criticized by later scholars as 'a loose paraphrase' and 'Franco-Cockney'. Crucially the Spanish word for pudding - 'budín', doesn't appear in the original Spanish text. It is doubtful that 'the proof of the pudding’ was a figurative phrase that was known to Cervantes. The earliest printed example of the proverb that I can find is in William Camden's Remaines of a Greater Worke Concerning Britaine, 1605: "All the proof of a pudding is in the eating."

It is worth remembering that, as the phrase is quite old, the pudding wouldn't have been a sticky toffee pudding from the sweet trolley, but a potentially fatal savory dish. In Camden's Listing of proverbs he also includes “If you eat a pudding at home, the dog may have the skin,” which suggests that the pudding he had in mind was some form of sausage. The Oxford English Dictionary describes
the mediaeval pudding as “the stomach or one of the entrails of a pig, sheep, or other animal, stuffed with a mixture of minced meat, suet, oatmeal, seasoning, etc., and boiled.”

Those of you who have ventured north of the border on Burns Night will recognize this as a fair description of a haggis - "the great chieftain o' the pudding-race", as Burns called it in the poem Address to a Haggis, 1786. Mediaeval peasants, faced with a boiled up farmyard massacre, might have thought a taste test to have been a wise choice.

Now that we appreciate the historical context of this timeless expression, we come to understand that the real value of an endeavor is in its ultimate form. It isn’t about the ingredients themselves. It isn’t even about the recipe we follow to combine those ingredients. It isn’t in the presentation on the plate. It … the success or failure of the endeavor as a whole … is in how well it sits with us, when all is said and done. It is in the “eating” – which, of course, is an Action! Once eaten, we can at last (and only then, at first) sample the flavor and subject the food to the ultimate test – whether it sits well with us.

We can apply this same thinking to the Project Schedule, believing that the ultimate “proof” of whether a Project Schedule is a good Project Execution Model is to be found in how well it functions as a model of Project Performance: before, during, and after the fact.

11B: How Time is Understood in a Project Schedule

Up to this point in the book, our Schedules have been completely theoretical. That is, the Logic and Activity Durations and Performance Restrictions have all been understood as reflecting the best thinking of the Project Executors as to how they intend to approach performance of the Project's defined Scope of Work. All Logic Diagrams that we have dealt with reflected a before-the-fact vantage point.

This chapter, however, invites us to understand the other dimension of information to be found in a Schedule as a Project Execution Model. For if, up until now, our Schedule's content has adopted an exclusively before-the-fact perspective, we will now consider an after-the-fact viewpoint.

For the first time in this book we will consider using our Project Schedule – as a model of Project Execution Strategy – to reflect more than just best thinking or clever intentions. Now we will incorporate into our Project Execution Model certain additional information about the Project itself, as it is unfolding. And, as we will soon learn, there is really no limit to the number of times that we can modify our Schedule to incorporate the realities
of the Project. This periodic adjustment to the Project Schedule — one which we can make as often as we care to — is what Dominant Project Management generally means when it uses the common term, Schedule Update. The ICS-Dictionary provides this definition of the term, Schedule Update:

**Schedule Update:** As used throughout Dominant Project Management, the term Schedule Update describes both a work product and a Project Time Management process. A Baseline Schedule, which is a virgin Project Schedule on a yet-to-be-performed Project, contains no actualized or historical data. However, once the Project has commenced, there is informational need in “updating” the Baseline Schedule to reflect what has transpired “to date” (up to the Data Date). Captured information includes actual Work Performance, actual Work conditions, changes in Work objectives, and so forth.

1B1: About Schedule Editions and Schedule Iterations

But Cognitive Project Management asserts that the term Schedule Update is too general, and needs to be subdivided into two new terms: Schedule Edition and Schedule Iteration. Per the ICS-Dictionary:

**Schedule Edition:** In Cognitive Project Management jargon, a Schedule Edition refers to any separate release of a Project Schedule in which Schedule Content has been modified from the Baseline Schedule, based on a different Data Date. A single Schedule Edition may be presented in different Schedule Formats. [Note: A single Schedule Edition that utilizes the same Schedule Report Format and same Data Date, but contains different Schedule Content, is known as a Schedule Iteration.]

**Schedule Iteration:** Schedule Iteration refers to a particular release of a Schedule Edition, where there happens to be more than one release of the Schedule Edition. All Schedule Iterations bear the same Data Date, and utilize one or more of the same Schedule Report Formats. They differ only in slight changes to Schedule Content, often as a “correction” to a previously rejected Schedule Iteration of the same Schedule Edition.

Just what are these two terms, and how do they differ from a Schedule Update? To answer this question, let’s think about the local television weather broadcast.

It occurs at the same time each night. It lasts the same number of minutes. It follows the same standard format, and it presents the same routine charts. But what changes from broadcast to broadcast are the specifics, the details. Now we know that some of those
Looking Backward in Time: The weather broadcast tells us what we just experienced. In this regard (recapping yesterday’s weather), rarely does the broadcast tell us anything we did not already know; it just quantifies our experience. It tells us how many inches of rain fell, how hot it got, how humid it was, what wind damage was inflicted in the area, and so forth.

Looking Forward in Time: The weather broadcast tells us what to expect: in the near term, and even further out in time. It speculates on the next day’s temperature range, humidity levels, precipitation expectations, and so forth. It also provides a general explanation of what is happening around us; sort of an educational piece to the presentation — things like explanations of the fronts, the Alberta Clipper, the El Niño effect, tropical depressions forming in the Atlantic Ocean, and so forth.

Now, let’s say that the overall broadcast segment is called The Channel 6 Weather Report, and there are two presented each night, one at the dinner hour and one right before bedtime. How many evening weather reports does the television station have? If we ask the station manager he would likely say, “We have two evening weather reports.”

But what about the fact that each of the two reports is being modified and broadcast 365 times each year. Doesn’t the station have 730 evening weather reports (per year)? This is where we make the distinction between (a) the program itself, and (b) the editions of the program, where all editions follow the same format, but differ principally in terms of content details.

The same distinction applies to magazines and newspapers. A publishing company may operate three different newspapers in a certain geographic region: the Daily News (a morning paper), the Sentinel (an afternoon paper), and the Frugal Flyer (a neighborhood throw-away). The company would say that they produce three newspapers.

But then, each “newspaper” would be issued daily, and each such release would be called an edition. Hence, the Sentinel’s edition for June 3rd might carry an article about pet thefts that would not appear in any other edition, or in either of the other two papers.

Cognitive Project Management thinks of the Project Schedule along these same lines: that each significant change in a Project Schedule’s substantive content constitutes a distinct Schedule Edition. This means that the content of the Schedule, at the detailed level, is different than any other release of Schedule Content. Look at Figure F1102.

Back to our earlier discussion, each Schedule Update process significantly changes Schedule Content at the substantive detailed level. So even if a release of a current
Schedule Update uses the same format as was used in the previous reporting period, this constitutes a new Schedule Edition – because the details have changed in a substantive way, and to a significant extent.

Schedule Report Formats, on the other hand, refer to how information from a single Schedule Edition is packaged and presented. Packaging differences result from the combination of Data Sorting, Data Selecting, and Data Grouping options that have been chosen. You will recall these terms from our previous discussion about Activity Codes in Chapter Nine.

Finally, Schedule Iteration refers to a particular release of a Schedule Edition, when there happens to be more than one release of that particular Schedule Edition. In Figure F1102, for example, the original Baseline Schedule is surely a Schedule Edition; indeed, it is the very first Schedule Edition of the Schedule. The Contractor submits it to the Owner for their review and acceptance as the Project's Baseline Schedule. Now, suppose that the Owner is willing to accept the Schedule as the Baseline – but only after some minor changes are made to its content or appearance. The Contractor may prepare a second issuance of the same Schedule Edition. This is what we are calling a Schedule Iteration.

**11B2: About Schedule Tense**

To better understand the categorization of Schedule Editions, we need to discuss something that Cognitive Project Management calls Schedule Tense. Per the ICS-Dictionary:

**Schedule Tense:** Every Project Schedule can contains information pertaining to the past, present, and future. Accordingly, Schedule Content can be better utilized when understood in the context of its place along the temporal continuum that spans Past Period, Current Period, and Future Period. Each Performance Period is characterized by different informational nuances and Data Type Perspectives. Schedules are divided into Past Segment, Current Segment, and Future Segment,
and these Schedule Segments correspond to Project Performance Periods: Past Period, Current Period, and Future Period.

In consideration of all of the above, in the context of Time, Cognitive Project Management felt the need to create the concept of Schedule Tense, temporal boundaries that separate the past, present, and future.

With few exceptions, virtually every Schedule Edition may contain data from one or more of these three tenses.

Said differently, a Project Schedule can be seen as having up to three different temporal segments: a Past Segment, a Current Segment, and a Future Segment. And these correspond to the three Performance Periods on every Project: a Past Period, a Current Period, and a Future Period. We will discuss each of these in the next subsection.

Just to be clear, you will not likely encounter any of these new terms (Schedule Tense, Schedule Segments, or Performance Periods) in Dominant Project Management literature; they are innovations of Cognitive Project Management. However, they are incorporated, without apology, into this and other ICS-Compendium volumes because we believe that they add to our (Project Executors and Project Facilitators) collective ability to effective manage the use of Time on Construction Projects – which is, after all, the primary reason that we create Project Schedules in the first place!

For now, though, we want you to understand that:

- A Project Schedule might function as a before-the-fact document (i.e., as long as Project Execution has not begun). Such a Schedule Edition is commonly known as an As-Planned Schedule. In such a Schedule there would be no Past Segment or Current Segment – only a Future Segment.

- Conversely, a Project Schedule might function as an after-the-fact document (i.e., once the Project is complete and Project Execution has ended). This Schedule Edition would be known as an As-Built Schedule, and such a Schedule would have no Future Segment or Current Segment, only a Past Segment.

- All other Schedule Editions – that have Data Dates some time after PROJECT START, but some time before PROJECT END – would contain all three segments: a Past Segment, a Current Segment, and a Future Segment.
11B3: Introducing the Data Date

You will notice that in that third bullet we used a term that is new to you, but is at the very epicenter of the Critical Path Method of Modeling: the Data Date. The Data Date is actually a conceptual value; a Moment in Time. In our everyday lives, we refer to this Moment in Time by a much more familiar name: “Now!” Here is what the ICS-Dictionary says about the Data Date:

**Data Date:** Less commonly referred to as the Time Now Date, the Data Date is not a date, per se, but rather a conceptual Moment in Time, that indicates the “cutoff date” for a schedule reporting period. In Cognitive Project Management, the Data Date is used to orient the Current Period along the Project Length continuum.

In the Project Time Management discipline, this theoretical Moment in Time – that separates the past from the future – is called a Data Date. For example, in a 365-day Project that begins on January 1st and ends on December 31st, if we were taking stock of where things stood half-way through the Project, the Data Date might be June 30th.

The Data Date serves an important function. It helps us to distinguish the past from the future, at least with respect to the Moment in Time we know of as “now.” You might think of the Project Schedule as having been split into the equivalent of two Schedules that are symbiotically linked together by way of the Data Date.

You realize, of course, that even though the Data Date is expressed as a Gregorian Date (or, in manual Forward Pass and Backward Pass calculations, as an Ordinal Date) the Data Date is not actually a date at all – neither is it a day; nor does it have any temporal dimension. The Data Date is actually just a Moment in Time – a dimensionless slice that cuts through Time and serves to separate the past from the future. Think about it: when we read a Data Date as an Actual Date, say, March 23rd, we cannot mean the entire length of the day. Instead, and depending on the Point-of-Day Perspective we choose to adopt, we may mean either the first moment of this date, or the last moment of this date.

Let us also give notice to the leading adjective, data, which prefixes this term. “Data” reminds us that this date is important to the interpretative value of the data contained in this Schedule Edition. In a one-year Project that begins January 1, for instance, a Schedule Update with a Data Date of July 31 tells us that the information (data) contained in the Schedule reflects the Project's best understanding of progress and conditions as of that Data Date.

11B4: Time Passage and Performance Achievement

On real Construction Projects, Schedule Updates are performed periodically, and create what Cognitive Project Management refers to as interim Schedule Editions, meaning
that they are neither the first nor the last Schedule Edition. Here are two things you need to know about interim Schedule Editions;

- **Time Passage**: First, each such interim Schedule Edition exists somewhere along the Project Timeline continuum and you need to know precisely where the Schedule Edition is rooted. The Data Date will indicate a Moment in Time, but you must also make the mental reference to where the Data Date is with respect to **PROJECT START** and **PROJECT COMPLETION**.

- **Performance Achievement**: Second, you must be forever mindful that all Activities in the Past Segment reflect work performed, whereas all Activities in the Future Segment remain just as theoretical as they were when they were originally conceived (in the As-Planned Schedule). Never let this distinction escape you. That is:
  - All of the Calculated Dates that are **after** the Data Date are **theoretical**: just as they have been since they were first developed as part of the initial Project Baseline Schedule. The Earliest Dates are still best guesses as to when, at their most hopeful, Activities *might* start or finish ... at some point in the future. And Latest Dates are still best estimates as to the latest that Activities can be started or finished without jeopardizing downstream Deadline Milestones.
  - Contrast this with all of the dates that are **prior to** the Data Date, which must all be **Actual** Dates. If the work has been performed, then the Actual Start and Actual Finish Dates are known. If they haven’t been performed then, when they eventually are performed, such performance would of course occur in the future – later than the current Data Date. Does this make sense?

As we said above, within every Schedule Edition there is what Cognitive Project Management calls Schedule Tense. We want to spend a few minutes to appreciate the role that Schedule Tense plays in Project Time Management. We will be working with two sets of correlated terms. Out on the real Project, the length of the Project can be divided into the Past Period of performance, the Current Period of performance, and the Future Period of performance. Correspondingly in the Project Schedule, we can identify the Past Segment, the Current Segment, and the Future Segment.

**11C: Temporal Context of Schedules: Schedule Segments**

In a few minutes we will get into the nitty-gritty of how to perform CPM Date Calculations in a Project Schedule that contains information pertaining to Project Performance. Before we do, however, we want to frame that discussion with a panoramic view of the Project Life Cycle and how different Schedule Segments (which reflect a Project's Performance Periods) correlate across time.
Schedule Segments: Providing a temporal context for Schedule Data, Schedule Segments correspond to Project Performance Periods, both of which are divided into three phases: Past, Current, and Future. The Current Segment, which corresponds to the Current Period, straddles the Data Date and separates the Past Segment from the Future Segment. The Current Segment is further divided into the Just Done Segment (immediately prior to the Data Date) and the Imminent Segment (immediately following the Data Date).

Let us examine these three Schedule Segments from different angles:

- Dimensions and Boundaries and Schedule Segments
- Contextual Perspectives of Each Schedule Segment’s Unique Data
- Contrasting Inherent Nature of Different Schedule Segment Data Types

11C1: Performance Periods: Their Dimensions and Boundaries

We will now examine each of the three Performance Periods, to learn about (a) their dimensions and boundaries with respect to one another, (b) the temporal context of data contained within each Performance Period, and (c) the inherent nature of the data types resident in each Performance Period.

Let's begin by providing ICS-Dictionary definitions for the key terms we will be discussing over the next several pages:

- Performance Periods: In the Cognitive Project Management model, the Project Length is divided into three sequential phases, each delineated by temporal boundaries that are relative to the Data Date. The Current Period, which straddles the Data Date, also serves to separate the Past Period from the Future Period. Each Performance Period correlates to a corresponding Schedule Segment.

- Past Period: One of three Performance Periods, the Past Period stretches from the start of the Project to the commencement of the Current Period. That portion of the Project Schedule that reflects the Past Period is referred to as the Past Segment. Past Period data is contextually retrospective.

- Current Period: Because the Current Period is the only one of the three Performance Period in which Work is actually performed, temporal boundaries for the Current Period are especially important. Cognitive Project Management recommends use of a fixed percent of Project Length, as opposed to a specific number of days. ICS-Protocols uses 5% as the default “width” of the Current Period. The Current Period separates the Past Period from the Future Period, and its data is contextually introspective.
future period: one of three performance periods, the future period stretches from the end of the current period to the project end date. that portion of the project schedule that reflects the future period is referred to as the future segment. future period data is contextually prospective.

11c1a: current period: dimensions and boundaries
we begin with the current period, because the past period and future period are subordinate to, and ultimately defined by, the current period.

the proper terms are project performance past period, project performance current period, and project performance future period. to ease communication, we drop the first two words, and instead refer to these as simply past period, current period, and future period.

we trust that, even before we get into any particulars, you have a general idea of what we mean by a current period. all project execution occurs in the present, never in the past or future. the esoteric question is: what is meant by “currently” or “current,” as in: “we are currently encountering…” or “our current plan is to….”?

11c1a-i: relationship of current period to project length
the ics-compendium development team held discussions on this point and concluded that it made little sense to stipulate a specific number of days for the current period because project lengths could range from quite short to quite long. instead, they felt that the current period might be better defined as a percent of the entire project length.

this of course led to a healthy discussion as to what that percent ought to be. in the end it was agreed that 5% was a fairly versatile value, at least as a default, and would work for most construction projects. it was also noted that each project team should decide for itself what percent of project length they would want to consider as the current period.

the first two columns of the table in figure f1104 cross-reference different project lengths with their corresponding current periods. (the other two columns will be described next.)

so, for instance, on an 18 month project, the current period would be 3.9 weeks. assuming five workdays per week, this equates to 22 workdays.

11c1a-ii: introducing just done period and imminent period
having developed a consistent formula for determining the width of the current period,
we next had to decide where the Data Date would fall within that period of Time. Was the Data Date to be at the start of the Current Period, at the end of the Current Period, or somewhere in between?

Interestingly there was almost immediate concurrence among the ICS-Compendium Development Team members that the Current Period must minimally include the most immediate past few days. Several different reasons were given, but the explanation that seemed to serve as a theme for the other reasons was that on any given day, a Project is pursing agendas set during the previous few days.

In other words, of all of the days within the Current Period, the most immediate of all days would have to be “today.” And rarely do we work on things today that were not Planned out and cooperatively blessed by the Project Team in the recent days leading up to “today.”

For this reason we decided that the Current Period should be split by the Data Date, with 20% of the Current Period stretching backward in Time from the Data Date, and the remaining 80% of the Current Period stretching forward into the immediate near term. We gave the name Imminent Period to the latter (last 80% of the Current Period) and Just Done Period to the former (first 20% of the Current Period).

These words were meant to describe the information contained in each Schedule sub-segment. Activities in the Just Done Period were … “just done,” while Activities in the Imminent Period were expected to take place in the very near future.

F1104: Project Length and Schedule Periods

<table>
<thead>
<tr>
<th>PROJECT LENGTH (Months)</th>
<th>CURRENT PERIOD (Weeks)</th>
<th>JUST DONE (Days)</th>
<th>WORK IMMINENT (Days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1.3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>2.6</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>18</td>
<td>3.9</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>24</td>
<td>5.2</td>
<td>5</td>
<td>24</td>
</tr>
<tr>
<td>30</td>
<td>6.5</td>
<td>7</td>
<td>30</td>
</tr>
<tr>
<td>36</td>
<td>7.8</td>
<td>8</td>
<td>36</td>
</tr>
<tr>
<td>48</td>
<td>10.4</td>
<td>10</td>
<td>48</td>
</tr>
<tr>
<td>60</td>
<td>13.0</td>
<td>13</td>
<td>60</td>
</tr>
</tbody>
</table>

- **Just Done Period:** The Just Done Period is that portion of the Current Period that occurs immediately before the Data Date. The “width” of the Just Done Period is a fixed percent of Project Length, as pre-determined during the Schedule Design.
Summit and recorded in the Scheduling Performance Specifications (SPS). For example, if the SPS sets the temporal width of the Just Done Period at 1% of Project Length, then on a Project lasting 200 days, the Just Done Period would be two days long. The Just Done Period ends at the Data Date.

Imminent Period: The Imminent Period is that portion of the Current Period that occurs immediately after the Data Date. The “width” of the Imminent Period is a fixed percent of Project Length, as pre-determined during the Schedule Design Summit and recorded in the Scheduling Performance Specifications (SPS). For example, if the SPS sets the temporal width of the Imminent Period at 4% of Project Length, then on a Project lasting 200 days, the Imminent Period would be eight days long. The Imminent Period commences at the Data Date.

Returning to Figure F1104, we see that it converts each Current Period value, expressed in weeks, into its equivalent Just Done Period and Imminent Period values, expressed in Workdays. So, to complete our example, an 18 month Project's Current Period would span from four days prior to (and exclusive of) the Data Date through 18 Workdays beyond (and inclusive of) the Data Date. You might better understand all of this by looking at Figure F1106, which illustrates the relationship between all of these terms.

11C1a-iii: Calculation of Current Period Width and Start/End Dates

The above discussion is a bit conceptual; we might want to include a formulaic description of the Current Period and its component parts, the Just Done Period and the Imminent Period. We begin with this clarification of the Data Date itself:

The Data Date, whether expressed as a Gregorian Date or Ordinal Date, actually refers to the Moment in Time that occurs at the Start-of-Day of the cited Data “Date.” Therefore, for a Schedule Edition bearing a Data Date of June 3, 2012, all information contained in the Schedule Edition is valid through the end of work, June 2, 2012.

Now we provide a technical description of the Just Done Period.

The width of the Just Done Period will be the lesser of (a) the difference between the Data Date and the Project Start Date (assumed to be zero), or (b) 1% of the total Project Length. All calculations should be consistently founded: either Workdays, or Calendar Days, but not a mixture of both. If the Data Date is Day 120, and the Project ends on Day 200, then the Just Done Period spans between Day 118 and Day 119, inclusive.
The difference between the Data Date (Day 120) and the Project Start Date (Day 0) is 120 days. Since 1% of 200 days is two days, the lesser of the two values is two days. Hence, the width of the Just Done Period is two days. We understand that the Just Done Period ends just before the Data Date, so it must therefore span Day 118 and Day 119, inclusive.

The width of the Imminent Period will be the lesser of (a) the difference between the Data Date and the PROJECT END DATE, or (b) 4% of the total Project Length. Since the Project Length is 200 days, then 4% would be eight days. The difference between the Data Date and the Project END DATE is 80 days. Eight days is less, so the width of the Imminent Period is eight days. The Imminent Period begins with, and includes, the Data Date day, so the span of the Imminent Period is Day 120 through Day 127, inclusive.

Combined, we now have the width of the Current Period, which begins at the start of the Just Done Period and ends at the close of the Imminent Period. In this example, the Current Period spans from Day 118 through Day 127, inclusive – a width of ten days … which equals 5% of the Project Length.

11C1b: Calculation of Past Period Width and Start/End Dates

The Past Period fills in the difference between the Project Start Date and the beginning of the Current Period. In the above example, since the Current Period begins on Day 117, then the Past Period spans from Day 1 through Day 116. The width of the Past Period is 116 days. If we divide the Past Period width by the Project Length, we get a percent figure that represents the portion of the Project that is now considered “history.” In this example, the Past Period comprises 58% of the Project Length.

11C1c: Calculation of Future Period Width and Start/End Dates

A similar thinking guides our understanding of the Future Period, which spans the difference between the PROJECT END DATE and the end of the Current Period. In the above example, since the Current Period ends on Day 127, then the Future Period spans from Day 128 through Day 200, the end of the Project. The width of the Future Period is 73 days. If we divide the Future Period width by the Project Length, we get a percent figure that represents the portion of the Project Length still available for Work Performance. In this example, the Future Period comprises 37% of the Project Length.

Notice that if we add up the three percentage values, we arrive at 100%. In the above example, the Past Period consumes 58%, the Current Period consumes 5%, and the Future
Putting Reality into the Project Schedule

11C2: Schedule Segment Data: Contextual Perspective

Now we switch our attention from the three Performance Periods (Past Period, Current Period, and Future Period) to their counterparts in the Project Schedule, the three Schedule Segments: the Past Segment, Current Segment, and Future Segment.

The proper terms are Schedule Past Period, Schedule Current Period, and Schedule Future Period. To ease communication, we truncate the terms to Past Segment, Current Segment, and Future Segment.

In particular, we are interested in what we call the Contextual Perspective of the data that are found in each of these three Schedule Segments.

When we speak of any Schedule Segment’s Data Contextual Perspective we are referring to the context in which those data are derived and used. We will discuss Data Perspective with respect to all three Schedule Segments, beginning with the Current Segment—because the other two Schedule Segments are substantively subordinate to the Current Segment.

11C2a: Current Segment: Contextually Introspective

The Data Perspective of the Current Segment of a Project Schedule is classified as an introspective. According to Dictionary.com, introspection means:

- **Introspection**: To look into or examine; the examination of one's own thoughts, impressions, and feelings, especially for long periods.

Indeed, certainly as contrasted with the Past Segment, which adopts a retrospective view of what was, or the Future Segment which continues to speculate on what might be, the Current Segment is preoccupied with the matters at hand. Daily conversations, immediate Actions, and ever-constant thinking all drive the Project forward one day at a time; sometimes, one hour at a time. And this is where mental focus ought to be.

This explains the ICS-Dictionary definition of the term, Current Segment:

- **Current Segment**: The Current Segment refers to that portion of the Project Schedule that falls within the timeframe of the Current Period, based on the Four Basic Calculated Dates. It is further subdivided into the Just Done Segment (which immediately precedes the Data Date) and the Imminent Segment (which immediately succeeds the Data Date).
The reason we are spending time thinking about the focus of the Project Executors is because it dictates the Contextual Perspective of the Project Schedule. We need to understand where the Project Executor’s head is at, at any given time – that is, if we as Project Facilitators hope for our Project Time Management products and services to be of any real value to them.

Completely unlike the other two periods, the Current Period is the Project. If we define a Project as being the *execution*[^3] of the Project – the performance of the Project – then such Action can only takes place in the present. The Past is static; the Past is recorded; the Past is history. The Future is mental; the Future is indeterminate; the Future is speculation and intentions, with promises of surprise and upset to greet us.

It is only in the Present that we live. It is only in the Present that we think, or observe, or strategize, or react, or duck, or risk. Only in the Present do we Collaborate, Cooperate, Coordinate, or Communicate. Only in the Present do we perform – do we execute the Project. Therefore, of all three Schedule Segments, only the Current Segment matters at the very moment we are actually performing the Work of the Project.

This puts special importance on the content and quality of information that is drawn from the Project Schedule during the Current Segment. For one thing, the information should be as accurate as possible. For another it should be as detailed as possible. The more detailed the better.

This perhaps explains why an unwritten, yet instinctive tendency among the most successful Contractors (even among those with no formal Project Time Management training) is to develop a short-life management tool called the **LOOK-AHEAD SCHEDULE**.

The Scope of the **LOOK-AHEAD SCHEDULE** is only a few weeks out from the Data Date, and its purpose is to reflect the Project Team’s current and best thinking of how to proceed over the short term. It is typically based on the Project Schedule, but elaborates the Activities in the Project Schedule into much more detail, usually down to the day, and sometimes into parts of a day.

Another common practice typical of any competent Contractor is the **WEEKLY COORDINATION MEETING**, wherein the Project Executors follow a strict agenda that addresses:

- What was accomplished in the preceding week;
- What is to be accomplished in the next few weeks (likely, referencing the **LOOK-AHEAD SCHEDULE**);

[^3]: Hopefully you will recall our earlier discussion wherein we concluded that the Project Schedule, as a model, does not merely reflect the Project itself, but rather the Execution Strategy of the Project.
What current issues challenge the Project;
What current concerns haunt the participants.

The Current Segment of the Project Schedule should provide a short-range look backward (Just Done Period), maybe only to the previous week or so, just enough to provide the current Operational Context of the Activities that are Scheduled to take place during the upcoming few weeks. These Imminent Period Activities span the balance of Schedule Content found in the Current Segment.

11C2b: Past Segment: Contextually Retrospective

Contrast the Current Segment with the Data Perspective of the Past Segment of a Project Schedule, which we classify as retrospective. According to Dictionary.com:

- **Retrospective**: Directed to the past; Contemplative of past situations, events, periods.

Here is the ICS-Dictionary definition of the term, Past Segment:

- **Past Segment**: The Past Segment refers to that portion of the Project Schedule that falls within the timeframe of the Past Period, based on the Four Basic Calculated Dates.

Let us be clear: the Past Segment of a Schedule has no functional effect on the CPM computations applied to the balance of the Schedule (Current Segment\(^4\) and Future Segment). The Past Segment is the burial ground for Activities, or portions of Activities, that have been finished. One could actually delete all completed Activities from a Project Schedule and still end up with the same Earliest Dates and Latest Dates for Activities in the remaining two Schedule Segments.

From a temporal perspective,\(^5\) the main and perhaps only practical reason for maintaining the Past Segment in a Project Schedule is convenience. That is, having the details of work performed and completed readily available for review and analysis serves to facilitate the kinds of discussions and thinking by Project Executors that should take place during the Current Period.

11C2c: Future Segment: Contextually Prospective

The Data Perspective of the Future Segment of a Project Schedule is classified as

---

4 But only the Imminent Period, though; the Just Done Period is just like the Past Segment, having no effect on CPM computations.

5 However, for non-temporal purposes, the Past Segment contains important information pertaining to actual performance: costs, resources, logs and notations, approved changes, modified logic, and so forth.
prospective. According to Dictionary.com, prospective means:

- **Prospective:** Likely to come about; expected to happen; related to or effective in the future; anticipated.

The Future Segment is extremely important for it provides the raw ingredients for Project success. The Future Segment of the Schedule is the meal plan to the chef, the recipe to the cook, the script to the actor and director, the flight plan to the pilot, the playbook to the quarterback, the lesson plan to the teacher, and so on. The Future Segment lays out the intentions of the Project Team, the Project Executors. It is influenced by the observations and insights of the Current Period, and informed by the statistics and inferences drawn from the Schedule’s Past Segment.

- **Future Segment:** The Future Segment refers to that portion of the Project Schedule that falls within the timeframe of the Future Period, based on the Four Basic Calculated Dates.

The Future Segment of the Schedule represents the essence of Project Management; the very heart and soul of Project Time Management. Recall the Cognitive Project Management summation of Project Time Management: Coordination, Collaboration, Cooperation, and Communication. It is here, in the Future Segment of the Project Schedule, that we skillfully align these four management objectives.

Through the meticulous steps in its creation, the Future Segment mandates and stimulates Communication among the Project Executors, as they contemplate the best **WAY FORWARD**. The essence of purposeful thought during this creative stage is consideration of the logistics and interactions between Project Players and results in a Coordination Plan.

If honored by all players (Cooperation), the Future Segment charts a route that leads to the Project's best chance of maximum success as measured against a mutually-beneficial scale. The Future Segment of the Schedule also provides the choreography for Project Execution that should yield the most harmonious Collaboration among the Project's participants.

**11C3: Inherent Nature of Different Schedule Segment Data Sets**

Informed by these distinctly different Contextual Perspectives, we are now able to characterize the types of data to be found in each Schedule Segment of the Project Schedule.
11C3a:  Past Segment: Realized Data and Gleaned Data

The Past Segment captures the history of Project Performance; this is what we learned from our discussion of the Contextual Perspective of the Past Segment. But if we look closer, we will notice that some of that capture is evidentiary, while the balance of Past Segment Data is inferred.

Dictionary.com defines evidence, from a legal perspective anyway, as:

Evidence: Data presented to a court or jury in proof of the facts in issue and which may include the testimony of witnesses, records, documents, or objects.

11C3a-i:  Introducing Realized Data

Cognitive Project Management labels such evidentiary data as Realized Data. Why “realized?” Again, from Dictionary.com, we learn that to realize means to:

Realize: Make real; give reality to.

Think about the starkest difference between Past Segment Data and that found in either the Current Segment or Future Segment of a Schedule; Activities in the latter two Schedule Segments have yet to happen. The Past Segment Activities did happen. The contents of the Current Segment and Future Segment represent best intentions, “the best-laid plans of mice and men.” These include Activity Duration estimates, Activity Sequencing schemes, staging and manpower stacking plans, delivery timetables, and so forth.

But the Past Segment tells us what actually transpired. It tells us what really happened. It tells us what the Project experienced; what the Project Executors, through their efforts, made real – from what were only plans and intentions. The Past Segment details what was realized.

Included in this narrow category of Realized Data are values such as Actual Start and Actual Finish Dates or Activity Percent Complete assessments. Also included would be any mutually-accepted changes to the Project in contract Work Scope, independently confirmed records of working conditions, and so forth. The essential attribute of Realized Data is that it is derived directly from the Project record.

Here is the ICS-Dictionary definition for Realized Data:

Realized Data: One of two categories of Past Segment Data, Realized Data captures the details of what has transpired in the Project Past Period, as contained in the Schedule Past Segment. Included in this category of Schedule Data would
be Actual Start and Finish Dates, assessments of partial Work Performance, introduction of authorized changes in Work Scope, changes in Project working conditions, and so forth. Realized Data is evidentiary in substance, whereas Gleaned Data is overwhelmingly inferred.

11C3a-ii: Introducing Gleaned Data

Contrast Realized Data with Gleaned Data, which comes from an interpolation of Realized Data. Here is the ICS-Dictionary definition Gleaned Data:

- **Gleaned Data:** One of two categories of Past Segment Data, Gleaned Data is an interpolation of Realized Data, in large part through recalculation of the Four Basic Calculated Dates. It also includes the findings of comparative analyses of planned and actual Work Performance, including Duration Variance, Start Date Variance, Out of Sequence Variance, Earned Value Variance, and the like. Gleaned Data is mainly inferred from Realized Data which, for its part, is overwhelmingly evidentiary.

This is the data category that contains the results of popular analytic processes, such as a variety of popular variance analyses:

- **Duration Variance Analysis:** This is a common analytic process whereby the Actual Durations of completed Activities are compared with the Planned Duration counterparts. Identified differences are thought to shed light on how the Project fell behind or accelerated ahead.

- **Start Date Variance Analysis:** Similar to the Duration Variance, a variance between the Planned Earliest Start Date and the Actual Start Date of an Activity is thought to suggest what contributed to Schedule slippages or gains.

- **Out of Sequence Performance Analysis:** This is a very popular observation made after-the-fact and points out instances where the Actual Sequence of Work Performance differed from the Planned Sequence. Dominant Project Management literature often considers working Out-of-Sequence an undesirable (even unacceptable) practice.

- **Earned Value Analysis:** This is a well-established set of analytic formulas that enjoys a rich and long history at the core of Dominant Project Management. Without getting into the details, Earned Value Analysis compares Planned and Actual Resource consumption levels and from these comparisons draws conclusions about the performance of the project. It is commonly called *Performance Measurement*. 
There are of course other types of analysis beyond what is listed above, but you get the idea. Gleaned Data is information that is inferred from the Realized Data. None of the above analyses would be possible without first knowing the details contained in the Realized Data.

You might say, then, that Realized Data is provided data, whereas Gleaned Data is inferred data. That is why we chose the word “glean,” which Dictionary.com defines as:

**Glean**: Learn, discover, or find out, usually little by little or slowly.

We especially liked the notion that Gleaned Data is discovered “little by little or slowly.” Doesn’t that accurately describe the inherent nature of conclusions drawn from raw statistics gathered over the course of multiple Schedule Updates? — that trends and patterns seem to stabilize over time (i.e., that wild pendulum swings of oscillating performance curves tend to level out over time) and that the denser the data population the more credible the inferences?

As we proceed with this discussion, let us be clear on the meaning of the word, “infer,” which Dictionary.com describes as:

**Infer**: To derive by reasoning; conclude or judge from premises or evidence.

Both Gleaned Data and Predictive Data (which we will introduce in the next subsection) are inferred from the “premises and evidence” contained in the Realized Data and Strategic Data (also discussed in the next subsection), respectively.

### 11C3b: Current Segment: Apparent Data

The Current Period, of course, refers to a very small window of Time, in the context of the entire Project Length. As previously explained, it is comprised of a very short Just Done Period (by default, 1% of Project Length) followed by an Imminent Period (by default, 4% of Project Length), strategically split by the Schedule Edition’s Data Date.

The information contained in the Current Segment of the Schedule has its own characteristics, to be sure, and we chose the word “apparent” to describe the essential nature of the data exposed and used during the Current Period.

Here is its meaning according to Dictionary.com:

**Apparent**: Readily seen; exposed to sight; open to view.
11C3b-i: Introducing Apparent Data

Throughout the ICS-Compendium we continue to stress a Cognitive Project Management philosophy, that the front lines of Project Execution are where the Project is won or lost. We believe that those on the front line will be the ones to first be aware of impending doom. They will also be the ones with the brightest ideas of how to ward off such doom, or to mitigate its impact, if it is not entirely avoidable altogether.

Cognitive Project Management holds the view that those who prosecute the work on a daily basis possess the most trustworthy eyes and ears of all, and that their observations and insights quite often offer the brightest suggestions of a WAY FORWARD for the Project. We prefer this view to the worn-out, proven-ineffective Command-and-Control philosophy of the World War II era. We have even given a name to this new management philosophy: Cognitive Project Management.

We think that the word Cognitive gets to the core of what we believe makes the difference between a management team that understands what it is up against, and one that clings (out of sheer faith or loyalty to a management dogma proven not to work) to the idea that the initial Plan is sacrosanct, and that compliance with the Plan is the essence of ideal Project Management. To folks of this mind set, the battle cry seems to be, “damn the icebergs, full steam ahead!”

Cognitive Project Management is based on the idea that those who prosecute the work on a daily basis – be they workers, foreman, superintendents, or managers – do so with their eyes and ears open! They are intelligent and responsible people who take pride in their work. And that combination of attributes – intelligence, responsibility, and integrity – is why they should be trusted to maneuver a Project through stormy seas.

Cognitive Project Management puts complete trust in the cognizance of the Project Executors and their army of dedicated workers. We know that they are continually taking stock of all that is around them. In this balanced partnership – the observer and the observed – we have a paradigm for Cognitive Project Management.

As we discuss the nature of information to be found in a Schedule’s Current Segment, the word “apparent” comes quickly to mind. As the ICS-Dictionary explains:

Apparent Data: Apparent Data is a term that describes the primary characteristic of all Current Segment Data. It is “readily seen,” “exposed to light,” and “open to view.” It is a call-to-action term that behooves all Project Participants to be forever aware—cognizant, if you will: of the conditions, opportunities, challenges, and potentials available in the Project in the Current Period.

This is what we mean by Apparent Data. It is the empirical data that lies all around us
at the time of the Schedule Update. It is more than just statusing Activities with Percent
Complete estimates, or assigning Actual Start and Actual Finish Dates to completed
Activities. For example, Apparent Data means being aware of the weather over the next
few days or weeks. But also:

☐ It means taking into account the latest, most credible information as to the likely
delivery of key materials, equipment, and supplies.

☐ It means considering Gleaned Data when sketching out the Short-Term Look-Ahead Schedule.

☐ It means taking into account known or anticipated changes to Scope.

☐ It means rethinking manpower and other Resource levels (required and likely available).

All of this is information that is apparent to a Project Team whose eyes and ears are open. Apparent Data should make its way into the Schedule, because this is the one and only opportunity to override the wishful thinking of the original Plan with the credible, practical, and enforceable approach to the immediate future, the Imminent Period.

11C3c: Future Segment: Strategic Data and Predictive Data

A similar set of data types exist at the other end of the Schedule spectrum. We have already noted that the Data Perspective of the Future Segment is prospective … looking forward. But, just like with the retrospective Past Segment, Future Segment Data can be segregated into two groups: that which is provided and that is inferred.

11C3c-i: Introducing Strategic Data

When you give it a little thought, the information contained in the Future Segment of a Project Schedule is either influential or consequential. One (influential) constitutes a cause (of Action), whereas the other (consequential) represents the effect (of Actions). Both are to be found in the Future Segment of the Schedule, and it would serve us well to distinguish the two from one another. Here are the ICS-Dictionary definitions:

!!! Strategic Data: One of two categories of Future Segment Data, Strategic Data appears as the purposeful integration of Mandatory Schedule Elements including Activities, Activity Durations, Schedule Logic, Date Constraints, and effective Scheduling Software Settings. Strategic Data is of an influential nature, whereas Predictive Data is of a consequential nature.

!!! Predictive Data: One of two categories of Future Segment Data, Predictive Data is mainly inferred from, and chiefly subordinate to, Strategic Data, the other Future
Segment Data category. Predictive Data is of a consequential nature, whereas Strategic Data is of an influential nature.

Strategic Data encompasses the Mandatory Schedule Elements that we discussed back in Chapter Four. Strategic Data is clearly influential, in that it lays out the intentions of the Project Executors. We are of course referring to the choice of Activities, the Activity Durations assigned to those Activities, the Logic between those Activities, the Date Constraints that pin those Activities, and any Software Settings that modulate those Activities.

The above description may sound a lot like a brand new, As-Planned Baseline Schedule, because a Baseline Schedule has a Data Date of Day 0, and the entire Schedule is a Future Segment! Well, remember that, earlier, we characterized a Schedule Update as a sort of re-Planning of the remainder of the Project. Well, what is the difference, if any, between Planning done before a Project commences, and the Planning performed during each Schedule Update? There are actually two huge differences!

- First, we have Gleaned Data. We have started to develop a performance track record on this Project – with this Project Team under these Project conditions.
- Second, we have Apparent Data; and this second cluster of information is the most valuable and influential of all. Apparent Data describes what is currently happening! Apparent Data details what we are currently doing, what conditions we are currently struggling with, what short-term goals we are chasing, what specific Actions we plan to take over the ensuing weeks (during the Imminent Period).

How could we ever hope to develop a strategy for the Future Segment that has any semblance of credibility, if we do not take into account where things currently stand or incorporate what we have gleaned from Realized Data to date? And so, it should be standard practice to include in the course of every Schedule Update a reappraisal of the integrity of every Schedule Edition’s Future Segment content.

11C3c-ii: Introducing Predictive Data

Predictive Data is what can be inferred from the Strategic Data of the Future Segment. This includes many of the most cited uses of the Schedule, such as predicting when the Project will finish or whether various Project Execution Commitments will be met.

What we find interesting is just how much importance Dominant Project Management places on Predictive Data as compared to the Strategic Data upon which it is based. In fact, some extremely well-known authorities on Project Time Management limit their reasons for having a Project Schedule at all to just this one data set, the Predictive Data.
They will say something along the lines that, “The purpose of the Project Schedule is to know whether the Project will finish on Time, or when it can be expected to finish.”

Our primary reason for separating Future Segment Data into two distinct groups (Strategic/Influential and Predictive/Consequential) is to draw your attention to the subordinate nature of Predictive Data, which is only as good as the Strategic Data upon which it is predicated. And we hope that you now appreciate that, in turn, Strategic Data should be rooted in Gleaned Data and Apparent Data. This may (now) seem like a no-brainer comment, but when you get out into the real world of Construction Project Management, we are certain that you will be shocked by how often (some estimates are upwards of 90% of the time) Schedule Updates do not include any recalibration of the forward Plan!


We hope you have enjoyed this chapter's discussion all about the three Schedule Segments and the three Performance Periods that they seek to model. We hope you have appreciated the distinctions we have drawn between what are essentially five different types of data to be found in a single Schedule.

Without your knowing it, we created five labels, the initial letters of which form a helpful mnemonic. The intent behind defining five data periods was to help you grasp an extremely important point about Project Schedules, about CPM as a Project Execution Model: a Project Schedule’s content is not homogenous. A Schedule’s Project data is not “composed of parts or elements that are all of the same kind,” as Dictionary.com defines the word, homogenous.

Rather, every Schedule Edition (except for the very first or the very last) contains a Mixture of Gleaned Data, Realized Data, Apparent Data, Strategic Data, and Predictive Data. Each data type has its own unique characteristics, and each serves a different set of informational objectives. By keeping all of this in mind, we believe you will begin to grasp the notion that a Schedule, as a Project Execution Model, is a heterogeneous and complex organism – one that lives and breathes. And now you know why the CPM Network-Based Schedule is the most dynamic form of Project Schedule of all.
CHAPTER TWELVE

12A The Schedule Data Credibility Profile
12A1 Credibility Profile Locates Data Date on Project Timeline
12A2 Schedule Data Integrity Improves Across the Project Length
12A2a Realized Data Accuracy Ties to Data Acquisition Frequency
12A2b Gleaned Data Accuracy Gets Better with Time
12A2c Apparent Data Endurance Ties to Width of Current Period
12A2d Strategic Data Inversely Proportional to Coverage Period
12A2e Predictive Data Inversely Proportional to Coverage Period
12A3 Credibility Profile Explains Project Model Usefulness
12A3a Time Surfing in the Data Date
12A3b The Flow of Essential Schedule Data
12A3c Essential Schedule Data Gets Better with Time
12A4 Understanding the Schedule Data Credibility Profile
12A5 The Schedule Data Credibility Profile on Construction Projects
12A6 What GRASP Tells Us about Dominant Project Management
12A6a What is Being Gleaned?
12A6b What is Not Being Gleaned?
12A6c Fixation on Compliance; No Interest in Assessing Potential
12A6d Dominant Project Management May Contain Failure Formula

12B Making a CPM Execution Model Really Dynamic!
12B1 CPM Schedule is Actually Three Project Execution Models
12B2 Secret to Most Effective Execution Model: Gleaned Data
12B3 GRASP the True Meaning of a Dynamic Project Schedule
12A: The Schedule Data Credibility Profile

With your heads full of these new terms – Performance Period/Schedule Segment, Past Period/Segment, Current Period/Segment, Future Period/Segment, Just Done Period/Segment, and Imminent Period/Segment – you may be wondering whether there is any practical use for them beyond the philosophical discussion we have just had.

Actually, there is! But we need to introduce one more new term: the Schedule Data Credibility Profile. \[1\] Before we tell you all about the Schedule Data Credibility Profile and how Project Facilitators and Project Executors can use it to achieve better Project Time Management, we need to make a few observations about the behavior of Schedule Data in different Schedule Segments, or Project Performance periods.

12A1: Credibility Profile Locates Data Date on Project Timeline

We begin by noting that within every Schedule Edition the Data Date anchors the Schedule Data Credibility Profile. Do you remember our earlier example, where the Data Date was Day 120, based on the Start-of-Day Perspective? In that example, the Past Period constituted 58% of Project Length, the Future Period spanned the last 37% of Project Length, and the Current Period of course consumed the default 5% of Project Length.

As for where the Data Date stood on the Project Timeline continuum, if we divide 120 by 200 (the Project Length), we get 60%. In other words, “now” (the Data Date) is 60% of the way across the allowable Project Length. I hope you would agree that this is a useful piece of information to know. Hold that thought, as we move to another important observation.

12A2: Schedule Data Integrity Improves Across the Project Length

Depending on data type (Realized, Gleaned, Apparent, Strategic, or Predictive), data quality mainly improves with the Passage of Time. Let’s consider the effect of Time on each Schedule Segment data type.

---

1 The Schedule Data Credibility Profile is an invention of Cognitive Project Management. You will not find its mention in Dominant Project Management literature.
12A2a:  Realized Data Accuracy Correlates to Data Acquisition Frequency

Have you ever tried to fill out an expense report or time sheet—six months after the fact? You know what we are talking about. Our recollections are much more accurate when the events are fresh in our minds. So, when it comes to statusing a Schedule with Actualized data, it is always best not to wait too long between when the Actions take place and when the Realized Data is recorded.

12A2b:  Gleaned Data Accuracy Gets Better with Time

At first you might think that Gleaned Data, which is based on Realized Data, should also be more accurate the more often we draw inferences from the Realized Data. But let’s recall the definition for the word, glean, which is:

- **Glean**: Learn, discover, or find out, usually little by little or slowly.

We noted earlier that as we accumulate more empirical data, the credibility of the inferences we make from that data tends to grow proportionately. That is why we chose the word “glean,” because whatever precious insights we may be able to take away from Actualized performance (Realized Data) increases “little by little,” its full meaning and implication building “slowly.”

12A2c:  Apparent Data Endurance Correlates with Width of Current Period

Whether the width of the Current Period in your Schedules is 5% of Project Length as we have suggested as a default, or some other percent, it remains a constant width—but only as a percent of Project Length. In terms of actual days or weeks, the Imminent Period may be too short or too long to be useful or reliable depending on Project Length. Recall that the Imminent Period, according to our recommended default width, is 4% of Project Length.

For Projects over two years in length, the Imminent Period can challenge the meaning of the word, “imminent.” For instance, on a three-year Project, applying the 4% basis, the Imminent Period would be seven weeks. On a four-year Project, the Imminent Period would be nearly ten weeks! We wouldn’t consider Activities ten weeks away as being “imminent,” would you?

In practice we are not overly concerned, as most long Projects are quite often further subdivided into phases that do not span the entire contract period. For purposes of establishing the Imminent Period, we would recommend treating each phase as a "mini-Project," and applying the 4% basis to each phase, updating accordingly.
The main point we wish to make under this heading is that the wider the Imminent Period, the less credible are short-term Plans for that timeframe. Even though short-term Plans tend to be far more detailed than their Future Segment/Strategic Data cousins, these **Look-Ahead Schedules** can still lose some of their credibility, the farther into the future they attempt to speculate on performance conditions and capacities.

**12A2d: Strategic Data Reliability/Coverage Period Inversely Proportional**

Strategic Data refers to provided information contained in the Future Segment of a Schedule Edition, whereas Predictive Data refers to inferred information for this same Future Period. The purpose of developing Strategic Data altogether is to lay a Plan for future Project Execution. But because the future remains an unknown to us, any attempts to lay Plans beforehand are inherently flawed, to some extent.

That said, we humans have no choice but to make our Plans nonetheless. However, as we do we would be wise to keep in mind that there is a correlation – an inversely proportional one to be precise – between the ultimate utility and reliability of our Plans and the length of Time across which our Plans span. We think you would agree that the Strategic Plans for a short, four-week Imminent Period would have much more credibility than Strategic Plans for a Future Period of 18 months, for instance.

Let us be clear about something: the uncertainty of the future (at least, with respect to Strategic Data) exists in two separate dimensions. In one dimension, we are not able to anticipate the conditions that we might face in the future, or to foresee the extent of our ability to handle those conditions. In the other dimension, this lack of clairvoyance seriously impairs our ability to create credible Plans for future Project Execution.

**12A2e: Predictive Data Credibility/Coverage Period Inversely Proportional**

Of course the same can be said for Predictive Data because Predictive Data is inferred from Strategic Data. Not only does the inferential relationship between these two data sets transfer the unreliability of the Strategic Data to the Predictive Data, such predictions themselves are subject to the uncertainty of life in general. Predicting future outcomes based on current trends makes a bold and unfounded assumption: that the past equals the future. But the very essence of a “surprise” is that things happen that are unexpected, out of the ordinary, departures from repeating patterns or trends. More importantly, though, this assumption assumes that we cannot, and do not, learn from our mistakes and experiences.
12A3: Credibility Profile Explains Project Model Gradual Usefulness

Take a good look at Figure F1202. The red arrow maps the natural progression of Schedule information. It starts with Actual performance, where the best intentions of the Project Executors are “realized” as Work is performed. This information is provided to the record, as Realized Data. Calculations and analyses are performed on the Realized Data and certain conclusions are inferred; these are called Gleaned Data.

In the Current Period, when all Actual Work occurs, Plans are made for the immediate future, which we call the Imminent Period. These Plans are the most detailed that the Project Team will ever see. We recall that these Plans are based on the insights gleaned from Past Segment Realized Data. That is, based on what we have learned from our performance to date, we lay Plans for the immediate future that reflect an appreciation for the limits of our ability to perform.

12A3a: Time Surfing on the Data Date

The comedian GREG FERGUSON has this saying: “tomorrow’s just a future yesterday.” We are intrigued by Time. Does Time move: Does it pass by? Does it fly by? Or does Time simply exist, like the great expanse of the Universe, with we humans instead merely passing through it?

Think of us as surfing through Time with the Data Date being the surfboard. At any given moment, the Data Date exists at some point along the Project Timeline continuum. But then, a little while later, the Data Date has “moved” to the right along that continuum.
What used to be *ahead of us* is instead *upon us* (and, soon, will be *behind us*). What used to be Activities in the Future Segment of the Schedule become Activities upon which we are now working (Current Segment). And with yet another movement of the Data Date, these same Activities will slowly gather dust in the Past Segment.

So this is the great **Catch-22**, the great symbiotic relationship. Why do we say this? Because, for as long as those Activities remain in the Future Segment, we must still be in the Current Segment, even though the “location” of the Current Segment rides atop the Data Date that is surfing across the Project Timeline continuum.

Recall, also, that during each Schedule Update we are advised by Cognitive Project Management protocols to revisit Future Segment Strategic Data. In fact, it is our recommendation that you develop your Imminent Period strategy and revisit your Future Period strategy in the same sitting. Make sure that what you Plan to accomplish in the immediate future supports your long-term Plans, and that your long-term Plans take into account Current Period endeavors that are currently underway or anticipated.

So, the great **Catch-22** is that Future Segment strategy depends on Current Period realities. Yet Current Period realities are very much affected by the Plans for this Current Period — when those same Activities were still Future Segment Activities!

If you are confused by the point we are trying to make, read the preceding paragraphs again. It’s like one of those optical illusion drawings that you have to keep staring at until, suddenly, it all comes into focus.

We’ll take one more crack at it. Let’s say that the Data Date is **June 29th**. As we develop our short-term Plans for the next few weeks, we also revisit the Future Segment Logic and Schedule Content for its sanity. We make a few changes, but mostly we maintain the Logic from the previous Schedule Edition. Let’s say that there are some deliveries that we are concerned may come in later than Planned or needed, but that we *don’t* make any real changes to the Future Segment Logic to mitigate the impact, should those items in fact arrive later than expected.

Slip ahead three months. It is now **September 30th** and we are laying out Plans for the next few weeks. Suddenly we find ourselves grappling with the unwanted effects of that equipment which, in fact, *did* turn out to be significantly late. Now we regret not having taken some mitigating steps three months earlier when we had the chance. And so, Actions three months ago (during *that* Schedule Edition’s Current Segment) have affected Actions we now must take in *this* Schedule Edition’s Current Segment.

**12A3b: The Flow of Essential Schedule Data**

Of the five Schedule Data sets, the ones that are most influential are the Gleaned Data, the Imminent Data, and the Strategic Data.
Predictive Data is not influential, merely speculative.

It is important to our next discussion that you are clear about the relationship between these three key Schedule Data sets. As Figure F1204 illustrates, Gleaned Data is important both to Imminent Data development and to Future Segment Strategic Data development. That is, we would not want to develop either short-term or long-range Project Execution strategies without taking into account what the Realized Data tells us about Actual performance to date. And the lessons learned from the Realized Data are what evolve into the Gleaned Data.

![F1204: Importance of Gleaned Data per Cognitive Project Management](image)

We also don’t want to develop short-term or long-term Plans without considering how they may affect one another. The short-term Plans should align and support long-term intentions, and long-term intentions should pick up where short-term, Imminent Period performance drops off.

12A3c: Essential Schedule Data Gets Better with Time

Now we put all of this into a meaningful context when we recall that the quality of the information within each of these three essential Schedule Data sets gets better with Time.

- Gleaned Data becomes more credible as the track record of Project Performance grows over Time.
- Strategic Data improves as the period over which the strategy is to be applied shortens. That is, as the Project lumbers along, the length of the Future Period gets smaller, which means the window of opportunity for unknowns is correspondingly less. The shorter the period across which we must guess about future eventualities, the better off we are, right?
- Apparent Data depends heavily on Gleaned Data and Future Segment Strategic Data, and it is indirectly affected by the Passage of Time.
12A4: Understanding the Schedule Data Credibility Profile

What all of this means is that our Schedules get better with Time. A visual depiction of what we refer to as the Schedule Data Credibility Profile is shown in Figure F1206. The take-away from a study of this Profile is that we should put the least faith in Schedule Data drawn from the start of a Project and only incrementally develop increasing confidence in our Schedule as the Project unfolds.

![F1206: Schedule Data Credibility Profile](image)

If you think that this means that the Schedule's Predictive Data Set isn't especially reliable until the Project is around 75% finished, you would be correct. As long as humans cannot predict the future, then the correlations between the Passage of Time and the credibility of Schedule Data must prevail.

12A5: The Schedule Data Credibility Profile on Construction Projects

So far, our entire discussion of the Schedule Data Credibility Profile has been based strictly on the nature of data and on the inability of humans to be clairvoyant. Now we need to consider additional factors that are introduced by Construction Projects in particular, and how Past Period performance may not be all that indicative of Project Performance in subsequent Time frames. For this discussion we will imagine an 18-month, high school Project.

- **Baseline Schedule Credibility Profile:**

  Before the Project begins, there is no Past Period Schedule Data. Even the Current Period has no Just Done Data. The Current Period’s Imminent Data are tightly tied to the Strategic Data of the Future Period. It should not be surprising that the first month’s performance is extremely consistent with all Imminent and
Strategic Planned dates. There are no issues to upset the execution of the Plan, and any known issues that do exist at the outset of the Project have been taken into account during Schedule Development.

- **Ignoring Reality; The Challenge of a Belated Baseline Submittal**

Most contracts allow the Baseline Schedule to be submitted as much as six to eight weeks after **NOTICE TO PROCEED**. The requirement (and also the challenge) of the Project Team developing the Schedule (Project Executors and Project Facilitators) is not to take into account any “realities” that have become known in the opening weeks of the Project. The Baseline Schedule must bear a Data Date of **Day 0**, and it must not contain any status. And yet here we are, a month of more into their Project — and we know what we know!

And because the Project has been operating for many weeks without an approved Schedule, it has been “going blind” so to speak. And it may be another couple of weeks before the Schedule is reviewed and approved (best scenario) by the Owner. If it is rejected, it could take another month before the Project has a functional Schedule that it can use.

- **Paperwork Comes First**

On most Construction Projects, and therefore in most Construction Schedules, the preponderance of Activities in the first few months deal with the submittal, review and approval, and ultimate fabrication/delivery of key Long-Lead Items. Consider who is performing these Activities:

- **Submittals** are handled by Home Office personnel for the General Contractor and for the Subcontractors. The responsibility rests with the General Contractor to submit details about the precise materials and equipment he plans to use. Since the majority of Project work is “brokered” (performed by others) these details must come from the Subcontractors. In neither case are we talking about General Contractor or Subcontractor field people; we’re talking about white collar folks back at the Home Office of each organization.

- **Reviews/Approvals** are handled by the Owner or its agent(s): the Architect, the Engineer, the Construction Manager, or the Owner’s Representative.

- **Fabrication/Deliveries** are performed by the vendors and manufacturers of the materials and equipment being procured.

Do you notice who is performing these Activities? Better yet, do you notice who is not? Three primary groups of players are involved in these early
Activities, players that will not be involved in the vast majority of the remaining Activities in the Schedule. Specifically:

- Once the submittals are in, the Home Office folks (of either the General Contractor or the Subcontractors) have next to no involvement on the Project.
- Once the submittals have been reviewed and approved, the extent of Owner involvement throughout the balance of the Project is comparatively less intense, and even then involves a very small percent of Activities in the Schedule.
- Once the materials and equipment are delivered, the vendors and manufacturers become non-players altogether.

So just how meaningful is it if Gleaned Data for the opening month or two indicate that the Project is, or is not, maintaining the pace of the Schedule? What does that tell us about the likely performance of other Project Players who have yet to arrive on the Project?

**Beware the Underground**

When physical activity commences on the Project, it almost always begins with earthwork. Before foundations can be sunk into the ground, the Project site must be “prepared.” This involves “Cut and Fill” operations that level the site by filling in the low spots and cutting down the peaks.

Any time you carve into virgin soil you run the risk of encountering unexpected subsurface conditions. Such conditions might include veins of rock that are difficult to extract or bore through, ancient burial grounds, foundations of forgotten structures, and so forth. Ask any Contractor and he will tell you that “until we are out of the ground, all bets are off.” The corollary to this is that, once out of the ground, things become far more predictable.

Once again we ask: can we legitimately interpolate from the performance of subsurface work as to how future above-ground work will progress?

**Super-Structure and Interiors are Two Different Animals**

Even once we start above-ground Work, many of the Subcontractors who perform the exterior aspects of the Project are not the ones who will be performing the majority of interior work. Grade level foundations (spread and continuous footings, and grade beams), structural steel building framing, metal decking and concrete slabs for elevated floors and roof, exterior windows and doors, exterior walls and finishes, and roofing – all of these are performed by trades that, for the most part, will not be involved after their work is done.
What does Gleaned Data for the super-structure work tell us about how the interior trades will perform in the months ahead?

**Late Deliveries Hit**

At this point we are probably six months into the Project and the building is framed out and enclosed. Some interior rough-in Activities may have commenced a month or so ago, but only now are we beginning to see the same trades working across more than a few Schedule Updates. Only now can we begin to speculate on upcoming performance during the Future Period based on trends identified from Past Period performance (Gleaned Data).

It is usually about this time that the major deliveries start coming in, and according to historical records around 20% of the items will come in later than expected. So now the Project Team must begin to think about how to do things differently in order to mitigate the negative impact of these late deliveries.

**Recovery Schedule and Change Orders**

By the time the Data Date reaches the half-way point in our 18-month Project, progress is beginning to fall seriously behind Schedule. The Owner calls for a revision of the Schedule, one that shows how the Contractor intends to recover the delays incurred to date.

Of course, the Contractor has a different take on why the Project is behind. From its perspective, the Project has been besieged by Owner changes. Maybe no single one of those changes can be blamed for extending the Project Length, but surely the sheer *number* of changes have had a measurable effect on productivity and Planned Activity Sequencing.

So the Contractor wants to incorporate all of these changes in the revised Schedule; not so the revised Schedule will be seen as a “recovery” of Contractor-caused delays, but rather as proof that only through aggressive Acceleration efforts can the Contractor offset the delays caused by the Owner.

Our point is that, before our eyes, the Schedule is transforming into a weapon of war. It is being commandeered by both sides to confirm or validate their respective contractual positions. Once these adjustments start finding their way into the Schedule, just how useful do you think Gleaned Data will be in informing Future Segment Strategic Data Planning?
Finish Line in Sight

Somewhere around the 75% point, the changes have stopped, the deliveries are all in, the Schedule has been revised (maybe more than once), and all parties seem unified in their focus on the finish line. By now a respectable volume of Realized Data is in the record. Perhaps for the first time, the Gleaned Data reflects consistent effort and compares apples to apples, with respect to the nature of the remaining work. Only now are the red arrows of Figure F1204 playing out.

12A6: What GRASP Tells Us about Dominant Project Management

It has been a long and hopefully interesting discussion about the different kinds of data that are found in the typical Construction Schedule. Many important lessons can be learned from the observations we have made. But one more that we suspect may not have hit home is that G.R.A.S.P. helps us understand why Dominant Project Management's approach to Project Time Management doesn’t work especially well on Construction Projects.

12A6a: What is Being Gleaned?

Here is why we say this: take a close look at the kinds of analyses that Dominant Project Management recommends be performed on Realized Data and how that Gleaned Data is actually used. Specifically, Dominant Project Management calls for a comparison between Planned Past Segment effort and Actual Past Segment effort.

As Figure F1208 shows, this Gleaned Data is then used in two ways:

- **Compliance**: The primary thrust is to keep the Project Team in line with the original Project Execution Strategy. Any detected variances between what was Planned and what actually occurred become cause for “corrective actions” to be taken during the Imminent Period. Schedule Variance Analyses, Earned Value Analyses and Total Float Analysis often form the technical assessment for this use of Gleaned Data.
Prediction: The other primary thrust for the Gleaned Data is to predict the Project’s likely temporal outcomes. Earned Value Analysis and Total Float Analysis are the basis for such predictions, too.

Notice that there is no real emphasis on Future Period Strategic Data, and Imminent Period Data is modified solely to “correct” for any deviations identified by the above comparison between Planned and Actual Past Segment performance. The focus is on adherence to the original Baseline Schedule.

Do you spot the fundamental difference in Ideologies?

Dominant Ideology: Under Dominant Project Management, each Schedule Edition’s newly-recorded status is compared with the Baseline Schedule’s Planned performance for the same Activities! Being compared are two different Schedule Editions: the Baseline Schedule and the current Schedule Update. Said bluntly, Dominant spends its energies looking backward.

Cognitive Ideology: Under Cognitive Project Management, rather than obsessing over compliance with the Baseline Schedule, Gleaned Data is valued for what it can tell us about our ability to perform Imminent and Future Period work. Cognitive spends its energies looking forward.

12A6b: What is Not Being Gleaned?

Question: “Are you really surprised when Actual performance doesn’t match the Baseline Schedule?” Didn’t the Schedule Data Credibility Profile sufficiently debunk the idea that early (or even mid-Project) trends are reliable indicators of future performance? Didn’t we establish, rather certainly, that the Schedule is essentially unreliable at the outset, and only slowly moves toward a point of credibility – long after a point in the Project when the window of opportunity to change the ultimate outcome of the Project has mostly sealed shut?

Of course the Actual performance and the Planned performance will not match! But does that mean that the Actual performance is necessarily the culprit; or, could it simply be that the Plan was just a human attempt to predict the future? So we are not surprised when variance analyses between Planned and Actual performance show incongruity.

What does surprise us, however, is that Dominant Project Management dogma often downplays efforts to draw inferences from Realized Data as to how we might differently perform in the future. That is, the critical linkages that we have drawn in Figure F1206 are not commonly discussed in Dominant Project Management literature.
12 Schedule Data Credibility Profile

12A6c: Fixation on Compliance; No Interest in Assessments of Potential

We understand why Dominant Project Management literature obsesses over the slightest variances between Planned and Actual performance. It is because the Dominant Project Management Ideology honors the Newtonian-inspired battle cry: *plan your work, and work your plan.*

For proponents of this thinking, it is all about Command-and-Control. To them it is quite simple: you develop a sound Plan, you monitor for compliance, you call out violators, and you ensure that the Plan is worked.

12A6d: Dominant Project Management May Contain Failure Formula

It should be abundantly clear to you why the Dominant Project Management approach might not work as intended or promoted:

- **First:** It assumes that a credible Plan, developed at the beginning of a long Project, can hold its value for the entire duration of the Project, when the Schedule Data Credibility Profile confirms that such Plans are not reliable for more than a month or two, at most.

- **Second:** It further assumes that the Plan is not only good for the entire Project Length, but that it is presumed to be a flawless Plan (until proven otherwise); one that can, should, and must be followed to the letter.

- **Third:** It assumes that mere compliance with the Schedule is the overwhelming requirement to insure a timely Project completion.

- **Fourth:** It puts next to no emphasis on assessing Project Team capacity as a prelude to mid-Project modifications to the Project Plan.

- **Fifth:** Concerning such modifications, it prohibits any changes to the Project Plan once it is devised, approved, and established as the Baseline Schedule, unless authorized by the Owner.

- **Sixth:** It considers “Schedule Control” as being tantamount to Baseline Schedule compliance. Work performed Out-of-Sequence is considered problematic, and chastised as non-compliance.

- **Seventh:** Summing the above, it forces the Contractor (by Contract) to (a) develop a Plan for the entire length of the Project, (b) follow that Plan even in light of indicators that the Plan may no longer reflect reality, and (c) only adjust the Plan after receiving Owner permission, but not on its own accord.
Eighth: The previous point means that the Contractor is not allowed to make mid-course corrections to the Schedule, but must wait until the disparity between Planned and Actual details is so great that a wholesale rebaselining is mandated.

12B: Making a CPM Execution Model Really Dynamic!

In order for you to fully appreciate what makes a CPM-based Project Execution Model really dynamic, there is one more important observation that we need to make about how the three Schedule Segments interact with one another.

12B1: CPM Schedule is Actually THREE Project Execution Models

Over lunch a few weeks ago, one of the members of the ICS-Compendium Development Team told the story of his first major sales presentation. He had just taken a position as a Scheduling Manager for a mediums-sized construction company. One of his duties in this managerial position was to participate in sales presentations to prospective Owner clients. He would be the fourth and final speaker in a 30-minute pitch as to why the Owner should retain their company as Construction Manager.

In the days leading up to the presentation, the Scheduling Manager wrote, practiced, rehearsed, rewrote, and practiced some more. The Big Day came, and the Sales Team drove three hours from Pittsburgh to the presentation venue. They arrived a little early, and the 45-minute wait in the lobby, as one of their competitors pitched its wares, seemed like hours.

Finally they were on! Their 30-minute presentation went quickly. The president led off with a five-minute spiel about the tributes of the company. Next the Vice President of Marketing discussed commercial advantages of hiring their company. At fifteen minutes into the presentation the assigned Project Manager gave a fairly detailed explanation of how they envisioned approaching the Project’s construction. At 20 minutes into the presentation, the Scheduler Manager had five minutes to explain how the Project Schedule and Budget would be ensured. The Project Manager closed with a five-minute Question and Answer session.

On the ride back to Pittsburgh there was the usual banter among the three senior men. But the Scheduling Manager sat silent in the back seat, almost sullen. Finally, after a short breath of silence, the Marketing VP leaned around from the front seat and said to the Scheduling Manager:

“You did fine, kid.” He paused, and then continued. “You know, every speech you give ... is really three speeches.” The Scheduling Manager looked up, confused. Then, counting on his fingers, the older gentleman continued. “There is the speech you rehearsed. There is the speech you gave. And then, there is the speech you wish you had given!”
The same can be said about the Project Execution Strategy that our Project Schedule struggles to model.

- There is the strategy you plan for the Future Segment.
- There is the Strategy that was Actually executed, as reflected in Past Segment Data.
- And then there is the extemporaneous Strategy that is conceived in the moment based on what is presently known; the Current Period strategy.

Three different Project Execution Strategy depictions: all inside one of Project Execution Strategy model! One could say: three Schedules in one. We say: one Schedule with three distinct Schedule Segments.

What makes a CPM-based Schedule so incredibly dynamic is in how seamlessly these three different Schedule Segments interact with one another. Look at Figure F1210. The jagged borders between the Schedule Segments signify this seamless interconnectedness. The different colors of the Schedule Segments reminds us that:

- Each Schedule Segment contains significantly different data sets.
- Each Schedule Segment’s data content is created at very different points during the Project Timeline continuum.
- Each Schedule Segment serves a different set of informational objectives.

It is easy not to fully appreciate the brilliance of the CPM Project Execution Model. Consider the implications of the above bullets.

- **Future Segment:** Here is an anticipatory plan of attack that is designed to coordinate future action and predict outcomes.
- **Current Segment:** It is symbiotically tied to a much more detailed current plan of attack that, in turn, influences and informs future strategy.
- **Past Segment:** And both Schedule Segments draw insights from the Project Execution that has already transpired!
Somehow, the information from each Schedule Segment flows effortlessly to the next – and even back again. The Critical Path Method of Project Modeling allows Project stakeholders at all stages of the Project Life Cycle to draw guidance from the Project Schedule, as well as to feed valuable information back into it, with equal ease.

This dynamic flow of information only works, of course, if the Schedule is used as designed. That is, the Schedule needs to be maintained in the manner we have informally described thus far as Schedule Updating. In the final section of this chapter we will provide a more finite description of what is entailed in Schedule Updating.

For now, though, please understand that without properly capturing Realized Data about Past Segment Project Execution the development of useful Gleaned Data is impossible. And without meaningful Gleaned Data, any Strategic Planning (be it for an Imminent Period or a Future Period) will suffer great discredit, for having not considered the Project's true ability to deliver on the promises of those Strategies.

12B2: Secret to Most Effective Execution Model: Gleaned Data

This gets to the very heart of how the Project Execution Model works. It manages two vital pieces of information:

- The Project Schedule, *properly maintained*, monitors and measures performance (Realized Data) and establishes the outer limits of Project Execution potential (Gleaned Data).
- The Project Schedule, *properly used*, modulates future Strategic Plans with a sober acknowledgement of Project Execution’s true potential and limitations.

Keep in mind that at the outset of every Construction Project, the real capabilities of the players, both individually and collectively, are largely unknown. The particular combination of Subcontractors and suppliers and consultants is fairly unique to this one Project. How well will this hastily-assembled team work together?

The labor pool is almost always drawn from local sources. For Projects performed in locations foreign to the General Contractor, the quantity and quality of human talent is mostly untested until Project Execution begins.

Finally, the management-level influences of all participating organizations – including the Owner, the Design Professionals, and the Project Executives of the various Constructor organizations – are also unknown factors.

Only after Project Execution commences is it possible to begin to recognize patterns – of competence, perfectionism, craftiness, duplicity, and so forth. We cannot stress enough the supreme importance of Gleaned Data to successfully capturing and assessing performance potential. Without reliable Gleaned Data, the Project Schedule is rendered
mostly useless.

12B3: GRASP the True Meaning of a Dynamic Project Schedule

This book is about CPM as a method of modeling Project Execution Strategy. This chapter has been philosophical, which may seem to be at odds with the title, CPM MECHANICS. How is any amount of philosophy appropriate for a book on the dry mechanics of a technical application?

The answer is that the main point of this book is about how the Critical Path Method can be used to model Project Execution Strategy. It’s the mechanics of the model that helps us to appreciate the potential as well as the limits of the model to guide Project Execution.

Of course those last three words, “guide Project Execution,” may be an unconfirmed assumption. That is, we assume that a major if not primary reason for creating a model is to guide Project Execution. But as we just noted, the Dominant Project Management application of the Critical Path Method as a model of Project Execution seems almost entirely fixated on performance measurement, monitoring, and compliance — and much less on guiding future work in any direction other than back to the Origin Plan.

We hope that you now appreciate the starkly different notions as to why a Project Execution Model is needed. As you get out into the world of Construction Project Management, you will be confronted by two very different centers of thought. There will be your technical colleagues, those who have been schooled in Dominant Project Management thinking. They will be expecting you to create Schedules that can be used to monitor performance and force compliance with a pre-conceived Plan. This is the world of Project Control.

But you will also encounter the Project Team, those who care mainly (if not only) about finishing the Project on Time and within Budget. These folks will harbor a disdain for the Schedule, because:

- It is like the classroom tattle-tail that shadows and reports on their every move.
- It is too often out of touch with reality, and therefore isn’t trust to guide the work!

We hope that Chapter Twelve has given you a different understanding of the Project Schedule — as a model of Project Execution Strategy, one that is malleable, responsive and, above all else, helpful!

Now you fully G.R.A.S.P. the value of an utterly dynamic, multi-segmented, entirely interactive, Network-Based CPM Schedule that is designed and built to model Project Execution Strategy — before, during, and after the fact. That is the true meaning of a dynamic Project Schedule!
CHAPTER THIRTEEN

13A Understanding Schedule Updates
13A1 Schedule Updating Process; in General
13A1a Past Segment Information Modifications
13A1b Current Segment Information Modifications
13A1c Future Segment Information Modifications
13A2 Schedule Updating Considerations; in General
13A3 CPM Terms Associated with Schedule Updating
13A3a Understanding Activity Percent Complete
13A3b Understanding Activity Remaining Duration

13B Data-Adjusting Calculations during Schedule Update
13B1 Two Essential Variables Important to Schedule Updating
13B1a CPM Calculations Involving Activity Remaining Duration
13B1b CPM Calculations Involving Activity Percent Complete
13B2 Understanding Out-of-Sequence Work
13B2a What Does It Mean to Be Working Out-of-Sequence?
13B2b Does Working Out-of-Sequence Hurt the Schedule?
13B2b-i Activity Paths with Different Total Float
13B2b-ii When Skipped/Performed Work Reside on Same Path
13B2b-iii When Skipped/Performed Work Reside on Different Paths
13B2c Is Working Out-of-Sequence a Bad Practice
13B2d Powerful Software Setting: Progress Override
13B2d-i Understanding the Progress Override Software Setting
13B3 How to Interpret a Schedule Update
13A: Understanding Schedule Updates

This will not be an especially detailed or complex chapter because the various processes involved in maintaining a Project Schedule are covered elsewhere in the ICS-Compendium where the Critical Path Method of Scheduling a Project Execution Strategy is discussed in far more detail. But in order for this chapter to make sense to you – especially the section on the mechanics of Schedule Updating calculations – we think you should have a general understanding of the processes involved in maintaining a Project Schedule.

13AI: Schedule Updating Process, in General

The primary purpose behind Schedule Updating is to keep the Project Schedule valid, as a Project Time Management tool. According to Dictionary.com, “valid” has several meanings that help us understand what steps are required for a Project Schedule to remain useful and reliable as a Project Time Management tool:

- **Valid**: Sound; just; well-founded
- **Valid**: Producing the desired result; effective
- **Valid**: Having force, weight, or cogency; authoritative.
- **Valid**: Legally sound, effective, or binding; having legal force
- **Valid**: (Logic) so constructed that if the premises are jointly asserted, the conclusion cannot be denied without contradiction.

Aren’t these all attributes we want in a Project Schedule?

- Don’t we want our Project Schedules to be “sound” and “well-founded?”
- Don’t we want them to produce the “desired results” that we expect from a Project Execution Model?
- Don’t we want them to be “effective?”
- Don’t we want their contents and conclusions to carry some “weight,” and be a “force” of persuasion and guidance in their application?
 Aren’t we demanding “cogency” – meaning that our Schedules will be “convincing or believable by virtue of forcible, clear, or incisive presentation?”

Finally, don’t we want our Schedules to be “so constructed” that, if their contents are taken as a whole (Logic, Activity Durations, Date Constraints, Software Settings, and so forth), any predictions, recommendations, and guidance emanating from the Project Schedule “cannot be denied without contradiction?”

Don’t we want our Project Schedules to be, and remain, valid throughout the Project Life Cycle? Well, in order to remain valid, they must accurately reflect the Past and reasonably anticipate the Future. If we go back to our three Schedule Segments and their five categories of Schedule Data, we can precisely identify the areas of information modification required during a typical Schedule Update.

This may seem all too obvious, but basically a Schedule Update adjusts the Project Schedule to:

- Reflect what has taken place
- Incorporate relevant new information is now available to the Project Team
- Determine how those two categories of information change future Plans, if at all.

Hopefully you have spotted the connection between these three areas of information modification and the three Schedule Segments that comprise every Schedule Edition.

13A1a:  Past Segment Information Modifications

Under the heading of Realized Data, during a Schedule Update we want to record details about whatever has taken place during the reporting period.

- For Activities that have been started, we want to record the Actual Start Date.
- If the Activity was completed during the reporting period, we would also record the Actual Finish Date.
- If there are special notes or other annotations concerning the performance, we might enter commentary in an Activity Log data field.

Past Segment information is limited to recording what was accomplished.

13A1b:  Current Segment Information Modifications

The focus within the Current Segment of the Schedule is to understand what the Project Team is currently involved with.

- Sometimes, during a reporting period, an Owner adds or deletes Contract Scope, relative to what had been the formal Scope during the previous reporting period.
Whether adding or deleting, the changes in Scope usually result in changes to Schedule Content. Activities may be added or removed, and Activity Durations may be increased or decreased. Restriction Linkages between Activities may need to be altered as well.

☐ Not all Activities are started and finished in a single reporting period. This means that some Activities may be underway (incomplete) as of the Data Date. For these Activities, it is important to (a) determine the extent of completeness (called Activity Percent Complete) and (b) to revisit the estimate of Time needed to complete the Activity. The latter process involves estimation of what is called an Activity Remaining Duration for the uncompleted portion of the Activity.

☐ In the last chapter we discussed Imminent Period Schedule Segment and described it as a tactical plan of attack for near-term Project Execution. We specifically identified the LOOK-AHEAD SCHEDULE as the most popular work product of this short-range Tactical Scheduling. Whether such Tactical Scheduling is part of Schedule Updating or merely done at the same time as the Schedule Update is perhaps a semantic distinction.

In one sense, the LOOK-AHEAD SCHEDULE is a Level 4 document that is based on the Level 3 Schedule. As such, it is not part of the Schedule Update, although it surely benefits from the new information provided by the Schedule Update. Yet as we have also noted several times, Imminent Period Tactical Scheduling should be done in conjunction with Future Period Strategic Planning. And that is where things get a bit gray.

13A1c: Future Segment Information Modifications

We approach each Schedule Update with the Future Segment of the Schedule already populated with Activities and Logic. During the Schedule Update, some of the Future Segment Activities relocate to the Current Segment (a function of Data Date surfing). As for the remaining Future Segment content, the portion that does not make this migration, the key question is whether we should assume that the Activities and Logic of the Future Segment will remain as rational and cogent as they were when first conceived during initial Schedule Development, or whether they warrant adjustment?

Experience has taught us that the actions or inactions of the Past directly affect the workload left for the Future. Accordingly, Cognitive Project Management encourages us to revisit the Schedule Content of the Future Segment as part of the Schedule Updating process, and refers to this revisiting of Future Schedule Content as Project Execution Strategy Recalibration.

Each Schedule Update should include a Recalibration of the Logic within the Future
Segment, based on what has been accomplished to date (Realized Data), what this tells us about our performance capacity (Gleaned Data), and what performance is Planned for the Imminent Period (Apparent Data). Since Recalibration includes adjustments to the Future Segment (Strategic Data), any changes to Schedule Updating variables might reasonably be expected to impact Project outcomes (Predictive Data).

For the previous paragraph doesn’t make a lot of sense to you, then flag it and come back to it after you complete the final section of this chapter. It will make complete sense by then, for sure.

13A2: Schedule Updating Considerations, in General

From all of this we can easily recognize the essential factors that have prominent roles to play during a Schedule Update. They are:

- Identifying Activities started and/or finished during the reporting period.
- Determining and recording Actual Start and Actual Finish Dates.
- Determining the Percent Complete of Activities started, but not yet finished.
- Determining Remaining Durations of Activities started, but not yet finished.
- Revisiting Future Segment Schedule Content and Recalibrating as warranted.
- Devising a short-term Project Execution Strategy for the Imminent Period

13A3: CPM Terms Associated with Schedule Updating

In recent pages we have used some new terms, like Activity Percent Complete and Remaining Duration. Before we go much further, we should make sure that we all have the same understanding of what those terms mean.

13A3a: Understanding Activity Percent Complete

The meaning of Activity Percent Complete may seem obvious to you, but there is a subtle distinction in its meaning that we need to highlight. An Activity may have a different Percent Complete for Costing purposes than for Scheduling purposes. Consider two different examples.

- **Example 1:** An Activity with the description, **Set/Install Boiler**, has a **two day** Duration. All that is involved in “setting” the boiler is simply to lift it off of the truck that brought it to the job site, and set it down on a concrete pad that has been prepared and is waiting for it. The “installation” portion of the Activity includes the connection of mechanical pipes and electrical conduit to the boiler. The boiler itself costs **$7,500** and the contract between the Owner and the Contractor
prohibits the Contractor from billing for the physical equipment until it has been delivered to site and set in place.

Let us assume that on the last day of the reporting period the boiler was delivered to site, offloaded from the truck, and set onto the pad. But let us also assume that none of the piping or conduit work was performed. What Percent Complete is the Activity? The “setting” portion of the Activity took less than 30 minutes! The minimum Duration unit is one day (i.e., we are not Scheduling in hours).

If we show the Remaining Duration at one day, then the corresponding Percent Complete would be 50%. If we show the Remaining Duration as two days, then the corresponding Percent Complete would be 0%. If the claimed Percent Complete is 0%, how can the Contractor take credit for the equipment delivery, since the Activity contains the word “set” and the Activity is 0% complete?

Example 2: We are building a gymnasium at an operating high school. One of the site work Activities involves running permanent power from a nearby electric company substation to a panel box on the side of the building. We have an Activity entitled, EXCAVATE/LAY/BACKFILL POWER SUPPLY TO BUILDING and it has a two day Activity Duration.

Notice that the first part of this Activity is the matter of excavating a small trench across the site into which the conduit will be laid. What is not referenced in the Activity Description is the need for a surveyor to “stake out” where the POWER SUPPLY LINE is to be located. It costs money to call out a surveyor each time one is needed on site. So a common practice is to “save up” a few surveying tasks until it justifies calling the surveyor out for a day of Work.

Now, suppose that the surveyor is on site for six such small surveying tasks, one of which is to stake out the POWER SUPPLY LINE. The surveyor invoices the Contractor $2500 for the day’s work. The Contractor wishes to recover this cost, but on any individual Activity, the surveying effort is (a) not even mentioned, and (b) even if allowed by the Owner, is only a fraction of the Activity Duration. What Percent Complete would you assign to the POWER SUPPLY LINE Activity, and to the other five similar Activities in order to be able to invoice for the surveyor’s charges?

What we are discussing is the relationship between what is recorded (as to Performance Execution) and how that information is used for Costing purposes. From a Scheduling perspective, either Activity is barely started. Yet from a Costing perspective, perhaps as much as 50% or more of the Activity’s dollar value has been “earned.”
Fortunately, most Scheduling Software programs allow the Activity Percent Complete and the Activity Remaining Duration to be “unlinked” from each other. By designating the Percent Complete to the Costing use, and the Remaining Duration to the Scheduling use, we can “claim” a higher Percent Complete than would be derived by comparing the Remaining Duration to its Original Duration.

And so, in the first example, we would report 75% complete, in order to recoup the cost of the boiler, yet report two days Remaining Duration for Scheduling purposes. In the second example, we would report 5% complete to recover the Surveyor’s cost, yet keep the Remaining Duration at two days for Scheduling purposes.

13A3b: Understanding Activity Remaining Duration

We have been using the term Activity Remaining Duration, but without really explaining what it is or how it is used. We certainly have not encountered it in any of the scores of figures throughout the book up to this point. That is because a Remaining Duration only comes into play on an Activity that is underway (started, but not yet finished). We would only encounter in-progress Activities during a Schedule Update, and almost always in the Imminent Period of the Project.

Per the ICS-Dictionary, a Remaining Duration is:

\[ \text{Remaining Duration: A Remaining Duration is an estimate of the amount of Time (in Continuous Crew Days) that will be required to complete that portion of the Activity Work Scope that has not yet been completed.} \]

The same thinking and basis that went into the estimation of an Original Duration goes into determination of an Activity’s Remaining Duration.

\[ \text{The only Duration you have encountered thus far in this book has been an Original Duration, which is an estimate of the Continuous Crew Days required to perform the entire Work Scope of the Activity.} \]

13B: Data-Adjusting Calculations during Schedule Update

At last it is time to examine CPM calculations associated with a Schedule Update. You will be happy to discover that there are no new, earthshaking concepts to learn. We will still be working with Earliest Dates, Latest Dates, Total Float, Free Float, and Path Tracing (including the Critical Path). As you will soon see, it is not the calculation processes that are significant to our examination, but rather the interpretation of those calculations.
Our discussion will begin with an examination of the two most essential variables when it comes to Schedule Updates: the Activity Remaining Duration and the Activity Percent Complete. Next we will consider a common situation where Work has been performed Out-of-Sequence, and how such departures from Schedule Logic might affect CPM calculations. Finally we will discuss nuances to Float calculations and Activity Path determination as a result of Schedule Updates.

13B1: Two Essential Variables Important to Schedule Updating

We begin by looking at an Activity Duration value that we had no need to speak about before now: the Activity Remaining Duration.

In truth, every Schedule Activity has a Remaining Duration, even Activities in a Baseline Schedule that contains no recorded progress. Of course, in such a Baseline Schedule the Remaining Duration always equals the Original Duration.

13B1a: CPM Calculations Involving Activity Remaining Durations

We will start with a simple example of how to incorporate Remaining Durations into Forward Pass and Backward Pass calculations. Take a look at Figure F1302, which shows three Activities, linked through Default Restrictions. In this example, the Forward Pass begins with a Start-No-Earlier-Than Date Constraint of Day 100. The Earliest Dates are calculated with ease and culminate in an Earliest Finish for Activity C of Day 122.

The Backward Pass starts with a Finish-No-Later-Than Date Constraint of Day 140 and concludes with the determination that Activity A has a Latest Start date of Day 118. Simple subtraction calculates a consistent Total Float value of TF +18 across all three Activities. So far, no surprises and nothing complicated, right?
Okay, now we need for you to imagine that it is ten days later and the Project Manager has asked us to perform a Schedule Update. The first thing we do as a Project Facilitator is ask the Project Executor to tell us what progress has been made. In response, the Project Manager sends us the email shown in Figure F1304.

---

**Mark Up Logic Diagram:** The first thing we do is mark up the previous Logic Diagram. Follow along in Figure F1306.

- **Step A, Eliminate Date Constraint:** We scratch out the Start-No-Earlier-Than Date Constraint because it no longer applies or matters.
**Step B, Add Actual Dates:** This is where we write in the Actual Dates that were provided, whether explicitly or implicitly. We are told that Activity A finished on Day 105 so we write Day 105A, with the “A” indicating an Actual Date.

Since we are told that Activity A took three consecutive days, then it must have started on Day 103. Since we are using an End-of-Day Perspective, we write in Day 102A. We are also told that Activity B started on the next day after Activity A finished, so we use Day 105A for the Actual Start of Activity B.

**Step C, Change Data Date:** We set the Data Date to Day 110, ten days since the previous Schedule Update.

**Step D, Remaining Durations:** Finally, we record the Remaining Durations for whatever Activities saw progress during the reporting period. Activity A is reported “finished” so its Remaining Duration is set to 0. As for Activity B, we are told that there are seven days of Continuous Crew Days left. We scratch out the 10 and replace it with 7.

Notice that we have blanked out the Total Float squares wherever Actual Dates have been recorded. Once an Activity has finished, it no longer factors into the calculation of Total Float. As for Activities that are underway, the start is no longer a factor to Total Float calculations, but its Earliest Finish is.

**Forward Pass/Backward Pass Calculations:** At last we are ready to perform the Forward Pass and Backward Pass calculations. Follow along in Figure F1308.

- The Forward Pass begins with the Data Date. Activity A is finished, so it does not participate in the Forward Pass calculations. As for Activity B, we
see that it has already begun, and that the Project Manager estimates another **seven days** to complete this Activity. What is the earliest that the remaining portion of **Activity B** can commence?

- Why, of course, the Data Date!! We cannot go back in Time (as much as we would like to). So the earliest that the remainder of **Activity B** can begin is **Day 110**. And by adding the Remaining Duration of **seven-days** to the Earliest Start, we get an Earliest Finish of **Day 117**. We complete the Forward Pass at **Activity C** with Earliest Start of **Day 117** and Earliest Finish of **Day 124**.

- The Backward Pass is fairly straight-forward. We have received no notice of any changes to the Deadline for this Project, so we continue to honor the Finish-No-Later-Than Date Constraint of **Day 140**. We perform our Backward Pass as we would normally. The only difference (with this being a Schedule Update) is the calculation a Latest Start for the remaining portion of **Activity B**. It should seem logical to you, but since the Latest Finish of **Activity B** is **Day 133** and the Remaining Duration for **Activity B** is given as **seven days**, then the Latest Start for the remainder of **Activity B** is **Day 126**.

**Total Float Calculations:** Total Float calculations follow the standard formulas: the difference between Latest Start and Earliest Start yields Start Total Float, and the difference between Latest Finish and Earliest Finish yields Finish Total Float.
All that remains is for us to analyze the results. We notice that Total Float has dropped by two days, from TF +18 in the previous Schedule Edition to TF +16 in the current Schedule Edition. Can you figure out why this is so?

Here’s the answer. Take a look at the progress that was reported as “done.” All five days of Activity A were completed, right? And how many days of Activity B’s ten days were completed? Well, if seven days remain, then they must have performed the equivalent of three days. So, in total, eight days of work were performed in a period of ten Workdays. The difference, two days, has consumed (reduced) available Total Float.

Now it is time for you to perform a Schedule Update on your own. Using Figure F1310, figure out whether the Project is falling behind Schedule or picking up Time on the Schedule? We also include Figure F1312 which shows the same Schedule from the previous Schedule Edition. Our solution is in Figure F1314.

13B1b: CPM Calculations Involving Activity Percent Complete

There is really nothing mysterious about using Activity Percent Complete. Unless informed otherwise, we simply assume that Activity Remaining Duration and Activity Percent Complete are correlated with one another. For instance, if we have an Activity with a Duration of 12 days, and we are told that its Activity Percent Complete is 25%, then that means that three days of the 12 days have been completed and that the Remaining Duration is nine days.
13B2: Understanding Out-of-Sequence Work

We are about to immerse ourselves into another philosophical discussion, this time about a performance condition that is the subject of much heated debate among seasoned Project Time Management practitioners: Out-of-Sequence work. While this discussion will surely make you much more aware of the issue itself, that is not our main reason for engaging in it. Rather, we hope that by following this discussion your overall understanding of CPM as a Project Modeling Methodology will be more rounded.

13B2a: What Does It Mean to Be Working Out-of-Sequence?

What exactly do we mean by Out-of-Sequence work? Simply put, it refers to work that is performed in a Sequence that does not follow or honor the Logic in the Schedule as Figure F1316 shows us. In the Upper Panel we see that Activity B was started before Activity A finished, even though the Restriction Linkage between the two Activities is a Default Restriction. We would say that Activity B started Out-of-Sequence.

Now let’s look at the details. Activity A had an Original Duration (OD) of eight and, according to the annotations in the illustration, five of those eight days were performed. As a result, Activity A has a Remaining Duration (RD) of three. The details for Activity B are: OD=5, RD=2. Activity B must have performed three of its five days. Between them, eight days of work were performed.
13B2b: Does Working Out-of-Sequence Hurt the Schedule?

What we are asking is whether by working Out-of-Sequence the Deadline Milestones are necessarily put into jeopardy? Pushed for a Yes-or-No response, we would have to say “No,” as we point to the word “necessarily.” The better answer is, “It depends: Working Out-of-Sequence might threaten or even impact downstream deadlines; but it might not, either.”

13B2b-i: Activity Paths with Differing Total Float

It probably should go without saying that it of course depends on whether the involved Activities enjoy sufficient positive Total Float as to whether they “might threaten or even impact downstream deadlines.” Since this understanding should be well ingrained in your mind at this point, the balance of this subsection will deal with another condition that could cause a downstream Deadline Milestone to be threatened or missed due to work performed Out-of-Sequence: what Activity Paths are involved.

Pay special attention to how we word this condition: that the skipped work and the performed (Out-of-Sequence) work do not reside on different Activity Paths. Notice that we are not testing whether the skipped and performed work are on the same Activity Path – but rather that, at any point, they are not sharing a common Activity Path.

This distinction matters because different Activity Paths typically have different Total Float values. And so, if the Activity Path where the work is skipped has a lower Total Float value than the Activity Path where the work is performed Out-of-Sequence, the latter may not offset the former. This condition is what the next few paragraphs are all about.
13B2b-ii: When Skipped and Performed Work Reside on the Same Path(s)

Let’s start with what we are given in Figure F1316. In the **Lower Panel**, we show what the progress would have looked like had the work been performed in-sequence, i.e., according to the originally designed Logic. The **eight days** of work would have been concentrated on the first Activity in the series, **Activity A**. **Activity A** would have been completed, and **Activity B** would not have started.

Now look at Figure F1318, where we do a quick Forward Pass on both scenarios shown in Figure F1316. Is there any difference in the Earliest Finish of **Activity B**? There isn’t, and that is because both the skipped work and the performed work reside on the same Activity Paths.

☐ In the **Lower Panel** we perform a Forward Pass. **Activity A** is finished, so it has neither an Earliest Start nor an Earliest Finish. The first Activity with work remaining to be performed is **Activity B**, which has not yet begun. The Earliest Date on which it could start is the Data Date, which is shown to be **Day 110**. Since **Activity B** has an Original Duration of **five**, and since it has not been started yet, its Remaining Duration is also **five**. Adding the Remaining Duration to the Earliest Start (**Day 110**), we derive an Earliest Finish for **Activity B** of **Day 115**.

☐ Now we will apply the same approach to our Forward Pass of the **Upper Panel**. The first Activity with work remaining to be performed is **Activity A**, which has started but not yet finished. The Earliest Date that the Remaining Duration of **three** can be performed is the Data Date, which is shown to be **Day 110**. Therefore, the Earliest Start for (the remaining work of) **Activity A** is **Day 110**, and the Earliest Finish is **Day 113**.

Next, we see that **Activity B**, which has an Original Duration of **five days** has been started Out-of-Sequence, and has a Remaining Duration of **two days**. Based on the Earliest Finish of **Activity A**, we set the Earliest Start of **Activity B** also to **Day 113**. By adding the Remaining Duration of **two days** to the Earliest Start (**Day 113**), we derive an Earliest Finish for **Activity B** of **Day 115**.

We realize that you most likely saw the obvious when you first compared the two panels. In either case, **five days** of work remained. It didn’t matter whether that remaining work was within a single Activity, or across two Activities; either way, **five days** of work remained to be performed. With an Earliest Start equal to the Data Date (**Day 110**), either way the remaining work would not finish earlier than **Day 115**.
13B2b-iii: When Skipped and Performed Work Reside on Different Activity Paths

We are now going to study and compare three different scenarios with respect to six Activities that reside on two different Activity Paths. We’ll start with the scenario depicted by the Schedule Edition prior to the current Schedule Update. Figure F1320 shows two Activity Paths, as follows:

- **The Upper Path**: Activity A ► Activity B ► Activity C ► Activity F
- **The Lower Path**: Activity A ► Activity D ► Activity E ► Activity F

We immediately recognize that Activity A and Activity F reside on two different Activity Paths, although not the same one in all four instances. Activity B and Activity C exclusively reside on the **Upper Path**, while Activity D and Activity E exclusively reside on the **Lower Path**. The **Upper Path** is the Critical Path with reported Total Float of **TF +0**. The **Lower Path** enjoys positive Total Float of **TF +4**.

Next we are going to compare two different scenarios, each in which work is performed Out-of-Sequence. In the first scenario, depicted in Figure F1322, the Activity Path upon which the Out-of-Sequence work is performed is different than the Activity Path where the skipped work resides.

Let’s take a close look at Figure F1322. For starters, where is the skipped work located? Well, that depends on where the Out-of-Sequence work is located … because the skipped
portion is any Activity or Activities (partial or entire) that, according to the prevailing Logic, should have happened before the work that was performed Out-of-Sequence happened. We immediately spot the Out-of-Sequence work, four days of accomplished work at Activity E. We know that it is four days because Activity E’s Original Duration is seven days and its Remaining Duration is three days.

So, now that we know where the Out-of-Sequence work is located, what do we consider to be the skipped work? We follow the Logic backwards from the Out-of-Sequence work. First to be encountered is Activity D which, according to the Default Restriction between it and Activity E, should have finished before Activity E began. Next, we see that in order for Activity D to even commence, Activity A had to be finished. So, the skipped portion of work is (a) the uncompleted portion of Activity A (two days), and (b) the entire portion of Activity D (four days). In total, six days of work were skipped.

Let us pay attention to which Activity Path(s) the skipped and Out-of-Sequence work reside on. It is clear that the Out-of-Sequence work resides exclusively on the LOWER PATH. As for the skipped work, the portion within Activity B also resides exclusively on the LOWER PATH. But as for those last two days of Activity A … well, they reside on both Activity Paths simultaneously.

Here is why this is such a significant distinction. It is because this means that, with respect to the UPPER PATH, there is a loss of Time (the skipped work) without an offsetting recovery of Time (the Out-of-Sequence work).

Before we move to the third scenario, take a more in-depth look at the Total Float values
for the two Activity Paths, and make sure you really understand what is happening. You will recall from Figure F1320 that the Upper Path came into this Schedule Update with Total Float of \( TF +0 \). Now we see that it has turned negative, with Total Float of \(-4\). Why is this so?

Well, the meaning of Total Float of \( TF +0 \) was that the Upper Path had no Time to spare with respect to meeting the Finish-No-Later-Than Deadline of Day 131. How much work was performed on the Upper Path during the reporting period? Well, six days of work were performed on Activity A, but that is all! No other work was performed on the Upper Path, and so – during a ten day reporting period only six days of work were accomplished. That means that four days of available Workdays were squandered. The Upper Path loses four days – and since it had no Time to lose, it is now four days behind Schedule.

And what happened down on the Lower Path? Well it came into the current reporting period with four days of positive Total Float, and it seems to be holding onto all of that Total Float! How so? Well, how many days of work were performed on the Lower Path? We have those same six days from Activity A, and we have four more days performed, albeit Out-of-Sequence, on Activity E. Combined, ten days of work were performed on the Lower Path across a ten day reporting period. No days were squandered and therefore the Lower Path’s Total Float holds steady at \( TF +4 \).

Now we are ready to consider the third scenario, as illustrated in Figure F1324. The only difference between this and the previous scenario is the location of the Out-of-Sequence work. Instead of four days of Out-of-Sequence work occurring in Activity E (on the Lower Path), those same four days of Out-of-Sequence work were performed in Activity C (on the Upper Path).

This means that the skipped work and the Out-of-Sequence work share the same Activity Path – and, therefore, that the loss of Time (skipped work) is offset by the recovery of Time (Out-of-Sequence work) on the same Activity Path! Let’s confirm that this is true.

In Figure F1324 we see that the Upper Path still reports Total Float of \( TF +0 \), the same amount that it had in the prior Schedule Update (see Figure F1320). This is because the Upper Path saw ten days of work performed within a ten days reporting period. Activity A saw a six-day reduction in its Remaining Duration, and Activity C reported another four-day reduction in its Remaining Duration.

But now look at what happened to the Lower Path. Coming into this Schedule Update you will recall that the Lower Path had four days of positive Total Float. Now it appears to have lost all of that spare Time and is now also reporting Total Float of \( TF +0 \), just like the Upper Path shows.
F1324: How Out-of-Sequence Really Works

Why is this so? Well, how much work was performed on the Lower Path? We know about the six-day of Activity A but … that is all! No other work was performed on the Lower Path. And so, even though ten days were available to the Lower Path, only six-day of work were accomplished. The Lower Path lost four days, so its Total Float dropped from $TF +4$ to $TF +0$. And now – both Activity Paths are critical!

So there you have it: two different scenarios, where the only difference between them is the location of Out-of-Sequence work with respect to the skipped work. Now do you understand why we said, “It depends?”

13B2c: Is Working Out-of-Sequence a Bad Practice?

Before we leave this topic of Out-of-Sequence work, let us ask you if you think that performing work Out-of-Sequence is a bad practice altogether? To be clear, we are asking a slightly different question than the one we just discussed. Before, we were considering whether the performance of Out-of-Sequence work necessarily threatens the Timely achievement of a Project. Here, we are asking whether working Out-of-Sequence ought to be viewed as a “bad practice.”

We realize that such a question – and answer – probably falls beyond the scope of a book on the mechanics of the Critical Path Method of Project Modeling. But we will entertain the question, nonetheless, (a) because some students of this book may not go on to read other volumes of the ICS-Compendium where this topic is dealt with in greater depth, and (b) because, as you will soon learn, like so many other aspects of Project Time
Management, the conventional teaching sends mixed messages about the function and purpose of Critical Path Method of Project Modeling.

Our answer to the question is quite simple and maybe innocent as well: if it doesn’t do any harm to the Schedule, and if it may even improve the Project outcome, why should it be outlawed? Now we appreciate one of the main arguments against ignoring prevailing Schedule Logic: if everybody ignores the Schedule’s pre-negotiated approach to the work, we will end up with chaos!

Fair enough! In deference to this argument, we would insist that all Out-of-Sequence work be preceded by careful thought and deliberation, and be approved by the Project’s key stakeholders. That said, we nonetheless take the position that under certain circumstances, not only is Out-of-Sequence work not harmful, it can be fully beneficial to the Project. Moreover, some contract language may even be interpreted as requiring Out-of-Sequence work. [1]

Return to Figure F1320. Suppose that, during the performance of work on Activity A, two pieces of information come to the attention of the Project Executors. First, they are told that the Architect is currently working a design change that, if approved, would require removal of work performed in Activity A. Second, they learn that equipment to be installed in Activity B will be arriving later than Planned? In terms of effective Project Time Management, what – if anything – should the Project Executors do?

For starters, all would agree that they should immediately halt work on Activity A, since they may just have to rip out whatever they are putting in. But then what? With respect to the Upper Path (which, you will recall, came into the current reporting period with Total Float of TF +0), should the Project Team perform Out-of-Sequence work on Activity C (remember: they can’t work on Activity B due to the late delivery of equipment)? Or should they simply abandon the Upper Path and put double attention to the Lower Path?

Even if they double their efforts on Activity D and Activity E, and finish both – will this better the Schedule with respect to the timely completion of Activity F? Of course not! Our recommendation, if asked, would be to perform as much Out-of-Sequence work on the Upper Path as possible, in order to offset the expected delivery delay.

13B2d: Powerful (Potentially Harmful) Software Setting: Progress Override

It is now time to discuss a very powerful Scheduling Software Setting that has the potential to dramatically reduce the quality of Current Segment and Future Segment

1 Many contracts require the Contractor to “mitigate delay” whenever possible. For example, if a Contractor learns that a critical piece of equipment will be arriving several weeks later than Planned (and needed), he may elect to perform other work (on the same Activity Path) in an attempt to offset the loss of Time that the delayed delivery is expected to cause.
Schedule Data – in particular any Predictive Data or Strategic Data.
We are referring to a toggle setting provided by various software programs that takes affect when work is performed Out-of-Sequence. In the following discussion, we will adopt P3’s names for these two settings: Retained Logic and Progress Override.

13B2d-i: Understanding the Progress Override Software Setting

As the name suggests, the effect of invoking the Progress Override Software Setting is that it changes the rules with respect to performing Forward Pass and Backward Pass calculations. In essence it neutralizes any Logic that had been disregarded when work is performed Out-of-Sequence.

F1326: Applying Progress Override

Figure F1326 applies Progress Override to the Schedule Update that we last saw in Figure F1322. In this example, work was commenced on Activity E before work on Activity D was completed; thus, violating the Default Restriction between them. Let us use this simple example to point out a couple of things about Progress Override.

☐ As noted two paragraphs back, the Progress Override Software Setting rewrites the rules pertaining to consideration of previously established Logic that, by virtue of the Out-of-Sequence work, seems no longer to be valid. After all, if Activity E commenced work before Activity D was completed, then the Default Restriction between them must not have been a very firm one. Notice at Circle C that the Default Restriction between Activity D and Activity E has been “overridden.”
Next we must appreciate the direct consequence of this dismissal of the Logic Tie. At Circle B, we see that now Activity D has no Restricted Activity. This would mean that Activity D’s Latest Finish would now be set to equal the end of the Project, or Day 131! Conversely, at Circle C we see that Activity E has no Restricting Activity, and so its Earliest Start drops back to the Data Date, Day 110.

The net effect of the Progress Override is that it has cut what used to be our Lower Path into two separate Activity Paths. We can call these two new Activity Paths, the Middle Path and the Bottom Path. The Middle Path is comprised of Activity A ► Activity D; and the Bottom Path is comprised of Activity E ► Activity F. Notice that the Middle Path does not connect with Activity F in any way whatsoever!

Because the Middle Path goes directly to the end of the Project, Activity D (the final Activity in the Middle Path) has a Latest Finish of Day 131. It continues to report the same Earliest Finish as it did before, Day 116. As a result, the Middle Path enjoys Total Float of TF +15!

As for the Bottom Path, while Activity E remains the Restricting Activity to Activity F, and therefore has a Latest Finish of Day 123 – reflective of Activity F’s Latest Start of the same Date – Activity E has no Restricting Activity. This means that the balance of its work can commence on the Data Date, Day 110. (Circle A) As a result, the Bottom Path has Total Float of TF +10.

Of course nothing has changed with the Upper Path, which remains the Critical Path, one bearing four days of negative Total Float.

The opposite toggle setting for Progress Override is called Retained Logic, the preferred setting by the majority of Project Facilitators. Set to Retained Logic, the software would continue to honor Logic Ties that were overlooked when the work was performed Out-of-Sequence. Return to Figure F1322, which shows Forward Pass and Backward Pass calculations with the Retained Logic setting invoked. Notice that there are no Logic disconnects, as we saw when the Progress Override setting was chosen. More importantly, though, notice that the Total Float for Activity D and Activity E are TF +4 – compared with the TF +15 value when Progress Override was chosen.

So how do you feel about the use of Progress Override? You don’t like it, do you? We don’t either – and neither do most experienced Project Facilitators. Truthfully, it is difficult to come up with a really good justification for its use, except maybe if you are performing some kind of theoretical What-If Study and want to know how the dates would shake out if overlooked Logic were to be permanently dismissed.
13B3: How to Interpret a Schedule Update

As we close out this chapter, we zero in on the reason for performing a Schedule Update in the first place. What we wish to know is whether the work performed during the reporting period was sufficient to maintain Schedule pace or — whether the Project slipped behind or slid ahead of its previous standing. Most Owners have high interest in these determinations and that explains why Predictive Data is so popular among them.

When we think of the CPM Schedule as a dynamic model of Project Execution Strategy (before, during, or after the fact) we come to understand what Project Time Management is really all about. For now we see that Project Time Management is much more than its ancestor, Dominant Project Management's basic Planning and Scheduling. Read any book on Planning, Scheduling and Project Control — it will invariably boil down to developing and maintaining the Project Schedule, and little more. That “little more” amounts to analyzing the Schedule Update to determine whether the Schedule is being followed.

That is not what Project Time Management professionals should care about, first and most. The Project Team should have its eyes squarely pointed forward to where the Project is headed — and not looking over its shoulder or caring all that much about where it has been. Proactive Project Time Management practitioners (both Project Executors and Project Facilitators) are forever fixated on whatever work remains to be accomplished.

This means focusing on Imminent Period and Future Period strategy. And, as (we hope) you will recall, both of those strategy-developing exercises depend heavily on two categories of real time information: Gleaned Data and Just Done Data.

We have spoken at length about Gleaned Data and how it tells us what our performance potential may be. But we have said little about Just Done Data. Don’t let the seemingly minor presence of the Just Done Period (around 1% of Project Length) fool you: what it
tells us about a Project in the Current Period could not be overstated.

Just Done Data is not just a matter of noting what performance was executed in the days leading up to the Data Date. Hardly! More than anything, it tells us about the contemporaneous operating conditions. One of the primary pieces of information under this heading is where the Project stands, with respect to Schedule? Are we behind Schedule? Are we ahead? Those aren’t the main questions to ask, but they represent a good start. Where are we behind or ahead; now that is important to know!

For example, if we can identify those Activity Paths where we enjoy sufficient positive Total Float, we might be able to divert Resources to other Activity Paths where we are either suffering negative Total Float or in serious jeopardy of the Path’s Total Float turning negative. So one of the important analyses to be performed as part of every Schedule Update should be a careful assessment of what the current Schedule Update tells us about our contemporaneous conditions.

To this end, please use Figure F1328, which is a reprint of the earlier Figure F0928. Let us assume that Figure F1328 represents the Schedule Edition from the previous reporting period. Based on the information provided in Figure F1330, we want you to use the blank Logic Diagram in Figure F1332 to develop the current Schedule Edition Logic. Based on the information in Figure F1330, determine all Remaining Durations. (Assume that we are not using Progress Override.)

Now perform a manual Forward Pass and a manual Backward Pass. Calculate Start Total Float and Finish Total Float for each Activity. Trace the various Activity Paths through the Schedule. And then answer the following questions. Our answers appear in Figure F1334.

☐ **Question 1:** With respect to **FNLT 92,** has the Earliest Finish of this Deadline improved, gotten worse, or stayed the same? What is the Total Float for this Deadline, last Update? This Update?

☐ **Question 2:** With respect to **FNLT 36,** has the Earliest Finish of this Deadline improved, gotten worse, or stayed the same? What was the Total Float for this Deadline, last Update? This Update?

☐ **Question 3:** Overall, from a Project Time Management perspective, has the Project improved, stayed the same, or gotten worse since the last Update?

☐ **Question 4:** Which Path or Paths, if any, have slipped further behind from where they were in the previous Schedule Edition? Identify each qualifying Path.
F1334: Solution to Figure F1332
CHAPTER FOURTEEN

14A  Arrow Diagramming Method: Original CPM Format
14B  PERT: Program Evaluation and Review Technique
14B1 PERT’s More Complex Duration Estimating Technique
14B2 PERT’s Flowcharting Format
14B3 PERT’s Legacy: Living on in the Critical Path Method
14A  Linear Scheduling and Line of Balance Scheduling
14B  Building Information Modeling (BIM)
In this chapter we will take a look at other Project Time Management modeling Methodologies and Technologies – that is, other than the Critical Path Method. Some of these were popular in the past, but have since fallen out of favor. Others have been sleeping giants for most of their existence, and are only now gaining in popularity. Whether you are (or hope to soon be) a Project Executor or Project Facilitator, you would be well served to know about these Other Project Time Management alternatives.

14A: Arrow Diagramming Method: Original CPM Format

The Critical Path Method was invented in the late 1950s. Back then, computers were in their infancy. Data storage was restricted to reel-to-reel magnetic tape, and this meant that data could only be read in a pre-determined (by the programmer) and serial order. Accordingly, in order to function efficiently, the instructions and information would have to be written into the program in the same order of their anticipated need and eventual supply. Imagine being at the front end of a program and needing information located at the other end of the reel? This early form of memory was called ROM, standing for Read Only Memory.

You can imagine, then, why programmers drooled over the next evolution in computer Technology, Random Access Memory, known more commonly, even today, as RAM. With the invention of the memory disk, a magnetic plate could spin on a drive at lightning speed, and a Read Head could move in and out from the center of the disk (not unlike its ancient counter-part, the diamond-tipped record needle riding across an old vinyl record), and thus quickly “access” any “random” spot on the disk – all in an instant. Suddenly a universe of possibilities sparkled in every garage programmer’s eyes!

But back to those days of Read Only Memory, the restrictive mandate of that archaic technology was that everything written into a program – or its use – had to keep in mind the rather sequential approach to computational operations that characterized these early computer programs.

When the Critical Path Method was initially conceived, it was devised first as a complex piece of programming codes (algorithms and such), and only secondarily as a graphical
system of symbols and notations (which, even then, were only developed as a way to pictorially explain to non-programmers how the CPM concept worked programmatically).

There were rules associated with the earliest form of CPM that do not exist today. One of them had to do with what we have called the Activity Identifier, in this book. Actually, in those days the Activity Identifier was (a) a two-part value, and (b) it was completely numeric.

Keep in mind that even the most basic computer functions required many lines of code. Simply sorting a small list of numbers could require several lines of codes that were then run in a loop several times over, in order to put the items in a numeric order. Imagine having a Schedule with 5,000 Activities that had to be organized numerically!

Each Activity was identified by a two-part number pair, called the I-J number. This nickname was clearly a product of the minds of the mathematicians who invented the Critical Path Method. In mathematics, then and now, it was/is common to refer to algebraic variables by letters, such as A, B, C, and so forth. Sometimes letters from the opposite end of the alphabet are used, such as W, X, Y, and Z. And if a third set of variables is required, they might be taken from the center of the alphabet, such as I, J, K, and L. And so it came to be that an Activity would be identified by a pair of numbers, those numbers being represented algorithmically by the letters I and J.

To give an example, let us return to an Activity Listing we have not seen since Chapter Two, reproduced as Figure F1402. In particular, let us consider the first four Activities and imagine how they might have been “numbered” back in the days of the Arrow Diagramming Method (ADM).

As we just noted, the Activity Identifier would have been written as a number-pair, such as:

- 10-20 for **Excavate Trench**
- 20-30 for **Lay Pipe in Trench**
- 30-40 for **Inspect Pipe**
- 40-50 for **Backfill Trench**

---

**SIMPLE TIMETABLE**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>BldgA-001</td>
<td>03JUN</td>
<td>04JUN</td>
</tr>
<tr>
<td>BldgA-002</td>
<td>05JUN</td>
<td>06JUN</td>
</tr>
<tr>
<td>BldgA-003</td>
<td>07JUN</td>
<td>07JUN</td>
</tr>
<tr>
<td>BldgA-004</td>
<td>08JUN</td>
<td>08JUN</td>
</tr>
<tr>
<td>BldgB-001</td>
<td>05JUN</td>
<td>06JUN</td>
</tr>
<tr>
<td>BldgB-002</td>
<td>07JUN</td>
<td>08JUN</td>
</tr>
<tr>
<td>BldgB-003</td>
<td>09JUN</td>
<td>09JUN</td>
</tr>
<tr>
<td>BldgB-004</td>
<td>10JUN</td>
<td>10JUN</td>
</tr>
<tr>
<td>BldgC-001</td>
<td>07JUN</td>
<td>08JUN</td>
</tr>
<tr>
<td>BldgC-002</td>
<td>09JUN</td>
<td>10JUN</td>
</tr>
<tr>
<td>BldgC-003</td>
<td>11JUN</td>
<td>11JUN</td>
</tr>
<tr>
<td>BldgC-004</td>
<td>12JUN</td>
<td>12JUN</td>
</tr>
</tbody>
</table>

**F1402: Simple Timetable**
Even if you are not a computer science major, we suspect you are able to see how an algorithm could be written that would “link” the four Activities together sequentially. We are imagining something along the lines of a routine that effectively says that, any Activity with 20 in its second part (J position) is a Restricting Activity to any Activity with 20 in its first part (I position).

Because this concept was entirely new to the world, its inventors found it necessary to explain the model by way of a simple diagram. They decided to use circles, which they called Nodes (another mathematics throwback) as a container for the I and J values, and an Arrow to represent the Activity itself. See Figure F1404.

The beautiful simplicity of the Arrow Diagramming Method caught on like wildfire. Before long, Construction Project Managers were using the Critical Path Method (with its Arrow Diagramming Method, ADM – at that time being the only diagramming method) to model Project Execution before the fact. Imagine the excitement in the dusty, dirty, gritty world of construction … using the shiny, blinking, humming computer to model a Project Execution Strategy before the first spade of dirt was turned!

But the mood was not idyllic everywhere. Back at the Contractor’s home office, hidden away in some dark, remote corner of the building … a lonely Project Scheduler poured over the Schedule’s Logic, pulling his hair out at the unnecessary complexity of the evolving Logic Diagram compared to the corresponding work that it was meant to model. Take a look at a Logic Diagram from a previous chapter, which we have reproduced here as Figure F1406. It has 22 Activities, as you can see.

That same Logic Diagram, drawn in the old Arrow Diagramming Method would have required 48 Activities, as shown in Figure F1408. This doubling of the Schedule’s size is due to a few simple, but collectively intrusive, rules required by ADM.

- No two Arrows can ever touch one another. Said differently, every Arrow must have a Node at its tail and another at its head. This makes sense, since inside those Nodes were the I-J values that served as the Activity Identifier. The natural consequence of this rule meant that whenever two Activities overlapped (as we would depict through either a Start-to-Start or Finish-to-Finish Dependency Type), the Activity would have to be “split up.”
No two Activities could have the same I-J number pair. This, too, makes sense since the I-J number pair served as the unique record identifier. (You wouldn’t want multiple records in a database to have the same identifier, would you?) To prevent two parallel Activities, ones that had the same Restricting Activities and same Restricted Activities, from having the same I-J pair, a bogus Activity (called a Dummy) was injected into the Logic. Dummies are shown in Figure F1408 as dashed Arrows with no Duration.

All I-J pairs had to follow a lower-higher numbering pattern, such that the I value was always numerically lower than the J value.

By the way, have you realized that with the I-J values being entirely numeric, this would mean that Activity Identifiers could not have what we call “intelligence?” Compare this to today’s Logic Diagrams, where the Activity Identifier can contain alphanumeric characters of virtually any length. For example, we might have an Activity Identifier that reads, B02W-0304, where “B” stands for “Building B” and “02” stands for the second floor, and “W” stands for the “west wing.”

You may be wondering what we call the current Diagramming Method. It is known as Precedence Diagramming Method, or PDM. Why they latched onto the word “precedence” is unclear to us, as even the Arrow Diagramming Method depicts the precedence of Activities. But PDM is the format we have been using throughout this book, and today is the prevailing Diagramming Method of CPM worldwide.

Anyway, there are certainly some historical carryovers from the old ADM days. As you might imagine, ADM was also known as Activity-on-Arrow (AOA for short). This made sense. But then, when PDM came around, it was alternatively known as Activity-on-Node (AON for short). The idea of course was to distinguish this newer Diagramming Method by diverting emphasis from the Arrow, which no longer represented the Activity but instead depicted the Performance Restriction. Still, we find reference to “Nodes” (when PDM does not contain any Nodes) to be a bit awkward, if not also confusing.

Arrow Diagramming fell out of favor once PDM showed up on the scene.

- With PDM's three additional Restriction Linkages there was no contest.
- Schedule sizes were cut in half (as you saw above).
- Each Activity could have an Activity Identifier that was singular and meaningful.
- The Logic Diagram itself looked more like a familiar flowchart and was therefore more intuitively understood.
- There was no need for Dummy Activities.
- Yet, ADM allowed for time-scaled logic diagrams, something PDM could not do.

1 Some might argue that a "dummy" is not actually an Activity, per se, but rather merely a logical link of sort.
All of that said, believe it or not, there are still old timers today that pine for the return of the Arrow Diagram. We are not one of them!

14B: PERT: Program Evaluation and Review Technique

At the very same time that the Critical Path Method of Modeling was being born, a competing Project Modeling Technology was in its incubator. It was called the Program Evaluation and Review Technique; PERT for short. Since PERT ultimately was not adopted by the construction community, and since the ICS-Compendium is directed exclusively toward the construction community, we will keep our comments about PERT fairly brief by merely noting PERT’s most distinguishing characteristics:

- PERT adopted a process for estimating Activity Durations that is markedly different than how Activity Durations are estimated in the Critical Path Method.
- PERT utilized a decidedly different graphical format than its CPM counterpart (at the time), Arrow Diagramming.
- PERT’s legacy lives on in the Critical Path Method.

14B1: PERT’s More Complex Duration Estimating Technique

PERT took a decidedly more scientific approach than the Critical Path Method when it came to how it estimated Activity Durations. In CPM’s approach to Project Execution Modeling an Activity Duration was simply determined (stated as a matter of fact, so to speak) and, at that, given as a whole number.

- Granted, this determination may have come after open discussion among Schedule Development participants.
- Granted, those contributing to that discussion may well have drawn on personal past experience having performed the same type of work as contemplated by the Activity Work Scope.
- Granted, industry references, such as historical productivity tables, might well have been cited.

But in the end, the group that had gathered around the Schedule Development table would settle on a number – on a single numeric value that would be assigned to the Activity as its Original Duration. This approach is often referred to as a deterministic approach.

Now contrast that with how PERT went about developing its Activity Durations, which is commonly known as a probabilistic approach. To the inventors of PERT, an Activity Duration could never be precise; it would always be, at its very best, an estimate. As such, while ultimately given as a single number, it really represents a range of possible
Activity Durations. As they saw it, there were three points along the continuum of that probability range: Optimistic Estimate, Pessimistic Estimate, and Most Likely Estimate.

- **Optimistic Estimate:** This is the value at the low end of the Activity Duration Range continuum, the smallest numerical value.

- **Pessimistic Estimate:** This is the value at the high end of the Activity Duration Range continuum, the largest numerical value.

- **Most Likely Estimate:** This is the value somewhere in the middle of the Activity Duration Range continuum, with a numeric value representing something less than pessimistic, but something more than optimistic.

So, for instance, in estimating the Activity Duration for making breakfast, the following three estimates might emerge:

- **Optimistic Estimate:** If it is a really small breakfast, like a Pop-Tart and a cup of rewarmed coffee, the Activity Duration might be as little as three minutes – just long enough to toast the Pop-Tart and microwave the coffee.

- **Pessimistic Estimate:** If we’re talking about a full-course breakfast – complete with juice, prepared eggs, freshly-baked croissants, and a cinnamon roll for the road – we’re looking at more like 20 minutes.

- **Most Likely Estimate:** Assuming that the our guests won’t have time for the full breakfast, and yet wanting more than warmed over coffee and Pop-Tart, probably 15 minutes is more reasonable to throw together a quick scrambled egg sandwich and a glass of juice.

What PERT does with the three estimates is where things get really interesting and explains why it is called the probabilistic approach. They apply a formula that averages all three estimates but – and this is the kicker – favors the middle value, the Most Likely Estimate. Here is the formula:

\[
\text{DE} = \frac{\text{PE} + \text{OE} + (4 \times \text{MLE})}{6}
\]

\[
\text{DE} = \frac{20 + 2 + (4 \times 15)}{6} = \frac{82}{6} = 14
\]

PERT does offer a distinct advantage over CPM in one particular area: estimating the probability that some downstream Deadline Milestone will be met. In a process known as Monte Carlo, regression analysis is applied to the three different Activity Duration estimates, and then a Forward Pass is run repeatedly (e.g., 30,000 times). The result is a probability distribution curve for each such Milestone.
We find it interesting that even this more scientific approach to estimating Activity Durations still favors the Most Likely Estimate which, you must agree, is an entirely subjective value. So, even though the attempt is to bring more precision and credibility to the Activity Duration Estimate, at the end of the day it remains just that, an estimate.

14B2: PERT’s Flowcharting Format

A second distinguishing characteristic of PERT became obvious when the two Methods’ graphical forms were laid side-by-side. PERT used boxes to represent Activities, and Arrows between the Activity Boxes to indicate the Dependencies between the Activities. Does this sound familiar? It should, because the Precedence Diagramming Method (PDM) – the one almost exclusively used today – was based on the PERT format. But back then, long before PDM, the two Methods could not look more different.

14B3: PERT’s Legacy: Living on in the Critical Path Method

As we just said, when PDM came around, it chose to use the PERT format of Diagramming Logic, with Activities shown as Activity Boxes, and Dependencies shown as Arrows. One need only reflect back on Figure F1406 and Figure F1408 to appreciate how much of a godsend this was to practitioners of the day. To this day, Schedule Logic, drawn out as a flowchart – as Activity Boxes and Arrows – is known affectionately as a PERT Diagram, even though there is nothing about the PERT Project Execution Modeling Technology at play in the CPM Schedule.

14C: Linear Scheduling and Line of Balance Scheduling

Two other approaches to Project Execution Modeling that have not acquired the immense popularity of the Critical Path Method (in Construction Project Time Management) but are definitely becoming more accepted in certain Project Management circles are Linear Scheduling and Line of Balance Scheduling. In the interest of keeping things simple, since there are only subtle differences in the two approaches (i.e., they are more alike than they are different), we will refer to them both with the same sweeping statements.

Both share the same orientation to space over Time as to how they model Project Execution. Where the Critical Path Method links Activities through Logic (in drawn form within graphical diagrams; and in coded form within Scheduling Software), these two Methods link Activities by virtue of the proximity of the Actions to which they refer – by their location.

Consider Figure F1410. Here we see a simple example of a Linear Scheduling output. See how it shows the spatial Relationship between four Activities that are repeated across six floors of a multi-story structure? Linear Scheduling/Line of Balance Scheduling can
be effective Project Execution Modeling tools on Projects that call for repetitive Actions. Project Types that fit this category include pipelines, road work, extensive utility runs in a large structure, and so forth.

This form of Project Execution Modeling, while filling a certain void that traditional Critical Path Method does not plug, has largely been disregarded by mainstream Construction Project Time Management up until now for a few important reasons:

☑ The percent of Project Types (or Project segments within traditional Project Types) that are repetitive in character or performance remain a distinct minority. The application opportunities for this modeling approach, therefore, are severely limited.

☑ Up until recently, practical and usable software for this modeling type either did not exist or, if they did, had serious limitations.

☑ Without software to make computer application of the technique viable, manual efforts to draw Linear Scheduling graphs or Line of Balance charts proved impractical, expensive, and not terribly responsive to the fast-paced nature of construction operations.

☑ In any event, neither Project Execution Modeling Technology takes into account the Criticality aspects of the Critical Path Method: Total Float, Free Float, or Critical Paths.
All of this said, be on the lookout for this Project Execution Modeling technology to continue to make gains in upcoming years, especially where they dovetail with Building Information Modeling (BIM) Technology.

14D: Building Information Modeling (BIM)

Building Information Modeling (BIM) is a fairly new technology that allows one to create a digital model of the operational attributes of a Project. Using BIM, one can construct a Project in virtual space, all the while making note of any “problems” that arise in the design (e.g., missing important data, missing or conflicting dimensions), construction (e.g., the Planned Sequence, while sounding good at the time, actually results in logjams of people and resources), or commissioning of a Project, to name a few examples.

The BIM model doesn’t merely provide an informational clearinghouse, however. Its main appeal is its integration of the Project's three-dimensional (3D) model with what have always been the dry, tabular aspects of Project Management: Cost Management and Time Management.

BIM constitutes a geometric platform on which these other systems can be merged together to form an informational powerhouse for the Project Team. It allows engineering analysis in the computer rather than out in the field. It allows this 3D model of the physical Project to be linked, data bit by data bit, with the Critical Path Method Schedule and its many hundreds or thousands of Activities.

And because a CPM Schedule contains dates, with a few mouse clicks the Project can materialize in three-dimension in accordance with the Logic in the Schedule. BIM also integrates costing information (whether stored in the Schedule or separately), such that budgets, cash flows, draw downs, and Commitments can change as the Project manifests.

One need only sit back and consider the gains being made in Linear Scheduling (Line of Balance) and BIM to appreciate that Project Scheduling as a model of Project Execution Strategy is in for some big changes.

We also observe, however, that these changes cannot (and will not) compensate for bad Logic, unrealistic Activity Durations, excessive or inadequate Date Constraints, ill-conceived Work Performance Calendars, inappropriate Software Settings or, perhaps worst of all, damaging or counter-productive Project Time Management practices.

Our treatment of PERT, Arrow Diagramming, and BIM are admitted light. They do not even cover all of their more salient features. Because this is a book on the mechanics of the Critical Path Method, we could have omitted any mention of these other techniques, altogether. However, we felt it was important for you to at least know that these concepts exist, and generally what they are all about.
CHAPTER FIFTEEN

15A What the Critical Path Method of Modeling is Not!
15A1 Virtually All Project Model Content is Speculative
15A2 Best Project Execution Model Never More than a Good Guess
15A3 Project Model: While Not Flawless, Better than the Alternative

15B What the Critical Path Method of Modeling Is!

15C Not All Project Execution Models are Schedules
15C1 Project Execution Modeling versus Project Scheduling
15C2 Project Time Management Lacks Comprehensive Training
15C2a Generalized Credentials Are Not the Silver Bullet
15C2b Software Training is Not the Entire Answer Either
15C2c Legitimate, Comprehensive, Quality Training IS the Answer
Interpreting the Project Execution Model

Well, you made it! The last chapter!

We have come a long way from the opening notion that a Project Execution Strategy can be modeled in a computer, and to some practical and useful end. Indeed, using the Critical Path Method of Modeling, we have found that we can indeed model both a Project's planned and accomplished Execution Strategy.

We have assembled the pieces and behold, we have a bicycle. Sir Isaac Newton would be proud that we now understand the whole ... for having analyzed the parts. But sometimes things are greater than the sum of their parts; and at other times they are not as magnanimous as the sum of their parts might first lead us to believe.

So on these closing pages we want to temper what you have learned with some pensive and sobering thoughts about what the Critical Path Method of Modeling is not. We want to draw boundaries around what you have learned, and we wish to suggest that you are but one-third\(^1\) of the way in your study of the Critical Path Method's role in Construction Project Time Management.

15A: What the Critical Path Method of Modeling is Not!

We can surely agree that the Critical Path Method of Modeling is very powerful and versatile. We can agree that its invention in the 1950s entirely revolutionized how Projects are managed and, more to the point, how to better achieve Project Time Management. But we would be wrong to put too much stock in the Project Schedule, even one built on the Critical Path Method platform. As Clint Eastwood warns us, “A man’s got to know his limitations.”

15A1: Virtually All Project Model Content is Speculative

Let us start with this truth. Virtually all of the information that you find in a Project Schedule is speculative in the first or second degree; that is, the data is either speculative

\(^1\) The other two-thirds (to learn) are: CPM as a Scheduling Method, and CPM as a Managing Method; subjects of other ICS-Compendium volumes.
in its own right, or it is based on speculative data.
Consider these examples of the former group:

- **Activity Duration**: The item that comes to mind most immediately is the Activity Duration. No matter how much deliberation, research, and discussion goes into its formation, every Activity Duration is still just a best guess of how long a particular Action (or set of Actions) will take on the Project.

  This is true even if the Activity Duration estimation is based on the PERT three-point approach – for even then, the three values are just guesstimates: of the Most Optimistic, the Most Pessimistic, and the Most Likely Duration.

- **Activity Description**: We bet you might not have thought of the Activity Description as speculative. Well, the Activity Description represents the Scope of Work of the Activity, and this is also based on a set of assumptions and generalizations about what is entailed in the performance of the work of that particular Activity.

  We are actually discussing two different points at once, here. First, we are suggesting that all of the Activities in the Schedule, taken in totality, do not actually cover every aspect of the Project. Second, and for the same reason (as we will shortly discuss), any single Activity contains more work that its Activity Description is able to fully describe.

  Consider the Activity Description, **Install Drywall**. What is needed in order to install drywall? Well, sheets of drywall for starters. What else? Consumable materials and hand tools, like drywall screws and a drywall screw gun. What about workers? How about a place to work – free of other trades, of debris, of inclement weather? All of this is assumed to be included in the Activity, **Install Drywall**.

  Of course it is possible that the drywall delivery was covered by another Activity, called **Deliver Drywall**, but tracking the delivery of basic materials in a Project Schedule is not commonplace. Rather, it is just assumed that those who will be performing the installation work will recognize, in their prior and timely review of the Schedule, that drywall will be needed “... next week! We had better go get some!”

  The same is true for availability of labor, consumable supplies, and even of having a place to work. More than most any other aspect of the Schedule, it is a key assumption that the Contractor will have a place to work. More often than you might believe, this is not the case. A preceding Contractor vacates a work area
and leaves it a royal mess. Or, a Contractor leaves behind a stack of materials because he can find no other place to store them.

Our point is that it is simply not possible or practical to include every little detail in a Project Schedule. In fact, more is left out of the Project Schedule than is ever included in it! Studies conducted by ICS-Research have found that the majority of Schedule delays occur between Activities, not during their performance. Many of those delays are due to conditions that were simply not adequate to allow the work to commence in the first place.

So when we read an Activity Description, however well worded, we are wise to remember that the Scope of Work described by the Activity Description’s wording is almost always less than the amalgamation of Actions that are required for the Activity to eventually be statused as 100% complete.

- **Remaining Duration**: Just like its cousin, the Original Duration, a Remaining Duration is always a best guess as to how much more Time will be required to complete the work of the Activity.

- **Restriction Linkages (Dependency Types)**: The Logic of the Project Schedule, the hundreds or thousands of individual Performance Restrictions between Activities, are all speculative. Every single one of them is an assumption about how the corresponding Actions out on the Project will be coordinated.
  - When a Default Restriction is assigned, this is an assumption that the Restricted Activity will wait for all of the Restricting Activities to complete.
  - When a Start Restriction is assigned, this is an assumption that the Restricted Activity will not wait for the Restricting Activity’s completion before starting, but will be proactive and get “a jump” on its work just as soon as the Restricting Activity has progressed through an early portion of its work.
  - When a Finish Restriction is assigned, this reflects a perception (which may not always be correct) that a certain portion of a follow-on Activity cannot occur until some prior Activity has finished.

- **Progressive Restrictions (Dependencies) in General**: Not unlike Activities themselves, not all of the Progressive Restrictions that exist on the Project are, or ever should be, reflected in the Project Schedule. In fact, it is both correct and advisable for us to say that if you do try to include all of the Restrictions between all of the Activities in the Schedule, you will render the Project Schedule utterly worthless as a Project Execution Modeling tool. In other words, there is such a thing as
over-tying a Schedule. This is discussed in another ICS-Compendium volume.

**Restriction Delays:** Quite apart from the choice of which Restriction Linkage to use between Activities that share an Immediately-Restricting Progressive Relationship, is the question of how much of a Restriction Delay to assign. If you thought that Activity Durations were speculative, when it comes to Restriction Delays there is virtually no guidance provided in Dominant Project Management literature as to how to determine the appropriate amount of “Leads and Lags.” It is fair to say that, whatever numerical value is assigned to the Restriction, it is entirely subjective and quite speculative.

**Date Constraints:** Many, maybe even most, Date Constraints are not speculative; they reflect contractual Deadlines that have no ambiguity about them. But then there quite often are other Date Constraints that are speculative. A classic example is a Start-No-Earlier-Than Date Constraint that is used to delay the start of Activities that cannot be performed during severe winter weather conditions. And so, some arbitrary date on the calendar is chosen to represent a “best guess” as to when winter will have ended. Likewise, a Finish-No-Later-Than Date Constraint might be used to represent the onset of winter. These are entirely speculative dates.

**Work Performance Calendars:** Work Performance Calendars provide a means for blocking out expected Non-Workdays. Some of those blocked-out days may not be speculative, such as Government holidays. But the Work Performance Calendar is a favorite choice among seasoned Project Facilitators to set aside Time Contingency for inclement weather. Both the actual number of Non-Workdays, and where they are placed on the Work Performance Calendar, are each highly speculative.

**Software Settings:** As we discussed when we were talking about them in Chapter Nine, the choice of Software Settings can dramatically affect all CPM calculations. There are no hard-and-fast rules as to which settings to choose under which conditions. This makes them entirely conjectural as a Scheduling variable.

Now consider these candidates from the latter group, the Schedule variables that are based on speculative data:

**All Schedule Dates:** Earliest Dates and Latest Dates are all affected by the above variables. If all of the underlying variables are speculative, then so are the resultant (calculated) dates.

**Total Float and Free Float:** Since both Total Float and Free Float values are determined
by comparing Schedule Dates, these too must be speculative, as a result.

**Critical Path:** All Path ratings, whether Critical, Near-Critical, or otherwise are based on Total Float values. Therefore, any declaration as to an Activity Path’s Criticality is entirely speculative.

15A2: **Best Project Execution Model Never More than a Good Guess**

The long and short of it all is that the Project Schedule, as a model of Project Execution Strategy, is simply never more than a *good guess*. We would be well advised to keep this reality in our minds at all times, lest we get too full of ourselves and think that we have some greater scientific model. This may seem like a bit of a chastisement, but we speak from years of experience – watching some of the biggest and brightest in the business split hairs over Total Float numbers from one month to the next.

Folks, it’s exponential. Total Float is a factor of Calculated Dates. Calculated Dates are a factor of Logic and Activity Durations. Logic is a factor of Restriction Linkages and Restriction Delays. When you start compounding those variables you quickly come to realize just *how much* Total Float can shift as well as *how easily*. There is no mystery as to why Total Float changes each reporting period, or why the Critical Path flips around like a dinosaur tail in a fight.

15A3: **Project Model: A Reflection of Compromise**

Let us never forget that the Project Schedule is not the creation of just one single individual. All of the variables we have mentioned on the last few pages would be true if just one person was compiling the Project Schedule in a vacuum. But good schedules — indeed, the best schedules — are the work product of the Project Team. So multiply the probability of errors and omissions resulting from a *group* effort.

Humans have known for thousands of years that "management by committee" is not the most effective approach to decision-making. Since it is unlikely that any group of individuals will agree on everything (or, sometimes, anything), it stands to reason that the Project Schedule represents necessary compromise, doesn’t it? For its part, compromise entails not incorporating everything you would have liked to see in the Schedule. And this is yet one more reason why the Project Schedule is not an absolute model of Project Execution Strategy.

15A4: **Project Model: While Not Flawless, Better than the Alternative**

From all of the above, you might be wondering if the Project Execution Model is worth
all of the trouble to create and maintain. Of course it is. Projects are challenging; by
definition they are unique and this is certainly true of Construction Projects. They also
take a long time to complete; they extend into the distant, unpredictable future. They
employ workers and companies that are quite often at first strangers to one another.

Projects are risky and they are speculative. Every competitive bidder who becomes a
successful Contractor is a risk-taker. He is betting his company’s profits (and sometimes
survival) on a strong belief that his company will not only perform the work within the
Budget and Schedule, but can turn a profit too.

If everything about the Project is speculative, why would we ever expect the Project
Schedule, which models that Project's Execution Strategy, to be one iota more certain? So
the Project Schedule is speculative! So what?

That’s why we routinely update the Schedule. That is why we gather Realized Data and
pour over it. That is why we develop Gleaned Data. That is why we recalibrate our WAY
FORWARD, based on the Gleaned Data. And with each Schedule Update, the remaining
Project Length shortens, our Gleaned Data becomes more reliable and credible, and our
Future Segment strategy becomes more viable and stable!

15B: What the Critical Path Method of Modeling Is!

The Critical Path Method of Modeling is at the very epicenter of Construction Project Time
Management which, in turn, is the very backbone of Construction Project Management.
Here is why we say this.

☐ Effective Project Management requires effective Project Time Management.

☐ Effective Project Time Management requires effective Processes, and qualified
personnel. Yet no combination of Processes or Personnel, no matter how effective,
can secure or sustain an effective Project Time Management Program that does
not have an effective Project Schedule.

☐ An effective Project Schedule must be an interactive, dynamic, responsive,
relevant, accurate, Timely, and usable model of Project Execution Strategy.

This is the view of Cognitive Project Management — that without a sound and effective
Project Schedule one simply cannot have effective Project Management, and surely
not effective Project Time Management. We also contend that the effectiveness of any
formal Project Management effort correlates directly with the coherence and integrity
of the Project Schedule, as a model of Project Execution Strategy.
Interpreting the Project Execution Model

We are hardly alone when we correlate the effective management of Time on Projects with an overall ability to manage the achievement of all other Project success objectives. To our thinking, we cannot imagine how it is possible to confidently insure the achievement of Cost Management, Risk Management, Procurement Management, Human Resources Management, and so forth ... without a competent Project Time Management program.

Consider these definitions of Project Management – found from a Google search on that term – and notice the overwhelming reference to the scope, principles, and objectives of Project Time Management, (notwithstanding the broad array of different word choices). We have bolded what we consider to be references to Project Time Management.

- The discipline of planning, organizing, securing and managing resources to bring about the successful completion of specific project goals and objectives.\(^2\)
- Body of knowledge concerned with principles, techniques, and tools used in planning, control, monitoring, and review of projects.\(^3\)
- A methodical approach to planning/guiding project processes from start to finish.\(^4\)
- A set of well-defined methods and techniques for managing a team of people to accomplish a series of work tasks within a well-defined schedule and budget.\(^5\)
- The planning and organization of an organization's resources in order to move a specific task, event or duty toward completion.\(^6\)
- Completing a project on paper before you begin, and then executing the project according to the plan.”\(^7\)
- The planning, monitoring and control of all aspects of a project and the motivation of all those involved in it to achieve the project objectives on time and to specified cost, quality and performance.\(^8\)
- The process of directing and coordinating human and material resources throughout the project life cycle using modern management techniques to achieve established objectives of scope, quality, time, cost and stakeholder satisfaction.\(^9\)

2 Wikipedia
3 Business Dictionary.com
4 Searchcio-midmarket.com
5 Visitask.com
6 Answers.com
7 Globalknowledgeblog.com, Brian Egan
8 MicroPlanning.co.uk
The planning, scheduling, and controlling of project activities to achieve performance, cost, and time objectives, for a given scope of work, while using resources efficiently and effectively.\[^{10}\]

Planning, monitoring and control of all aspects of a project and the motivation of all those involved in it to achieve the project objectives on time and to the specified cost, quality and performance.\[^{11}\]

We readily appreciate a possible counter-argument by would-be critics of our assertion that Project Time Management is central to effective Project Management. They might insist that the majority of the above quotes only identify Project Time Management as a key, but not necessarily central, element of Project Management.

Our rebuttal comes from the application of simple common sense. If Project Management involves the orchestration of people that are performing Actions which utilize Resources, then how can such orchestration be accomplished without some organizational structures and processes designed to coordinate these people and their Actions?

We believe that the Chartered Institute of Building said it best when they noted in their seminal work, GUIDE TO GOOD PRACTICE IN THE MANAGEMENT OF TIME IN COMPLEX PROJECTS, that:

Without effective Time Management there can be no effective resource management, Cost Management, nor allocation of liability for slippage, its recovery, or accountability.

And this assertion requires little defense. Cost Management, at its heart, is about the efficient use of money. Prices are set by the marketplace; one does not control prices. They can only control cost at the point and time of their being incurred. When work is performed inefficiently, unnecessary costs are incurred. True Cost Management is all about performing the work as efficiently as possible.

Likewise, Resource Management involves using Resources as wisely as possible. Not only does the waste of Resources drive up costs, it could deplete a limited supply of such Resources, the eventual extinction of which could kill a Project. The other primary areas of Project Management concern are also directly dependent on the effective performance of Project Activities — an outcome that cannot be accomplished without a well-orchestrated strategy.

\[^{10}\] MaxWideman.com — Brochure by Management Concepts, 1999
\[^{11}\] MaxWideman.com — Association of Project Management, Symposium, 2000
15C: Not All Project Execution Models are Schedules

Let us be perfectly clear about this last point. All Project Schedules are models of Project Execution Strategy. But not all Project Execution Models are Project Schedules. The difference is in the application of the Project Execution Model. It’s like the difference between people in general, and employees in particular. All employees of a company are people, but not all people in the world are employees of a particular company.

Have you ever been to a theme park where, as you engage in a certain attraction, the venue takes your picture? Maybe you and the family are careening through raging water rapids, or flying over a high peak on a roller coaster ride, or perhaps just standing in line to board a train. Later, when the ride is done, and you are shuffling among a sea of fellow participants to leave, and you’re still smiling from the event, you are confronted by a cork board on which you spot a picture of you and the family at that very moment of extreme ecstasy.

If you purchase the picture, it becomes a souvenir. If you don’t it ends up in their trash bin at the end of the night. In this example, all photographic souvenirs that you bring home begin as snapshots taken at random, but not all snapshots taken end up as photographic souvenirs you bring home.

Do you see the difference now? A Project Execution Model is just that — a model of the planned or actual execution of the Project. But it is only after you apply that model to a given Project that it become a Project Schedule. Let’s follow this line of thinking a little further.

This book has taught you how the Critical Path Method works and what strengths and limitations it possesses. But what we have not discussed in this book is just how to use this raw, mechanical Project Execution Modeling Technology in a real-life operational context; that is, to actually Schedule and Manage a Project!

The importance of the distinction we are trying to make (between Project Execution Model and Project Schedule) is that it clarifies what this book is about, and what it is not about. Said differently, other volumes of the ICS-Compendium Series are designed to cover what this book has not discussed.

15C1: Project Execution Modeling versus Project Scheduling

This book is about the Critical Path Method of Modeling Project Execution Strategy. ICS-Compendium Volume 3, Construction Planning and Scheduling, is all about the Critical Path Method of Scheduling. It is in that book where we learn how to take a Project Execution Model and, through various well-tested processes, use it to design, develop, and maintain a Project Schedule.
The following bullets may help you further distinguish between the *theoretical concepts* of *CPM Mechanics* discussed in this book, and the *practical application* of the Project model, as discussed in other ICS-Compendium volumes. For instance, in this book we have discussed:

- What an **Activity Identifier** is – but not how to actually create an Activity Identifier that is meaningful, functional, and effective.
- What an **Activity Description** is – but not how to word one so that it is concise, comprehensive, and resilient to change.
- What an **Activity Duration** is – but not how to determine an Activity Duration that reflects the Project Team's best, collective estimate of the amount of Time required to perform the Work Scope of the Activity.
- What a **Project Execution Strategy** is – but not how to extract this Strategy from the minds of numerous Project Executors, each of whom comes to the Project with his or her own personal and/or company agenda.
- What a **Network Diagram** is – but not how to divide the Project Schedule into meaningful Subnetwork categorizations.
- What a Schedule's three inherent **Relationship Types** are (Communal, Symbiotic, and Progressive) – but not how these distinctions can be used to achieve more effective Project Time Management.
- What the three primary **Performance Restriction Linkages** are (Default Restriction, Start Restriction, and Finish Restriction) – but not how to decide which one to use in any particular Immediately-Restricting Progressive Relationship situation.
- What a **Restriction Delay** is – but not how to establish an appropriate numeric value for one.
- What **Date Constraints** are – but not how to know when you have too few, or too many, or how to know which one to use in a given circumstance.
- What various **Software Settings** are – but not how to choose one Software Setting option over another – the pros and cons of each, under different Project circumstances.
- What a **Work Performance Calendar** is – but not how to use the Work Performance Calendar (e.g., to allocate Non-Workdays for inclement weather, hunting season, holidays, and so forth).
- What **Activity Codes** are and generally how they are used – but not how to put Activity Codes to their best use in order to tailor Schedule Reports and work products for maximum value to Project Time Management goals.
15 Interpreting the Project Execution Model

- What a **Work Breakdown Structure** is – but not how to create and use a WBS without weakening the Schedule's Project Time Management effectiveness.
- What a **Schedule Update** is – but not how to implement a responsible Schedule Updating regimen, including standard processes and protocols.
- What **Realized Data** are – but not how to gather that data, and how to competently monitor and measure performance without building resistance among Project Executors to the inherent oversight nature of such actions.
- What **Gleaned Data** are – but not how to properly draw inferences, interpolations, trends, and other conclusions from Realized Data.
- What a **Current Period** is – but not how to set an appropriate width (percent of Project Length) for the Current Period.
- What **Total Float** is – but not how to use Total Float values in a responsible, meaningful, and helpful manner.
- What **Free Float** is – but not how to use Free Float in support of Project Time Management objectives.
- What a **Path, Path Segment, and Critical Path** are – but not how to use this knowledge to support Project Time Management goals and objectives.
- What **Project Facilitators** and **Project Executors** are – but not how these two key Project Management players can work together to serve as the central catalyst for effective Project Time Management.

We could go on, but we think you get the point. This book has dealt with **CPM Mechanics**, the foundational elements and rudimentary operations of a Project Execution Modeling Technology that has transformed modern Construction Project Management. But knowing what a Project model is and how it works internally is a far cry from knowing how to use that Project model to actually schedule and manage a Project.

15C2: Project Time Management Lacks Comprehensive Training

In recent years, several different, independent surveys have raised a common complaint about the state of Project Time Management: that the discipline lacks sufficient training.

15C2a: Generalized Credentials Are Not the Silver Bullet

A number of leading Project Management organizations have tried to respond. While we sincerely admire their initiative, we nevertheless believe that their responses may be off target. Instead of developing and offering the much-needed Training first, and Certification second, instead they have rushed to offer Scheduling Credentials aimed at standardizing how Project Time Management is (or should be) performed, and then have
encouraged professional Trainers to prepare applicants to pass the Certification exams. The rationale is not with reasoning; by establishing performance standards first, a cottage industry would form, comprised of training services racing in to backfill the educational hole. Yet there is a fundamental flaw in this thinking. The anticipated training services have indeed rushed in to provide coursework, but their course content is narrowly aimed at preparing the student to pass the credentialing exam, and little more. So, the ultimate value of the training, in terms of teaching best practices and processes of Schedule Design, Schedule Development, Schedule Maintenance, and Schedule Usage, extends only as far as the credentialing exams cover these same points.

Therein lies the real deficiency in this schema. By and large, none of these credentialing organizations offer their own comprehensive set of standards, best practices, or recommended procedures that fully explain how to design, develop, maintain, and use Project Time Management products and services. Most certifying bodies simply provide a reading list of available literature on Project Time Management subjects and, as we have noted throughout this book, there is great inconsistency among these literary works. One wonders how a candidate preparing for a credentialing exam would know which is the right answer to a technical question, when there are multiple conflicting ones to pick from! [12]

As for those organizations that have offered original content of their own, such work products tend to be scattered materials, either covering the full breadth of Project Time Management but with little or no specificity, or covering certain aspects of Project Time Management in sufficient depth (while ignoring other aspects of Project Time Management entirely). [13]

15C2b: Software Training is Not the Entire Answer Either

For many years now there has been widespread criticism, we believe rightfully placed, that training provided by Scheduling Software publishers and vendors has proved to be mainly inadequate. Commonly offered seminars range from two to five days, and cover the basic steps necessary to create a few Activities, type them into the software, assign Activity Durations and Activity Codes (WBS), define and appoint a Work Performance Calendar, and generate a few standard Scheduling Reports.

These classes simply do not have the time – and quite often their instructors do not have sufficient practical experience – to teach all that you have learned in this book alone.

12 Even so called “Best Practices” documents routinely present multiple ways of accomplishing a certain process, again leaving it to the reader to decide which one is the best one for a given circumstance.

13 Reread the last three paragraphs carefully. What we are saying is that they either (a) cover all important topics but somewhat superficially, or (b) they provide sufficient depth but only for selected topics.
Now add to this book the contents of the other three books in the DOMINANT PROJECT TIME MANAGEMENT SERIES within the ICS-Compendium, and it is easy to see why many regard even five-day software seminars as hopelessly superficial.

In the end, the need and demand for quality training in the art and science of Project Time Management remains, notwithstanding a flurry of Scheduling credentials and the many different software training seminars that can be found with a simple Google search.

15C2c: Legitimate, Comprehensive, Quality Training Is the Answer

We believe that the ICS-Compendium, once complete, will represent the first full and comprehensive treatment of Project Time Management concepts, principles, terminology, practices, and processes made available to the Construction Industry in North America. The sole motivation behind the ICS-Compendium has been to clarify, to illuminate.

We have no hidden agenda, aimed at some overflowing pot of gold at the end of the credentialing rainbow. If our eyes are ever adrift, away from this direct goal, perhaps it is the occasional dream of leaving a legacy behind when our work is done, that we have made a positive difference in how Construction Projects are managed: so that the very best use is made of that one fleeting, uncontainable Resource: Time.

15D: Where Do You Go From Here?

Having finished this book, you are now completely knowledgeable of how to use the Critical Path Method to model the various elements of a Project Execution Strategy. With this Technology under your belt, you are now ready to learn how to design, develop, maintain, and effectively use Project Schedules that are built on the Critical Path Method platform you have just learned.

CPM MECHANICS has focused on the Technical aspects of a modeling technology. The next two volumes in the DOMINANT PROJECT TIME MANAGEMENT SERIES teach you all you will need to know in order to achievement maximum Project Time Management on your Projects. While your order of attack is entirely optional, we tend to recommend Volume 3 before Volume 2, But, again, the choice is yours.

Volume 3, CONSTRUCTION PLANNING AND SCHEDULING, deals with the Methodological aspects of “the human factor” — i.e., processes, practices, policies, procedures, standards, and so forth.

Volume 2, UNDERSTANDING PROJECT TIME MANAGEMENT, offers an Ideological approach to how the Project Team (and, in particular, the Project Executors) can utilize the work products of the Project Facilitation group to achievement the highest degree of Project Time Management. See you in the next volume — whichever one you choose.
Alphabetical Index

Index

Acceleration  219, 234, 288, 374
Action Schedule  30–31
Action versus Activity  15–16, 30–33, 43–47, 77, 140, 239, 323, 361, 424
Activity Bar  41, 42, 43, 44, 45, 49, 55, 119
Activity Box  125, 126, 127, 128, 129, 131, 132, 133, 134, 140, 142, 146, 164
Activity Criticality  8, 9, 235, 236
Activity Description  11, 30, 32, 33, 34, 41, 121, 126, 179, 206, 387, 424, 425, 431
Activity Bar  41–46
Activity Cost-Loading  13
Activity Crew Configuration  50–55, 120
Activity Dependency  16, 53–54
Activity Duration Range  418
Activity Duration Variance  358
Activity Identifier  41
Activity Log  314, 384
Activity-on-Arrow  415. See Arrow Diagramming Method
Activity-on-Node  415. See Precedence Diagramming Method
Activity Path  14, 290, 389
Activity Record  311
Activity Resource-Loading  331–334
Activity Sequencing  16, 29, 34, 53, 54, 56, 357, 374
Activity Title  34
Actual Cost of Work Performed (ACWP)  330
Actual Dates  347, 390
Actual Finish Date  384
Actual Start Date  358
ACWP  330. See Actual Cost of Work Performed (ACWP)
ADM  412–413, 415. See Arrow Diagramming Method
Agile Project Management  14
AOA  415. See Arrow Diagramming Method
AON  415. See Precedence Diagramming Method
Apparent Data 359–362, 366–370
Arrow Diagramming Method 84, 89, 197, 199, 411–415
As-Built Schedule 345
As-Planned Schedule 345–347
Backward Pass Multiple Restricted Rule 165–166, 170, 173
Bar Chart Schedule 42
BCWP 330–331. See Budgeted Cost of Work Performed (BCWP)
BCWS 329–330. See Budgeted Cost of Work Scheduled (BCWS)
Bid/Proposal Schedule 26
BIM 421. See Building Information Modeling (BIM)
Budgeted Cost of Work Performed (BCWP) 330
Budgeted Cost of Work Scheduled (BCWS) 329
Building Information Modeling (BIM) 421
Calculated Dates 14, 40, 66, 87, 116–119, 137–140, 260, 267, 276, 347, 427
Calculated Schedule Elements 121–123
Calendar Day 293–295, 352
Cognitive F.A.C.E. Diamond 3, 27
Cognitive’s T.I.M.E. Framework 3
Collaboration 313, 314, 321, 356
Command-and-Control 72, 360, 377
Commitment 3, 80, 82, 163, 212, 332, 335, 421
Commitment Schedules 29
Communication 24, 40, 41, 313, 314, 321, 356. See Project Communication
Complex Arithmetic Formula 144, 148–149, 153–154
Computed Choice 153–154, 157, 159, 170, 173
Concept of Time 15, 35, 36
Conditional State 126–127, 129, 146–147, 240
Connecting Arrow 127, 128, 134. See Logic Tie; See also Restriction Linkage
Consensus Plans 25
Construction Project Management 2–7, 322, 363, 381, 428, 433
Constructor 4
Contextual Perspective 348, 353–354, 357
Continuous/Contiguous Activity Duration 300–309
Continuous Crew Days 16, 49–53, 326, 391
Coordination 11, 313, 314, 321, 356. See Project Coordination
CPM Dates 12, 71, 117, 137. See Calculated Dates
CPM Network. See Network Diagram
CPM Scheduling 8–14, 21, 210
Crew Configuration. See Activity Crew Configuration
Criticality 8, 210, 235–236, 420, 426
Critical Path 8, 12, 37, 121, 179–180, 204, 223–250, 285, 288, 298, 299, 388, 420, 426, 427, 433
Critical Path Method of Managing 10, 21, 26, 202, 216, 308, 310
Critical Path Method of Scheduling 9–10, 13–14, 21, 26, 49, 308, 431
Current Segment Data 359
Data Date 13–14, 346, 359, 385, 392, 405, 407
Data Filtering 310, 326
Data Grouping 310–311, 344
Data Selecting 310, 344
Data Sorting 310–311, 344
Date Calculations
Date Calculation Arithmetic Formula 123, 142–156, 164–165, 170
Date Calculation Computational Rule 137–139
Date Calculation Procedures 137–138
Date Calculation Processes 137–138
Date Calculation Variables 137–139, 158, 166
Delay 219
Delay Period 91
Deliverables 25, 320–322
Dependency. See Activity Dependency
Design Schedule 26
Driving Path 224, 231, 290, 292
Dummy Activity 415
Dynamic Project Time Management 180
Dynamic Schedule 83, 84
Earliest Finish Date 117, 278, 281, 292
Earliest Start Date 117, 358, 392
Early Dates. See Earliest Dates
Early Finish Date. See Earliest Finish Date
Early Start Date. See Earliest Start Date
Earned Value Management System 328–329
Element of Time 29
End-of-Day Perspective 15, 135, 141, 143, 294, 391
Event 29, 58–63
Eventually-Impacting Relationship 74, 76, 237, 267
EVMS. See Earned Value Management
System
Expected Finish Date Constraint 278, 283
Feasibility Plans 25
Finish Dates 56, 57, 71, 119, 161, 176, 347, 357, 361
Finish Float. See Finish Total Float
Finish-No-Later-Than Date Constraint 278–279, 283–290, 306, 389, 392, 399, 426
Finish-to-Finish Dependency 82. See also Finish Restriction
Finish-to-Start Dependency 82, 93, 98, 104, 129. See also Default Restriction
Finish Total Float 12
Fixed Condition Date Constraints 278
Float 179–180
Forward Pass Process 137, 140–143, 146, 159, 166
Four Basic Calculated Dates 8, 137–139, 187
Fragmented Activity 197
Finish Free Float 261, 262
Most Critical Free Float 255
Start Free Float 253, 261, 262
Future Segment Data 361, 363
General Conditions Cost 326
G.R.A.S.P. 13, 363, 381
Gregorian Dates 296, 298, 346, 351
Hammock Activity 311
Hard Logic 80
Holdback Restriction 82, 98, 100, 101
ICS-Global 1–3, 72, 90, 112, 308
ICS-Research 1, 13, 425
Ideology 2–3, 38, 73
I-J Number 412, 415
Imminent Data 369–371
Impacted Activity Duration 46–52, 215, 268
Independent Float 261
Interferring Float 261
Interruptible Activity Duration 300–304
Just Done Data 371, 405
Key Schedule Elements 121–122, 179
Late Dates. See Latest Dates
Late Start Date. See Latest Start Date
Latest Finish Date 161, 212, 225, 281
Latest Start Date 281
Leads and Lags 82, 94
Lag Value 94, 95, 111
Lead Value 94, 95
Leapfrogging 56
Least Total Float Path 228–234, 286, 288, 290
Level of Detail 49, 336
Linear Scheduling 419–421
Line of Balance Scheduling 14, 419–421
Logic 12, 80, 84, 88, 99, 141, 146, 150, 164, 196, 213–216, 341, 362, 383, 384, 419
Logic Code. See Restriction Code
Logic Development Session 50, 118, 120, 200
Logic Diagram 125–126. See Network Diagram
Logic String 213, 216, 225, 232
Logic Tie 12, 91, 92, 93, 100, 127–134, 143, 153, 163, 245, 254, 402. See Restriction Linkage
Longest Path 224, 228–236, 288–292
Long-Lead Items 372
Mandatory Finish Date Constraint 280
Mandatory Restriction 81–82
Mandatory Schedule Elements 121
Mandatory Start Date Constraint 279–280
Maximum Condition Date Constraints 276, 283
Methodology 2, 73, 216, 394, 381–382
Milestone
  Anchored Milestone 64
  Finish Milestone 236, 240, 243
  Floating Milestone 64
  Start Milestone 228
Minimum Condition Date Constraints 275, 283
Moment in Time 59, 125–134, 146, 194, 196, 240, 300, 351, 352
Momentology 6, 72
  Momentum Control 72
  Momentum Management 3, 6
Monitored Paths Ranking System 236
Most Likely Activity Duration 424
Most Optimistic Activity Duration 216, 347, 424
Most Pessimistic Activity Duration 216, 424
Multi-Path Residency 267–268
Natural Restriction 81
Network-Based Schedule 12, 37, 82, 179, 290, 363
Network-Based Scheduling 199
Network Schedule 69, 263. See Network-Based Schedule
Non-Fragmented Activity 197
Open Start Condition 241
Operational Context 2, 355, 431
Optimization Plans 25
Optional Schedule Elements 121
Ordinals 346, 351
Original Duration 206, 388–389, 394, 396, 397, 417, 425
Paramount Path 234, 236–237, 241–248
Participant Perspective 23, 24
Passage of Time  38, 70, 104–108, 197, 365, 370, 371
Past Period  345–351, 365, 371
Past Segment Data  357
Path Ends  239–241, 268–269
Path Finish  239–241, 269, 288
Path Start  239–246, 269, 286, 288
Path Tracing  290, 388
PDM. See Precedence Diagramming Method
Percent Complete  13, 324–325, 357–360, 385–388, 393
Performance Determinant  76–80
Performance Restriction  78–79, 81–93, 100, 101, 115–116, 120, 129, 142, 145, 155, 274, 279, 415, 432
PERT. See Program Evaluation and Review Technique
PERT Diagram. See also Network Diagram
PINS. See Points of Navigational Significance
Planner  27
Planning and Scheduling  1, 6, 9, 25–28, 336, 337, 405
Point in Time  42, 55, 99, 94
Point-of-Day Perspective  12, 135–136, 212, 299, 307, 346
Point of Origin  66, 92
Point of Termination  66, 92
Precedence Diagramming Method  84, 89, 199, 415–419
Predictive Data  359–364, 367–369, 386, 401, 405
Procurement Schedule  26
Program Evaluation and Review Technique  14, 417–419, 424
Progress Override  401–407
Project  4, 23–24
Project Budget  322–328
Project Length  331, 359, 360, 374, 376, 377, 405, 428, 433
Project Life Cycle  8, 13, 25, 347, 380, 384
Project Phases  237, 309
Project Players  356
Project Scope  33, 72, 127, 207, 321, 328
Project Collaboration. See Collaboration
Project Communication. See Communication
Project Controls  8
Project Coordination. See Coordination
Project Execution Plans  25–26
Project Execution Strategy Recalibration  385
Project Facilitation Analyst  27
Project Facilitation Planner  27
Project Facilitation Scheduler  27
Project Management  1–14, 23–27, 36,
Project Management Domains 3, 38
Project Performance 322, 338–341, 347, 357, 365, 370–371
Project Performance Determinant 76–80
Provided Schedule Elements 121–123
RDCPM. See Relationship-Driven CPM (RDCPM)
Relationship Types (Cognitive) 75–89, 88, 180, 263–265, 432
Relationship Types (Dominant) 88, 115. See also Restriction Type
Remaining Duration 13–14, 206, 278, 385–399, 425
Resource Management 331–334
Restriction. See Performance Restriction
Restriction Abbreviation 91
Restriction Arrow 56
Restriction Delay Stipulations 108–111
Restriction Flow 85–86, 90–92, 127
Restriction Name 91
Restriction Statement 92, 93, 99, 100, 127–129
Restriction Type 82, 89, 103, 144, 151, 155, 165, 170, 172, 394
Retained Logic 401–402
Scale of Path Criticality 236
Schedule
Schedule Flexibility 216–217
Schedule Forms 15, 30, 34, 69
Schedule Notations 314
Schedule Content 337, 341, 343, 355, 369, 385–386
Schedule Control 1, 25, 72, 377
Schedule Credibility 13, 371
Schedule Data 13, 274, 308, 365–371, 376, 384, 401
Schedule Database 314
Schedule Data Credibility Profile 13, 365–371, 376
Schedule Data Perspective 353–359
Schedule Design 14, 433
Schedule Edition 235, 265–266, 342–
Supplemental Schedule Elements 12, 273–274
Symbiotic Relationship 2, 368, 369
Tabular Listing 309
Technology 2, 8, 25, 28, 73, 411, 417–421, 431, 433
Temporal End Point 116
Time Management 1, 24, 28, 36, 108–109, 217, 322, 326, 327, 421, 430
Time Now Date. See Data Date
Timetable 15, 30, 34, 40, 54, 71, 87
Total Float 8, 37, 71, 121, 142, 179–222, 251–262
Most Critical Total Float 207–210, 217, 288, 290, 292, 299
Unit of Time 40
WBS. See Work Breakdown Structure
Work Breakdown Structure 12, 311, 312, 313, 314, 319, 320, 321, 328, 432
Work Performance 40, 92, 104–106, 330, 353, 358
Work Performance Calendar 293, 294, 295, 297, 299, 331, 426, 432, 434
Work Scope 30–32, 41–42, 70, 326, 327, 357
Work Scope Element 15, 30, 32, 34
Zero Free Float 265, 303–305, 306–308
Zero Sum Game 77–80
Zero Total Float 217, 303–308