K2's Excel Best Practices

Continuing Professional Education Series

K2 Enterprises www.k2e.com

K2E Canada Inc. www.k2e.ca

K2 Enterprises
1905 W. Thomas St., Suite D #363
Hammond, LA 70401-2901
985.542.9390

K2E Canada Inc.
174 Brewer Court
Burlington, Ontario, L7L4G4
905-746-1581

CONTRIBUTORS: The shareholders, associates, and staff of K2 Enterprises and K2 Canada Inc. contributed to producing these materials.

Table of Contents

Introduction

Three Rules of this Seminar

1. **There Are No Dumb Questions.** Please ask questions if you need further explanation. We welcome your questions and will do our best to provide accurate and meaningful answers.

2. **Obtain the Information for Which You Came.** If you have a specific interest in a topic, please let us know. If your topic is not part of the course materials, we will do our best to accommodate your request. Your instructor is anxious to meet your needs.

3. **Share and Share-Alike.** Your instructor is here to share relevant information about the course. We hope that you will be willing to do the same. If you have helpful tips or experiences, please share them with the other participants and us.

Course Development

The shareholders, associates, and staff of K2 Enterprises and K2E Canada Inc. developed this course and the related course materials. We aim to deliver high-quality educational practices that help you use and understand the tools and concepts discussed in this session more effectively.

We designed our course presentation style for busy professionals. Through numerous short case studies and feature reviews, we deliver concise, easy-to-understand, easy-to-absorb information.

To maximize your time and the value of attending this course, we have carefully filtered out trivial and obscure features. Our goal is not to cover every keystroke and menu option, but to provide a fast-paced course that focuses on state-of-the-art information technology and how it can benefit you and your staff.

About K2 Enterprises

For over three decades, our team members have delivered more than 45,000 technology-focused seminars, webinars, and conferences worldwide. To stay on top of technology, our team members read numerous technology trade magazines, attend conferences, and speak with technology companies each year to stay current with the latest information for our courses. Also, we experience everything we teach and recommend to our audiences. Because our shareholders and associates are based in different states and provinces, using technology efficiently and appropriately is essential to our business's success and survival.

Our Mission Statement

K2 aims to develop and deliver the highest-quality technology seminars and conferences for business professionals. We work cooperatively with professional organizations such as state CPA societies, associations of Chartered Professional Accountants, and technology vendors. We make every effort to maintain high integrity, family values, and friendship among all involved.

Our Instructional Team

For a complete listing of instructors and bios, please visit:

K2 Enterprises

Instructors - K2 Enterprises

K2E Canada Inc.

Instructors - K2E Canada Inc

The K2 Seminar Curriculum

Full-day Seminars

- K2's Advanced Excel
- K2's Business Intelligence, Featuring Microsoft's Power BI Tools
- K2's Case Studies In Fraud And Technology Controls
- K2's Excel Best Practices
- K2's Excel Essentials For Staff Accountants
- K2's Excel PivotTables For Accountants
- K2's Excel Tips, Tricks, And Techniques For Accountants
- K2's Next Generation Excel Reporting
- K2's QuickBooks For Accountants
- K2's Small Business Internal Controls, Security, And Fraud Prevention And Detection
- K2's Technology For CPAs – Don't Get Left Behind

Half-day Seminars

- K2's Artificial Intelligence For Accounting And Financial Professionals
- K2's Best Word, Outlook, And PowerPoint Features
- K2's Better Productivity Through Artificial Intelligence and Automation Tools
- K2's Case Studies In Fraud And Technology Controls
- K2's Data Analytics For Accountants And Auditors
- K2's Ethics And Technology
- K2's Excel Charting And Visualizations
- K2's Implementing Internal Controls In QuickBooks Online Environments
- K2's Improving Productivity With Microsoft 365Cloud Applications And Services
- K2's Mastering Advanced Excel Functions
- K2's Microsoft Teams
- K2's Securing Your Data: Practical Tools For Protecting Information
- K2's Technology Update
- K2's Top PDF Features You Should Know

Besides the seminars listed above, K2 also produces numerous one and two-day technology conferences annually. You can find a complete listing of our course offerings, including dates, locations, and course descriptions, on our website at:

K2 Enterprises
Technology Conference - K2 Enterprises

K2E Canada Inc.
Technology Conference - K2E Canada Inc.

On-Site Training

Working cooperatively with state CPA organizations and associations of Chartered Professional Accountants, we are pleased to offer all our seminars and conference sessions as on-site training opportunities. On-site training provides companies of all sizes with numerous advantages, including the following.

- ***Customization***. K2 can customize the content of all on-site training events to meet your organization's specific needs. For example, we can combine selected parts of existing courses into a focused learning experience to meet your team's needs. Likewise, we can develop custom content from the ground up and deliver it to your team members, ensuring we meet your training objectives.

- ***Cost savings***. For organizations of all sizes, on-site training can provide significant cost savings compared to public training venues. In addition to potentially reduced registration fees, these cost savings materialize in the form of reduced travel costs and reducing the amount of time team members spend away from the office.

- ***Convenience***. Because you select your on-site training event's date, time, and location, on-site training is more convenient for you and your team than traditional training options. Many organizations schedule on-site training with staff meetings, retreats, and customer/client events to leverage the value of all participants' time.

Customization, ***cost savings***, and ***convenience*** are the three C's of "training to go." For more information on how you and your organization can experience these benefits or schedule an on-site training event, please visit:

K2 Enterprises
On-site Training - K2 Enterprises

caroline@k2e.com

K2E Canada Inc.
Seminars - K2E Canada Inc

caroline@k2e.ca

Web-Based Training

K2 offers web-based CPE and CPD to accounting and business professionals throughout the United States and Canada. K2's award-winning instructors deliver two-, four-, and eight-hour webinars on numerous technology-focused topics.

Additionally, K2 provides on-demand training. On-demand courses offer you the advantage of receiving top-quality training at a time and location that fits your busy schedule. You can start a class today and complete it in the future. For more information, please visit:

For more information on either platform, please visit:

K2 Enterprises
Webinars - K2 Enterprises

K2E Canada Inc.
Webinars - K2E Canada Inc

K2 Internet Sites

K2 maintains multiple websites containing information and links relevant to CPAs and other business professionals as a service to the profession. Further, the content of these sites can enrich your experience during and after our seminars. In addition, our pages contain links to some of the top accounting and content management software solutions and even to a few sites that are just fun places to visit. If you have any comments or questions regarding our web pages, please email *webmaster@k2e.com*.

K2 has two primary websites. You can view information regarding CPE and CPD opportunities, access a library of technology tips and articles, and link to K2-related websites.

www.k2e.com

www.k2e.ca

www.cpafirmtech.com

CPA Firm Technology contains technology-related information explicitly targeted at public accounting firms. We include product reviews, hardware and operating system discussions, affinity programs for CPA firms, and Cloud computing resources.

www.accountingsoftwareworld.com

Accounting Software World provides information related to accounting and financial management solutions. Here, you can find software and service reviews, product comparisons, white papers, and other material of interest to those seeking information about accounting software and service solutions.

www.totallypaperless.com

For information about document management and paperless processes, please visit **Totally Paperless**. In addition, you will find reviews on hardware and software used by all types of businesses in document imaging and management processes.

Designing And Constructing Spreadsheets

Chapter 1

Many Excel users begin constructing a spreadsheet by opening a blank Excel workbook and entering data and formulas with little forethought regarding design, control, and accuracy. While in the short term, this might seem efficient, over time, this practice can lead to lost productivity and, more damagingly, erroneous output. This chapter addresses and provides best practices for designing and constructing spreadsheets.

Upon completing this chapter, you should be able to:

- List five fundamental rules of spreadsheet design;
- Describe how to use specific techniques such as hyperlinks, IFERROR, conditional formatting, and Inquire to advance the rules of spreadsheet design; and
- Explain how to use default and application templates to streamline creating Excel workbooks.

Understanding Five Fundamentals of Spreadsheet Design

Documenting Your Workbook

While it may seem like this is stating the obvious, you should add at least some minimal documentation to each Excel workbook you create. Documenting the workbook will help you and others understand the workbook's nature, purpose, scope, and status. The level of documentation will vary based on many factors, including those listed below:

- The complexity of the workbook,
- The sensitivity of the data in the workbook, and
- Whether others use the workbook.

However, at a minimum, you should incorporate each of the following items into every Excel workbook you create:

- Your name and contact information,
- The general purpose of the workbook, and
- Any special considerations associated with the workbook.

You can easily document each of the three items listed above in Excel's **Backstage View**, shown in **Figure 1**. You can access the Backstage View by clicking the **File** tab of the Ribbon. Upon doing so, Excel presents you with the Backstage View.

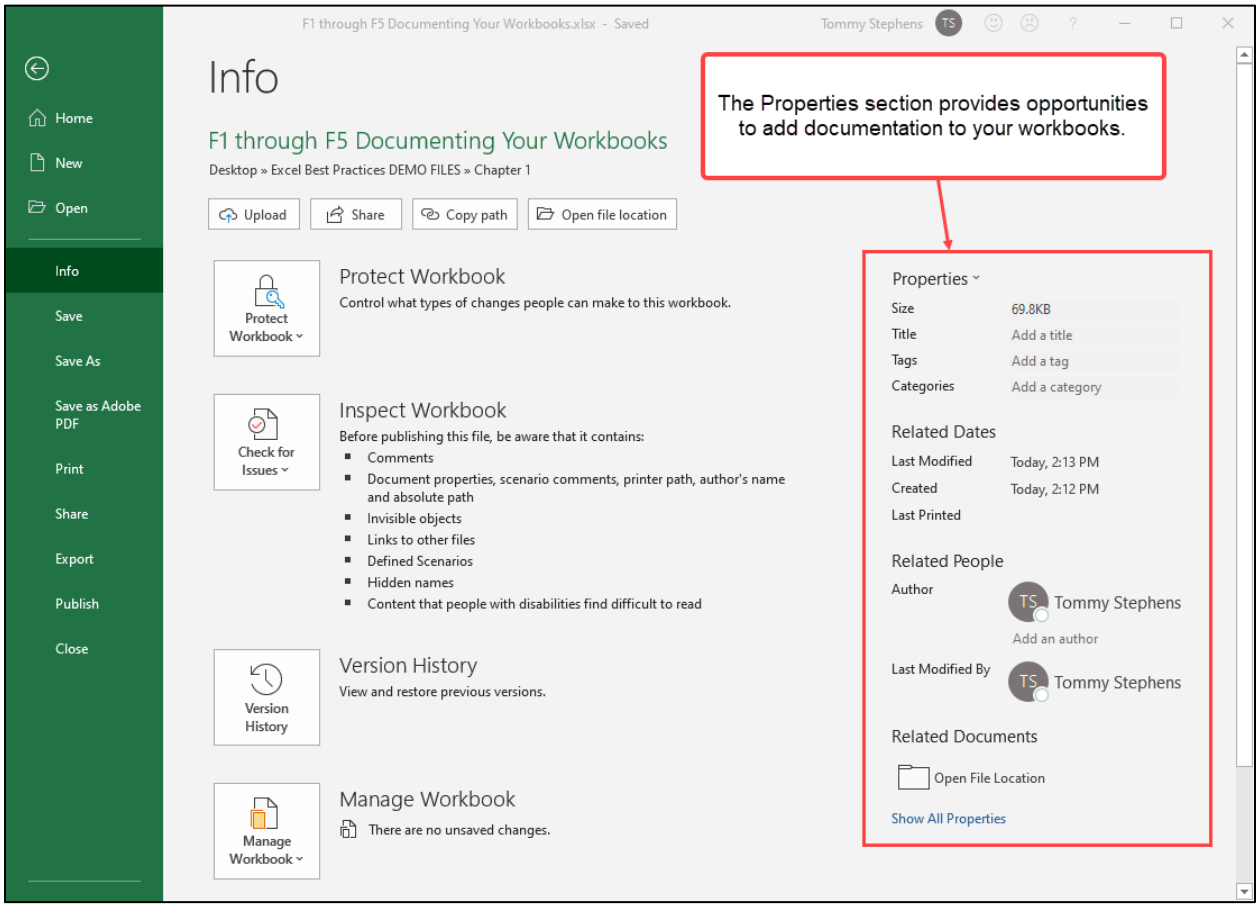


Figure 1 - Accessing Properties from Excel's Backstage View

You can edit each property in Backstage View; however, if you click the dropdown arrow next to **Properties**, you can access an even deeper level of documentation elements known as **Advanced Properties**, as shown in **Figure 2**. You may enter details such as who checked the workbook, client information, and other necessary documentation.

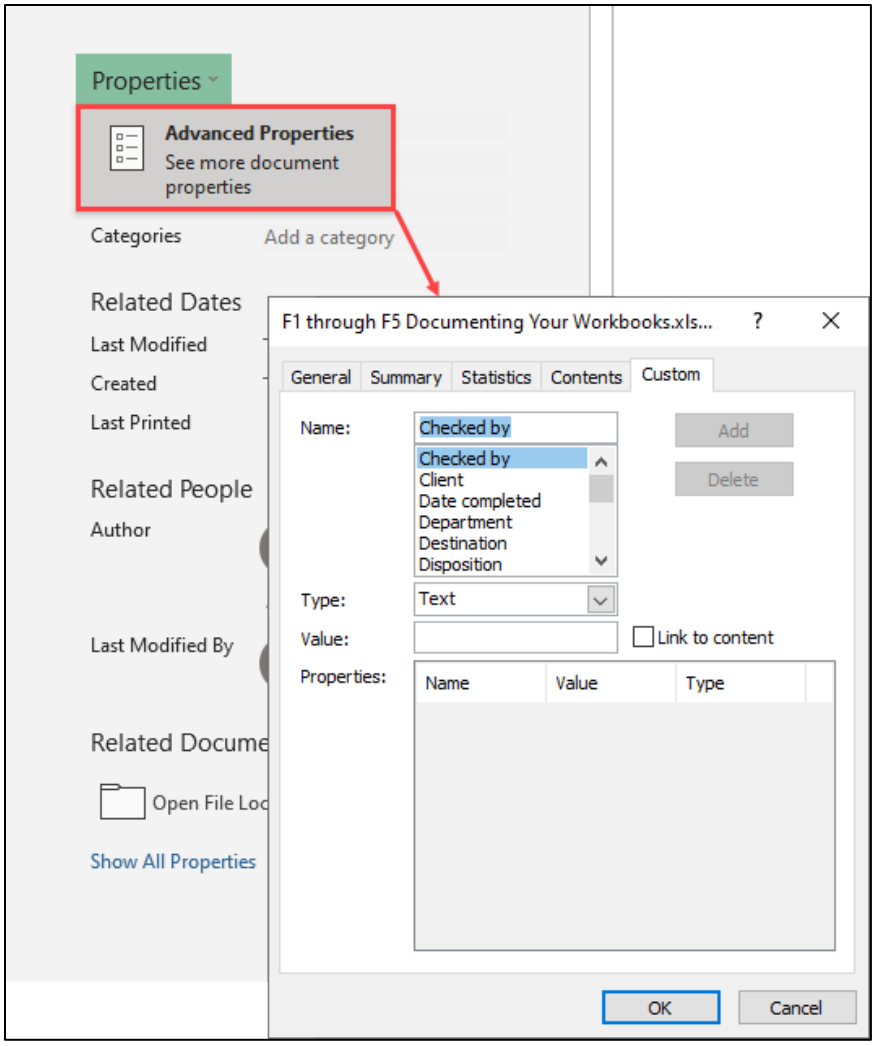


Figure 2 - Accessing Excel's Advanced Properties

Additionally, document the key assumptions that drive the formulas in your workbooks. Common methods used for this task include adding ***Comments*** or ***Notes*** to cells. While there are numerous ways to insert comments and notes, perhaps the easiest is to right-click a cell and choose **New Comment** or **New Note** from the resulting pop-up menu. As shown in **Figure 3**, adding a comment or note makes the origin of information for the assumption abundantly clear to all who access the spreadsheet.

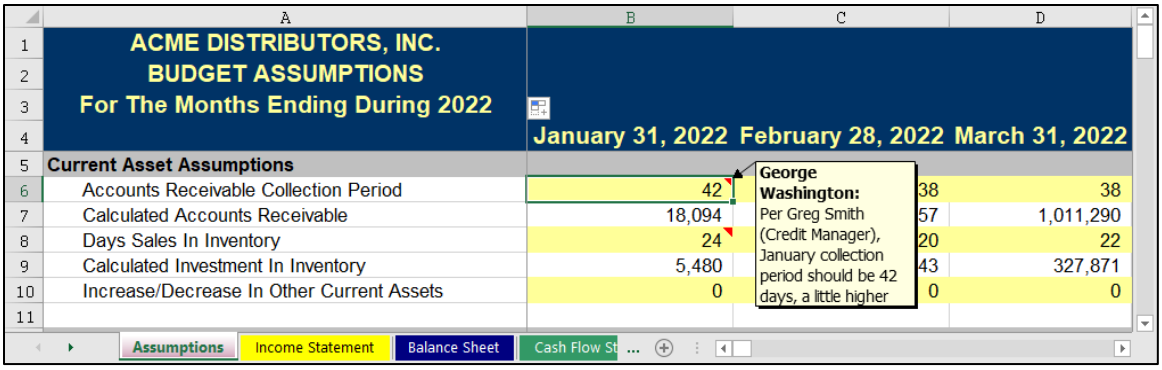


Figure 3 - Using Comments to Document Assumptions

One oft-cited reason for not using comments and notes is that they may distort the workbook's appearance. However, accessing Excel's Advanced options allows you to control how comments and notes appear in workbooks. As shown in **Figure 4**, perhaps the most helpful setting is that of *Indicators only, and comments on hover*. With this option enabled, a cell containing a comment or note will only show a small indicator in the upper right corner until you hover your mouse over that cell. At that point, the comments and notes appear and disappear when you move your mouse to another cell.

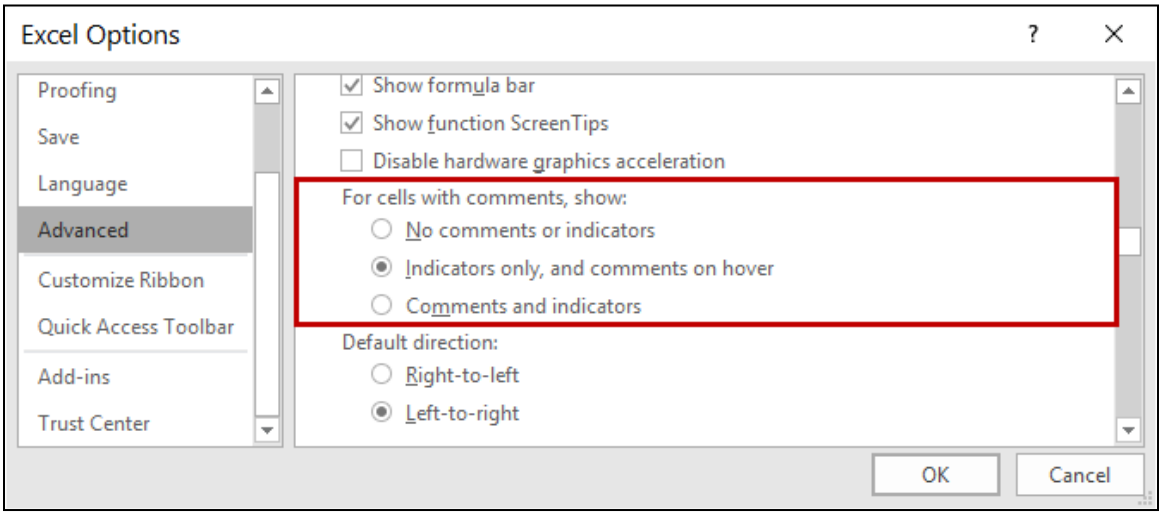


Figure 4 - Controlling How Comments Appear in Workbooks

Likewise, you can control how comments and notes print by accessing the **Page Setup** dialog box from the Ribbon's **Page Layout** tab. As shown in **Figure 5**, you can suppress them from printing, indicate that all comments and notes should print at the end of the print sequence, or print comments and notes as they appear on the worksheet.

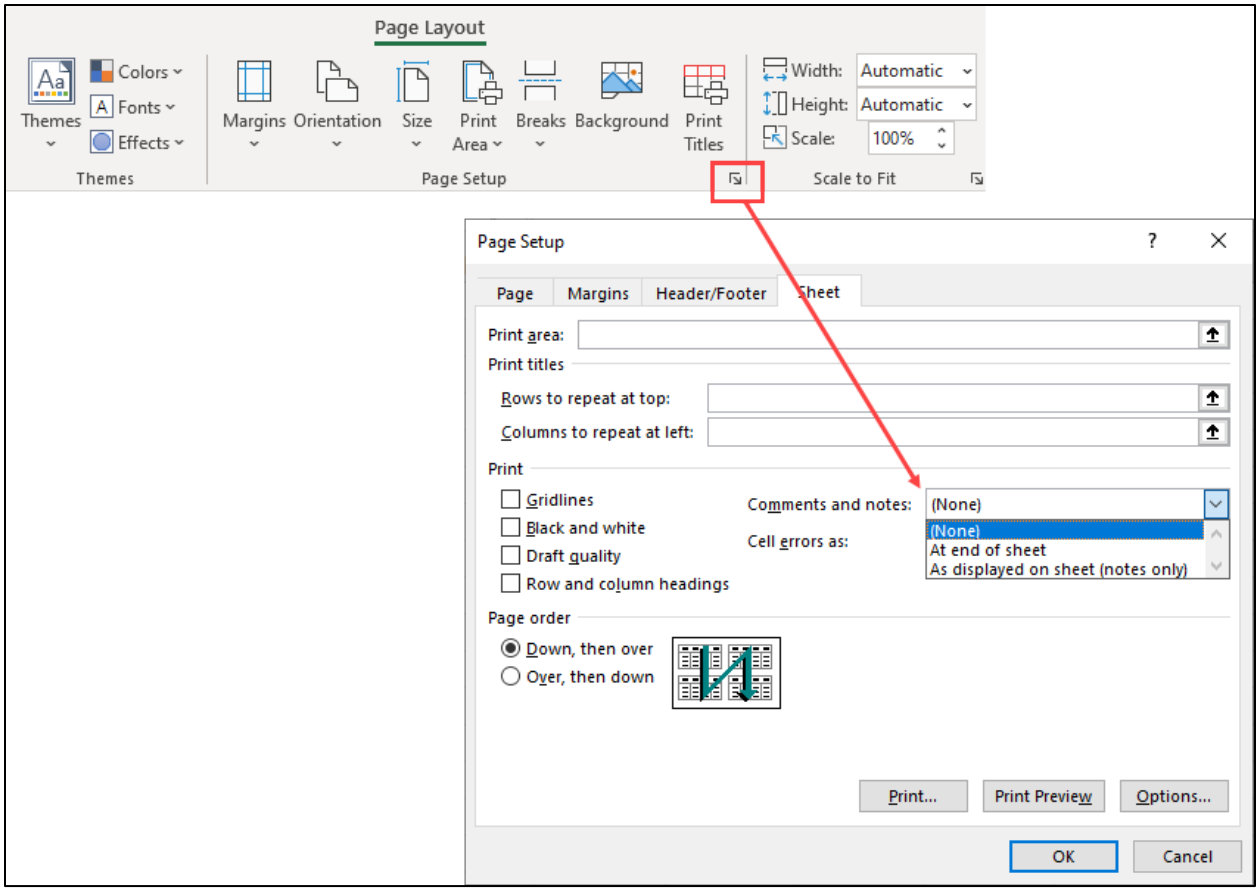


Figure 5 - Controlling How Comments Appear When Printed

Using Excel's Inquire *Tool to Generate Documentation*

Dating back to the 2013 release of Excel, Microsoft added the ***Inquire*** tool to the **Microsoft Office Professional Plus** version of the suite and **Office Pro Plus** subscribers in Microsoft 365/Office 365 subscriptions. Inquire provides a set of tools that allows you to, among other things, document certain aspects of your workbooks. This feature offers seven tools to assist you when designing and documenting your spreadsheets.

1. Workbook Analysis
2. Workbook Relationship
3. Worksheet Relationship
4. Cell Relationship
5. Compare Files
6. Clean Excess Cell Formatting
7. Workbook Passwords

Each of these tools is available from the **Inquire** tab of the Ribbon.

If you run a version of Excel that provides Inquire but do not see the **Inquire** tab, you most likely need to activate the Inquire add-in. To do so, click **File**, **Options**, and **Add-ins**. Select **COM Add-ins** in the **Manage** box and then click **Go**. Finally, check the box next to **Inquire**.

Workbook Analysis

The **Workbook Analysis** tool provides an Excel-based report detailing a workbook's structure, formulas, cells, ranges, and warnings. To use the Workbook Analysis feature, click **Workbook Analysis** on the Ribbon's **Inquire** tab to open the **Workbook Analysis Report** dialog box shown in **Figure 6**.

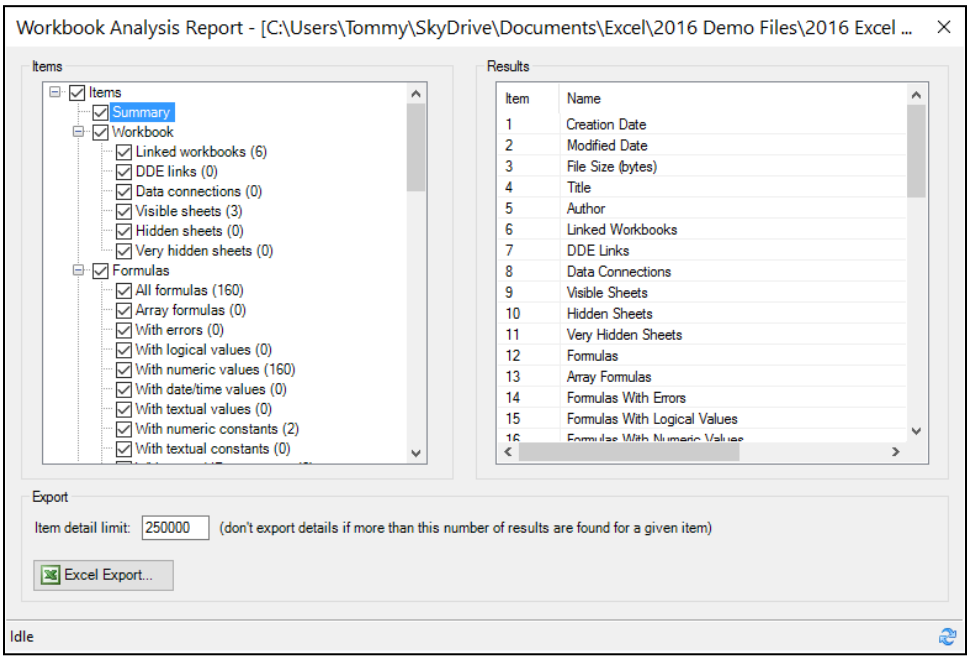
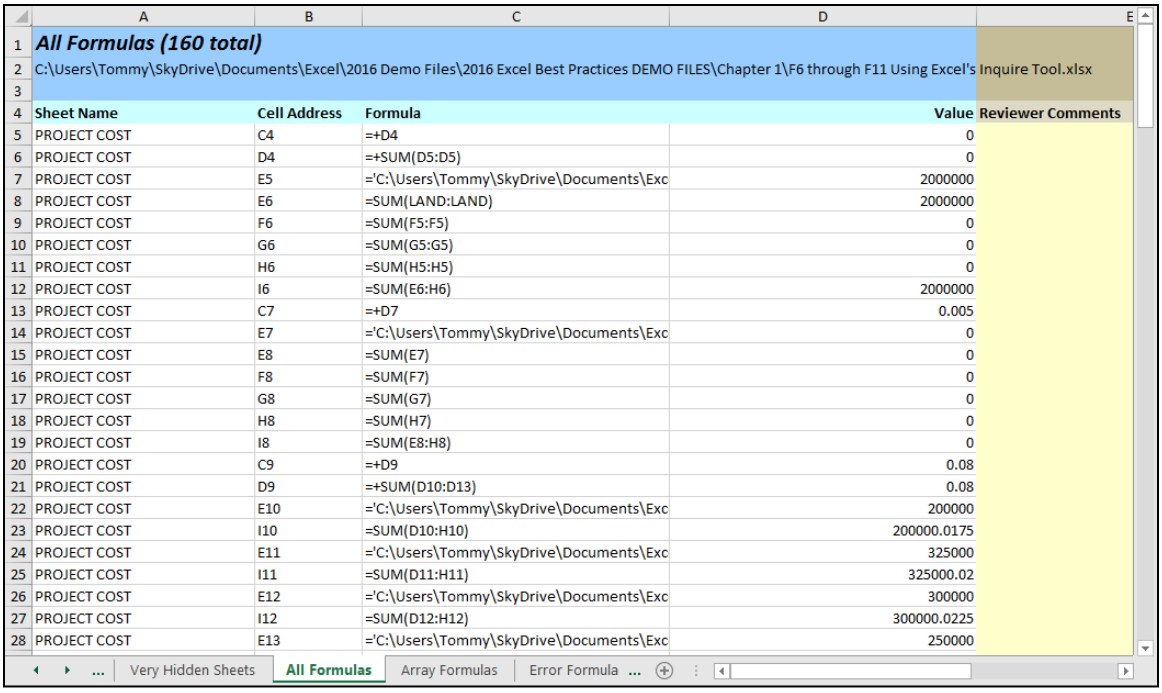


Figure 6 - Workbook Analysis Dialog Box

In the Workbook Analysis dialog box, select the items for which you want to generate documentation and click the **Excel Export** button in the dialog box's lower-left corner. Upon doing so, Excel creates a separate workbook containing the results of your query. As shown in **Figure 7**, the output workbook contains a separate worksheet tab for each item you selected for your query.



	A	B	C	D	E
1	***All Formulas (160 total)***				
2	C:\Users\Tommy\SkyDrive\Documents\Excel\2016 Demo Files\2016 Excel Best Practices DEMO FILES\Chapter 1\F6 through F11 Using Excel's Inquire Tool.xlsx				
3					
4	**Sheet Name**	**Cell Address**	**Formula**	**Value**	**Reviewer Comments**
5	PROJECT COST	C4	=+D4	0	
6	PROJECT COST	D4	=+SUM(D5:D5)	0	
7	PROJECT COST	E5	='C:\Users\Tommy\SkyDrive\Documents\Exc	2000000	
8	PROJECT COST	E6	=SUM(LAND:LAND)	2000000	
9	PROJECT COST	F6	=SUM(F5:F5)	0	
10	PROJECT COST	G6	=SUM(G5:G5)	0	
11	PROJECT COST	H6	=SUM(H5:H5)	0	
12	PROJECT COST	I6	=SUM(E6:H6)	2000000	
13	PROJECT COST	C7	=+D7	0.005	
14	PROJECT COST	E7	='C:\Users\Tommy\SkyDrive\Documents\Exc	0	
15	PROJECT COST	E8	=SUM(E7)	0	
16	PROJECT COST	F8	=SUM(F7)	0	
17	PROJECT COST	G8	=SUM(G7)	0	
18	PROJECT COST	H8	=SUM(H7)	0	
19	PROJECT COST	I8	=SUM(E8:H8)	0	
20	PROJECT COST	C9	=+D9	0.08	
21	PROJECT COST	D9	=+SUM(D10:D13)	0.08	
22	PROJECT COST	E10	='C:\Users\Tommy\SkyDrive\Documents\Exc	200000	
23	PROJECT COST	I10	=SUM(D10:H10)	200000.0175	
24	PROJECT COST	E11	='C:\Users\Tommy\SkyDrive\Documents\Exc	325000	
25	PROJECT COST	I11	=SUM(D11:H11)	325000.02	
26	PROJECT COST	E12	='C:\Users\Tommy\SkyDrive\Documents\Exc	300000	
27	PROJECT COST	I12	=SUM(D12:H12)	300000.0225	
28	PROJECT COST	E13	='C:\Users\Tommy\SkyDrive\Documents\Exc	250000	

Figure 7 - Sample Output of the Workbook Analysis Feature

Workbook Relationship

Choosing Inquire's **Workbook Relationship** feature allows you to visualize all other workbooks that link to or from the workbook with which you work. As shown in **Figure 8**, the Workbook Relationship tool generates a graphical representation of how the current workbook relates to other workbooks, including providing selected metadata about all linked workbooks.

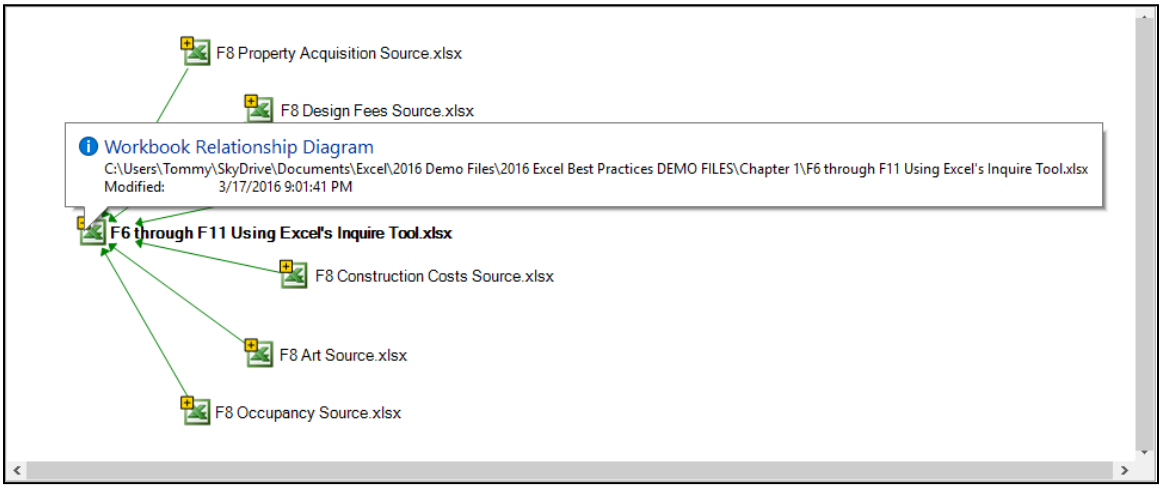


Figure 8 - Sample Output from Excel's Workbook Relationship Feature

Worksheet Relationship

Like the Workbook Relationship tool, Inquire's **Worksheet Relationship** feature visually depicts all worksheets' relationships in the selected workbook. Further, as shown in **Figure 9**, to the extent that external worksheets affect worksheets in the workbook, they are included in the visualization.

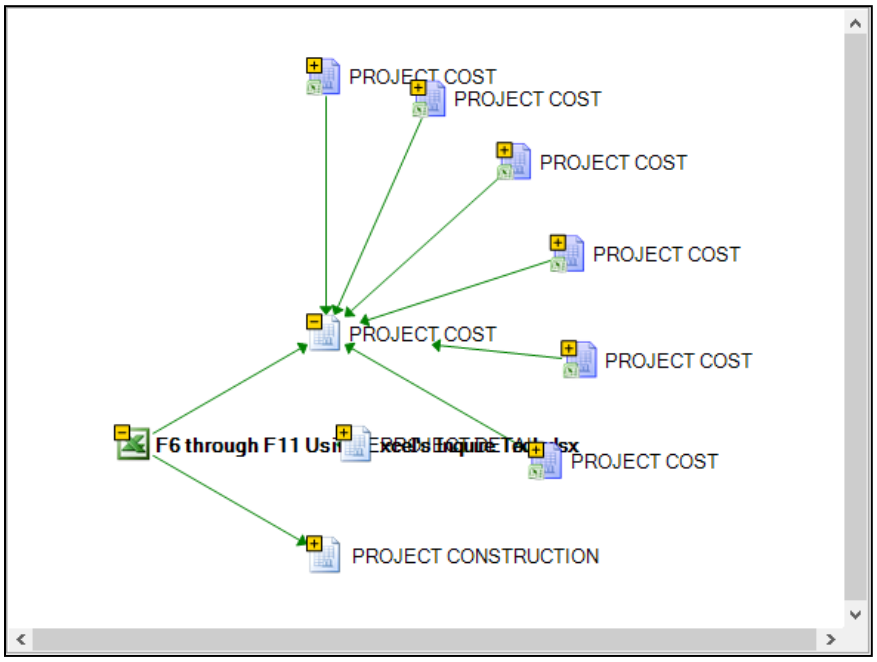


Figure 9 – Sample Output from Excel's Worksheet Relationship Feature

Cell Relationship

As its name implies, the **Cell Relationship** feature included in Inquire creates a diagram of how a specific cell relates to other cells. Upon selecting Cell Relationship from the Ribbon's Inquire tab, Excel presents the **Cell Relationship Diagram Options** dialog box in **Figure 10**.

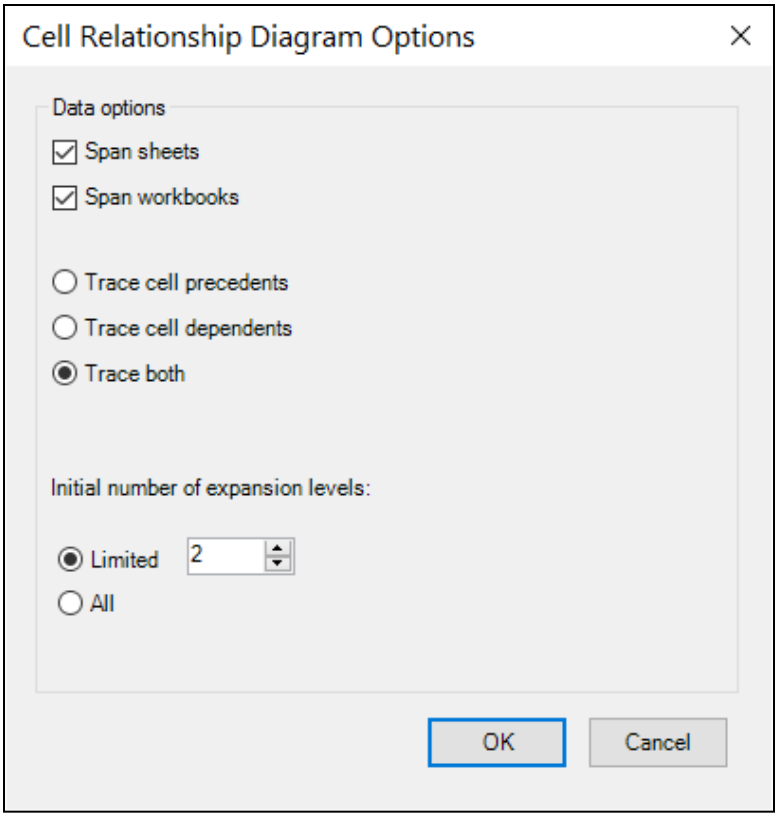


Figure 10 - Cell Relationship Diagram Options Dialog Box

In the Cell Relationship Diagram Options dialog box, enter whether you want your diagram to extend into other worksheets and workbooks, precedents or dependents shown, and how many levels of detail you want Inquire to generate. Once you select, click **OK** to create a graphic similar to the one shown in **Figure 11**. As you can see, the graphic illustrates the relationships between the current cell and all related cells.

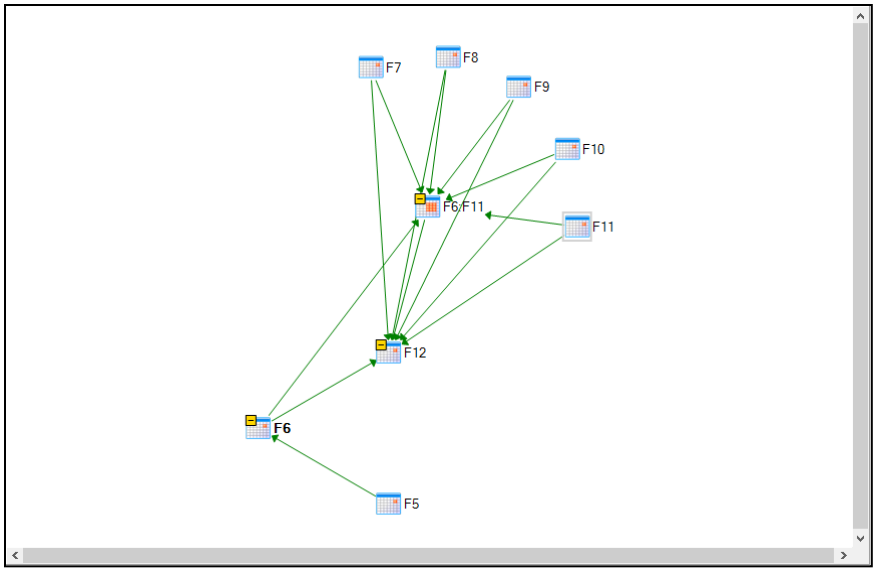


Figure 11 – Sample Output from Excel's Cell Relationship Feature

Compare Files

Inquire's **Compare Files** feature provides capabilities long sought by many users. For example, you can electronically compare two workbooks with Compare Files and instantly identify all differences between the compared workbooks. To use Compare Files, begin by opening the two workbooks you want to compare. Then, choose **Compare Files** from the **Inquire** tab of the Ribbon and select the two workbooks you want to compare, as shown in **Figure 12**.

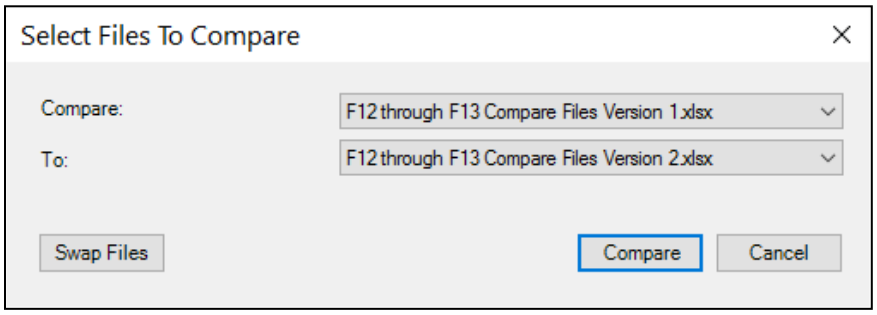


Figure 12 - Selecting Excel Workbooks to Compare

Upon clicking **Compare** in the lower right corner of the **Select Files To Compare** dialog box, Excel analyzes the two workbooks and generates a detailed analysis highlighting differences in data, formulas, formatting, and other characteristics. **Figure 13** presents a sample of that analysis.

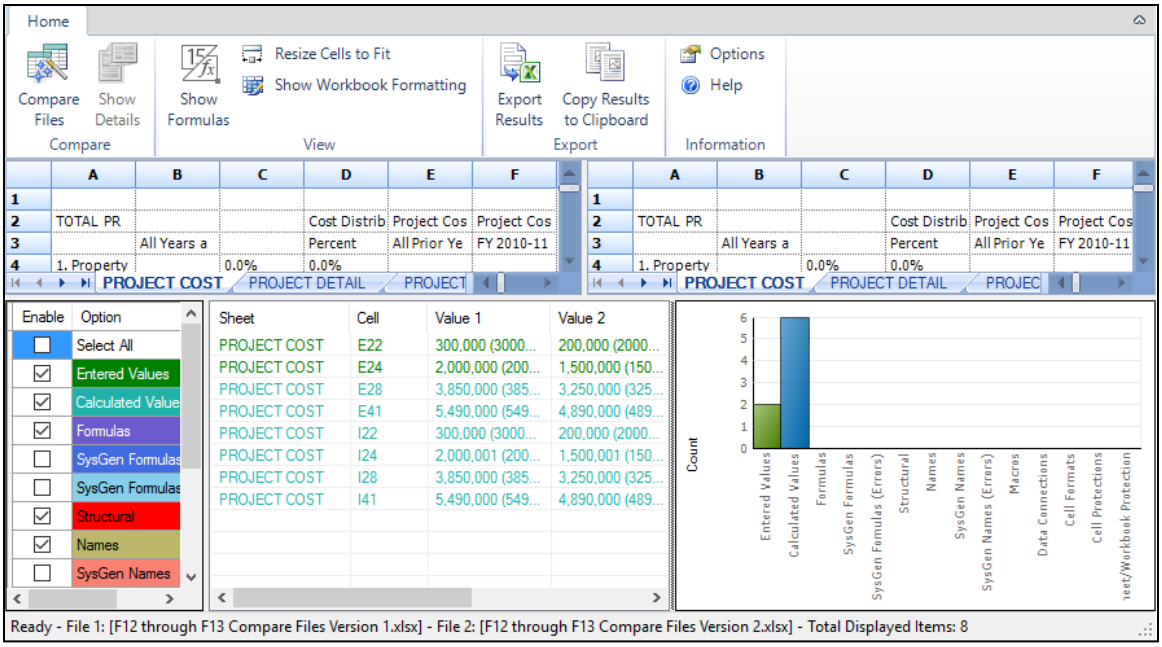


Figure 13 - Sample Output from Excel's Compare Files Feature

Clean Excess Cell Formatting

On occasions, workbooks become "bloated" with excess cell formatting. For example, suppose a user applies conditional formatting to an entire worksheet column, yet data only extends to row 100 in that column. In that case, all the conditional formats from row 101 through row 1,048,576 are extraneous and unnecessary. Further, these excess formats occupy resources and can cause the workbook to open, save, and recalculate more slowly. Excessive formatting can also contribute to printing issues, including causing blank pages to print. The **Clean Excess Cell Formatting** tool addresses these issues by removing cells from the worksheet beyond your data's last cell.

> ⓘ You can identify the active worksheet's last cell by pressing **CTRL+SHIFT+END** or selecting **Find & Select** on the Ribbon's Home tab, **Go To Special**, and **Last Cell**.

When you choose **Clean Excess Cell Formatting** from the Ribbon's **Inquire** tab, Excel displays the **Clean Excess Cell Formatting** dialog box in **Figure 14**. In this dialog box, indicate whether you want to remove excess formatting from all worksheets in the current workbook or just the active sheet, then click **OK**. Upon doing so, Excel analyzes the selected sheet(s) and strips them of any excess formatting.

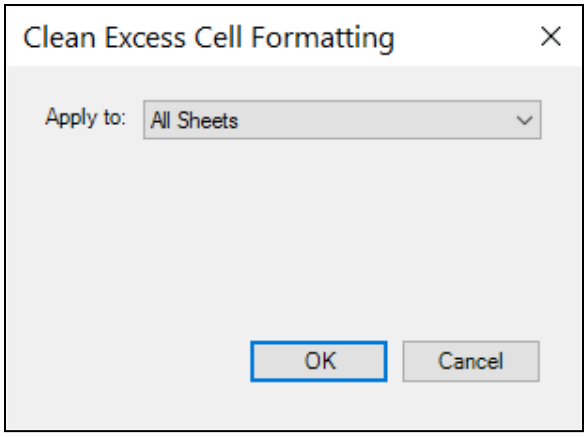


Figure 14 - Clean Excess Cell Formatting Dialog Box

Workbook Passwords

The last feature associated with Inquire is that of **Workbook Passwords**. Using this feature, you can store passwords associated with workbooks for which you intend to use the Workbook Analysis or Compare Files tools repeatedly. For security purposes, Excel encrypts these passwords.

Managing Spreadsheet Assumptions

A frequent spreadsheet design flaw is embedding assumptions into formulas. To illustrate, compare the two formulas presented in **Figure 15**. The left formula contains embedded assumptions, whereas the one on the right uses cell references to them. If you or another user of this spreadsheet wanted to change the interest rate to 4.5% and see the impact on the monthly payment, which scenario would be easier to edit? Clearly, "what-if" analysis is easier with the design on the right because changing assumptions requires only editing assumption cells in the spreadsheet. Changing assumptions mandates editing the entire formula with the scenario on the left.

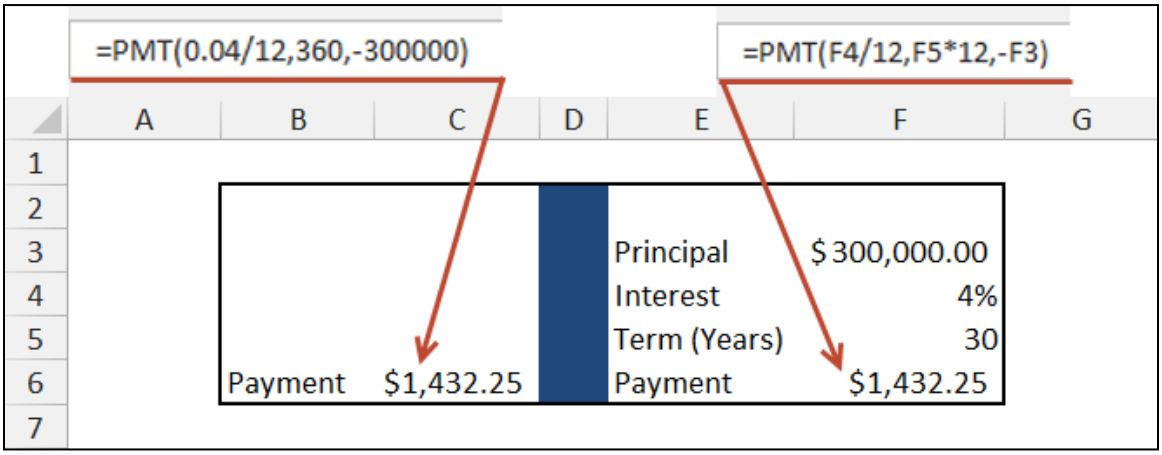


Figure 15 - Comparing Formulas with and without Embedded Assumptions

In addition to the ease of editing, changing assumptions in the scenario on the left presents another issue – it is more prone to error. Whenever users edit formulas, there is a risk that they will inadvertently alter the formula or its components, resulting in an incorrect formula result. Consequently, to help ensure that the integrity of formulas remains intact, you should never force other users to edit formulas to change assumptions.

Another issue associated with embedding assumptions in formulas is that they never appear anywhere in a workbook. Thus, unless a user has access to the original Excel workbook, all embedded assumptions are invisible to that user, making it more difficult for them to comprehend the workbook's entire construct.

Also related to managing assumptions, each assumption should appear only once in a workbook. For example, the interest rate shown in cell F4 in Figure 15 should only appear in that cell, and no other cell should provide an opportunity to enter the same variable. Suppose there are multiple entry points for the same variable. In that case, the risk that a user changes it in some locations but not all increases, leading to some calculations relying on erroneous input.

Lastly, clearly label all assumptions in a spreadsheet so that you or other users will not mistake what data they need to enter. The scenario on the right side of Figure 15 shows the labels for each assumption, removing any doubt about which information belongs in cells F3 through F5.

Easing Workbook Navigation

If your workbook contains many worksheets or some worksheets contain large volumes of data, ensure it is easy to navigate. Multiple options exist to facilitate navigation, including inserting tables of contents and hyperlinks, and standardizing workbook and worksheet layouts.

Inserting Tables of Contents and Hyperlinks

Hyperlinks are commands that "jump" to other locations. Most Excel users are familiar with and use hyperlinks to navigate through the Internet, but few use them to navigate workbooks. Inserting a table of contents with hyperlinks like the one shown in **Figure 16** can make navigating the workbook easier for many users.

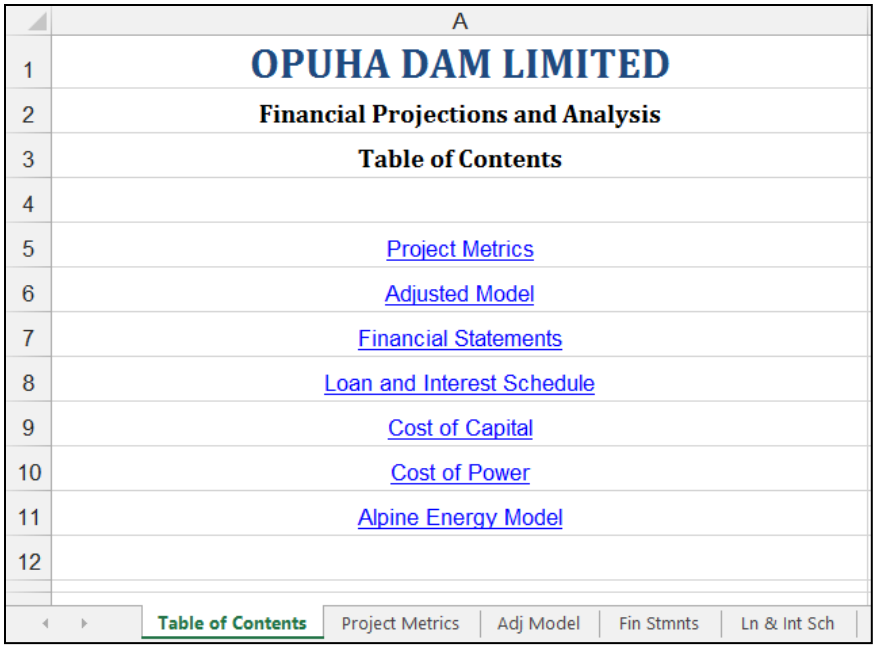


Figure 16 - Sample Table of Contents in a Workbook

To create the hyperlinks shown in cells A5 through A11, open the **Insert Hyperlink** dialog box pictured in **Figure 17**. You can choose **Hyperlink** from the Ribbon's **Insert** tab using the **CTRL+K** keyboard shortcut or right-click in a cell and choose **Hyperlink** from the resulting pop-up menu.

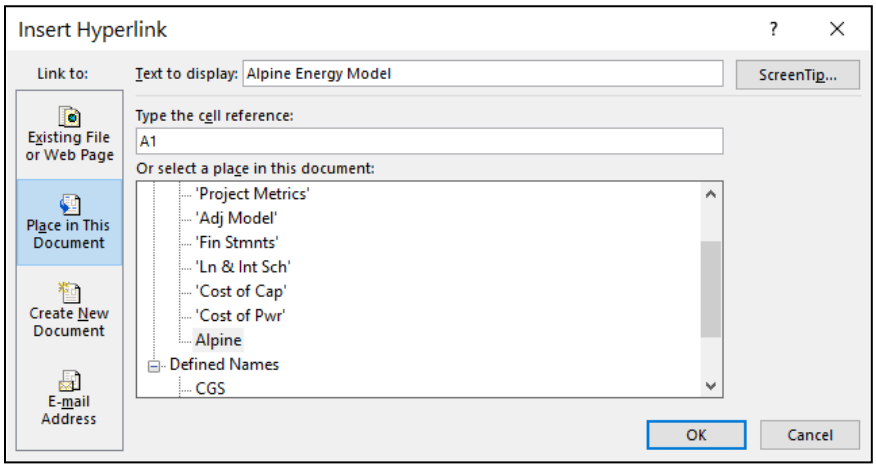


Figure 17 - Inserting a Hyperlink in Excel

In the **Insert Hyperlink** dialog box, choose to link to a **Place in This Document** and add the **Text to display** and the location to which the hyperlink should point.

If you insert a table of contents with hyperlinks to make navigating through a workbook easier, you should include return hyperlinks to take users back to the table of contents. The hyperlink in cell **O1** of **Figure 18** provides an example of this best practice.

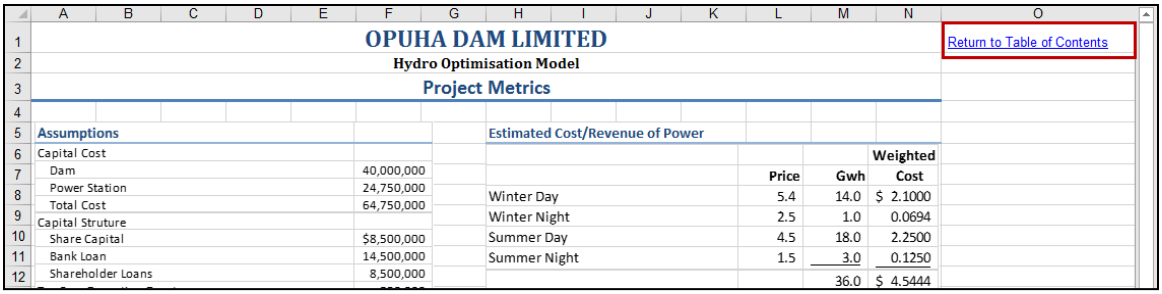


Figure 18 - Hyperlink to Return to Table of Contents

Hyperlinks can point to locations other than the current Excel workbook. For example, if you routinely access a website for data you enter into an Excel spreadsheet, you can build a hyperlink that points to that website. To do this, choose the **Existing File or Web Page** option under **Link to**. Then, enter the address for the website in the **Address** box, as shown in **Figure 19**. A key advantage of adding a hyperlink to your Excel workbook is to click on the hyperlink from within the workbook and navigate directly to the specified website.

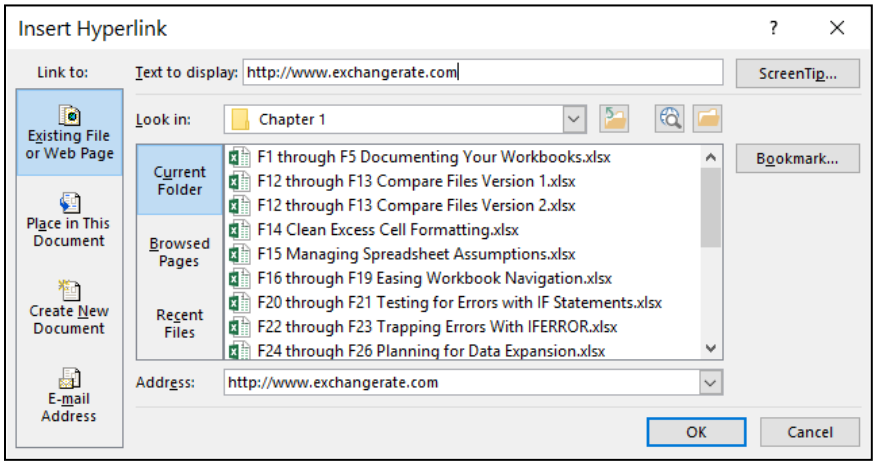


Figure 19 - Inserting a Hyperlink to a Website

Standardizing Workbook and Worksheet Layouts

Whenever practical, you should standardize your workbook and worksheet layouts to foster familiarity with the location of assumptions, formulas, data, and results. For example, you might consider adopting a standard workbook layout similar to **Table 1**.

Sample Excel Workbook Layout	
Sheet 1	Table of Contents
Sheet 2	Data
Sheet 3	Assumptions
Sheet 4	Calculations
Sheet 5	Reports

Table 1 - Sample Workbook Layout

Adding this type of standardization to all workbooks in an organization enables everyone who interacts with them to understand each workbook's overall layout. Further, standardizing items such as formatting on individual worksheets also proves beneficial. For example, formatting cells containing assumptions in one color while formatting cells containing formulas in another distinguishes these two data types for all spreadsheet users.

Using Styles to Streamline Formatting

One way to standardize the layout of your workbooks is by using **Styles** to color-code the workbook. Excel's Styles are formats that you can create and apply to cells. While you do not have to use Styles to apply formatting, using them helps to ensure consistency within a single workbook and across all workbooks.

A sample application of Styles to color-code a workbook might be to format all cells containing assumptions in one color, all cells containing formulas in another, and the final output in a third. By following this methodology, you and your team can see the workbook layout at a glance.

Building Error-Checking Routines into Your Workbooks

Despite best efforts, errors may still occasionally creep into your workbooks. Therefore, as a best practice, incorporate routines into your workbooks to alert you to erroneous conditions.

Testing for Errors with IF Statements

Multiple options exist for this purpose. One such option is to use an **IF** statement to alert you to an erroneous condition. For example, when creating a balance sheet, you could use the formula shown in **Figure 20** to compare the value of Total Assets (cell B35) to Total Liabilities and Equity (cell B61) and generate a warning message when these amounts differ.

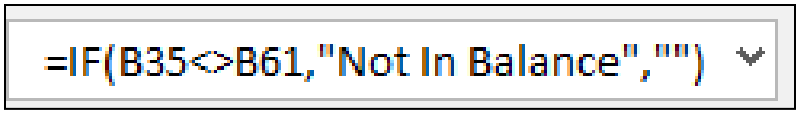


Figure 20 - Formula to Test for Out-of-Balance Condition

Testing for Errors with Conditional Formatting

Another approach to test for an out-of-balance condition is to use Excel's **Conditional Formatting** tool to compare Total Assets, Total Liabilities, and Equity. To build a conditional formatting rule for this task, select the cell(s) to which you want to apply the conditional formats and then click **Conditional Formatting** on the Home tab of the Ribbon, followed by **New Rule**, and then select **Use a formula to determine which cells to format**. In the resulting New Formatting Rule dialog box, enter the formula that serves as the basis for the test and the formatting characteristics you want to apply when the formula evaluates to "True," as shown in **Figure 21**.

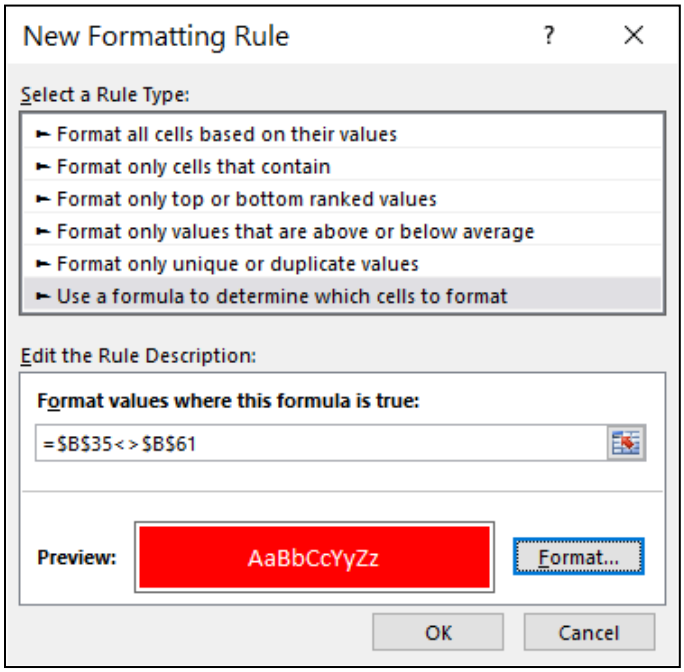
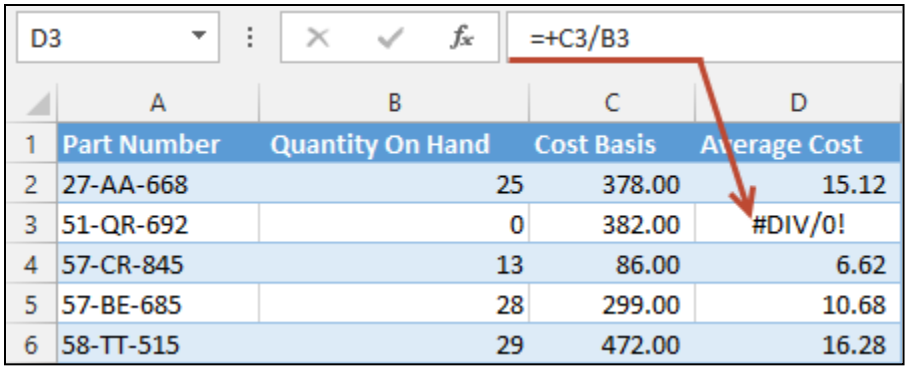


Figure 21 - Conditional Formatting Rule Testing for Out-of-Balance Condition

The conditional formatting rule shown above will apply a red background and white text when the value in cell B35 does not equal that in cell B61.

Preventing Downstream Errors with IFERROR

Another error-checking option is to use Excel's **IFERROR** function to identify and trap "upstream" errors and replace Excel's error messages with the content of your choice. You can use IFERROR in virtually all types of formulas. For example, suppose you work with a spreadsheet like the one shown in **Figure 22** to calculate the average cost for each inventory item.

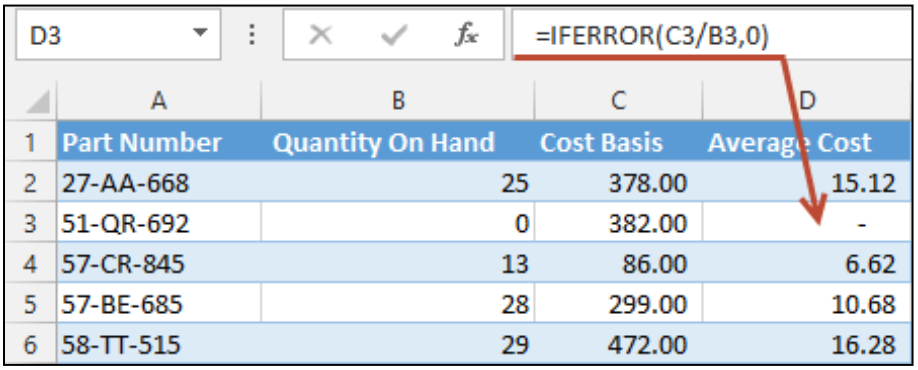


	A	B	C	D
1	Part Number	Quantity On Hand	Cost Basis	Average Cost
2	27-AA-668	25	378.00	15.12
3	51-QR-692	0	382.00	#DIV/0!
4	57-CR-845	13	86.00	6.62
5	57-BE-685	28	299.00	10.68
6	58-TT-515	29	472.00	16.28

Figure 22 - Sample Division by Zero Error

The simple formula that divides the Cost Basis by the Quantity On Hand results in an error when the Quantity On Hand is zero. Further, all other cells that depend directly or indirectly on cell D3 generate an error message because cell D3 is in error. Thus, an "upstream" or precedent cell causes all "downstream" or dependent cells to calculate as errors whenever an error exists in the upstream cell.

As shown in **Figure 23**, you can use IFERROR in your formulas to trap errors and replace them with values or text labels. By doing so, all downstream cells that depend on cell D3 will now calculate correctly.



	A	B	C	D
1	Part Number	Quantity On Hand	Cost Basis	Average Cost
2	27-AA-668	25	378.00	15.12
3	51-QR-692	0	382.00	-
4	57-CR-845	13	86.00	6.62
5	57-BE-685	28	299.00	10.68
6	58-TT-515	29	472.00	16.28

Figure 23 - Using IFERROR to Trap Upstream Errors

The syntax for the IFERROR function is:

=IFERROR(value, value_if_error)

where

- *Value*, a required field, is the formula checked for error.
- *Value_if_error*, a required field, is the value to return if the formula calculates to one of the following errors: #N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!.

Planning for Additional Data

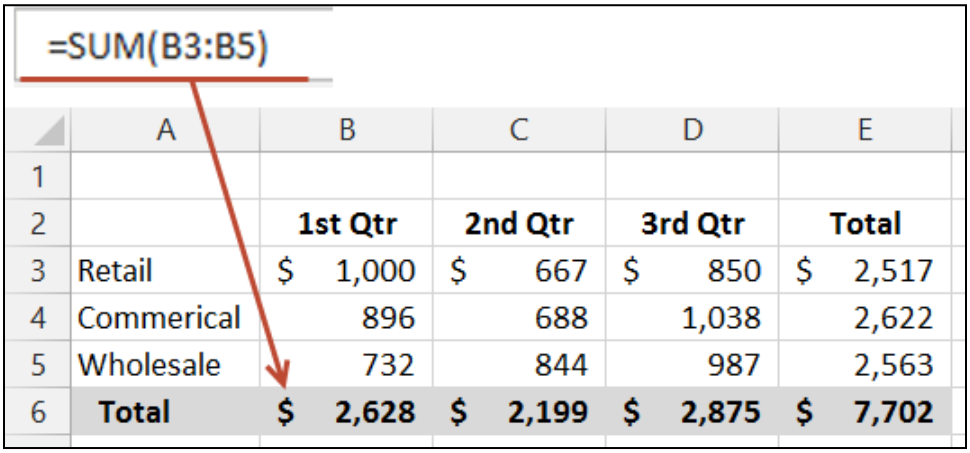
When creating a spreadsheet, you should assume that the model will grow over time and that you, or others, will add additional data in the future. Given this assumption, you should incorporate expansion principles into the model. Examples of best practices in this area include:

- Using Tables for storing data;
- Using Defined Names and Dynamic Defined Names; and
- Modifying Spreadsheet Layout and Formulas to Accommodate Future Data Expansion.

Modifying Spreadsheet Layout and Formulas to Accommodate Future Data Expansion

One of the simplest ways to accommodate future data expansion is to include blank rows and columns in formulas. While this may seem counterintuitive at first, the following example should help you understand why it is a best practice.

The sample spreadsheet presented in **Figure 24** does not readily accommodate data expansion. In other words, if you insert a row between the current row 5 and the current row 6 for another division, the formulas in cells B6 through E6 will not incorporate the newly added data.



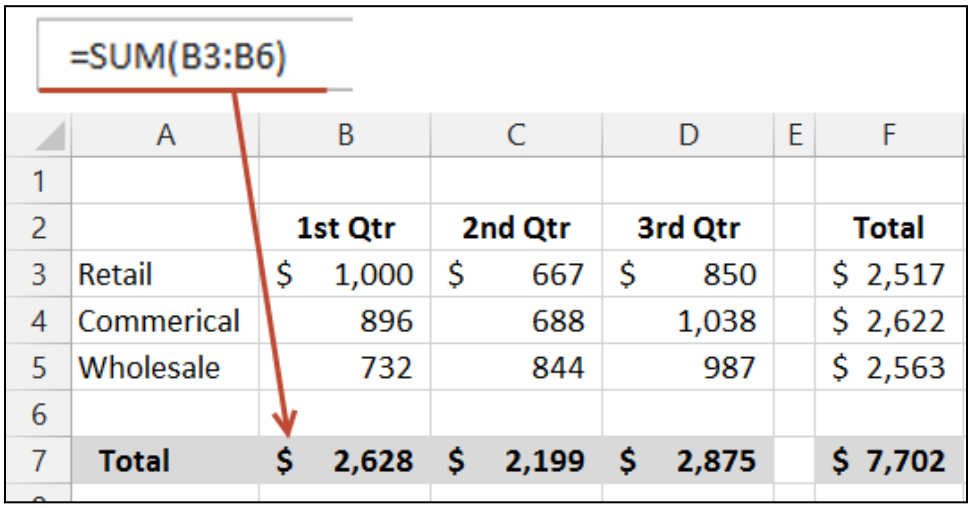
	A	B	C	D	E
1					
2		**1st Qtr**	**2nd Qtr**	**3rd Qtr**	**Total**
3	Retail	$ 1,000	$ 667	$ 850	$ 2,517
4	Commerical	896	688	1,038	2,622
5	Wholesale	732	844	987	2,563
6	**Total**	**$ 2,628**	**$ 2,199**	**$ 2,875**	**$ 7,702**

Figure 24 - Sample Spreadsheet Failing to Accommodate Data Expansion

On the other hand, you can slightly alter the spreadsheet's construction to accommodate the potential for additional data by following these three steps.

1. Include a blank row before the Total row and a blank column before the Total column.

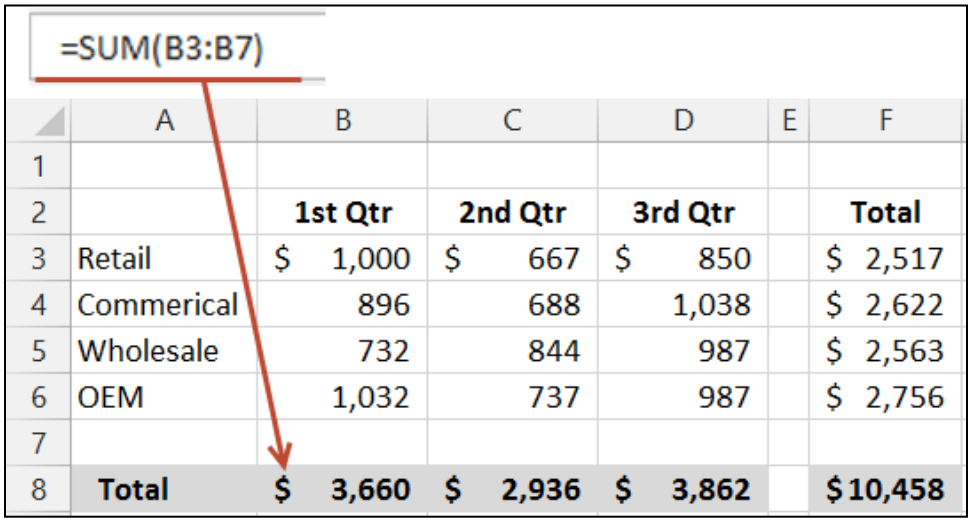
2. Include the blank rows and columns in your totaling formulas for each row and column, as shown in **Figure 25**.



	A	B	C	D	E	F
1						
2		**1st Qtr**	**2nd Qtr**	**3rd Qtr**		**Total**
3	Retail	$ 1,000	$ 667	$ 850		$ 2,517
4	Commerical	896	688	1,038		$ 2,622
5	Wholesale	732	844	987		$ 2,563
6						
7	**Total**	**$ 2,628**	**$ 2,199**	**$ 2,875**		**$ 7,702**

Figure 25 - Spreadsheet and Formula Modified to Accommodate Data Expansion

3. Insert new rows (or columns) in the spreadsheet at the existing blank row (or column) position. Then, the total formulas adjust automatically to incorporate the newly added data, as shown in **Figure 26**.



	A	B	C	D	E	F
1						
2		**1st Qtr**	**2nd Qtr**	**3rd Qtr**		**Total**
3	Retail	$ 1,000	$ 667	$ 850		$ 2,517
4	Commerical	896	688	1,038		$ 2,622
5	Wholesale	732	844	987		$ 2,563
6	OEM	1,032	737	987		$ 2,756
7						
8	**Total**	**$ 3,660**	**$ 2,936**	**$ 3,862**		**$ 10,458**

Figure 26 – Inserted Row Included in Total Row with Formula Automatically Adjusted

This methodology ensures that your total formulas automatically include the newly added data without additional manual effort. Additionally, you can use Tables, Defined Names, Dynamic Defined Names, and whole column and row references, all discussed later in this course, to accommodate expanding data.

Streamlining Spreadsheet Construction with Templates

An Excel ***template*** is a preconfigured workbook containing formatting, formulas, and content that you can use as a model for future workbooks. Microsoft makes templates available for download at no charge from www.office.com, and numerous other organizations offer paid and free templates.

Some advantages of working with templates include the following.

- A template helps to ensure that all workbooks based on that template are consistent in format and content;
- Templates speed the process of creating new workbooks; and
- Templates containing complex formulas help ensure that these formulas exist without error in workbooks based on these templates.

To create a workbook from a template, choose **File**, **New**, and select the template you want to use as the basis for your new workbook.

When considering how to use templates, you should understand that there are two broad categories: ***default*** and ***application templates***. Default templates control how future blank workbooks and worksheets appear. In contrast, application templates can contain items specific to a spreadsheet application, such as a financial statement or amortization schedule.

Creating Excel's Default Templates

Excel provides two default templates – *book.xltx and sheet.xltx*. However, before you can work with either of these templates, you must create them first. Book.xltx controls how all future blank workbooks will appear and sheet.xltx directs how all future blank worksheets you insert into a workbook appear.

To create a default template, open a blank workbook and configure it with your desired settings. In your default templates, you can save the following attributes:

- Cell and sheet formats;
- Page formats and print area settings for each sheet;
- Cell styles;
- The number and type of sheets in a workbook;
- Protected and hidden areas of the workbook;
- Text you want to repeat, such as page headers and row and column labels;
- Data, graphics, formulas, charts, and other information;
- Data validation settings;
- Macros, hyperlinks, and ActiveX controls on forms; and
- Workbook calculation options and window view options.

Once you have tailored the workbook to your specifications, use the **Save As** feature in Excel to save it as a template file. Most importantly, save these default templates – book.xltx and sheet.xltx – in the **XLStart** folder. The standard location for XLStart is:

C:\Users\ *user name* \AppData\Roaming\Microsoft\Excel\XLStart

By saving the book.xltx template to the XLStart folder, every time you open Excel, a blank workbook with settings inherited from this template appears automatically. Likewise, having saved the sheet.xltx template to the XLStart folder, every time you insert a new worksheet into an existing workbook, that sheet will inherit the template's settings. Additionally, every time you use the **CTRL+N** keyboard shortcut to create a new workbook, that workbook will inherit the template's settings.

Any workbook, template, or workspace you save in the XLStart folder will automatically open whenever you open Excel. As a best practice, if you always want a specific workbook to appear whenever you open Excel, you should save that workbook in the XLStart folder.

Ensuring Consistency with Application Templates

In addition to using Excel's default templates, you can also use application templates to speed up creating preconfigured workbooks, such as those you would use as the foundation for a financial statement or expense report. Application templates often include prefilled text, formulas, print settings, and macros.

Before using application templates in Excel, you must specify the location of your template files. To do this, click the **File** tab of the Ribbon, **Options**, and **Save** to open the **Excel Options** dialog box shown in **Figure 27**. Then, as shown, enter the default location to save your application templates.

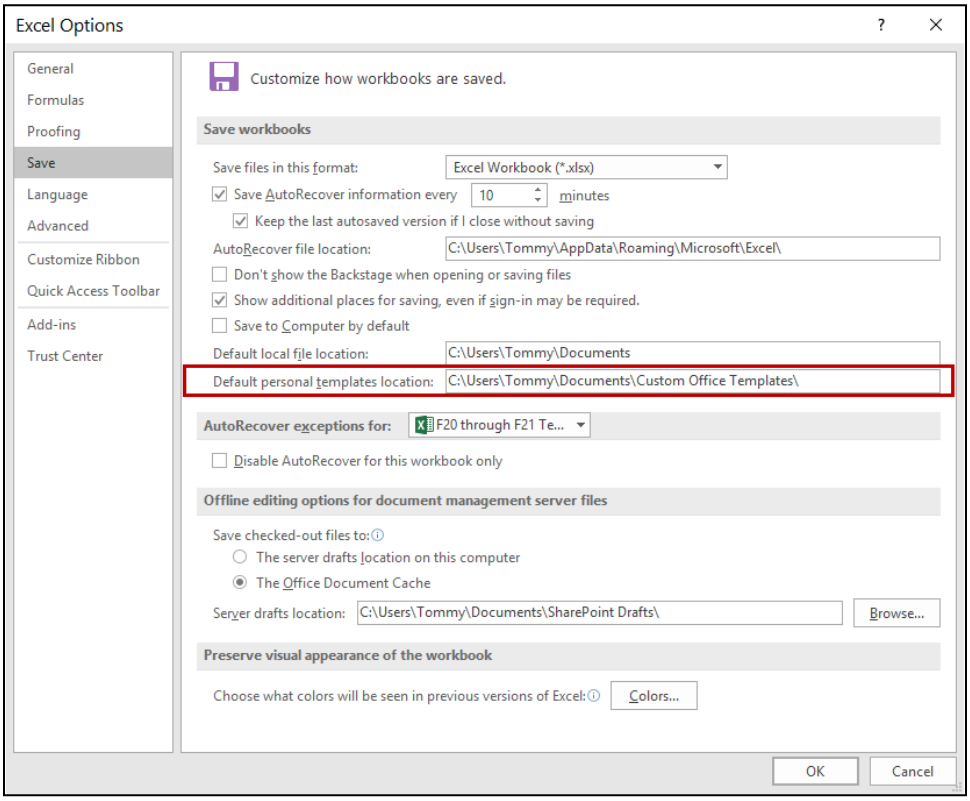


Figure 27 – Configuring Excel to Work with Application Templates

The process of creating an application template is similar to creating a default template. Begin by creating a workbook that contains your desired settings, formats, formulas, and content. Then, use Excel's **Save As** function (keyboard shortcut **F12**) to save the document as a template into the default or alternate locations.

Once you create and save a template, you can create a new workbook that uses the template as its foundation. First, choose **File**, **New**, and select the template serving as the workbook's base. Notably, all workbooks you create using a template are consistent for formats, formulas, content, and other settings inherited from that template.

?	What are some compelling reasons why it is an Excel best practice to place and label assumptions in their cells and not embed them inside formulas?

Improving Spreadsheet Performance

Chapter 2

No one likes a sluggish spreadsheet! Therefore, spreadsheet performance, including stability and fast recalculations, is prioritized for all Excel users. This topic is even more significant when working with large spreadsheet models such as multi-department budgets and forecasts. This chapter examines spreadsheet performance optimization processes, including troubleshooting performance issues.

Learning Objectives

Upon completing this chapter, you should be able to:

- Describe the fundamentals of how Excel calculates spreadsheets and why keeping recalculation times to a minimum is essential;
- List the three types of recalculation methods in Excel and explain how to change the one in use;
- List and describe five techniques for improving spreadsheet performance; and
- Describe how to identify bottlenecks in spreadsheet calculations using a systematic approach based on Excel macros.

Understanding Spreadsheet Calculation Fundamentals

Before discussing how to identify and correct bottlenecks in spreadsheet performance, it is necessary to understand how Excel calculates worksheets and workbooks and why improving calculation speed is essential. According to Microsoft's research, your productivity and ability to focus on tasks diminish as spreadsheet calculation speeds degrade.

- Users feel the system responds instantaneously when calculation times are less than one-tenth of a second.
- Users can successfully maintain thought processes when calculation times are between one-tenth and one second, although they notice slightly decreased performance.
- When calculation times are between one and ten seconds, users become annoyed, errors increase, and it becomes challenging to maintain a train of thought.

- When calculation times exceed ten seconds, users become impatient and multitask between recalculations.[1]

Excel offers three calculation modes, and you can configure It to use the one that best suits your specific needs. These three methods include the following.

1. **Automatic**. In automatic calculation mode, which is Excel's default setting, Excel recalculates the entire workbook every time certain events occur. These events include:

    - Entering new data,
    - Deleting or inserting a row or column,
    - Saving a workbook when the **Recalculate workbook before save** option is active,
    - Filtering data,
    - Double-clicking a row or column divider,
    - Adding, editing, or deleting a defined name,
    - Renaming a worksheet,
    - Rearranging the order of worksheets in the workbook, and
    - Hiding or unhiding rows, but not columns.

    If recalculation times are less than one second, most users should leave the calculation mode set to Automatic.

2. **Automatic except for data tables**. Your workbooks recalculate as they do in Automatic mode, with one exception. In **Automatic except for data tables** mode, all the workbook components recalculate based on the events described above, except for data tables. Therefore, when this mode is active, you must select your data tables and manually force calculations.

3. **Manual**. If your workbook recalculation time exceeds 1 second, you should consider changing the recalculation mode to **Manual**. In this mode, your workbooks do not automatically recalculate when any of the above events occur. Instead, it would help if you forced manual recalculation.

Changing Workbook Calculation Options

To change your calculation options in Excel, click **Calculation Options** from the Ribbon's **Formulas** tab. Then, choose your desired calculation settings, as shown in **Figure 28**.

[1] "Improving Performance in Excel 2007," http://msdn.microsoft.com/en-au/library/aa730921.aspx.

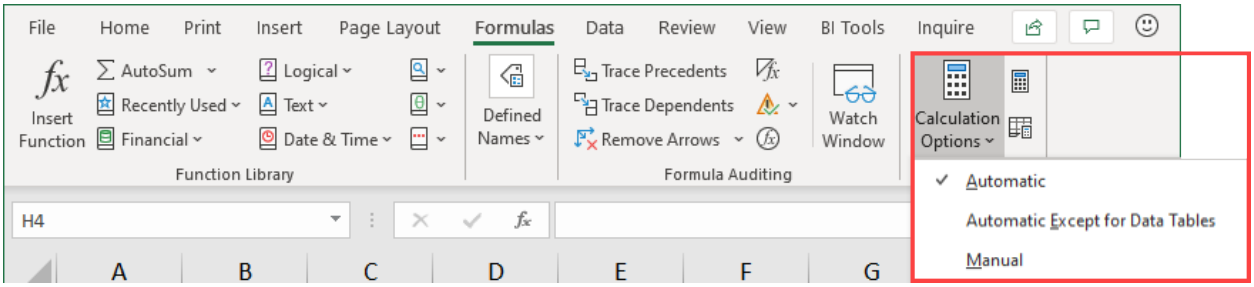


Figure 28 - Changing Excel's Calculation Options from the Ribbon

You can change Excel's Calculation Options by selecting **Options** from the Ribbon's **File** tab and then choosing **Formulas** to open the **Excel Options** dialog box shown in **Figure 29**. Next, choose the option that suits your needs in the Calculation Options section of the dialog box.

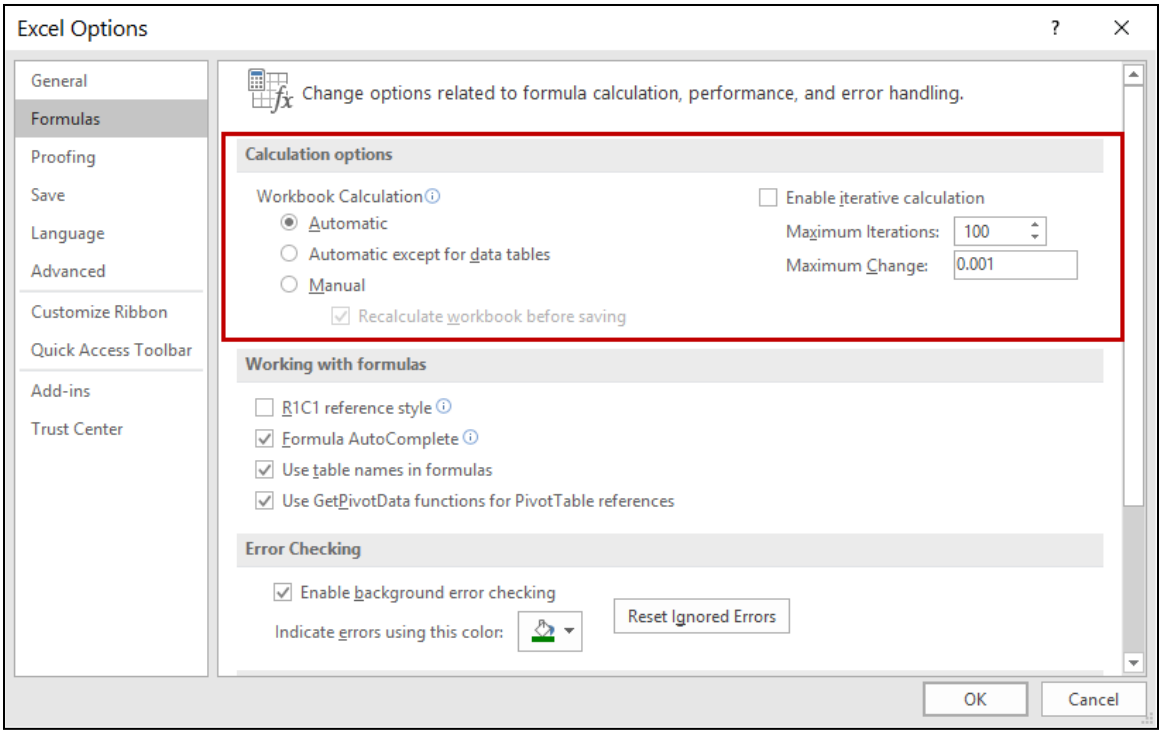


Figure 29 - Editing Workbook Calculation Options in Excel

Forcing Manual Calculations

When setting your workbook calculation mode to **Manual**, you must force recalculations. To do this, you can choose from the following two primary options.

- **Recalculate all open workbooks**. Use the **F9** keyboard shortcut or choose **Calculate Now** from the **Calculation** group on the Ribbon's **Formulas** tab to force all available workbooks to recalculate.

- **Recalculate the active worksheet**. To recalculate only the selected worksheets, use the **SHIFT+F9** keyboard shortcut or choose **Calculate Sheet** from the **Calculation** group on the Ribbon's **Formulas** tab.

Understanding Excel's Calculation Process

Excel uses a specific, three-step process when performing calculations. First, Excel creates a ***dependency tree*** that identifies which cells depend on others. Second, from the dependency tree, Excel creates a ***calculation chain*** that lists all cells containing formulas in the order in which Excel should calculate these cells. Note that when you change the structure of a workbook – for example, by adding a formula – Excel reconstructs the dependency tree and the calculation chain. Third, Excel calculates all formulas in the order the calculation chain sets. However, as part of this process, Excel revises the calculation chain if it encounters a formula that depends on a cell it has not yet calculated. Thus, Excel can recalculate the same cell multiple times during a single calculation.

In large workbooks, subsequent calculations are often substantially faster than the initial one. Reasons for this include the following.

- Excel's "smart" calculation engine typically only recalculates cells that have changed since the last calculation, their dependent cells, volatile functions, and conditional formats.

- Excel stores and reuses the most recent calculation chain to save time when determining the calculation sequence.

- Microsoft Windows and Excel cache recently used data for faster access in an Excel session.

Additionally, if your computer runs a multiple-core processor, you can configure Excel to take advantage of the processor's additional processing capabilities. To do so, access Excel's **Options**, and in the **Advanced** group, check the box next to **Enable multi-threaded calculation,** as shown in **Figure 30**.

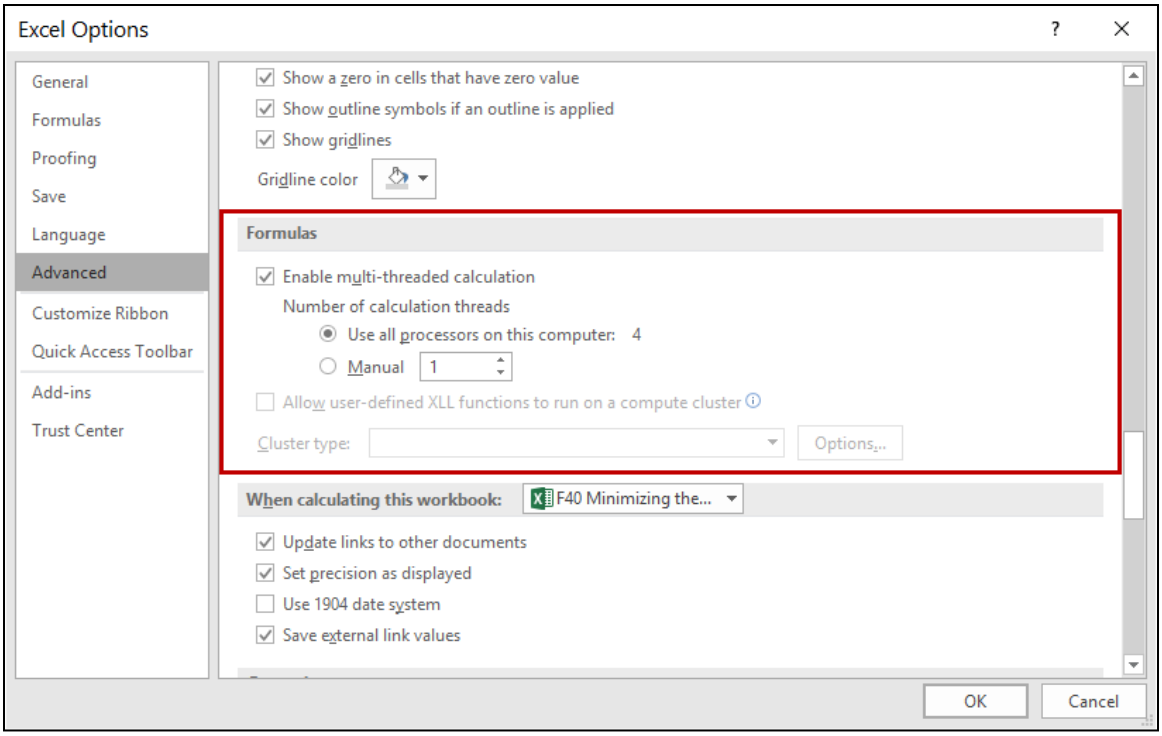


Figure 30 - Enabling Multi-Core Calculations in Excel

Forcing Advanced Recalculations

Previously, we addressed calculation options in Excel. Understanding how Excel calculates workbooks, let us examine two advanced recalculation options. First, you can force Excel to rebuild the dependency tree and the calculation chain for a given workbook and recalculate all cells in that workbook containing formulas. To do so, use a **CTRL+ALT+SHIFT+F9** keyboard shortcut.

Second, you can force a complete calculation of all cells in a workbook – not just those marked for recalculation – by using a **CTRL+ALT+F9** keyboard shortcut.

Handling Iterative Calculations

Though it is usually preferable to avoid circular references, they occasionally appear when building complex calculations. One example is tax accrual calculations, where federal income tax is deductible on state tax returns. Here, a circular reference arises from the calculation's very nature: federal tax is deductible on the state return, and state tax is deductible on the federal return. By default, Excel treats such calculations as errors. However, when necessary, you can enable circular references, also known as *iterative calculations*.

An iteration repeatedly recalculates a worksheet until it satisfies a specific numeric condition. However, Excel cannot calculate a formula that refers to a cell – either directly or indirectly –

containing the same formula. This condition is a circular reference, and circular references, by default, can iterate indefinitely, leading to worksheet errors, as shown in **Figure 31**.

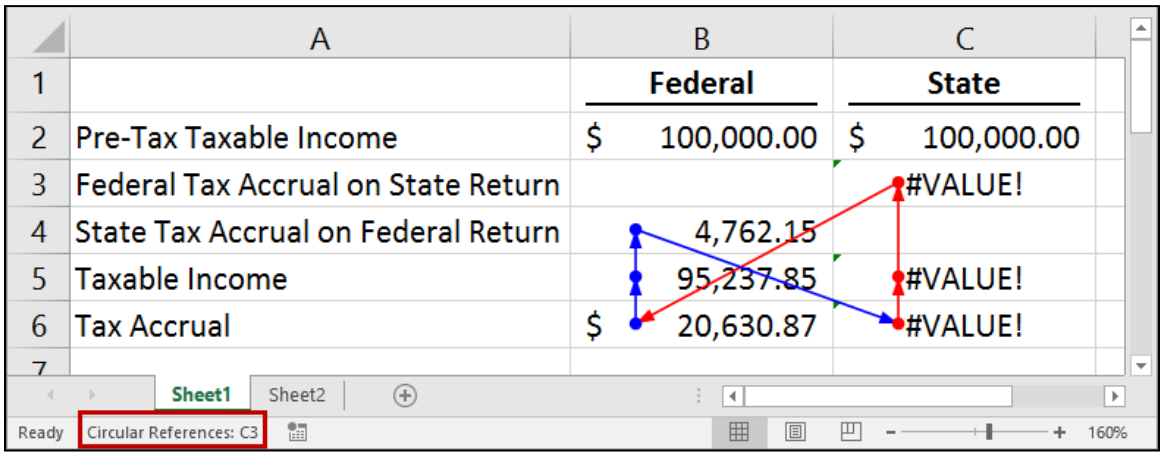


Figure 31 - Example of Erroneous Circular Reference in Excel

Many use Excel's **Formula Auditing** functions to identify circular referencing formulas and change them so the circular references disappear. However, eliminating these references is not always necessary if you understand why they appeared, their impact, and how to ensure that the presence of these formulas does not affect the worksheet's accuracy. The most common way to accommodate circular references in formulas is to adjust the number of iterations Excel calculates.

You can check the **Enable iterative calculation** box by accessing Excel's **Options** and choosing the **Formulas** tab, as shown in **Figure 32**.

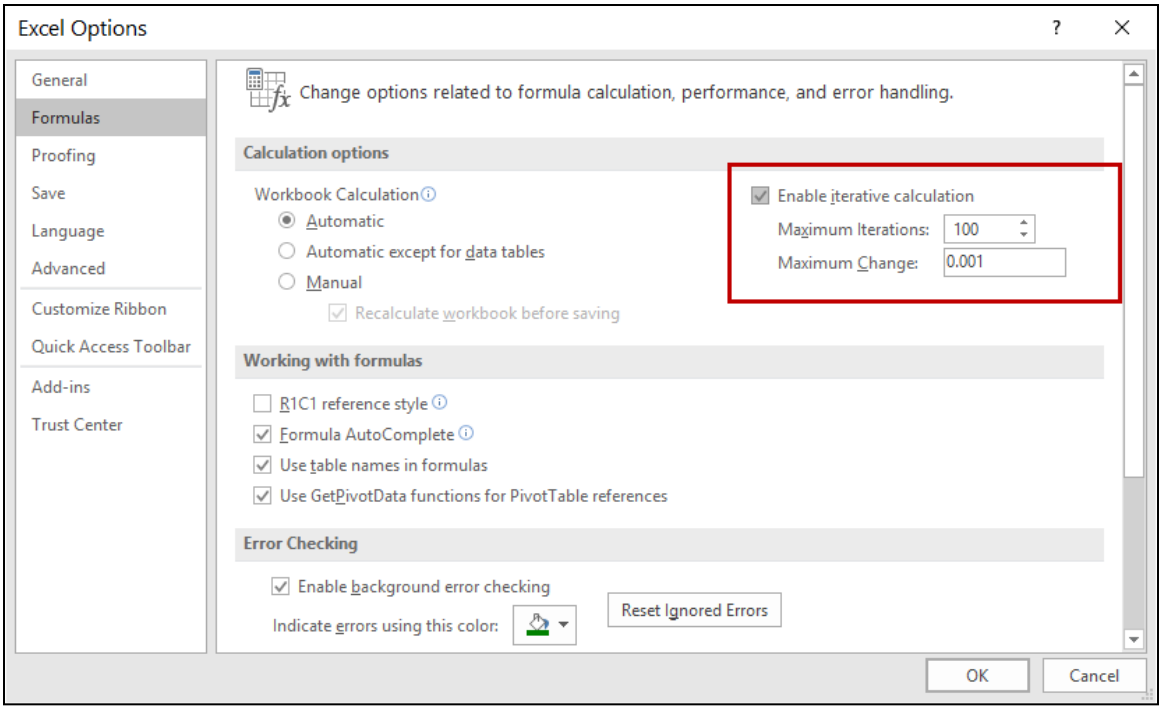


Figure 32 - Enabling Iterative Calculations in Excel

Enabling iterative calculations causes Excel to recalculate worksheets containing circular-referencing formulas up to the number specified in the Maximum iterations field, or until it reaches the value in the **Maximum change** field. One hundred iterations is the default maximum number of iterations. Likewise, 0.001 is the default maximum change value. However, Excel usually reaches sufficiently accurate results in as few as three iterations.

Returning to the worksheet presented in Figure 31, you will find that after enabling iterative calculations in the workbook, the formulas function correctly and calculate the correct amount of tax for both the federal and state income tax returns, as shown in **Figure 33**.

	A	B	C
1		**Federal**	**State**
2	Pre-Tax Taxable Income	$ 100,000.00	$ 100,000.00
3	Federal Tax Accrual on State Return		20,630.87
4	State Tax Accrual on Federal Return	4,762.15	
5	Taxable Income	95,237.85	79,369.13
6	Tax Accrual	$ 20,630.87	$ 4,762.15

Figure 33 - Properly Functioning Circular Referencing Formulas

Many Excel users view circular reference formulas as errors. However, while we should generally avoid circular references, they are not always errors. In fact, circular reference formulas are

sometimes necessary to address complex situations. As you can see, the key to working with circular references is to enable Excel's iterative calculation tool. Upon doing so, you should find that your circular referencing formulas function as you intend.

Optimizing Excel Performance

With a fundamental knowledge of how Excel calculates spreadsheets and how to manage associated settings, we can now focus on best practices for optimizing Excel performance, including minimizing calculation times. Specifically, we will address five techniques that will help to ensure your spreadsheet models calculate quickly and efficiently:

1. Providing adequate hardware resources;
2. Organizing data vertically, with raw data followed by formulas;
3. Segregating formulas from large data sets;
4. Understanding the effect of Data Tables; and
5. Minimizing the use of volatile functions.

Ensuring Adequate Hardware Resources

While obvious to most, the first step to optimizing Excel performance is to ensure you have adequate hardware resources and to configure them appropriately. The two most important considerations in this area are 1) processor capabilities and 2) available memory.

Processor Capabilities

Modern versions of Excel can take advantage of multiple physical or logical processors. Fortunately, most computers in today's business environments are multiprocessor systems. Therefore, as shown in Figure 30, assuming your computer offers multiprocessor calculations, configure it to utilize this capability. This configuration causes the device to split the calculations across the designated number of processors, potentially reducing calculation times.

Available Memory

The amount of Random Access Memory (RAM) installed on your computer can significantly affect the performance of all applications, including Excel. Ensure you have enough RAM installed on your computer to comfortably handle your operating system, Excel, the workbooks you anticipate having open simultaneously, and any other applications and services that will run while Excel is active. Without adequate RAM, your computer will page data to and from the swap file on the hard disk, a very slow process.

As shown in **Figure 34**, you can use the **Task Manager** in Windows to determine how much RAM you are using in total and on an application-by-application basis.

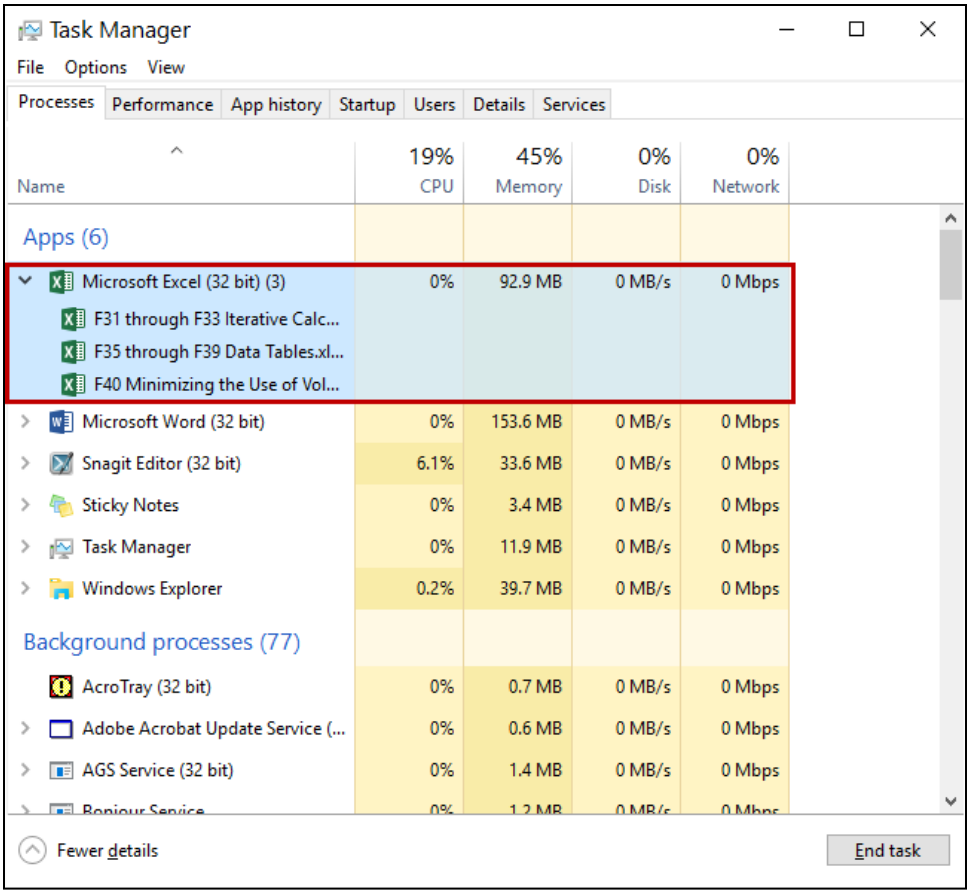


Figure 34 – Using Task Manager in Windows to Determine Memory Allocations

Note that the size of a saved Excel file is smaller than the amount of RAM that file will consume when you open it. This condition occurs because Excel compresses the data when it saves the workbook to your hard disk. Therefore, when you open the file, Excel decompresses the data, and the decompressed data will consume more memory than the stated file size. So, if you have an Excel workbook that consumes 5 megabytes (MB) of hard disk space, it might consume, for example, 10 MB of RAM when you open it in Excel.

32-Bit and 64-Bit Versions of Excel

Microsoft offers Excel as both a 32-bit and a 64-bit application. The capabilities of the 32-bit version of Excel are more than adequate for many – if not the majority – of Excel users. However, if you routinely work with large, complex workbooks, you may consider using the 64-bit version of Excel instead.

Technically, the 32-bit version of Excel can address up to 2 gigabytes (GB) of RAM. Therefore, regardless of how many workbooks are open, the application can only use 2 GB of RAM. Because of overhead in Excel, the amount of memory available for data and formulas is less than the stated 2 GB limit. Potentially, these limitations could cause issues if you need to work with large workbooks.

If you are a power user and need to work with large workbooks, install the 64-bit version of Excel on a device running a 64-bit version of Windows. This version of Excel addresses up to 8 terabytes (TB) of memory. To put this amount of memory into perspective, 8 TB equals 8,000 GB; in other words, from a practical standpoint, the 64-bit version of Excel addresses an almost unlimited amount of memory, allowing you to work efficiently with large workbooks.

Note that you can install the 32-bit version of Excel on a computer running either a 32-bit or a 64-bit version of the Windows operating system. However, you must install the 64-bit version of Excel on a computer running a 64-bit version of Windows.

Suppose you are running the 32-bit version of Excel and routinely experience RAM issues with a specific Excel workbook. In that case, consider the following best practices to reduce the memory your workbook requires.

- Check that Excel's last cell on each sheet is in the right place, as discussed on page 11. When it is not where you want, for instance, because of additional formatting or recently deleted cells, reset the last cell's location by deleting all columns to the right and all rows below the correct last cell.

- Convert formulas to values.

- Disable **Track Changes** for **Shared** workbooks.

- Reduce the number of formulas referencing other worksheets. Formulas that refer to other worksheets use significantly more memory for dependency trees than formulas that refer to cells or ranges on their worksheet.

- Use smaller ranges in formulas such as **INDEX** and **VLOOKUP**.

- Avoid references to ranges containing unused or blank cells.

- Use multi-cell array formulas.

- Store repetitively used formulas as **Defined Names** and use the Defined Names instead of formulas.

- Simplify formatting, including reducing the number of fonts in use.

- Reduce the number of graphics or embedded objects in a workbook.

Organizing Worksheets Vertically with Raw Data Followed by Formulas

Organize your worksheets vertically, filling data from top to bottom as a best practice. Further, place formulas at the bottom of your data sets. Adhering to this practice generates two benefits.

1. Navigating your worksheets is easier because users only need to scroll from top to bottom, not from top to bottom and side to side.

2. You avoid placing the last cell of the worksheet in a distant location. When the last cell of a worksheet resides many rows below and many columns to the first cell's right, the workbook's size can increase significantly. This situation often contributes to bloated workbook file sizes and slower calculations.

An additional, less noticeable benefit of vertically organizing workbooks is that you may reduce calculation times, though the time saved might be negligible. Returning to the previous discussion on dependency trees and calculation chains, the time necessary to generate a calculation chain decreases when formulas refer to data above them, thus improving calculation times. Therefore, as a best practice, build "backward-facing" formulas instead of "forward-facing" ones, particularly if you work with large data sets.

Tables are perhaps the best tool for organizing your worksheets vertically, with formulas at the bottom.

Segregating Formulas from Large Data Sets

While simplified spreadsheet models are quicker to build, easier to understand, and usually calculate faster, when working with large data sets, you may consider segregating your data from your formulas by moving your data onto a separate worksheet. For example, if your data consists of 300,000 rows of transactional information, consider storing it in a single worksheet. Then build formulas on another sheet that reference the data to produce a better-performing spreadsheet model than one where the formulas and data reside on the same worksheet. Although you might experience a small amount of performance degradation because of the linked data, reduced navigation time when observing the spreadsheet's results will likely compensate for any increase in calculation time.

Understanding the Effect of Data Tables

Data tables are a familiar feature to many Excel users and provide valuable tools for performing what-if analyses. To illustrate, consider the example pictured in **Figure 35**. In this example, the formula in cell B6 calculates the monthly payment required to amortize a loan, based on the assumptions in cells B2 through B4. However, the Data Table in cells D3 through E11 shows the monthly payment for various interest rates ranging from 3% to 7%.

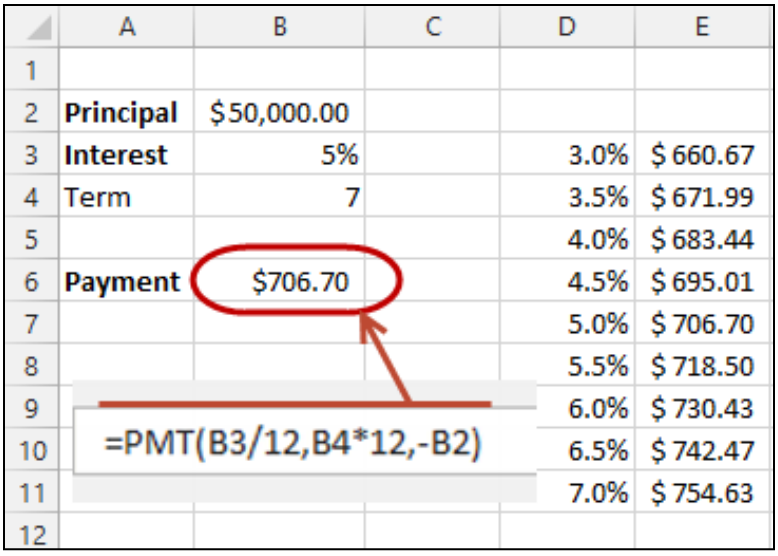


Figure 35 - Simple Single Variable Data Table

To build the Data Table shown above, perform the following five steps.

1. Enter the interest rates you wish to "flex" or insert into the calculation in cells D3 through D11.

2. In cell E2, create a cell reference back to cell B6.

3. Hide the output of cell E2 by changing the font color to white or applying Excel's **Hidden Format** code of **;;;** as a **Custom** format in the **Format Cells** dialog box, as shown in **Figure 36**.

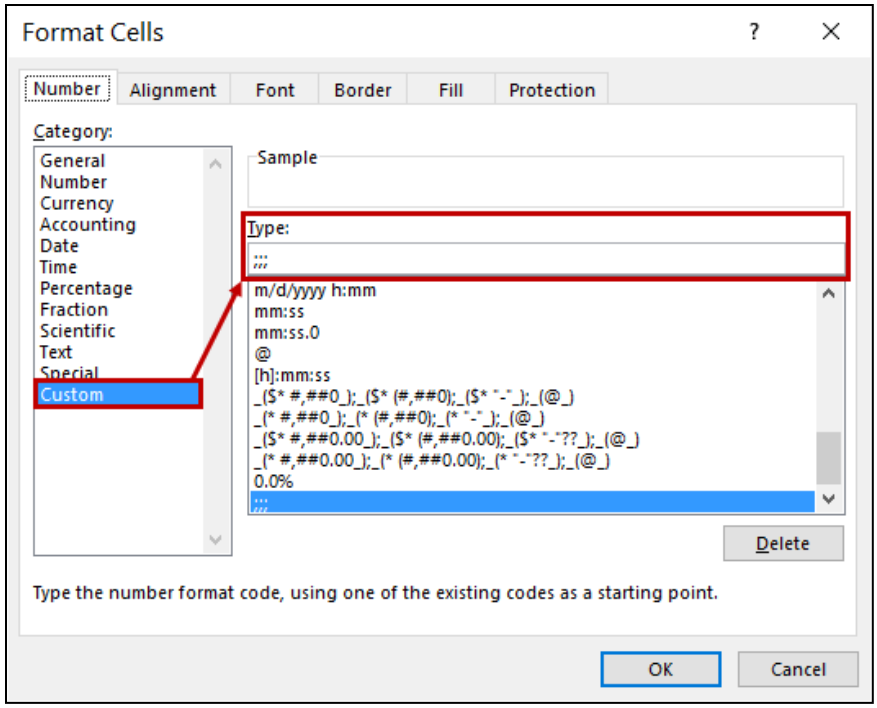


Figure 36 - Applying Excel's Hidden Format Code

4. Select cells D2 through E11 and choose **What-if Analysis** from the Ribbon's **Data** tab, followed by **Data Table**.

5. As shown in **Figure 37**, in the **Data Table** dialog box, enter $**B$3** as the **Column input cell**, leave the **Row input cell** blank, and click **OK**.

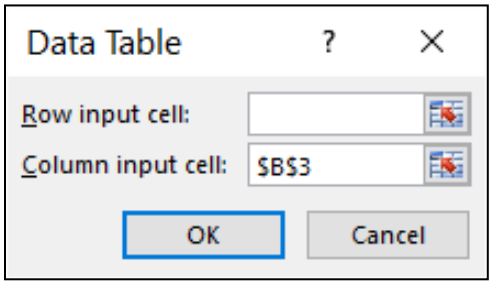


Figure 37 - Specifying the Column Input Cell

Completing these steps causes Excel to build the single variable Data Table shown in Figure 35 that takes each interest rate in cells D3 through D11, substitutes them into the formula in cell B6, and provides the corresponding monthly payment in cells E3 through E11. Notice that Excel performs the requisite calculations without requiring you to copy any formulas.

You can also build data tables that use two variables, such as interest rate and term, as shown in **Figure 38**.

	A	B	C	D	E	F	G	H	I	J
1										
2	**Principal**	$50,000.00			5	6	7	8	9	10
3	**Interest**	5%		3.00%	898.43	759.68	660.67	586.48	528.85	482.80
4	Term	7		4.00%	920.83	782.26	683.44	609.46	552.05	506.23
5				4.00%	920.83	782.26	683.44	609.46	552.05	506.23
6	**Payment**	$706.70		5.00%	943.56	805.25	706.70	633.00	575.86	530.33
7				5.00%	943.56	805.25	706.70	633.00	575.86	530.33
8				6.00%	966.64	828.64	730.43	657.07	600.29	555.10
9				6.00%	966.64	828.64	730.43	657.07	600.29	555.10
10				7.00%	990.06	852.45	754.63	681.69	625.31	580.54
11				7.00%	990.06	852.45	754.63	681.69	625.31	580.54

Figure 38 – Two-Variable Data Table

The procedure for building a two-variable Data Table is mostly the same as for a one-variable Data Table:

- Enter the variable you wish to flex – in this case, interest rate and term – as the first row and the first column of the Data Table.

- Select the range of **D2** through **J11** before choosing **What-if Analysis**, **Data Table** from the Ribbon's **Data** tab.

- In the **Data Table** dialog box, enter **$B$4** for the **Row input cell** and **$B$3** as the column input cell, as shown in **Figure 39**.

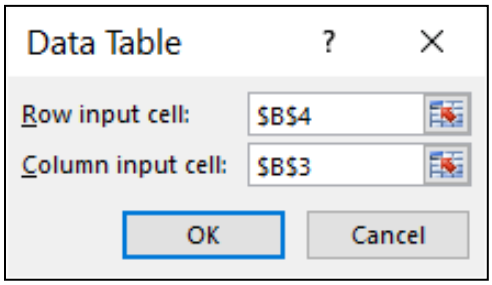


Figure 39 - Creating a Two-Variable Data Table

The resulting two-variable Data Table will appear in Figure 38, providing sensitivity analysis for interest rates and the proposed loan term.

Based on the brief discussion above, if you have not used Data Tables in the past, you might decide to begin using them to assist with performing sensitivity analyses of large datasets. However, before doing so, understand that Data Tables can adversely affect calculation times for a workbook. This trait exists because Data Tables cause the workbook to recalculate many times – an initial calculation and then separate calculations for each pair of row and column values Excel substitutes into the original formula. Thus, if you have a two-variable Data Table consisting

of 50 columns and 1,000 rows, you add 50,000 additional calculations to the spreadsheet every time the spreadsheet recalculates; these calculations are in addition to those required elsewhere in Excel.

Setting Data Table Calculation to Manual

Fortunately, if you wish to use Data Tables and want to avoid the calculation issues described above, you can take advantage of the following best practice. As discussed, one of Excel's calculation options is **Automatic Except Data Tables**. When this option – available from the **Formulas** tab of the Ribbon and Excel's **Options** in the **Backstage View** – is active, recalculating the workbook does not include any data tables. When this option is active, use the **F9** keyboard shortcut or select **Calculate Now** from the Ribbon's Formulas tab to recalculate Data Tables. You can also recalculate the worksheet using the **SHIFT+F9** keyboard shortcut or selecting **Calculate Sheet** from the Ribbon's **Formulas** tab.

Minimizing the Use of Volatile Functions

A ***volatile function*** in Excel is one whose values may change from one moment to the next, even if none of the function's arguments change. For example, Excel's **NOW** function returns the date and time into a cell as of the time at which the formula containing the function recalculates. As shown in **Figure 40**, entering **=NOW( )** into a cell is necessary to execute this function; notice that no arguments are present within the parentheses.

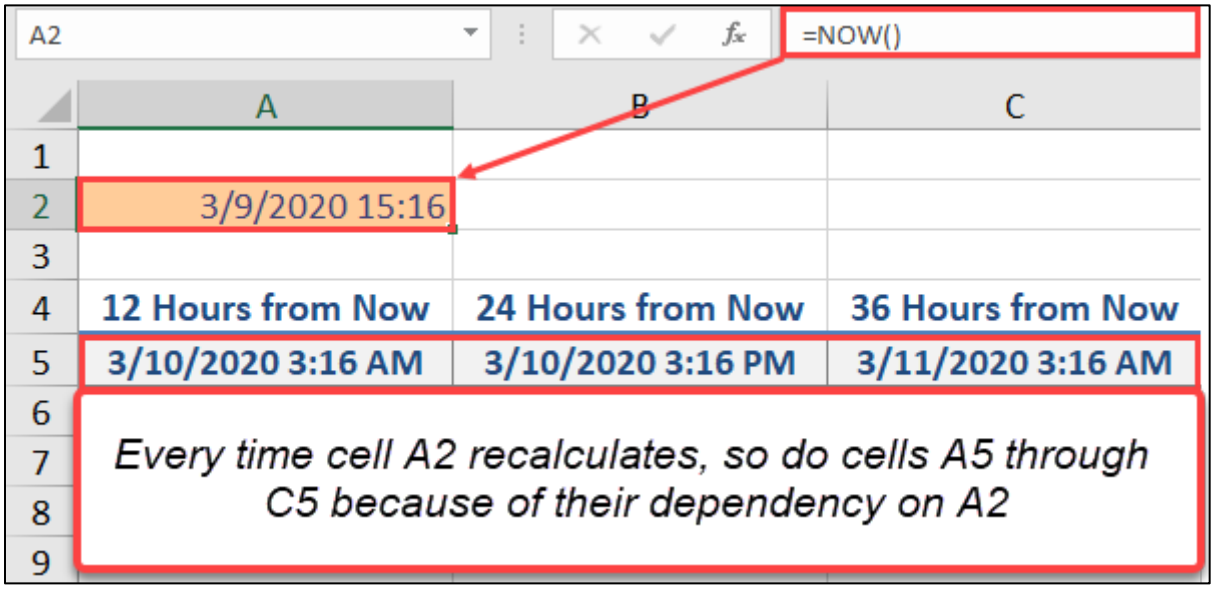


Figure 40 - Excel's NOW Function, an Example of a Volatile Function

Whenever Excel recalculates a workbook, it recalculates all volatile functions and any cells depending on those functions. Thus, in the simple example shown in Figure 40, Excel recalculates cells A5 through C5 every time cell A2 recalculates. Therefore, in situations where your workbook

contains many volatile functions or many cells that depend on them, extended recalculation times may occur. For this reason, you should limit the use of volatile functions in your workbooks.

Other examples of volatile functions in Excel include:

- **TODAY**,
- **RAND**,
- **OFFSET**, and
- **INDIRECT**

Additionally, **INFO** and **CELL** may be volatile depending upon the circumstances in which you use them.

Identifying Bottlenecks in Spreadsheet Performance

Notwithstanding previous discussions, bottlenecks will appear in your Excel spreadsheets, impeding their performance. Thus, there is a need for a systematic, sensible approach to identifying the sources of these issues so you can correct them. In October 2006, Microsoft published an article titled "Improving Spreadsheet Performance in Excel 2007."[2] Although the article is a bit dated, the information it contains is still valuable. This article presents a pragmatic approach to detecting and addressing calculation bottlenecks; the following discussion builds on those concepts.

Measuring Calculation Times

Before taking corrective action with poorly performing spreadsheets, you must identify the source of the problem(s). To do so, you must measure the calculation times of workbooks, worksheets, and ranges. Fortunately, Visual Basic for Applications (VBA) code exists for these tasks and is included with the **CalcTimers.xlsm** file available for download in this course's demo file package.

Without going into details on each line of code, the following code creates a user-defined function (UDF) named **MICROTIMER**.

```
Private Declare Function getFrequency Lib "kernel32" _
Alias "QueryPerformanceFrequency" (cyFrequency As Currency) As Long
Private Declare Function getTickCount Lib "kernel32" _
Alias "QueryPerformanceCounter" (cyTickCount As Currency) As Long
'
Function MicroTimer() As Double
'Returns seconds.
'
```

[2] Improving Spreadsheet Performance in Excel 2007, October 2006, Charles Williams, http://msdn.microsoft.com/en-au/library/aa730921.aspx.

```
    Dim cyTicks1 As Currency
    Static cyFrequency As Currency
    '
    MicroTimer = 0

'Get frequency.
    If cyFrequency = 0 Then getFrequency cyFrequency

'Get ticks.
    getTickCount cyTicks1

' Seconds
    If cyFrequency Then MicroTimer = cyTicks1 / cyFrequency
End Function
```

Additionally, the following code generates four subroutines (macros) that measure the time required to calculate a range, the time required to recalculate a sheet or all open workbooks, or the full calculation time for all open workbooks.

```
Sub RangeTimer()
    DoCalcTimer 1
End Sub
Sub SheetTimer()
    DoCalcTimer 2
End Sub
Sub RecalcTimer()
    DoCalcTimer 3
End Sub
Sub FullcalcTimer()
    DoCalcTimer 4
End Sub

Sub DoCalcTimer(jMethod As Long)
    Dim dTime As Double
    Dim dOvhd As Double
    Dim oRng As Range
    Dim oCell As Range
    Dim oArrRange As Range
    Dim sCalcType As String
    Dim lCalcSave As Long
    Dim bIterSave As Boolean
    '
    On Error GoTo Errhandl

' Initialize
    dTime = MicroTimer

    'Save calculation settings.
    lCalcSave = Application.Calculation
    bIterSave = Application.Iteration
    If Application.Calculation <> xlCalculationManual Then
        Application.Calculation = xlCalculationManual
    End If
    Select Case jMethod
    Case 1
```

```
        'Switch off iteration.

        If Application.Iteration <> False Then
            Application.Iteration = False
        End if

        'Max is used range.

        If Selection.Count > 1000 Then
            Set oRng = Intersect(Selection, Selection.Parent.UsedRange)
        Else
            Set oRng = Selection
        End If

        'Include array cells outside selection.

        For Each oCell In oRng
            If oCell.HasArray Then
                If oArrRange Is Nothing Then
                    Set oArrRange = oCell.CurrentArray
                End If
                If Intersect(oCell, oArrRange) Is Nothing Then
                    Set oArrRange = oCell.CurrentArray
                    Set oRng = Union(oRng, oArrRange)
                End If
            End If
        Next oCell

        sCalcType = "Calculate " & CStr(oRng.Count) & _
            " Cell(s) in Selected Range: "
    Case 2
        sCalcType = "Recalculate Sheet " & ActiveSheet.Name & ": "
    Case 3
        sCalcType = "Recalculate open workbooks: "
    Case 4
        sCalcType = "Full Calculate open workbooks: "
    End Select

'Get start time.
    dTime = MicroTimer
    Select Case jMethod
    Case 1
        If Val(Application.Version) >= 12 Then
            oRng.CalculateRowMajorOrder
        Else
            oRng.Calculate
        End If
    Case 2
        ActiveSheet.Calculate
    Case 3
        Application.Calculate
    Case 4
        Application.CalculateFull
    End Select

'Calc duration.
    dTime = MicroTimer - dTime
```

```
    On Error GoTo 0

    dTime = Round(dTime, 5)
    MsgBox sCalcType & " " & CStr(dTime) & " Seconds", _
        vbOKOnly + vbInformation, "CalcTimer"

Finish:

    'Restore calculation settings.
    If Application.Calculation <> lCalcSave Then
         Application.Calculation = lCalcSave
    End If
    If Application.Iteration <> bIterSave Then
         Application.Calculation = bIterSave
    End If
    Exit Sub
Errhandl:
    On Error GoTo 0
    MsgBox "Unable to Calculate " & sCalcType, _
        vbOKOnly + vbCritical, "CalcTimer"
    GoTo Finish
End Sub
```

To use these subroutines and functions, you should copy them into a VBA module. Open the VBA editor with the **ALT+F11** keyboard shortcut and select **Module** on the **Insert** menu. Then, copy and paste the code into the module. To make the code available in all workbooks, store it in a module in your **PERSONAL.xlsb** workbook.

With the appropriate VBA code, you can run the macros to measure the time required to calculate a workbook, worksheet, or range. You can do this using any of the following three techniques.

1. Use the **ALT+F8** keyboard shortcut to open the **Macro** dialog box pictured in **Figure 41**. From the Macro dialog box, select the macro you want to execute and click **Run**.

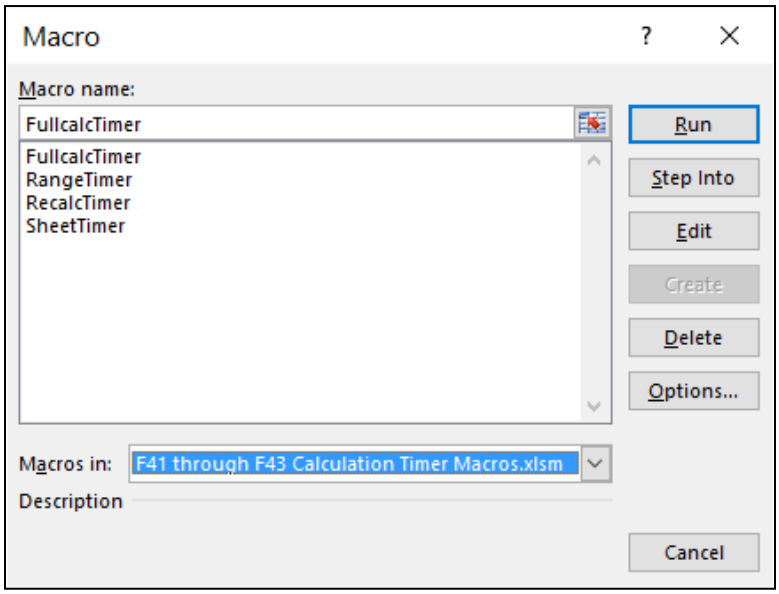


Figure 41 - Running Calculation Timer Macros

2. From the **View** tab of the Ribbon, click **Macros** followed by **View Macros** to open the Macro dialog box pictured in Figure 41. Then select the macro you want to execute and click **Run**.

3. From the Ribbon's **Developer** tab, click **Macros** to open the Macro dialog box pictured in Figure 41. Next, select the macro you want to execute and click **Run**.

To achieve the best results, you should run the calculation process three to five times, record your results for each test, and average the results.

Finding Spreadsheet Bottlenecks

You should adopt a systematic approach to measuring and identifying spreadsheet bottlenecks when using the macros outlined above. Specific steps for doing so include:

1. Ensure you have only one workbook open and no other tasks are running.

2. Set the calculation mode of the workbook you are testing to **Manual**.

3. Check the used range by pressing **CTRL+END** on each worksheet in turn. This action shows the last used cell. If this is beyond what you expect, consider deleting excess columns and rows and saving the workbook.

4. Make a backup copy of the workbook.

5. Open the workbook that contains the calculation timer macros.

6. Run the **FullCalcTimer** macro. Calculating all the formulas in the workbook is usually the worst-case time.

7. Run the **RecalcTimer** macro. A recalculation immediately after a full calculation usually gives you the best-case time.

8. Calculate workbook volatility as the ratio of recalculation time to full calculation time. This calculation measures the extent to which volatile formulas and the evaluation of the calculation chain are bottlenecks.

9. Run the **SheetTimer** macro on each worksheet in turn. Because you just recalculated the workbook, this gives you the recalculating time for each worksheet. This action should help you identify which worksheets are slowing the workbook calculation.

10. Run the **RangeTimer** macro on selected blocks of cells in the worksheets that appear to be calculating slowly.

    a. For each problem worksheet, divide the columns or rows into smaller blocks.
    b. Select each block in turn and then run the **RangeTimer** macro on the block.
    c. If necessary, drill down further by subdividing each block into smaller blocks.

11. Prioritize the bottlenecks and begin to take corrective actions, as discussed below.

To illustrate, running the **FullCalcTimer** macro on a workbook might indicate a total calculation time of 11.97747 seconds, as shown in **Figure 42**.

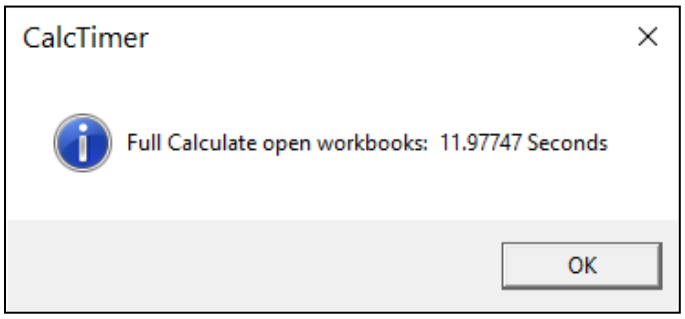


Figure 42 - Results of FullCalcTimer Macro

Immediately after running the FullCalcTimer macro, launching the **RecalcTimer** macro on the same workbook indicates a recalculation time of 0.02688 seconds, as shown in **Figure 43**. This calculation yields a workbook volatility ratio of 0.0022442 (0.02688/11.97747). The volatility ratio measures the extent to which volatile formulas and the evaluation of the calculation chain hinder calculation times.

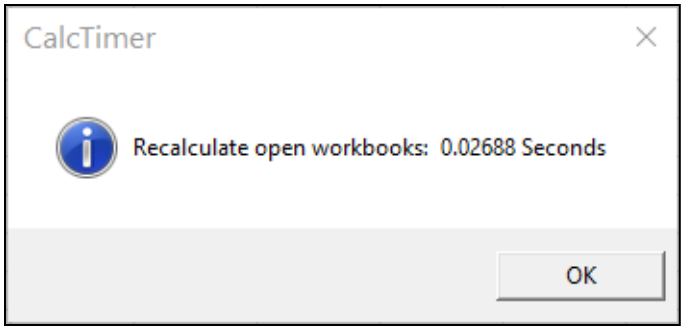


Figure 43 - Results of RecalcTimer Macro

Running the **SheetTimer** macro on each of the four worksheets in the workbook yields calculation times of 0.03718, 0.14702, 0.07239, and 0.04168 seconds, respectively. Of course, these points identify the second and third worksheets as the potential culprits for the workbook's relatively poor overall performance. For each of the two offending worksheets, you could run the **RangeTimer** macro to draw focus to the cells that offer the most opportunities for improvement.

Correcting Spreadsheet Bottlenecks

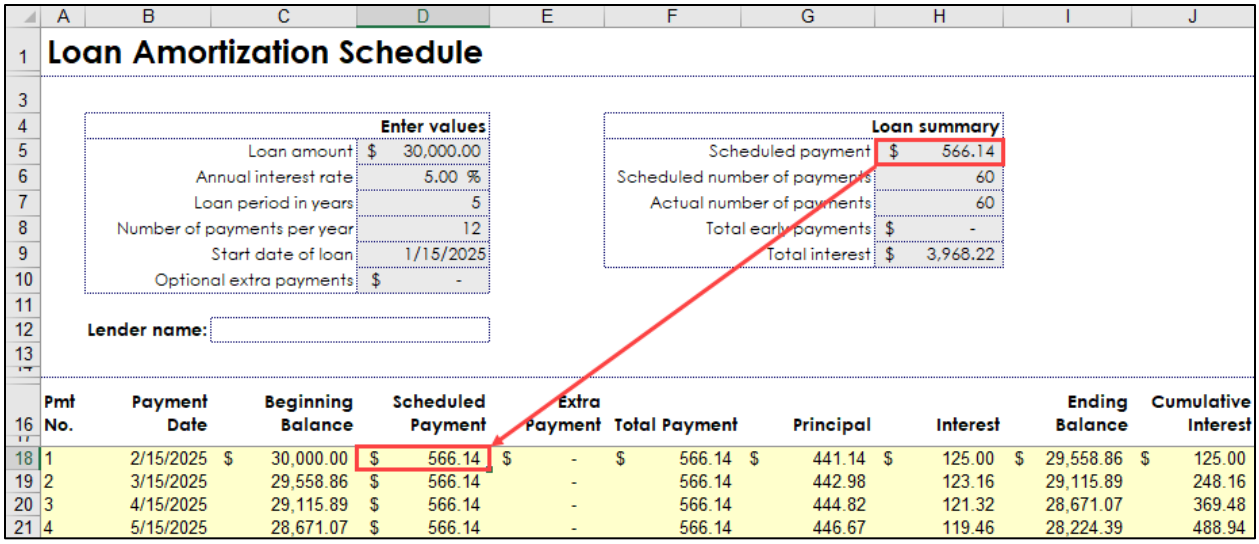
The size of a workbook and the number of formulas in it do not determine calculation time exclusively. In addition, the number of cell references and the efficiency of the functions used in calculations significantly impact calculation times. Therefore, you should take the following steps to identify the worksheets and cells that appear to have the greatest adverse effect on calculation times.

Removing Duplicated and Unnecessary Calculations

Analyze the spreadsheet for duplicated, repeated, and unnecessary calculations. Then, consider how you might obtain the same result with fewer references and formulas. Usually, this involves one or more of the following steps.

- Reduce the number of references in each formula.
- Move the repeated calculations to one or more helper cells and then reference the helper cells from the original formulas.
- Use additional rows and columns to calculate and store intermediate results and reuse them in other formulas.

For example, suppose you are working with the loan amortization schedule shown in **Figure 44**. As a best practice, notice how the monthly payment in cell D18 (and continuing for all payments in the schedule) is not a =PMT formula. Instead, it links to a cell that contains a =PMT formula. Calculating the scheduled payment once and only once, and then referring to the cell containing that calculation, relieves Excel from performing the same calculation for every payment in the amortization schedule, potentially improving spreadsheet performance.



Loan Amortization Schedule

Enter values	
Loan amount	$ 30,000.00
Annual interest rate	5.00 %
Loan period in years	5
Number of payments per year	12
Start date of loan	1/15/2025
Optional extra payments	$ -

Lender name:

Loan summary	
Scheduled payment	$ 566.14
Scheduled number of payments	60
Actual number of payments	60
Total early payments	$ -
Total interest	$ 3,968.22

Pmt No.	Payment Date	Beginning Balance	Scheduled Payment	Extra Payment	Total Payment	Principal	Interest	Ending Balance	Cumulative Interest
1	2/15/2025	$ 30,000.00	$ 566.14	$ -	$ 566.14	$ 441.14	$ 125.00	$ 29,558.86	$ 125.00
2	3/15/2025	29,558.86	$ 566.14	-	566.14	442.98	123.16	29,115.89	248.16
3	4/15/2025	29,115.89	$ 566.14	-	566.14	444.82	121.32	28,671.07	369.48
4	5/15/2025	28,671.07	$ 566.14	-	566.14	446.67	119.46	28,224.39	488.94

Figure 44 - Sample Loan Amortization Schedule

Using the Most Efficient Function Possible

Excel offers over 500 built-in functions to handle a wide variety of tasks. Thus, it can be challenging to determine the best function for a particular situation. When you find a bottleneck that involves a function or an array formula, determine if there is a more efficient way to achieve the same result. For example, consider the following methods.

- Minimize the number of used cells in formulas incorporating functions like **SUM** and **SUMIFS**. Calculation time is proportional to the number of used cells.
- Lookups on sorted data can be tens or hundreds of times more efficient than lookups on unsorted data; therefore, sort your data first.
- Although carefully written VBA functions can be fast, VBA-based user-defined functions are often slower than built-in functions in Excel.
- Consider replacing array formulas that calculate slowly with user-defined functions.

For example, if you need to add the values in cells F2 through F100 in a given worksheet, **=SUM(F2:F100)** is more efficient than **=SUM(F:F)**.

Taking Advantage of Smart Recalculation

As discussed previously, Excel's "smart" recalculation engine recalculates only those cells that require it at any time. You can help this recalculation engine work more efficiently by adhering to a few principles. First, as discussed on page 39, avoid volatile functions such as INDIRECT unless they are significantly more efficient than alternatives. A volatile function recalculates each time

the workbook or worksheet recalculates, regardless of whether any of its arguments have changed. Thus, a large number of otherwise unnecessary volatile functions will slow calculations.

Consider breaking them into multiple segments and joining their results in smaller formulas if your formulas are lengthy. Further, if you use constants in a formula, calculate the constant outside the formula and then reference the result of that calculation inside the formula. To illustrate, =B3*(.0765) is a more efficient formula than =B3*(0.062+0.0145).

Timing and Testing Each Change

Having timed various spreadsheet components and identified potential bottlenecks, you can make spreadsheet changes to improve performance. However, the effects of these changes may surprise you: they may degrade performance rather than enhance it. Therefore, you should time and test each change before pronouncing the workbook as "improved." The four steps for doing this are as follows.

1. Time the formula or range you want to change using the **RangeTimer** macro.

2. Make your change(s) to the formula or range.

3. Time the changed formula or range using the **RangeTimer** macro and compare it to the timing recorded in step 1.

4. Of course, you should verify that the results are correct if the change(s) improve overall performance.

You should "finalize" the change in the spreadsheet only after completing each of the four outlined steps.

Applied Example of Concepts

Consider the following example illustrating how the techniques described so far can improve spreadsheet performance.

Running Totals

The spreadsheet from which the excerpt shown in **Figure 45** was taken contains 2,000 rows of data in column A and two different types of formulas in columns B and C, respectively, generating running totals. Both formulas provide correct answers, but one of the approaches is more efficient than the other and reduces recalculation time.

	A	B	C
1	Data	SUM Function	"+" Formula
2	1	=SUM($A$2:$A2)	=A2
3	2	=SUM($A$2:$A3)	=C2+A3
4	3	=SUM($A$2:$A4)	=C3+A4
5	4	=SUM($A$2:$A5)	=C4+A5
6	5	=SUM($A$2:$A6)	=C5+A6
7	6	=SUM($A$2:$A7)	=C6+A7
8	7	=SUM($A$2:$A8)	=C7+A8

Figure 45 - Excerpt of Running Total Worksheet

The formula in column B is a traditional SUM function. Using relative and absolute addresses, you can build the formula shown in cell B2 and copy it down the range. For example, in cell B2, the formula refers to only one cell; however, in cell B2001, it refers to 2,000 cells. Since the average number of cells to which the formula refers is 1,000, and the number of rows is 2,000, multiplying these two numbers together tells us that the formulas in column B refer to 2,000,000 cells.

On the other hand, the formula in column C is a simple “plus” formula. In cell C2, the formula refers to only one cell; in all other cells, it refers to just two cells. The formulas in column C reference 3,999 cells [1 + (2*1999)], 500% fewer than the formula in column B. Accordingly, we expect the formulas in column C to calculate faster than the formulas in column B, and they do. As shown in **Table 2**, the average amount of time to calculate the formulas in column B was roughly **five times greater** than the average time to calculate Column C’s formulas.

Number of Seconds Required to Complete Calculations		
	Formula in Column B	**Formula in Column C**
Test 1	0.04605	0.00977
Test 2	0.04705	0.01025
Test 3	0.05303	0.00910
Test 4	0.05326	0.00836
Test 5	0.04226	0.00901
Average	**0.04833**	**0.00930**

Table 2 - Results of Calculation Times of Running Total Formulas

Of course, as the spreadsheet's size increases, so does the disparity in the recalculation results, not only in relative terms but also in absolute terms. If, for example, the data range expands to 50,000 rows, the average amount of time to recalculate column B increases to approximately 22 seconds. However, the average time to recalculate column C increases to only 0.22 seconds, ***one hundred times less than column B!*** These test results show how formulas can significantly affect spreadsheet performance.

?	Why is it essential to construct Excel workbooks in such a fashion to minimize recalculation times? Which of the techniques you learned about in Chapter Two can you use to help achieve that objective?

Creating, Auditing, And Verifying Formulas

Chapter 3

Formulas are one of the cornerstones of Excel spreadsheets; therefore, auditing and verifying them are critical issues when addressing the accuracy of your Excel models. This chapter describes best practices for auditing and verifying formulas in Excel workbooks.

Learning Objectives

Upon completing this chapter, you should be able to:

- Describe how tools such as Insert Function, the Data Analysis ToolPak, and Flash Fill can speed the process of creating formulas in Excel and reduce the need to create formulas;
- Explain how to identify and mark all the formulas in an Excel workbook;
- List and describe each of the seven options in the Formula Auditing group of the Formulas tab of the Ribbon;
- Describe how to create defined names and how to use defined names to help with building formulas;
- Discuss how working with Tables in Excel improves formula accuracy and efficiency; and
- Describe the key differences between Excel's SUM and SUBTOTAL functions.

Creating Formulas

One of Excel's cornerstones is the ability to enter formulas to perform calculations. Though a detailed discussion of all formula possibilities is beyond this course's scope, let us examine best practices for building better formulas and opportunities to minimize the number of formulas we create.

Using Insert Function

One commonly overlooked feature for improving Excel formulas is the **Insert Function** tool. This tool becomes even more helpful when you work with an unfamiliar function. You can access Insert Function from the Ribbon's **Formulas** tab; you can also access it by clicking the ***fx*** icon next to the Formula Bar in Excel. Upon accessing Insert Function, the **Insert Function** dialog box

appears. As shown in **Figure 46**, the tool allows you to search for a function and guides you through building a formula that contains that function.

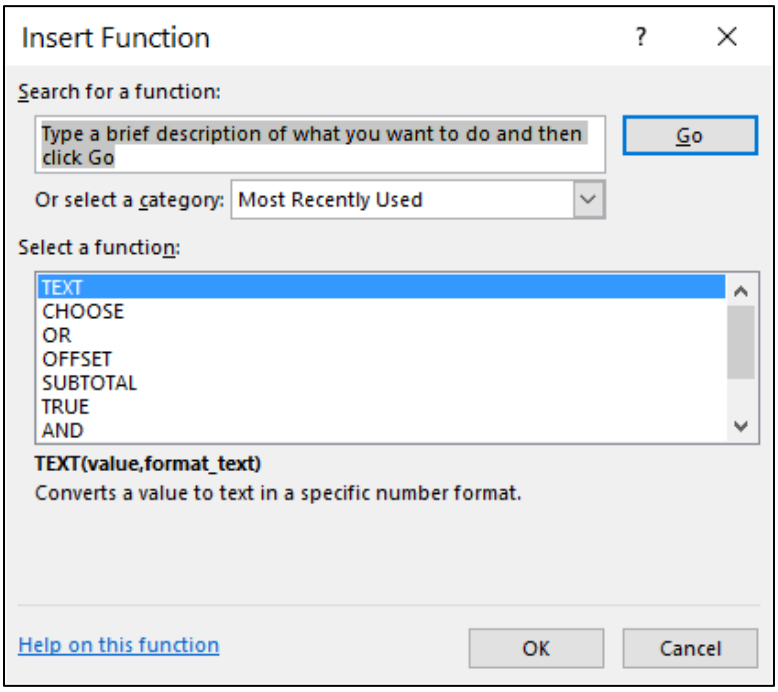


Figure 46 - Insert Function Dialog Box

Following the example in Figure 46, if you select the **Text** function and click **OK**, the **Function Arguments** dialog box pictured in **Figure 47** appears, guiding you through building the formula containing that function. Using the Insert Function feature, you can help ensure that the formulas you create that use functions you may not be familiar with are accurate and provide the correct result.

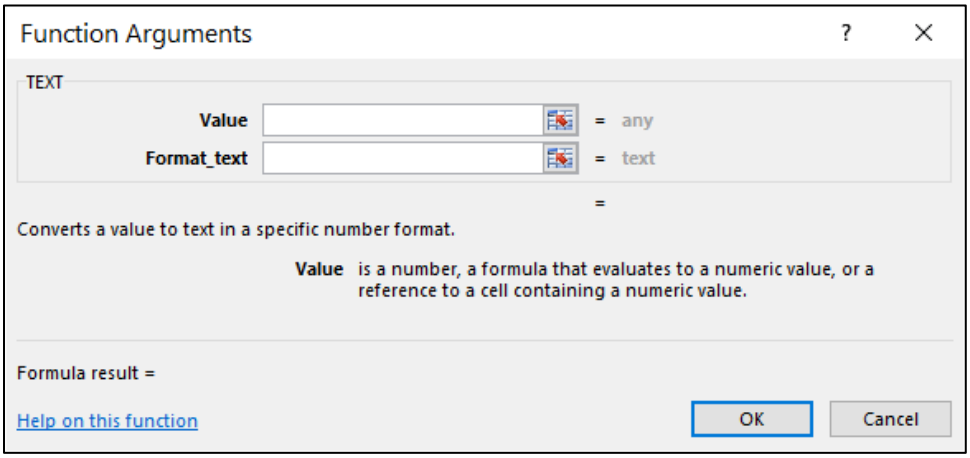


Figure 47 - Function Arguments Dialog Box

Using Alternatives to Formulas

Sometimes, using formulas and functions is unnecessary to obtain the desired result. In this section, we examine some best practices for alternatives to formulas.

Data Analysis ToolPak

The ***Data Analysis ToolPak*** is an Excel add-in that is not activated by default. On the Ribbon's **Data** tab, if you see **Data Analysis** in the **Analysis** group, the Data Analysis ToolPak is active. If the option is not on the Ribbon, activate it by accessing Excel's **Options**, choosing **Add-Ins**, and clicking **Go** next to **Manage Excel Add-ins** to open the **Add-Ins** dialog box. As shown in **Figure 48**, check the boxes next to **Analysis ToolPak** and **Analysis ToolPak – VBA**.

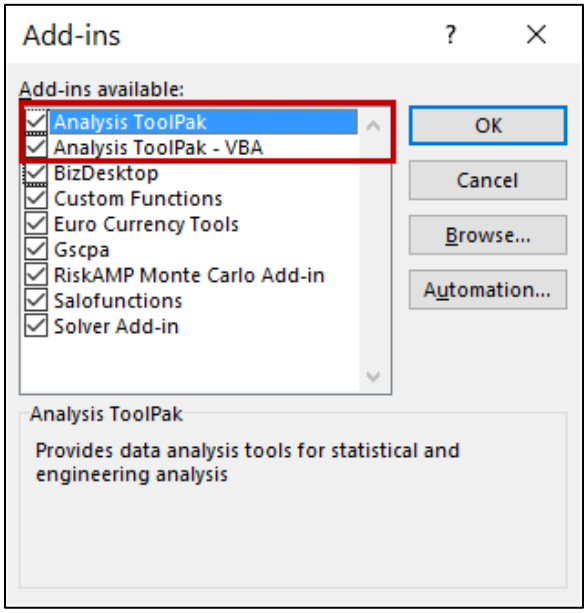


Figure 48 - Activating the Data Analysis ToolPak

The Data Analysis ToolPak provides access to many tools that automate calculations and procedures that could otherwise be complex to build and manage. For example, the ToolPak includes functions to automate calculating statistics, creating histograms, performing regression analysis, generating random numbers, and computing moving averages.

For example, suppose you need to generate a histogram to support a Benford's Law analysis of accounts payable transactions. You can complete this task using the Histogram tool in the Data Analysis ToolPak. First, open the **Data Analysis ToolPak**. Then click Histogram, then **OK** to create a histogram. Next, enter the **Input Range** and the **Bin Range** in the **Histogram** dialog box. Finally, select the desired outputs, as shown in **Figure 49**.

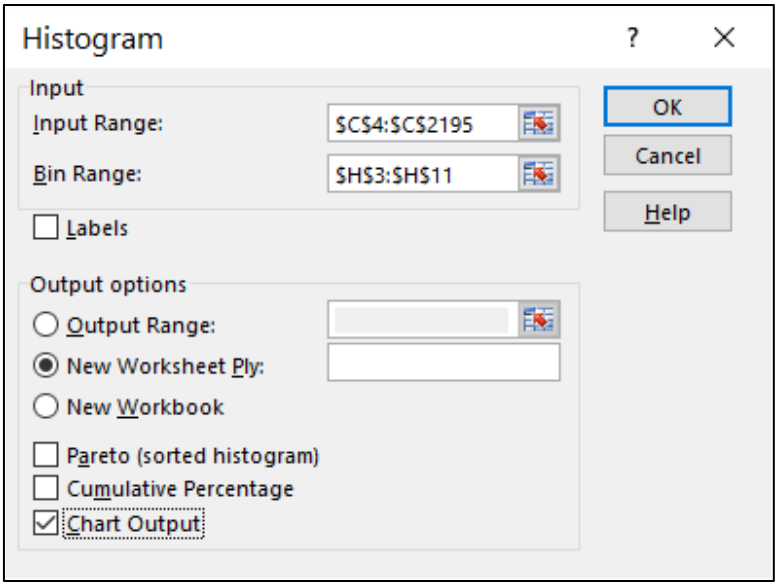
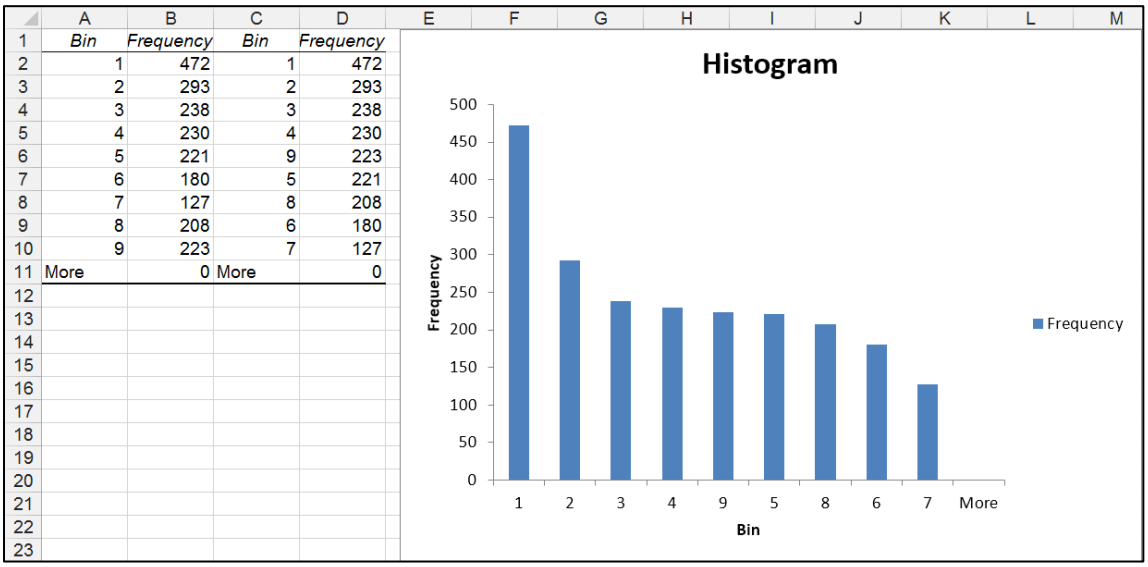


Figure 49 - Creating a Histogram for Benford's Law Analysis

Excel creates a histogram summarizing the data when you click **OK**, as shown in **Figure 50**. This output lets you determine whether the data aligns with Benford's Law.



	A	B	C	D
1	*Bin*	*Frequency*	*Bin*	*Frequency*
2	1	472	1	472
3	2	293	2	293
4	3	238	3	238
5	4	230	4	230
6	5	221	9	223
7	6	180	5	221
8	7	127	8	208
9	8	208	6	180
10	9	223	7	127
11	More	0	More	0

Figure 50 - Histogram Created by Data Analysis ToolPak

Flash Fill

Introduced in Excel with the 2013 release, Flash Fill lets you avoid using formulas for some delimiting and concatenation needs. Before learning to use Flash Fill, we will examine formula-based approaches to these same functions. For example, suppose you open an Excel workbook that contains information similar to that shown in **Figure 51,** and you need to separate name, address, city, state, and postal code information.

	A
1	Nancy Freehafer, 123 Main St, Atlanta, GA, 30303
2	Andrew Cencini, 456 Elm St, Houston, TX, 80808
3	Jan Kotas, 789 Oak Ave, San Francisco, CA, 40404
4	Mariya Sergienko, 1234 Hill Blvd, Los Angeles, CA, 20202
5	Steven Thorpe, 6789 Biscayne Ln, Miami, FL, 50505
6	Michael Neipper, 135 Ave K, New York, NY, 10101
7	Robert Zare, 79-B Central Ave, Minneapolis, MN, 70707
8	Laura Giussani, 246 Park Ave, Dallas, TX, 60606
9	Anne Hellung-Larsen, 468 Spuce Dr, Seattle, WA, 90909
10	Anna Bedecs, 689 Pine Ave, Portland, OR, 12121
11	Antonio Gratacos Solsona, 297 Vista Dr, Denver, CO, 23232
12	Thomas Axen, 638 Golden Peak, Salt Lake City, UT, 34343

Figure 51 - Name and Address Information in One Field

You could separate the information using a formula-based approach incorporating Excel's **LEN**, **LEFT**, **RIGHT**, **MID**, and **FIND** functions. For example, **Table 3** presents sample formulas to extract the desired information from the spreadsheet pictured in Figure 51.

In cell B1, enter this formula to extract only the name from cell A1…	=LEFT(A1,FIND(",",A1)-1)
In cell C1, enter this formula to extract only the address from cell A1…	=MID(A1,FIND(",",A1)+2, FIND(",",A1,LEN(B1)+2)-LEN(B1)-3)
In cell D1, enter this formula to extract only the city from cell A1…	=MID(A1,LEN(B1)+LEN(C1)+5, LEN(A1)-LEN(B1)-LEN(C1)-LEN(E1)-LEN(F1)-8)
In cell E1, enter this formula to extract only the state from cell A1…	=MID(A1,LEN(A1)-8,2)
In cell F1, enter this formula to extract only the postal code from cell A1…	=RIGHT(A1,5)

Table 3 - Sample Formulas to Extract Information

Though some formulas in Table 3 They are relatively simple to understand and construct, but some are somewhat tedious and prone to error. Therefore, you may prefer a more straightforward approach to extracting the desired elements from the spreadsheet.

You can use **Flash Fill** to extract the desired elements from the spreadsheet. Flash Fill uses pattern recognition algorithms to identify patterns in the data. Once it identifies patterns, it uses them

to manipulate the data based on their relationships to those patterns. To illustrate how easily you can extract data using this tool, explore the modified spreadsheet pictured in **Figure 52**.

	A	B	C	D	E
1	Nancy Freehafer, 123 Main St, Atlanta, GA, 30303	Nancy Freehafer	123 Main St	Atlanta, GA	30303
2	Andrew Cencini, 456 Elm St, Houston, TX, 80808				
3	Jan Kotas, 789 Oak Ave, San Francisco, CA, 40404				
4	Mariya Sergienko, 1234 Hill Blvd, Los Angeles, CA, 20202				
5	Steven Thorpe, 6789 Biscayne Ln, Miami, FL, 50505				
6	Michael Neipper, 135 Ave K, New York, NY, 10101				
7	Robert Zare, 79-B Central Ave, Minneapolis, MN, 70707				
8	Laura Giussani, 246 Park Ave, Dallas, TX, 60606				
9	Anne Hellung-Larsen, 468 Spuce Dr, Seattle, WA, 90909				
10	Anna Bedecs, 689 Pine Ave, Portland, OR, 12121				
11	Antonio Gratacos Solsona, 297 Vista Dr, Denver, CO, 23232				
12	Thomas Axen, 638 Golden Peak, Salt Lake City, UT, 34343				

Figure 52 - Spreadsheet Modified for Flash Fill

In cells B1, C1, D1, and E1, indicate the pattern you would like Excel to identify for Flash Fill. Upon doing so, select any cell in cells B1 through B12. Then, on the **Data** tab of the Ribbon, click **Flash Fill**. Excel fills the remainder of the column with the extracted names from column A. Repeat the process for columns C, D, and E, and your spreadsheet will appear as shown in **Figure 53**.

	A	B	C	D	E
1	Nancy Freehafer, 123 Main St, Atlanta, GA, 30303	Nancy Freehafer	123 Main St	Atlanta, GA	30303
2	Andrew Cencini, 456 Elm St, Houston, TX, 80808	Andrew Cencini	456 Elm St	Houston, TX	80808
3	Jan Kotas, 789 Oak Ave, San Francisco, CA, 40404	Jan Kotas	789 Oak Ave	San Francisco, CA	40404
4	Mariya Sergienko, 1234 Hill Blvd, Los Angeles, CA, 20202	Mariya Sergienko	1234 Hill Blvd	Los Angeles, CA	20202
5	Steven Thorpe, 6789 Biscayne Ln, Miami, FL, 50505	Steven Thorpe	6789 Biscayne Ln	Miami, FL	50505
6	Michael Neipper, 135 Ave K, New York, NY, 10101	Michael Neipper	135 Ave K	New York, NY	10101
7	Robert Zare, 79-B Central Ave, Minneapolis, MN, 70707	Robert Zare	79-B Central Ave	Minneapolis, MN	70707
8	Laura Giussani, 246 Park Ave, Dallas, TX, 60606	Laura Giussani	246 Park Ave	Dallas, TX	60606
9	Anne Hellung-Larsen, 468 Spuce Dr, Seattle, WA, 90909	Anne Hellung-Larsen	468 Spuce Dr	Seattle, WA	90909
10	Anna Bedecs, 689 Pine Ave, Portland, OR, 12121	Anna Bedecs	689 Pine Ave	Portland, OR	12121
11	Antonio Gratacos Solsona, 297 Vista Dr, Denver, CO, 23232	Antonio Gratacos Solsona	297 Vista Dr	Denver, CO	23232
12	Thomas Axen, 638 Golden Peak, Salt Lake City, UT, 34343	Thomas Axen	638 Golden Peak	Salt Lake City, UT	34343

Figure 53 - Extracted Data Using Excel Flash Fill Tool

You can also use Flash Fill to concatenate or join information together. For example, consider the spreadsheet pictured in **Figure 54**.

	A	B	C	D	E
1	Nancy Freehafer	123 Main St	Atlanta, GA	30303	Nancy Freehafer, 123 Main St, Atlanta, GA, 30303
2	Andrew Cencini	456 Elm St	Houston, TX	80808	
3	Jan Kotas	789 Oak Ave	San Francisco, CA	40404	
4	Mariya Sergienko	1234 Hill Blvd	Los Angeles, CA	20202	
5	Steven Thorpe	6789 Biscayne Ln	Miami, FL	50505	
6	Michael Neipper	135 Ave K	New York, NY	10101	
7	Robert Zare	79-B Central Ave	Minneapolis, MN	70707	
8	Laura Giussani	246 Park Ave	Dallas, TX	60606	
9	Anne Hellung-Larsen	468 Spuce Dr	Seattle, WA	90909	
10	Anna Bedecs	689 Pine Ave	Portland, OR	12121	
11	Antonio Gratacos Solsona	297 Vista Dr	Denver, CO	23232	
12	Thomas Axen	638 Golden Peak	Salt Lake City, UT	34343	

Figure 54 - Sample Data for Joining Data with Flash Fill

Cell **E1** contains a pattern for Flash Fill to use to join data. After entering the pattern, click any cell in E1 through E12. Then, choose **Flash Fill** from the **Data** tab of the Ribbon. Upon doing so, Excel fills the range with joined data based on the pattern established in cell E1, as shown in **Figure 55**.

	A	B	C	D	E
1	Nancy Freehafer	123 Main St	Atlanta, GA	30303	Nancy Freehafer, 123 Main St, Atlanta, GA, 30303
2	Andrew Cencini	456 Elm St	Houston, TX	80808	Andrew Cencini, 456 Elm St, Houston, TX, 80808
3	Jan Kotas	789 Oak Ave	San Francisco, CA	40404	Jan Kotas, 789 Oak Ave, San Francisco, CA, 40404
4	Mariya Sergienko	1234 Hill Blvd	Los Angeles, CA	20202	Mariya Sergienko, 1234 Hill Blvd, Los Angeles, CA, 20202
5	Steven Thorpe	6789 Biscayne Ln	Miami, FL	50505	Steven Thorpe, 6789 Biscayne Ln, Miami, FL, 50505
6	Michael Neipper	135 Ave K	New York, NY	10101	Michael Neipper, 135 Ave K, New York, NY, 10101
7	Robert Zare	79-B Central Ave	Minneapolis, MN	70707	Robert Zare, 79 B Central Ave, Minneapolis, MN, 70707
8	Laura Giussani	246 Park Ave	Dallas, TX	60606	Laura Giussani, 246 Park Ave, Dallas, TX, 60606
9	Anne Hellung-Larsen	468 Spuce Dr	Seattle, WA	90909	Anne Hellung-Larsen, 468 Spuce Dr, Seattle, WA, 90909
10	Anna Bedecs	689 Pine Ave	Portland, OR	12121	Anna Bedecs, 689 Pine Ave, Portland, OR, 12121
11	Antonio Gratacos Solsona	297 Vista Dr	Denver, CO	23232	Antonio Gratacos Solsona, 297 Vista Dr, Denver, CO, 23232
12	Thomas Axen	638 Golden Peak	Salt Lake City, UT	34343	Thomas Axen, 638 Golden Peak, Salt Lake City, UT, 34343

Figure 55 - Using Flash Fill to Join Data

Flash Fill is a potent tool that is also relatively easy to use. Accordingly, Flash Fill is a best practice for extracting, concatenating, and reformatting static data.

? How can you use Flash Fill to speed up everyday tasks? For what types of activities should you use Flash Fill? For which activities should you avoid using Flash Fill?

Text to Columns

You can use **Text to Columns** to transform “pseudo dates” into “real dates,” convert text-based numbers into numerical values, extract data from one cell into others, and change Julian dates into Roman dates, among other actions. Because of its power, some refer to Excel’s Text to

Columns tool as “Excel’s Swiss Army knife.” The following example illustrates the power of Excel’s Text to Columns feature.

In Excel, “real dates” are numbers formatted to look like dates. Because dates are numbers, you can perform any mathematical operation on them you desire, including subtracting one date from another to determine the time that elapses between the two. However, occasionally, those who import data from other applications and databases into Excel will find that dates appear as text strings, also known as “pseudo dates.” Because these are text strings, you cannot perform mathematical operations on them. Accordingly, you will need to convert the pseudo dates into “real” dates, and Text to Columns can be a best practice for accomplishing this task.

To illustrate, consider the entries in column A in **Figure 56**. Note the entries *appear* as dates, but closer inspection in the formula bar shows that an apostrophe prefaces each entry, a tell-tale sign that these entries are text labels and not “real” Excel dates.

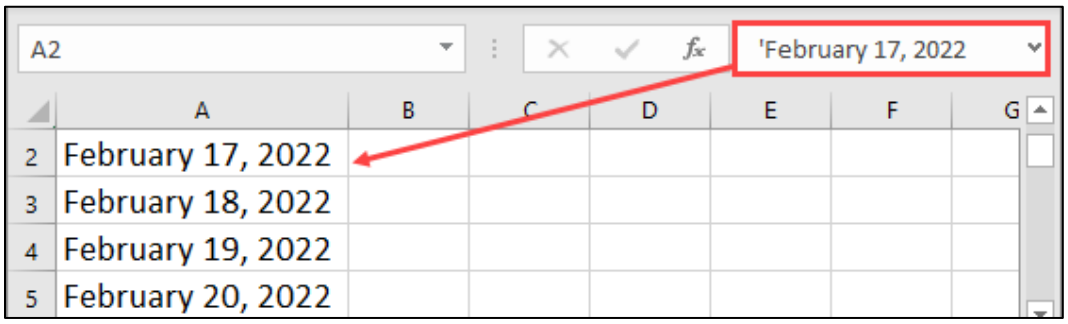


Figure 56 - Pseudo-Dates in Excel

To convert the entries into Excel dates, select cells A2 through A6. On the **Data** tab of the Ribbon, choose **Text to Columns**. Select **Delimited** in the **Step 1 of 3 menu** and click **Next** in the **Convert Text to Columns Wizard** dialog box**. Click Next in** the **Step 2 of 3** menu to arrive at **Step 3 of 3** in the Wizard. As shown in **Figure 57**, choose **Date**, indicate your desired format for the output, and specify a destination for the results.

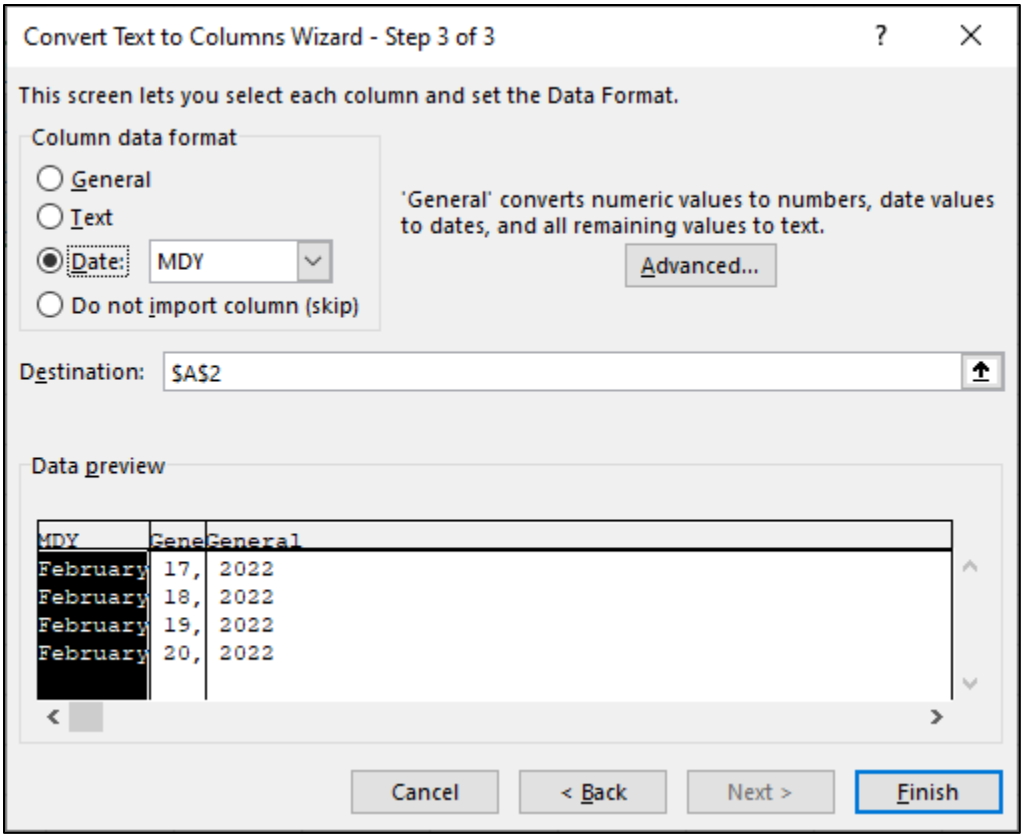


Figure 57 – Using the Text to Columns Wizard to Convert Text to Dates

Upon clicking **Finish**, Excel converts the text entries into Excel dates, as shown in **Figure 58**.

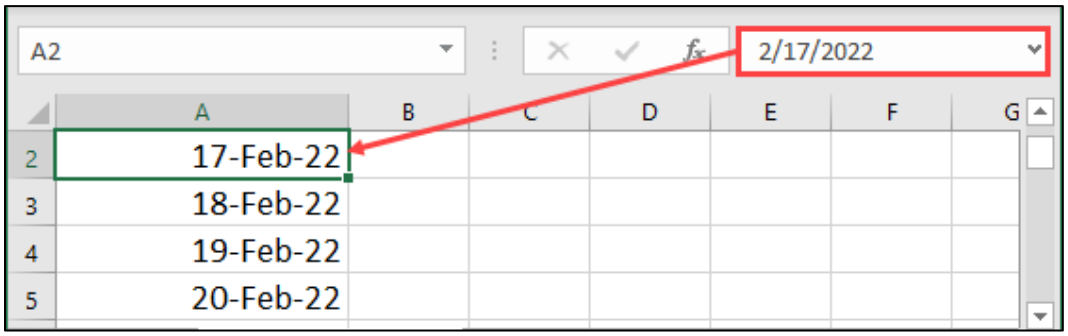


Figure 58 - Dates Converted by Text to Columns Wizard

If you use Flash Fill or Text to Columns, remember that these tools do not operate dynamically. If your source data changes, neither Flash Fill nor Text to Columns updates prior results based on the new data. If your source data may change, use a formula-based approach to extract and concatenate the data.

Identifying All Formulas in a Workbook

You first need to know which cells contain formulas to audit and verify them. This need can sometimes prove problematic when working on spreadsheets that others have prepared. This section examines best practices for identifying all formulas in a workbook.

Selecting Cells with Formulas

Selecting all cells containing formulas in a spreadsheet is relatively easy to complete. To determine all the cells that contain formulas, begin by selecting the range you wish to examine; otherwise, Excel will act on the entire worksheet. Then, choose **Find & Select** on the **Home** tab, then **Formulas**. Then, Excel selects all cells containing formulas, allowing you to identify them quickly, as shown in **Figure 59**.

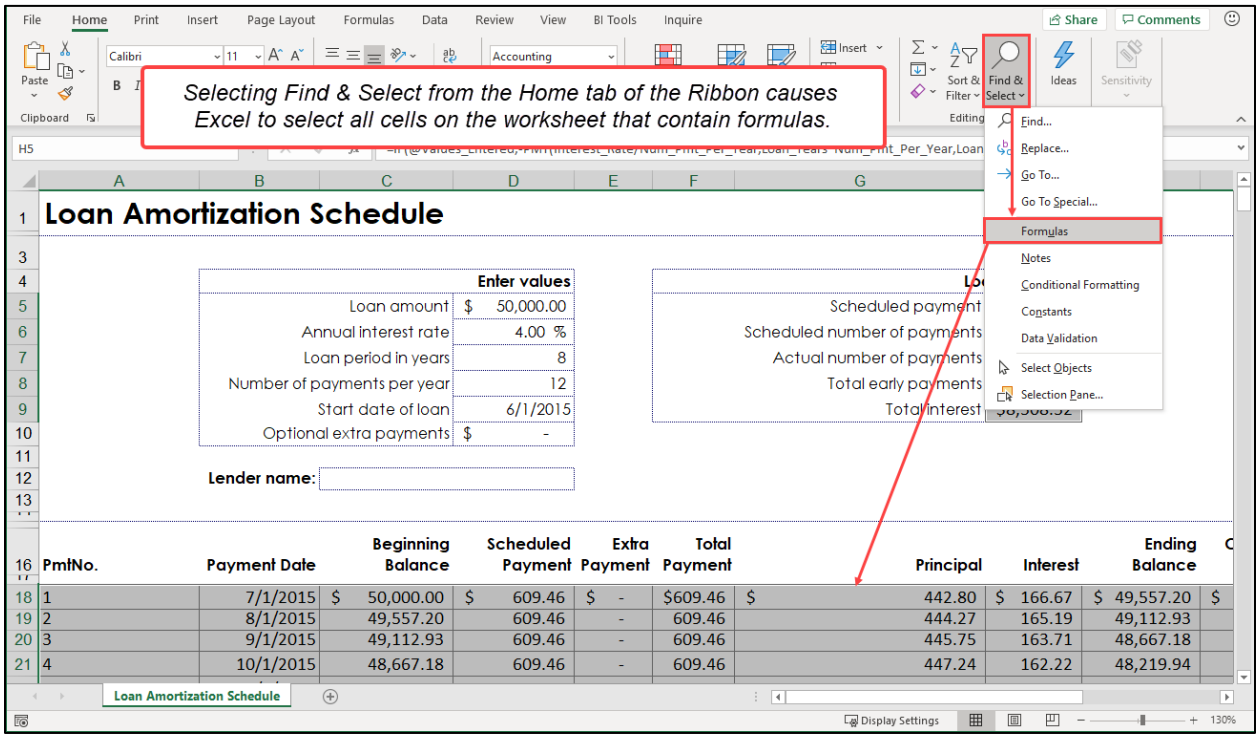


Figure 59 - Selecting All Cells That Contain Formulas

Adding Formatting to Cells Containing Formulas

While selecting all cells containing formulas might be helpful, remember that this is only a temporary selection. The selection disappears once you click any cell or press the **Esc** key. Therefore, as a best practice, you may consider adding formatting – such as shading – to all cells containing formulas. This action lets you and other users see which cells have formulas without repeatedly invoking the **Find & Select, Formulas** technique described above.

To add formatting to all cells containing formulas, use the **Find & Select**, **Formulas** to select all the cells in a range or worksheet containing formulas. Then choose the individual formatting characteristics or Styles you wish to apply, and Excel will apply them to all the selected cells. After applying all desired formats, press Esc or click any cell to deselect all selected cells.

Viewing Excel in Formula View

The previous illustrations help identify which cells contain formulas, but do nothing to help you see these formulas' contents. When you want to see all formulas' details in a worksheet, you can place the worksheet in **Formula View**. From the Ribbon's **Formulas** tab, choose **Show Formulas** and Excel changes to **Formula View**, shown in **Figure 60**.

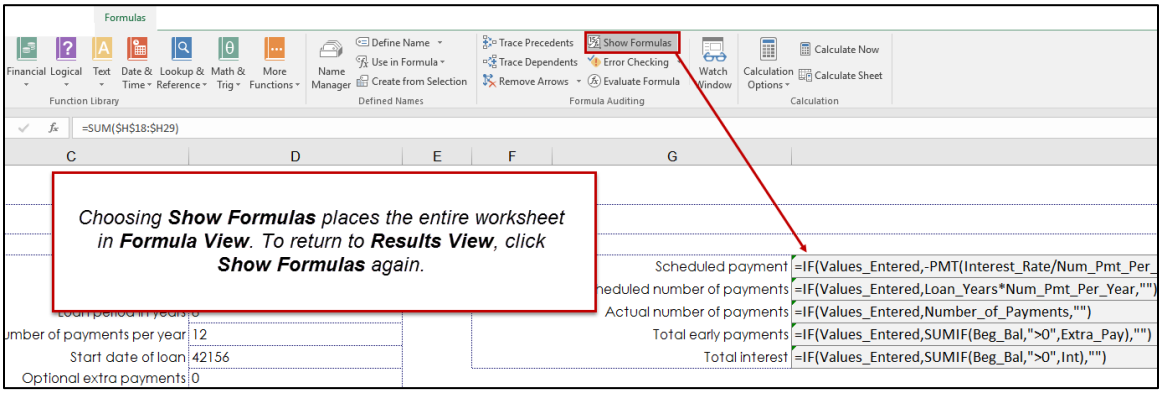


Figure 60 - Switching a Worksheet Formula View

If you are more inclined to use keyboard shortcuts, you can use a **CTRL +** ` keyboard shortcut to toggle between Formula and Results Views.

Viewing Formulas and Results

As valuable as Formula View is, it would be even more effective if you could see formulas and results simultaneously. You can do this by opening the worksheet in a second window. Begin by clicking **New Window** on the Ribbon's **View** tab to open a second window of the same worksheet. Next, select **Arrange All** on the **View** tab to open the **Arrange Windows** dialog box, pictured in **Figure 61**.

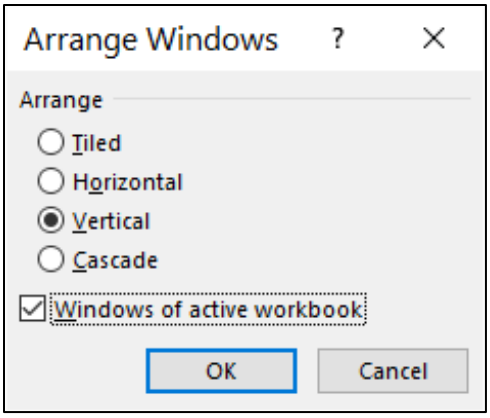


Figure 61 - Arrange Windows Dialog Box

In the Arrange Windows dialog box, choose **Vertical** and check the box next to **Windows of the active workbook** to arrange the workbook's two windows, as shown in **Figure 62**.

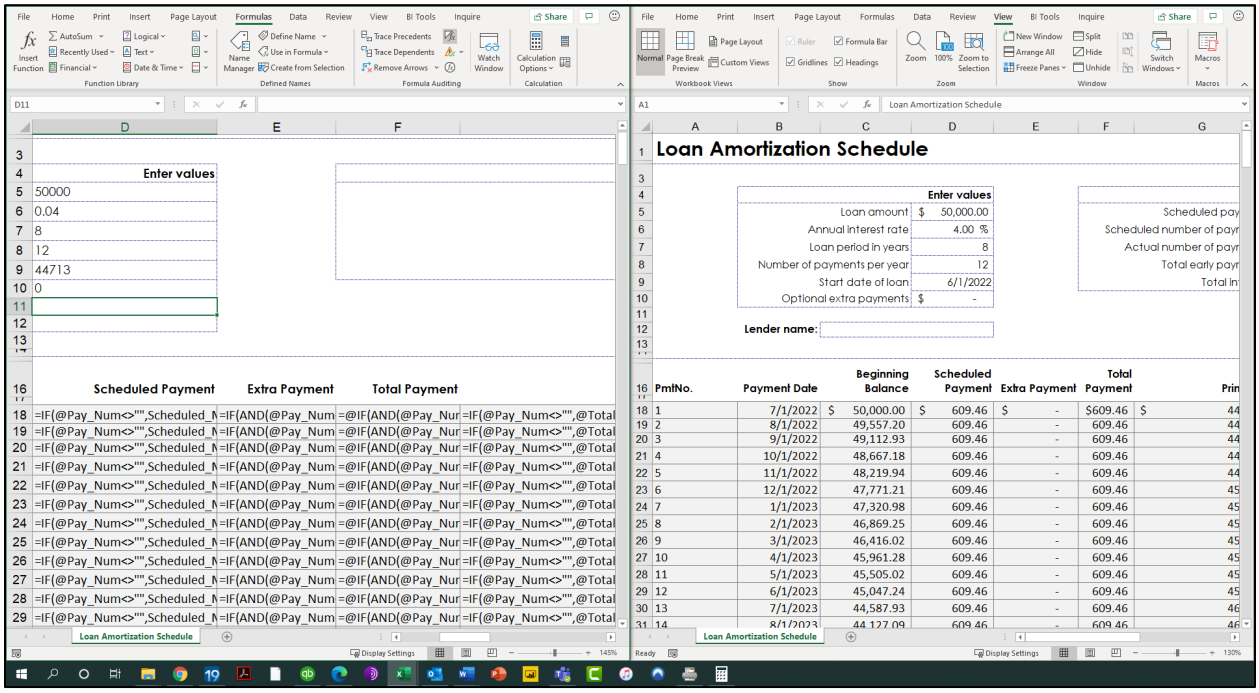


Figure 62 - Viewing Two Windows of the Active Workbook

Auditing and Verifying Formulas

Of course, to ensure your spreadsheets' accuracy, you need to see the formulas and audit and verify them. Fortunately, Excel provides numerous tools for verifying that the formulas in your workbook deliver the results you intend.

Tracing Precedents and Dependents

In addition to Excel's Inquire tool, all Excel versions provide tools to help you understand the relationships between cells in your spreadsheets. These tools help you identify which cells supply data to the current cell – ***precedents*** – and which cells depend on the current cell – ***dependents***. As its name implies, **Trace Precedents** highlights all cells that feed data into the selected cell. Likewise, the **Trace Dependents** feature identifies all cells that depend on the output of a specific cell. These tools reside in the **Formula Auditing** group on the Ribbon's **Formula** tab.

First, select the cell of interest to trace the data that feeds into it. Then, click **Trace Precedents,** and Excel draws lines from all the source cells into the destination cell, as shown in **Figure 63**.

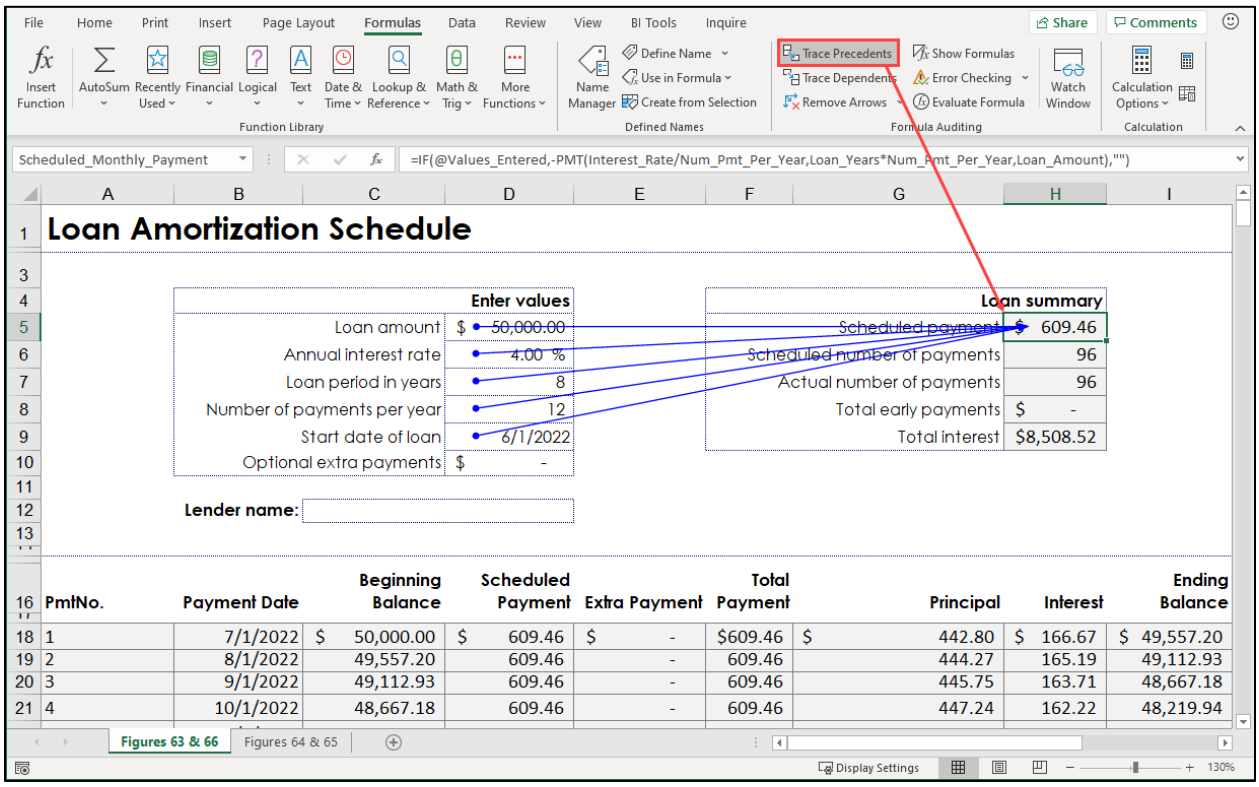


Figure 63 - Tracing Precedents for a Cell

One challenge with Trace Precedents (and Trace Dependents) is identifying cells on other worksheets that link to the selected cell. For instance, suppose the assumptions feeding the loan amortization schedule in **Figure 63** reside on another worksheet, and you desire to run Trace Precedents on cell H5. In that case, upon selecting Trace Precedents, Excel presents a dotted line with an origin that is an icon of a spreadsheet, as shown in **Figure 64**.

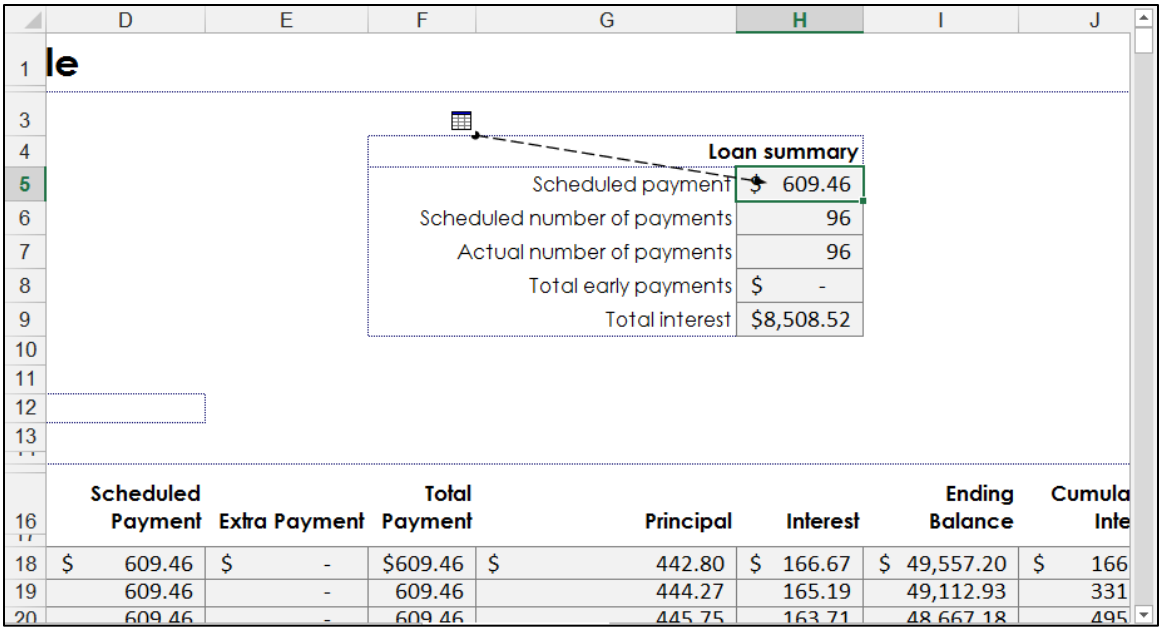


Figure 64 - Tracing Precedents from Another Worksheet

Your natural inclination might be to click the spreadsheet icon to see where the data feeding into cell H5 originates, but that does nothing. Rather, double-click the dotted line extending from the spreadsheet icon to cell H5 to open the **Go To** dialog box pictured in **Figure 65**.

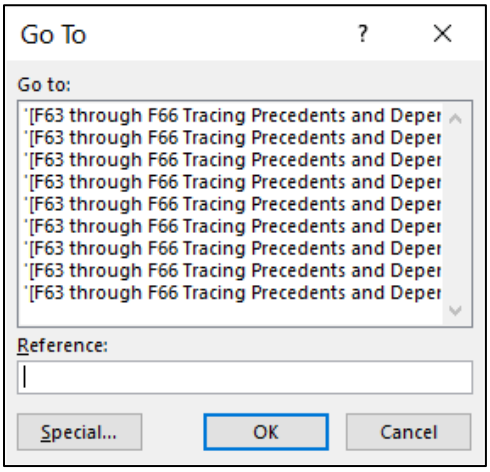


Figure 65 - Identifying Off-Worksheet Precedents

The **Go To** dialog box lists all off-worksheet precedent cells that feed into cell H5. To navigate to any location listed in the Go To dialog box, double-click on that cell reference in the list.

Select the cell of interest to see all the cells depending on the data in a specific cell. Then, as shown in **Figure 66,** click **Trace Dependents** on the Ribbon's **Formulas** tab. Finally, you can click **Remove Arrows** on the Ribbon's **Formula** tab to clear all the relationship lines.

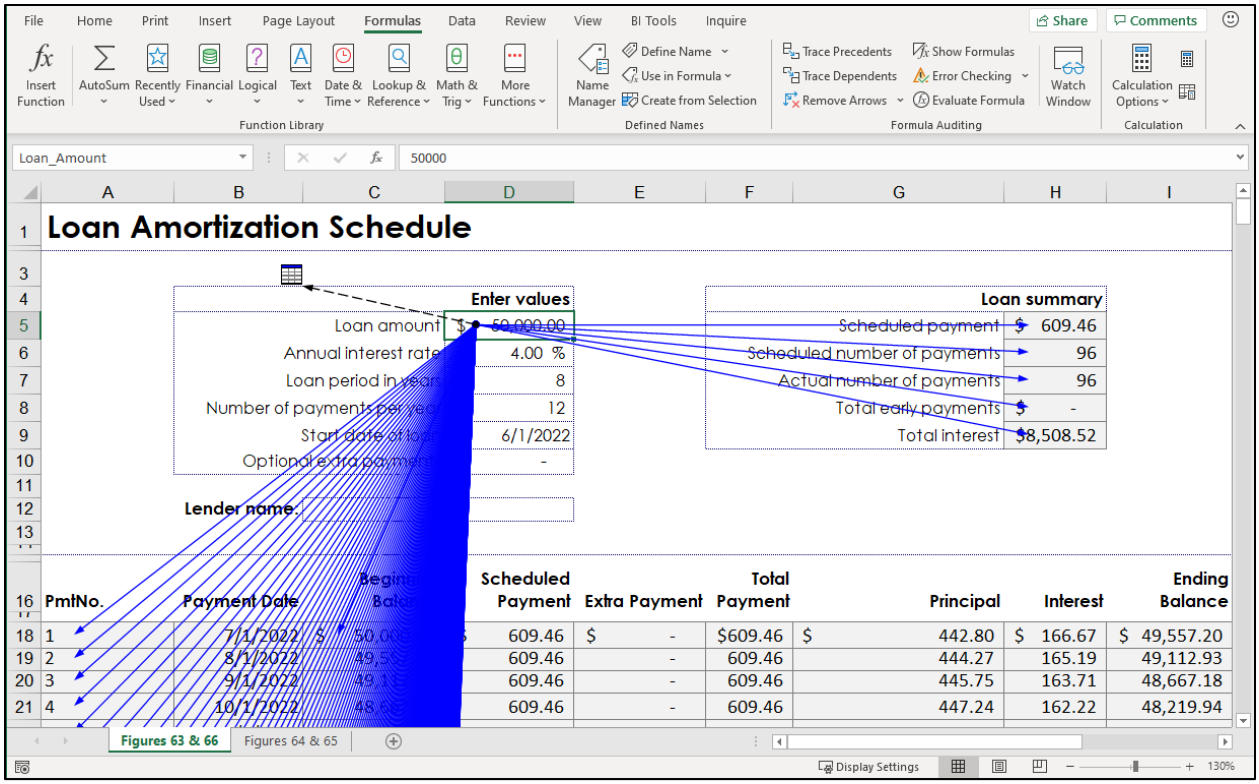


Figure 66 - Tracing Dependents of a Cell

Managing Excel's Error Checking Feature

Excel's **Error Checking** tool, located on the Ribbon's Formulas tab in the Formula Auditing group, provides many useful tools for checking individual cells for errors based on a customizable set of rules. Using this tool is a best practice for ensuring that cells containing formulas do not violate any defined rules.

To customize Excel's rules when checking for errors, visit Excel's **Options**. Then click **Formulas** to access **Error checking rules**, as shown in **Figure 67**. Once you access the rules, you can activate or deactivate any of the nine rules listed. Finally, click **OK** to save your changes.

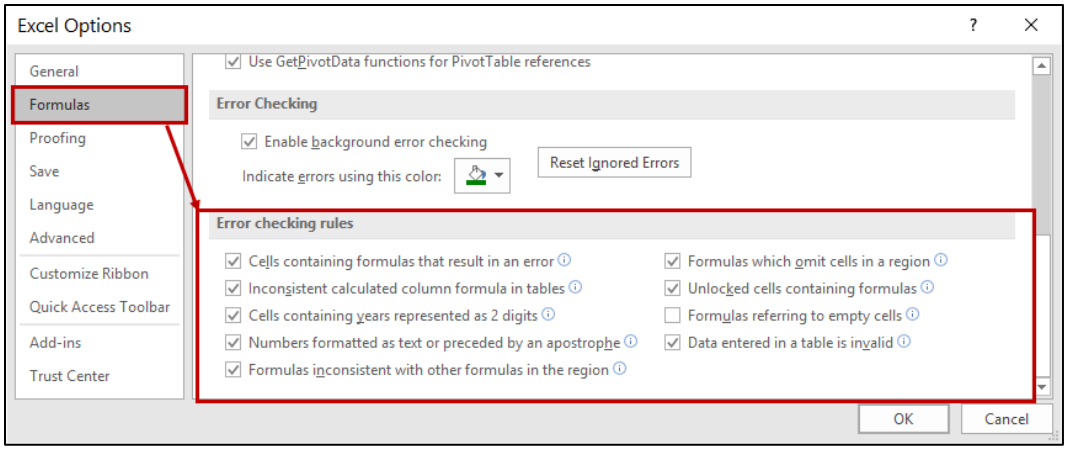


Figure 67 - Customizing Excel's Error Checking Rules

After customizing the rules, you can use Error Checking to identify and correct potential problems in your spreadsheet. For example, running Error Checking on cell H5 in the loan amortization schedule causes Excel to present the **Error Checking** dialog box pictured in **Figure 68**.

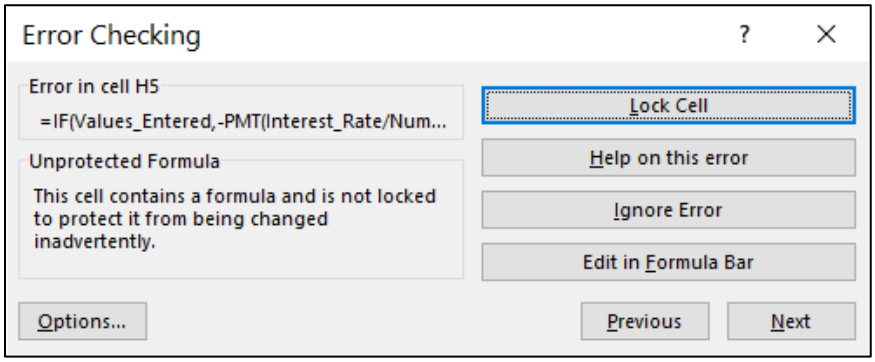


Figure 68 - Error Checking Dialog Box with Unprotected Formula Error Reported

Notice in the dialog box pictured in Figure 68 that Excel displays the formula under consideration and an error message about it. In this case, the error message indicates that the formula remains unprotected because the cell is unlocked.

Following is a second illustration of the usefulness of Error Checking. In this example, running Error Checking on cell E18 causes Excel to generate the Error Checking dialog box pictured in **Figure 69**. In this case, Error Checking reports that one of the precedent cells is empty. Clicking **Trace Empty Cell** in the dialog box immediately navigates to the empty cell so you can enter the appropriate data.

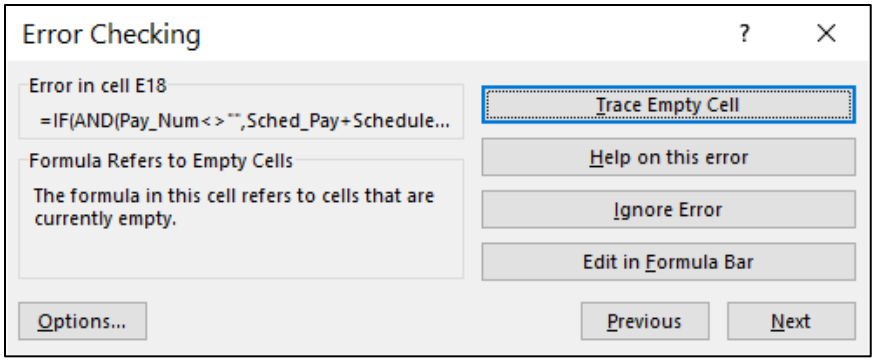


Figure 69 - Error Checking Dialog Box with Empty Cell Error Reported

In sum, running Excel's Error Checking routine on critical formulas in your workbooks provides an excellent option for helping to ensure the accuracy and integrity of formulas and, in turn, the final output of your spreadsheet models. As such, we recommend using it on all critical formulas.

Evaluating Formulas Step-by-Step

In addition to tracing precedent and dependent cells and programmatically checking formulas, Excel provides another tool in the Formula Auditing group of the Formula tab of the Ribbon that can also assist you in validating the accuracy and integrity of formulas you enter into a spreadsheet. This tool, **Evaluate Formula**, lets you watch your formulas calculate step by step. You can test and validate that your formulas calculate correctly, and identify errors in them, using this tool.

To use Evaluate Formula, select the cell that contains the formula you wish to diagnose. Then, click **Evaluate Formula** on the Ribbon's **Formula** tab to open the **Evaluate Formula** dialog box pictured in **Figure 70**.

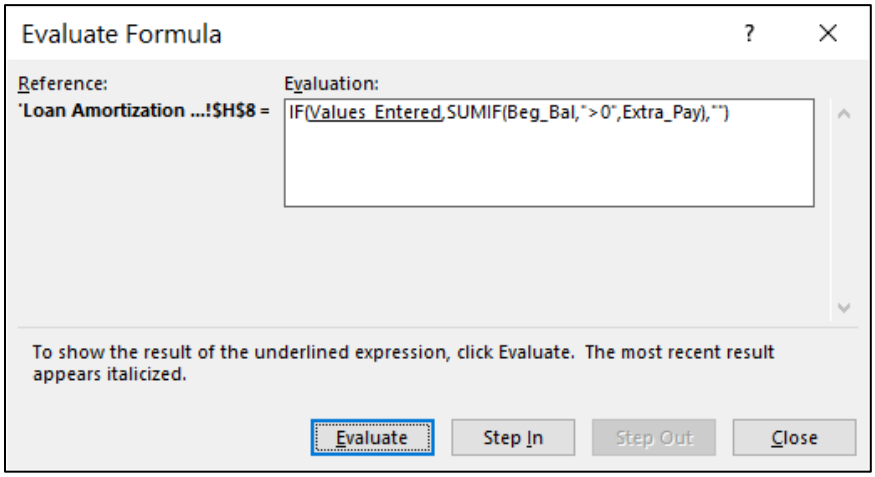


Figure 70 - Evaluate Formula Dialog Box

Excel underlines the first variable it will evaluate when calculating the formula in the **Evaluate Formula** dialog box. You can see the calculation's result by clicking the **Evaluate** button in the dialog box, as shown in **Figure 71**.

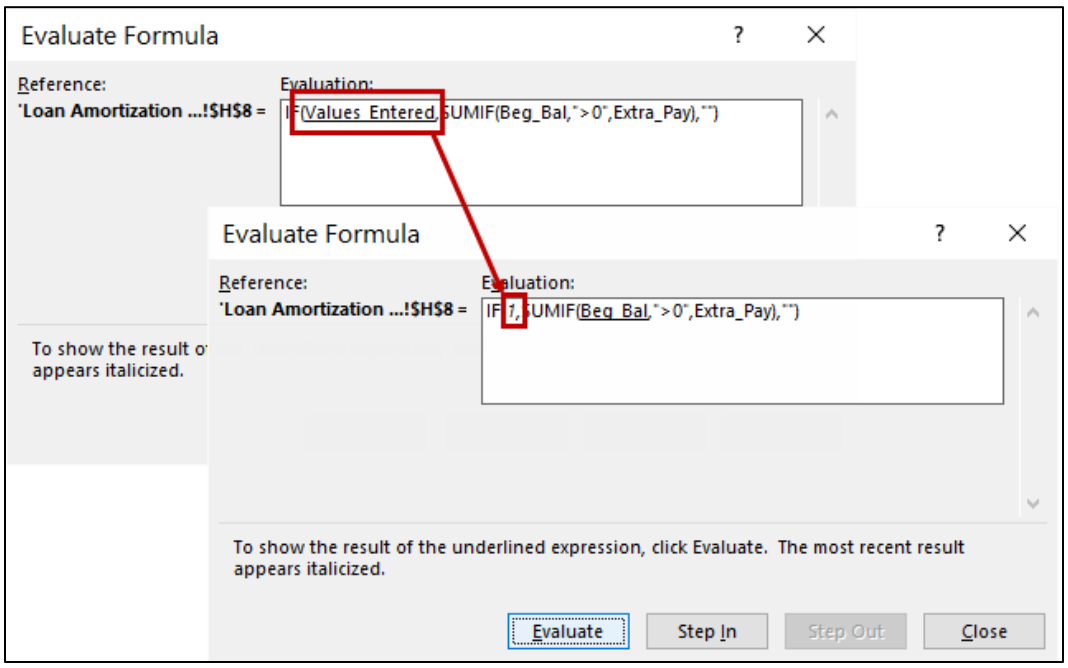


Figure 71 - Evaluating a Section of a Formula

After evaluating the first calculation of the formula, Excel underlines the second calculation, and you can click **Evaluate** again to assess it. You can continue this process until you have evaluated all the formula's calculations and seen its results.

A valuable capability of Evaluate Formula is the ability to separate each reference in the formula into a separate box, which further assists you when validating a formula's accuracy. To illustrate, consider the example presented in **Figure 72**. With Evaluate Formula focused on "Beg_Bal," clicking the **Step In** button causes Excel to show that "Beg_Bal" is a defined name that references cells C18 through C377. To close that section of the Evaluate Formula dialog box, click **Step Out,** and you can continue to evaluate the formula step-by-step.

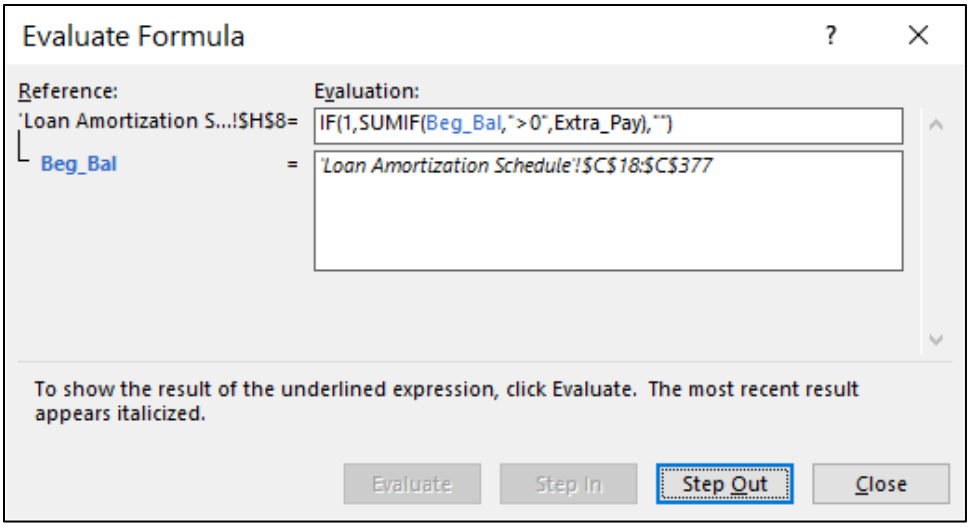


Figure 72 - Step In to a Cell Reference While Evaluating a Formula

Given the complexity of many formulas in Excel workbooks, using Evaluate Formula to verify and troubleshoot formulas is a best practice for ensuring the accuracy of your spreadsheets.

Monitoring Formula Results with the Watch Window

Excel's **Watch Window** feature is the last of the seven tools available in the **Formula Auditing** group on the Ribbon's **Formula** tab. With Watch Window, you can monitor the effect of changes you make to assumptions and formulas on other cells. This capability eliminates the need to repeatedly navigate to these results cells to verify that your "upstream" changes function as intended.

To illustrate, suppose you wanted to monitor how much interest you would pay over the life of a loan in a loan amortization worksheet. To do so, you might place a formula at the bottom of the worksheet that sums the amount of interest; however, since your loan contains 360 payments, you must repeatedly navigate the worksheet to determine life-of-loan interest as you change assumption variables. Alternatively, you could place the cell that sums the interest into a **Watch Window**, a floating window that displays the current values of all cells you monitor.

To place a cell into the Watch Window, click **Watch Window** from the **Formulas** tab of the Ribbon and **Add Watch**. Then, enter the cell you want to monitor in the Watch Window in the **Add Watch** dialog box. It is important to note that the Watch Window does not restrict you to a specific worksheet or workbook; therefore, you can monitor a cell or cells from any open worksheet or workbook, as shown in **Figure 73**.

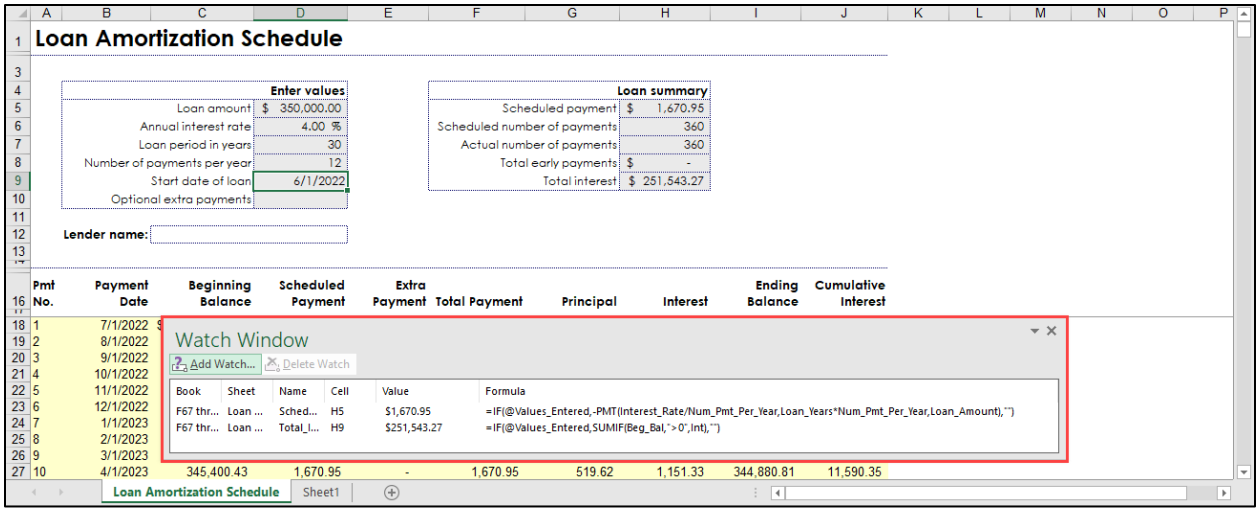


Figure 73 - Using the Watch Window to Monitor Results from Multiple Workbooks

Creating and Working with Defined Names

The concept of ***defined names*** – also known as *named ranges* and *range names* – is not new to spreadsheet users. In fact, in the early days of spreadsheets, using defined names was all but mandated when building anything other than the simplest workbooks. Unfortunately, many Excel users are unaware of the power of defined names and do not use them to streamline and simplify their spreadsheets.

At their core, defined names are nothing more than *aliases* or *nicknames* for a cell or range of cells. Once you create a defined name, you can use it virtually anywhere in Excel you would otherwise use a traditional row and column address. For example, if you assign the defined name of "Cash" to cell B35, you can use "Cash" in any place in Excel – including formulas – where you would traditionally use "B35." This capability makes it evident that one of the primary uses of working with defined names is to *generate self-documenting formulas*. Suppose you assign the defined name "AcctsRec" to cell B36 in the previous example. You could then build a formula that adds the two cells based on their defined names rather than their row and column addresses. The formula would appear as "=Cash+AcctsRec," not "=B35+B36." Because the formula uses defined names, all workbook users can look at that formula and immediately understand the source data.

In addition to generating self-documenting formulas, you can use defined names for other purposes, including:

- Using defined names to speed navigation through large workbooks,
- Storing values, such as allocation percentages,
- Performing two-way lookups,
- Maintaining the integrity of linked data between multiple workbooks, and
- Saving complex formulas that you use repeatedly.

Given the above, working with defined names constitutes an Excel best practice. The following examples reinforce that point.

Establishing Defined Names

Excel provides multiple options for creating defined names. Perhaps the most direct method of creating a defined name is to select the cell or cells to which you want to assign a defined name. Then, with the cell(s) selected, enter the desired name in the **Name Box**, as shown in **Figure 74**.

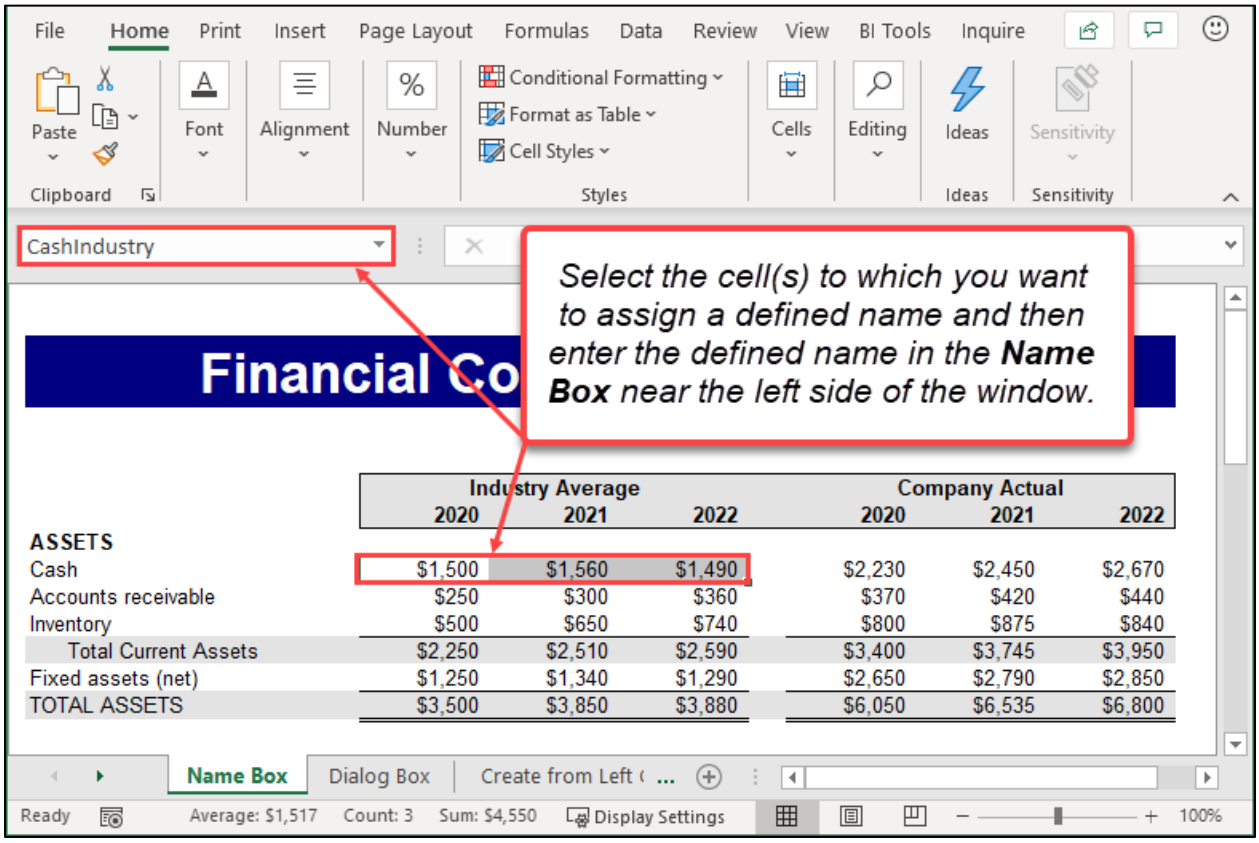


Figure 74 - Creating a Defined Name in the Name Box

When creating defined names, keep in mind several points:

- A defined name can refer to a single cell or a range of cells. If the defined name includes multiple cells, those cells do not need to be contiguous.

- Defined names must adhere to certain naming conventions. For example, defined names cannot exceed 255 characters, and they cannot contain spaces. Likewise, although defined names can include numbers, they cannot begin with them.

- Defined names, by default, possess a workbook-level scope. This characteristic means you can use a defined name on any worksheet in a workbook. However, you can restrict the

scope of any defined name to a specific worksheet. If you do, you can use that defined name in another worksheet within the same workbook only if you preface its reference with the worksheet's name.

- Defined names cannot include combinations of letters and numbers that could otherwise represent a valid row and column address. For example, “A1” cannot be a defined name because “A1” is a valid cell address in Excel.

You can also create defined names from the Formulas tab of the Ribbon. To do so, click **Defined Names** on the **Formulas** tab of the Ribbon, followed by **Define Name** to open the **New Name** dialog box pictured in **Figure 75**. In the **New Name** dialog box, enter the name and the cell references to include in the defined name and click **OK** to create the defined name.

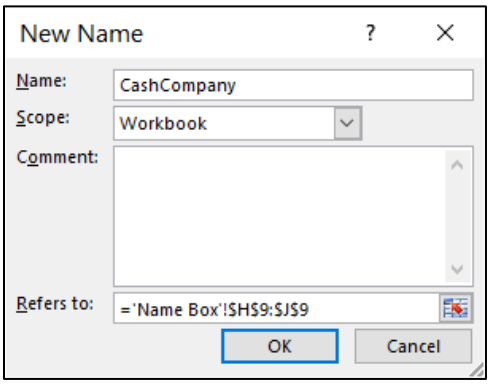


Figure 75 - Creating a Defined Name from the Formulas Tab of the Ribbon

Another option for creating defined names is to have Excel generate them based on column headings and row labels. To illustrate this technique, consider the simple financial statement pictured in **Figure 76**.

	A	B
1	**Balance Sheet**	
3		**December 31, 2022**
4	**ASSETS**	
5	Cash	$2,670
6	Accounts receivable	$440
7	Inventory	$840
8	**Total Current Assets**	**$3,950**
9	Fixed assets (net)	$2,850
10	**TOTAL ASSETS**	**$6,800**
11		
12	**LIABILITIES & EQUITY**	
13	Accts & notes payable	$3,975
14	Accrued & other payables	$650
15	**Total Current Liabilities**	**$4,625**
16	Long-term debt	$1,825
17	**Total Liabilities**	**$6,450**
18	**Shareholders' equity**	**$350**
19	**TOTAL LIABILITIES & EQUITY**	**$6,800**

Figure 76 - Sample Balance Sheet

To create defined names for each entry in cells B5 through B19 based on the row labels in cells A5 through A19, begin by selecting the range of A5 through B19. Next, click **Create from Selection** on the Ribbon's **Formulas** tab with the range selected. This action opens the **Create Names from Selection** dialog box shown in **Figure 77**.

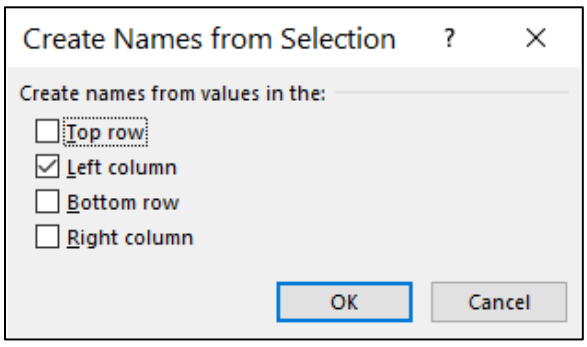


Figure 77 - Creating Defined Names from Selected Cells

In the Create Names from Selection dialog box, check the box next to **Left column** and click **OK**. Upon doing so, Excel creates defined names for the selected cells in column B based on the corresponding row labels in column A.

Identifying Defined Names

You can see all the defined names in a workbook by clicking **Name Manager** on the Ribbon's **Formula** tab. Doing so opens the **Name Manager** shown in **Figure 78**. Notice that not only does the Name Manager list all defined names, but it also lists what cells each defined name refers to and the current value associated with each defined name.

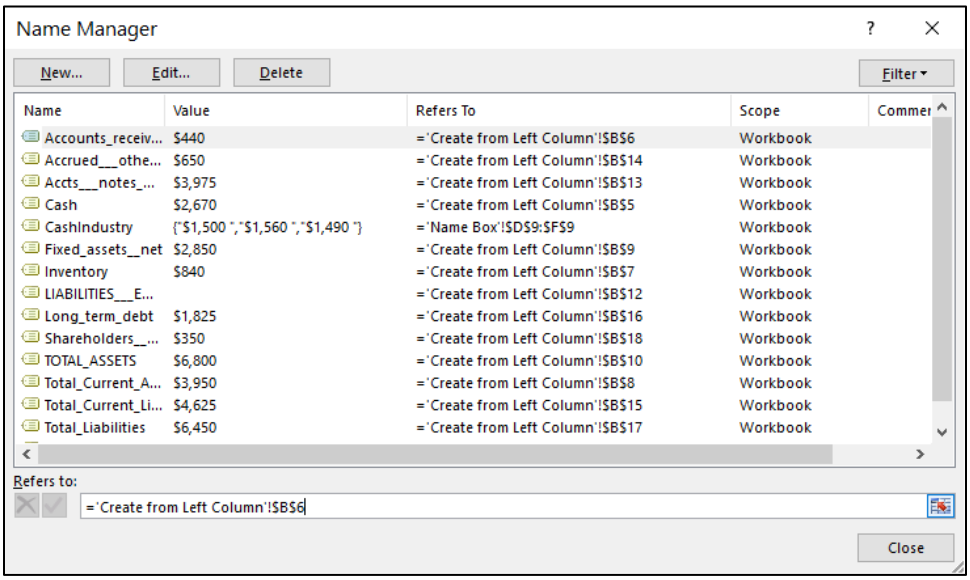


Figure 78 - Identifying all Defined Names in the Name Manager

Alternatively, you can create a "directory" of defined names on an Excel worksheet. To do so, click the arrow next to **Use in Formula** on the **Formulas** tab of the Ribbon and choose **Paste Names** from the resulting pop-up menu to open the **Paste Name** dialog box. Then, in the Paste Name dialog box, click **Paste List** to paste a list of defined names into the worksheet, as shown in **Figure 79**.

I	J
Accounts_receivable	='Create from Left Column'!$B$6
Accrued___other_payables	='Create from Left Column'!$B$14
Accts___notes_payable	='Create from Left Column'!$B$13
Cash	='Create from Left Column'!$B$5
Fixed_assets__net	='Create from Left Column'!$B$9
Inventory	='Create from Left Column'!$B$7
LIABILITIES___EQUITY	='Create from Left Column'!$B$12
Long_term_debt	='Create from Left Column'!$B$16
Shareholders__equity	='Create from Left Column'!$B$18
TOTAL_ASSETS	='Create from Left Column'!$B$10
Total_Current_Assets	='Create from Left Column'!$B$8
Total_Current_Liabilities	='Create from Left Column'!$B$15
Total_Liabilities	='Create from Left Column'!$B$17
TOTAL_LIABILITIES___EQUITY	='Create from Left Column'!$B$19

Figure 79 - Pasting a List of Defined Names to a Worksheet

"Retrofitting" a Worksheet with Defined Names

If you have not been working with defined names, you may now think using them is a good idea. However, you might be reticent to edit formulas in existing spreadsheets to replace cell references with defined names. If so, be aware that Excel provides an option you should consider. This "retrofit" option allows you to define names after you construct your formulas, and then have Excel incorporate them programmatically.

To retrofit a worksheet with defined names, open the worksheet and create the defined names you want to use to replace cell references. Next, select the cell(s) you wish to retrofit with newly-created defined names. Then click the arrow next to **Define Name** on the Ribbon's **Formulas** tab and select **Apply Names** to open the **Apply Names** dialog box shown in **Figure 80**.

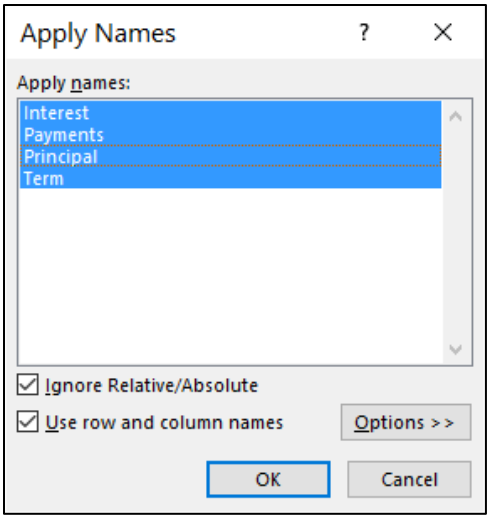


Figure 80 - Apply Names Dialog Box

Select the names you wish to substitute for cell references in the worksheet and click **OK** in the **Apply Names** dialog box. Upon doing so, Excel substitutes the selected defined names for cell references in the worksheet, effectively retrofitting the worksheet with the defined names, as shown in **Figure 81**.

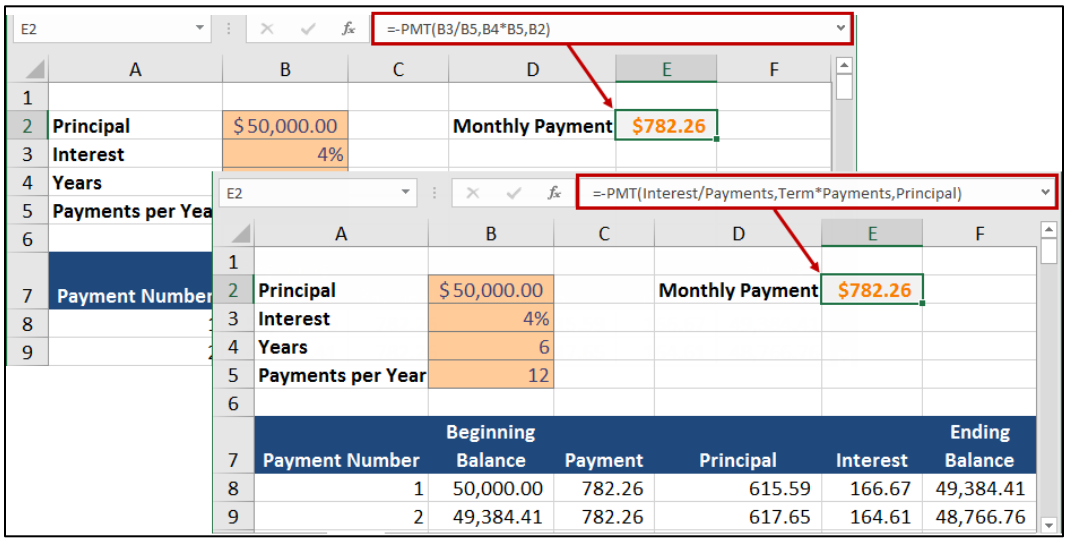


Figure 81 - Retrofitting Worksheets with Defined Names

Using Defined Names to Speed Navigation

One of the benefits of using defined names is their ability to speed navigation, particularly in large workbooks. To use defined names to speed navigation, select a defined name from the Name Box, and Excel will instantly "jump" to that defined name. Alternatively, you can use an **F5** keyboard shortcut to open Excel's **Go To** dialog box and select the defined name from the dialog box; again, Excel will instantly jump to the defined name you choose.

Storing Values as Defined Names

In addition to using defined names as aliases for cells and ranges of cells, you can also use them to store values such as sales tax rates, overhead allocation percentages, and payroll burden rates. Once you save these values as defined names, you can incorporate the stored values into cells and formulas by inserting the defined name into the cell or formula. Using defined names helps ensure that you consistently use correct values in your spreadsheet models.

To illustrate, suppose you routinely use the overhead allocation percentages shown in **Table 4** in spreadsheet calculations.

Department	Overhead Allocation Percentage
Accounting & Finance	18.29%
Executive	16.14%
Manufacturing	38.65%
Marketing	26.92%
Total	**100.00%**

Table 4 - Sample Overhead Allocation Percentages

You could opt to store each of these allocation percentages as a defined name. Then, you could insert the defined name whenever you need to use the percentage value. Store each percentage as a defined name by clicking **Define Name** on the Ribbon's **Formula** tab to open the **New Name** dialog box. Then, enter a label for the defined name in the **Name** box and a value in the **Refers to** box. Click **OK** to accept your entry. **Figure 82** illustrates this process.

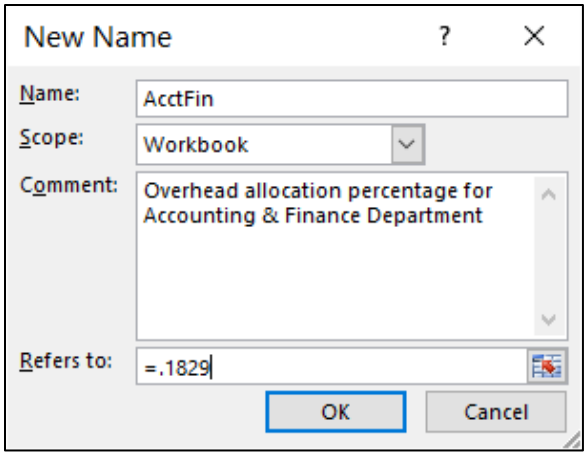


Figure 82 - Storing a Value as a Defined Name

After assigning each percentage a defined name, you may use that name anywhere in the workbook where you would otherwise have used the value. For instance, **Figure 83** illustrates using a value stored as a defined name in a formula.

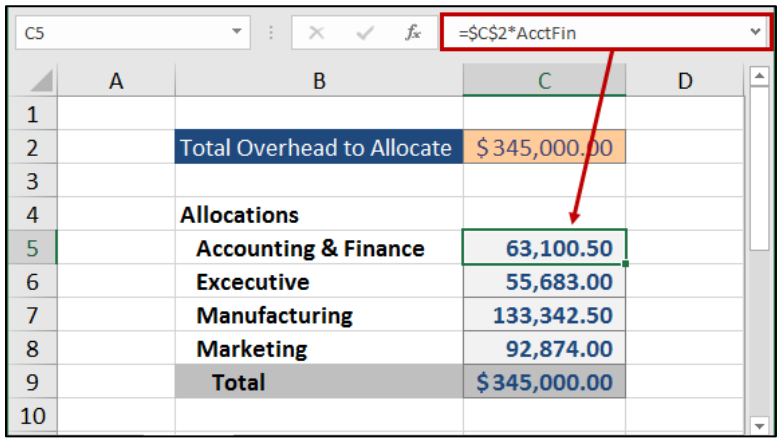


Figure 83 **- Using a Value Stored as a Defined Name in a Formula**

Performing Two-Way Lookups with Defined Names

You can also use defined names to simplify performing two-way lookups on a range of data. To illustrate, consider the simple range of sales data presented in **Figure 84**.

	A	B	C	D	E	F
1						
2			Qtr1	Qtr2	Qtr3	Qtr4
3		Anderson	471,146	293,522	352,034	445,888
4		Chen	373,424	408,363	413,088	275,348
5		Harris	458,256	358,555	316,537	357,319
6		Lopez	459,297	343,333	486,845	397,836
7		Smith	295,158	426,505	489,858	462,785

Figure 84 - Sample Data for Two-Way Lookup

In this sample data, nine defined names exist.

1. *Anderson* refers to cells C3 through F3.
2. *Chen* refers to cells C4 through F4.
3. *Harris* refers to cells C5 through F5.
4. *Lopez* refers to cells C6 through F6.
5. *Smith* refers to cells C7 through F7.
6. *Qtr1_* refers to cells C3 through C7.
7. *Qtr2_* refers to cells D3 through D7.
8. *Qtr3_* refers to cells E3 through E7.
9. *Qtr4_* refers to cells F3 through F7.

To extract total sales for salesperson Chen for the second quarter, you need only to enter the following formula in your spreadsheet:

=SUM(Chen Qtr2_)

Note that the key to this formula is a single space between the two defined names. Upon entering that formula, Excel extracts the value of 408,363, which is the value at the intersection of the *Chen* and *Qtr2_* defined names.

Maintaining the Integrity of Linked Workbooks with Defined Names

Another helpful feature of defined names is maintaining data integrity when linking data from other workbooks. For example, suppose you have linked data from a “source” workbook into a “destination” workbook. In that case, there is some likelihood that your links will become invalid. This problem arises when the source workbook's location changes while the destination workbook is closed. This problem often occurs when a user inserts or deletes rows or columns in the source workbook while the destination workbook is closed.

Use defined names instead of cell references to build the links between workbooks to ensure data integrity when linking from one workbook to another. For example, if you create links based on defined names, and someone changes the data cells' location in the source workbook, the defined names assigned to those cells “follow” those cells to their new location in the workbook.

Thus, when the destination workbook refreshes the links, it imports the correct data, even though that data's location has changed.

Storing Complex Formulas as Defined Names

Another advantage of using defined names is that you can store potentially complex formulas as defined names. Once stored, you can recall these formulas by referencing the defined name. To illustrate, suppose you need a formula to determine which day of the week the current date falls on. Unfortunately, Excel offers no direct method for creating this calculation. However, one option is to build a formula similar to the following.

=CHOOSE(WEEKDAY(TODAY()),"Sunday","Monday","Tuesday", "Wednesday","Thursday","Friday","Saturday")

The formula above combines three Excel functions – *Choose*, *Weekday*, and *Today* – to return the numerical value of the current date and convert it to a text label. However, building that formula whenever you want to insert it into a workbook is unattractive. Instead, you could create and store it as a defined name. Once stored, you only need to enter the defined name in the workbook to use the formula.

Storing the formula as a defined name is similar to saving a value as one. Begin by choosing **Define Name** from the **Formula** tab of the Ribbon. In the **New Name** box, enter the desired name for your formula, and then enter the formula, as shown in **Figure 85**. After clicking **OK**, insert the formula anywhere in the workbook by entering the "=DayOfWeek."

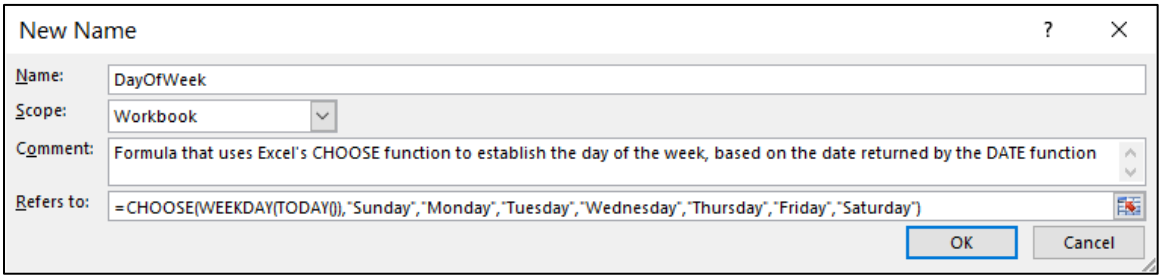


Figure 85 - Storing a Formula as a Defined Name

Using Defined Names and Conditional Formatting to Identify Formulas

Previously, we discussed identifying formulas in spreadsheets using several alternatives. Now that we have also discussed defined names, let us use them to help identify formulas.

Another method for identifying all formulas in a range or worksheet is to use a defined name to apply ***conditional formatting*** to the range or worksheet. As the name implies, conditional formats change as the conditions in a cell change. For instance, you can use conditional formatting to highlight all cells with unfavorable budget variances exceeding 15% or identify the

top ten selling items in an inventory list. You can also use conditional formatting to identify all cells that contain formulas by performing the following steps.

1. Create *CellHasFormula* as a defined name by choosing **Define Name** from the **Formulas** tab of the Ribbon.

2. In the **Refers to** box, enter **=GET.CELL(48,INDIRECT("rc",FALSE))** as shown in **Figure 86**. The **GET.CELL** function essentially instructs Excel to evaluate each cell in the selected range, and the value of **48** causes Excel to return "True" if the cell contains a formula. Lastly, the **INDIRECT** function references each cell in the selected range. Click **OK** to accept the entry.

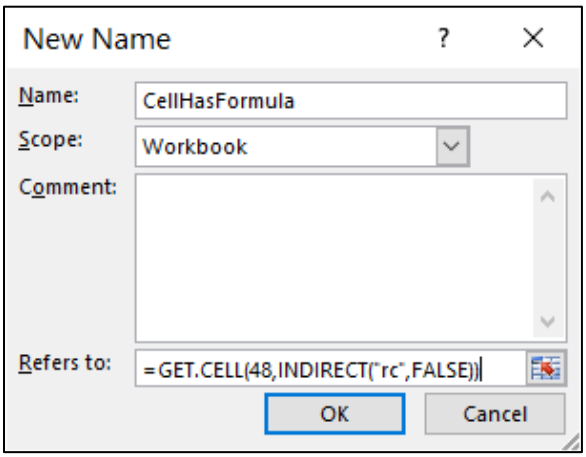


Figure 86 - Creating "CellHasFormula" Defined Name

3. Select the range of cells to which you want to apply conditional formatting.

4. From the Ribbon's **Home** tab, choose **Conditional Formatting**, **New Rule**, **Use a formula to determine which cells to format**.

5. In the **New Formatting Rule** dialog box, enter **=CellHasFormula**, as shown in **Figure 87**. Additionally, click the **Format** button and establish the formats you want to apply to the cells that contain formulas.

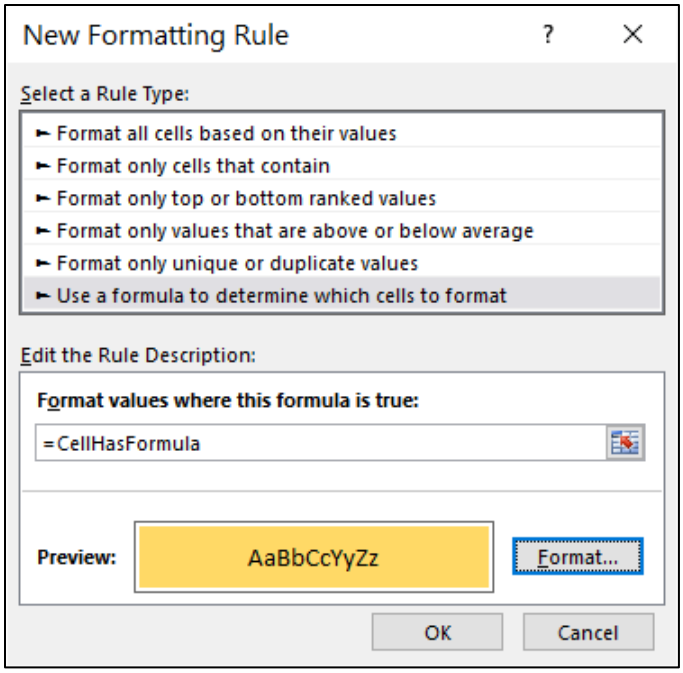


Figure 87 - New Formatting Rule Dialog Box

6. Click **OK** to complete the process and close the dialog box.

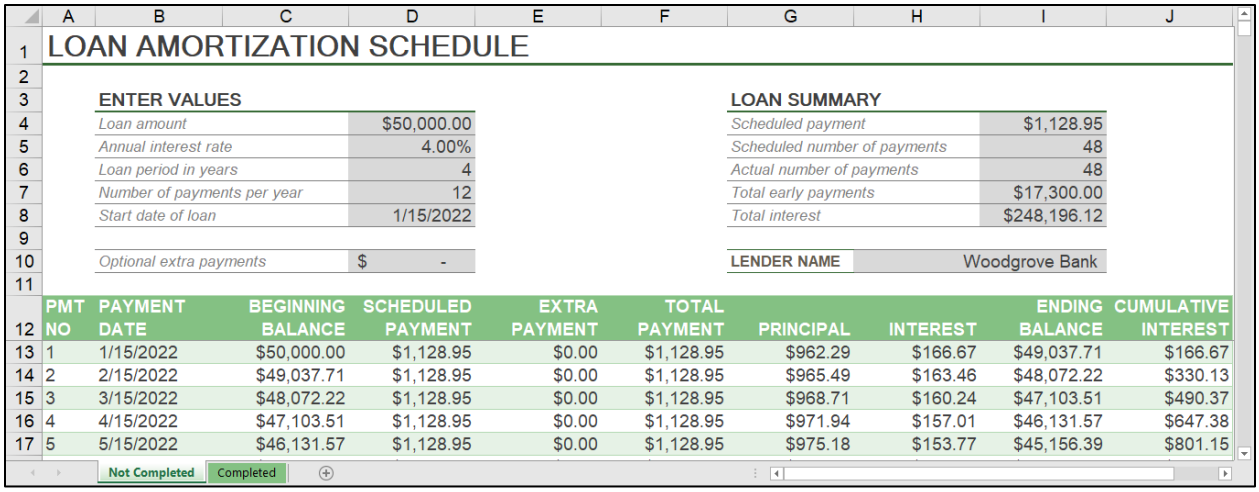
Having established the conditional format rule and selected the cells you want to apply it to, Excel evaluates each cell and applies the conditional format only in cases where the results are "true." An advantage to using this process as opposed to those described beginning on page 60 is that the results update dynamically as entries in the selected cells change by using conditional formats. In contrast, the methods discussed previously provide static results.

Minimizing Data Input Errors

In addition to following best practices for creating, auditing, verifying, and maintaining formulas, we should ensure that data entered into spreadsheets is accurate. The following techniques represent some best practices for minimizing data input errors.

Validating Data Input

Excel's **Data Validation** tool lets you specify allowable values for entry in your spreadsheet. Further, you can use Data Validation to speed up data entry for some users. To demonstrate the power of building Data Validation rules, consider the loan amortization worksheet pictured in **Figure 88**.



LOAN AMORTIZATION SCHEDULE

ENTER VALUES	
Loan amount	$50,000.00
Annual interest rate	4.00%
Loan period in years	4
Number of payments per year	12
Start date of loan	1/15/2022
Optional extra payments	$ -

LOAN SUMMARY	
Scheduled payment	$1,128.95
Scheduled number of payments	48
Actual number of payments	48
Total early payments	$17,300.00
Total interest	$248,196.12
LENDER NAME	Woodgrove Bank

PMT NO	PAYMENT DATE	BEGINNING BALANCE	SCHEDULED PAYMENT	EXTRA PAYMENT	TOTAL PAYMENT	PRINCIPAL	INTEREST	ENDING BALANCE	CUMULATIVE INTEREST
1	1/15/2022	$50,000.00	$1,128.95	$0.00	$1,128.95	$962.29	$166.67	$49,037.71	$166.67
2	2/15/2022	$49,037.71	$1,128.95	$0.00	$1,128.95	$965.49	$163.46	$48,072.22	$330.13
3	3/15/2022	$48,072.22	$1,128.95	$0.00	$1,128.95	$968.71	$160.24	$47,103.51	$490.37
4	4/15/2022	$47,103.51	$1,128.95	$0.00	$1,128.95	$971.94	$157.01	$46,131.57	$647.38
5	5/15/2022	$46,131.57	$1,128.95	$0.00	$1,128.95	$975.18	$153.77	$45,156.39	$801.15

Not Completed | Completed

Figure 88 - Sample Loan Amortization Schedule Used for Data Validation Rules

You would like to restrict a loan origination officer's opportunity to enter values when creating the amortization schedule in the loan amortization worksheet. More specifically, you would like to insert the following rules:

- The loan amount must be between $100,000 and $500,000,
- The interest rate must be between 2% and 6%, and
- The loan term must be 10, 15, 20, 25, or 30 years.

To ensure the worksheet meets the abovementioned rules, select cell **D4** and click **Data Validation** on the Ribbon's **Data** tab. This action opens the **Data Validation** rule dialog box pictured in **Figure 89**.

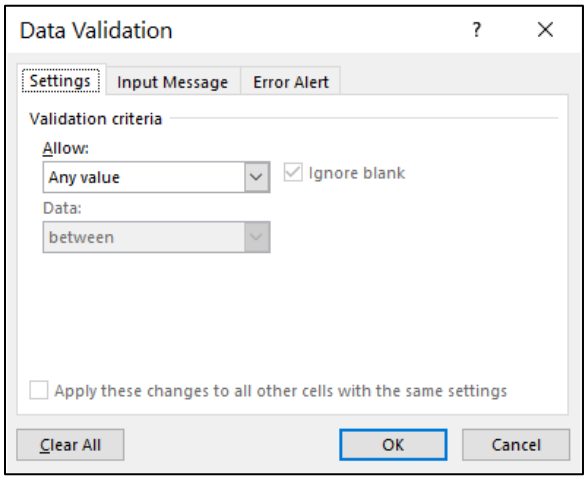


Figure 89 - Data Validation Rule Dialog Box

On the **Settings** tab, in the **Allow** box, choose **Whole number**; in the **Data** box, choose **between**; in the **Minimum** box, enter **100,000**; and in the **Maximum** box, enter **500,000**, as shown in **Figure 90**.

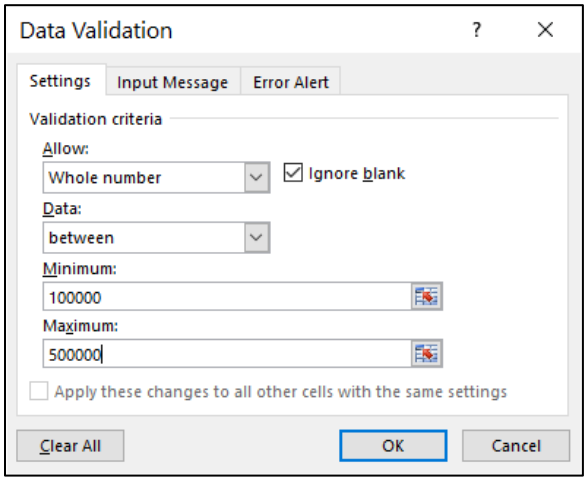


Figure 90 - Creating a Data Validation Rule for the Loan Amount Input

On the **Input Message** tab, enter any desired message to help guide users of the worksheet to enter data that falls into acceptable boundaries. For example, you might populate the fields for the Input Message, as shown in **Figure 91**.

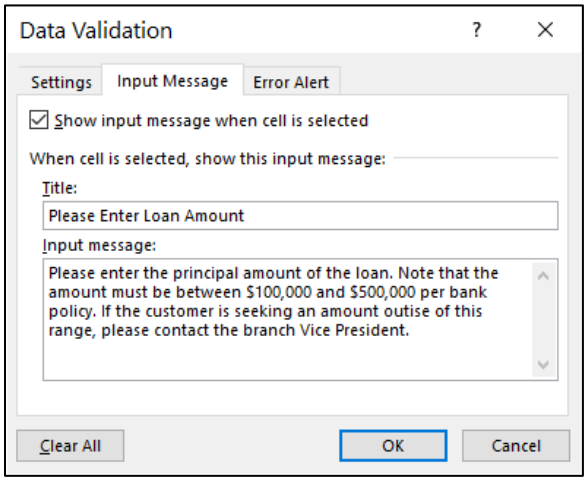


Figure 91 - Creating an Input Message for a Data Validation Rule

Finally, select the **Error Alert** tab to specify the message Excel displays when the user enters a value outside the established tolerances. When entering an Error Alert message, you have three options.

1. **Stop**. If you choose this option, the user cannot enter a value outside the established tolerances.

2. **Warning**. If you choose this option, the user can enter a value outside the established tolerances but must confirm their intention.

3. **Information**. If you choose this option, Excel will display a message indicating that the value is outside the established tolerances, but allows the user to accept the value without confirmation.

Choose the **Stop** option in the Style box for maximum control over data inputs, as shown in **Figure 92**.

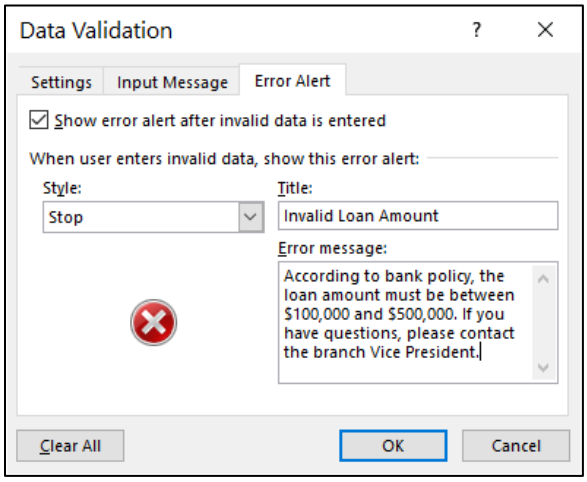


Figure 92 - Creating an Error Alert Message for a Data Validation Rule

Upon creating the Data Validation rule for the Loan Amount, Excel presents the dialog box shown in **Figure 93** when a user attempts to enter a value for the Loan Amount outside the established tolerances.

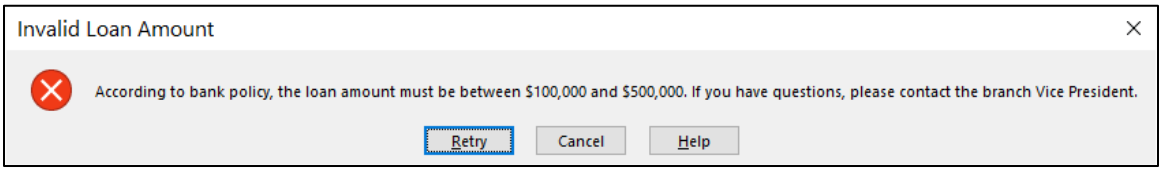


Figure 93 - Error Message Received upon Entering Invalid Loan Amount

The process for building a Data Validation rule to ensure that any interest rate a user enters into the amortization schedule is in the allowable range of 2% to 6% is virtually identical to that described for the Data Validation rule associated with the loan amount. The only substantive difference is selecting **Decimal** in the **Allow** box of the **Data Validation** dialog box, as shown in **Figure 94**.

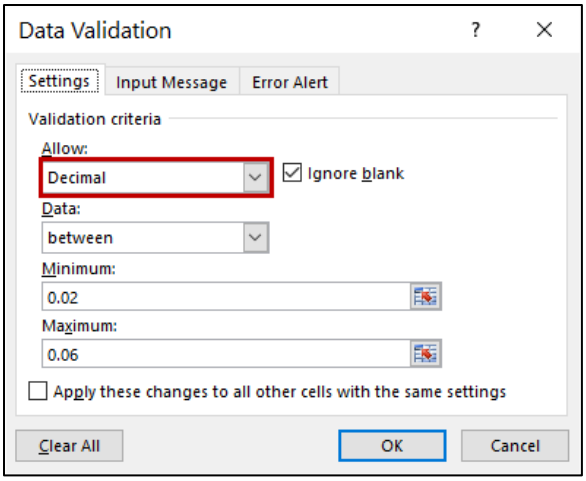


Figure 94 - Creating a Data Validation Rule Based on Percentages

In addition to using Data Validation rules to ensure user input accuracy, you can also build data validation rules to make it easier for users to enter data. For example, you can create a Data Validation rule to provide a dropdown list from which users can select a value to enter into the worksheet. The following five steps describe how to create a dropdown list that allows users to choose from the valid loan terms of 10, 15, 20, 25, and 30 years.

1. Select cell D6 and then click **Data Validation** from the Ribbon's **Data** tab to open the **Data Validation** dialog box.

2. On the **Settings** tab in the **Allow** box, choose **List**. Then, in the **Source** box, enter the allowable options of *10*, *15*, *20*, *25*, and *30*, as shown in **Figure 95**.

3. On the **Input Message** tab, enter an appropriate message to guide users through selecting a loan term.

4. Enter an appropriate message regarding potentially invalid data on the **Error Alert** tab.

5. Click **OK** to complete the process.

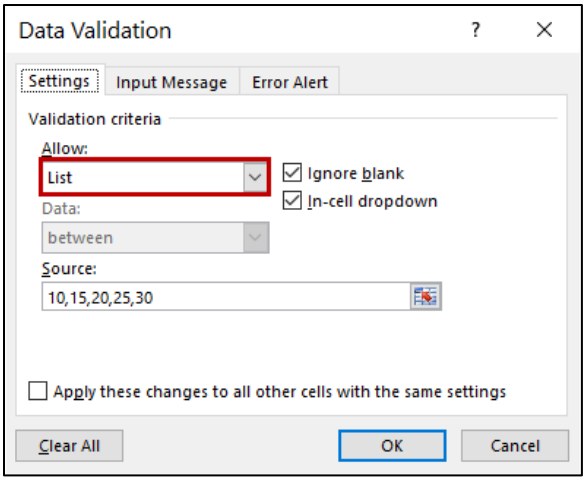


Figure 95 - Building a Dropdown List from a Data Validation Rule

Upon completing the process described above, when a user selects cell D6, Excel presents a drop-down list from which the user can select a valid loan term, as shown in **Figure 96**.

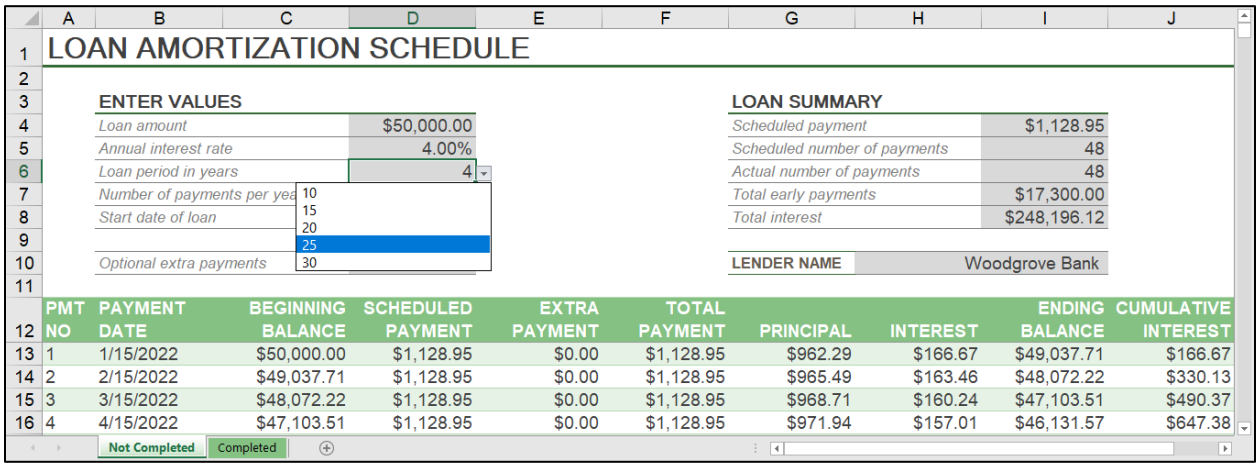


Figure 96 - Selecting Data from a Dropdown List Created from a Data Validation Rule

Highlighting Cells with Missing Data

To get a quick visual prompt that one or more input cells in your worksheet are missing data, you should consider applying conditional formatting to highlight all cells for which no data has been entered. As described below, this technique is relatively simple and consists of the following steps.

1. Begin by selecting the cells you want to apply the conditional format to.

2. Select **Conditional Formatting** from the **Home** tab of the Ribbon and choose **New Rule**, followed by **Use a formula to determine which cells to format**.

3. Enter the rule shown in **Figure 97** in the **New Formatting Rule** dialog box, substituting the first cell of your selected range for B7.

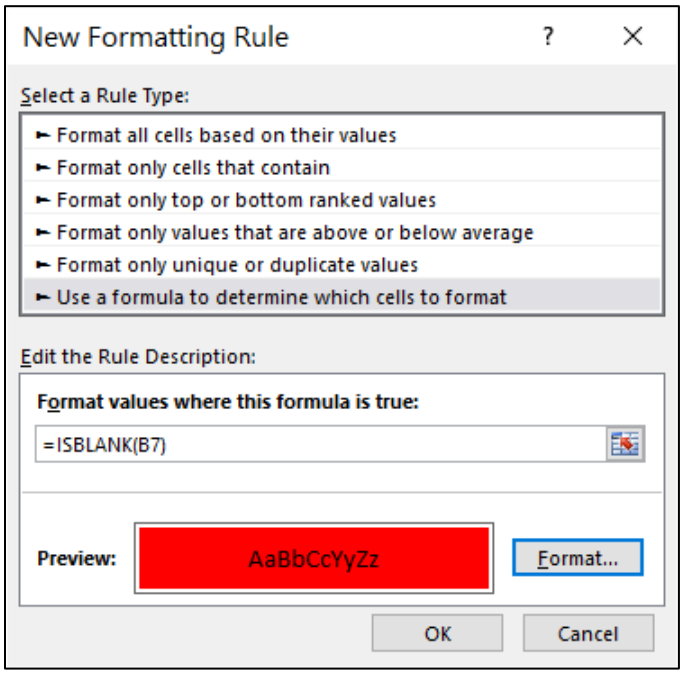


Figure 97 - Conditional Formatting Rule Testing for Blank Cell

4. Click **Format** in the **New Formatting Rule** dialog box and establish the formats you want Excel to apply to the selected range's blank cells.

5. Click **OK** to complete the process.

Introducing Tables

With the 2007 release, Excel added the ***Table*** feature. Tables contain data in worksheet rows and columns, which are converted to a Table object. You can convert a range of data to a table using any of the four techniques outlined below.

1. Click the range you wish to convert to a table and choose **Format as Table** from the Ribbon's **Home** tab.

2. Click the range you wish to convert to a table and choose **Table** from the Ribbon's **Insert** tab.

3. Click in the range you wish to convert to a table and use either the **CTRL+T** keyboard shortcut or the **CTRL+L** keyboard shortcut.

4. From the **Data** tab of the Ribbon, build a data connection to an external data source and, when prompted, choose to import the data as a Table.

Though a complete discussion of Tables is beyond this course's scope, numerous advantages are associated with converting traditional data ranges into tables. Using tables provides benefits that include *auto-expansion*, *formula replication*, and *structured references*.

Using Tables to Store Data

Tables serve as dynamically expanding and contracting ranges of data. Once you convert a range of data to a table, Excel automatically updates the table's definition if the number of rows or columns of data in the table changes. Upon doing so, all formulas that use the table as their data source are automatically updated, eliminating the need to manually edit them.

To illustrate, consider the table shown in **Figure 98**. More specifically, focus on the results of the formula in cell E434.

	A	B	C	D	E
1					
2	Date	Check Number	Vendor	Account	Amount
3	Jan-22	2154	Washington Properties, LLC	Rent	$ 2,400.00
4	Jan-22	2155	Adams & Jefferson, CPAs	Payroll Expense	$ 7,423.68
429	Dec-22	2580	Farmer's Produce	Food	$ 1,018.10
430	Dec-22	2581	Farmer's Produce	Food	$ 809.82
431	Dec-22	2582	Farmer's Produce	Food	$ 1,394.15
432	Dec-22	2583	Fresh Meats, Inc.	Food	$ 976.07
433					
434					**$778,025.19**

Figure 98 - Sample Table Data

Now, focus on the table shown in **Figure 99**. It includes an additional row of data, and the formula in cell E434 includes that row in the summation, with no manual effort required by the user. In other words, as the table expands (or contracts), every object and formula that uses the table as its data source updates to reflect the new volume of data in the table.

	A	B	C	D	E
1					
2	Date	Check Number	Vendor	Account	Amount
3	Jan-22	2154	Washington Properties, LLC	Rent	$ 2,400.00
4	Jan-22	2155	Adams & Jefferson, CPAs	Payroll Expense	$ 7,423.68
429	Dec-22	2580	Farmer's Produce	Food	$ 1,018.10
430	Dec-22	2581	Farmer's Produce	Food	$ 809.82
431	Dec-22	2582	Farmer's Produce	Food	$ 1,394.15
432	Dec-22	2583	Fresh Meats, Inc.	Food	$ 976.07
433	Dec-22	2584	RSI, Inc.	Supplies	$ 875.36
434					$778,900.55

Figure 99 - Table Expanded to Accommodate Extra Row

Using tables as a data source for your formulas and other objects is a best practice because using them eliminates the need to edit these formulas and objects when the volume of data changes. This feature helps ensure spreadsheet accuracy and reduces the time required to build and maintain the model.

Building Structured Referencing Formulas

Tables offer an alternative approach to constructing formulas known as *structured referencing*. For example, the formula in cell E434 in **Figure 100** is a structured reference formula. Notice that it does not refer to a range of cells. Instead, it refers to the table – *Table1* – and a specific column in the table – *Amount*.

Although you can continue to use traditional row and column addresses if you prefer, structured referencing is a best-practice approach because it facilitates automatic formula result updates as the table size changes. Building structured reference formulas also helps generate self-documenting formulas, such as those created with defined names.

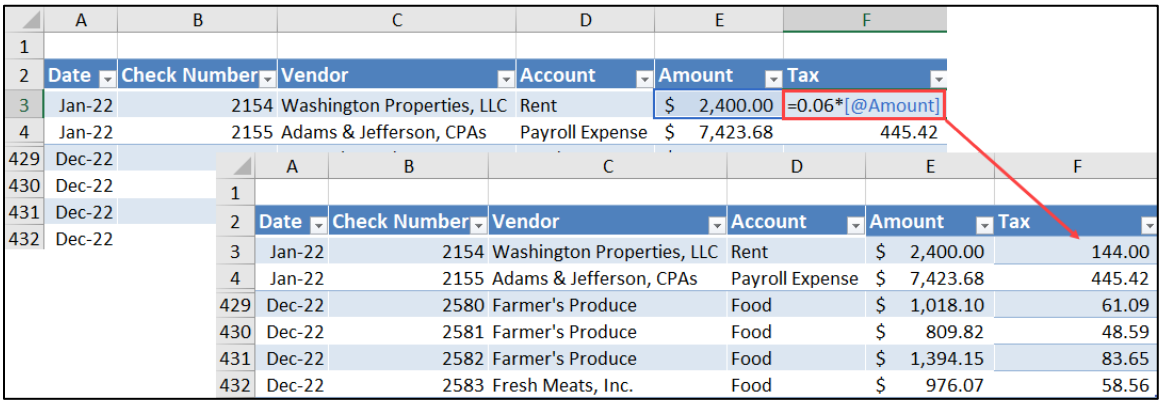
	A	B	C	D	E
1					
2	Date	Check Number	Vendor	Account	Amount
3	Jan-22	2154	Washington Properties, LLC	Rent	$ 2,400.00
4	Jan-22	2155	Adams & Jefferson, CPAs	Payroll Expense	$ 7,423.68
429	Dec-22	2580	Farmer's Produce	Food	$ 1,018.10
430	Dec-22	2581	Farmer's Produce	Food	$ 809.82
431	Dec-22	25		ood	$ 1,394.15
432	Dec-22	25	=SUM(Table1[Amount])	ood	$ 976.07
433	Dec-22	25		upplies	$ 875.36
434					$778,900.55

Figure 100 - Structured Reference Formula

Automating Calculations with Formula Replication

Formula replication, another feature of using tables, improves accuracy. With formula replication, whenever you enter a formula in a table column, Excel automatically copies that formula throughout the entire column. This action eliminates the need to manually copy and paste the formula.

For example, if you add a **Tax** column to the table pictured in Figure 100 and insert the formula **=.06*[@Amount]** to any row, Excel automatically copies that formula throughout the entire column, as shown in **Figure 101**.



	A	B	C	D	E	F
1						
2	Date	Check Number	Vendor	Account	Amount	Tax
3	Jan-22	2154	Washington Properties, LLC	Rent	$ 2,400.00	=0.06*[@Amount]
4	Jan-22	2155	Adams & Jefferson, CPAs	Payroll Expense	$ 7,423.68	445.42
429	Dec-22					
430	Dec-22					
431	Dec-22					
432	Dec-22					

	A	B	C	D	E	F
1						
2	Date	Check Number	Vendor	Account	Amount	Tax
3	Jan-22	2154	Washington Properties, LLC	Rent	$ 2,400.00	144.00
4	Jan-22	2155	Adams & Jefferson, CPAs	Payroll Expense	$ 7,423.68	445.42
429	Dec-22	2580	Farmer's Produce	Food	$ 1,018.10	61.09
430	Dec-22	2581	Farmer's Produce	Food	$ 809.82	48.59
431	Dec-22	2582	Farmer's Produce	Food	$ 1,394.15	83.65
432	Dec-22	2583	Fresh Meats, Inc.	Food	$ 976.07	58.56

Figure 101 - Formula Replication within a Table

Formula replication is a significant benefit associated with using tables, particularly those with changing volumes of data, because it means you no longer need to be concerned with ensuring that all table rows have the correct formula. Accordingly, formula replication is a significant advantage of tables and enhances their use as an Excel best practice.

? What are the significant advantages of converting ranges of data into Tables? How can you use Tables to help automate summarizations of data in Excel?

Understanding SUBTOTAL vs. SUM

Perhaps the most commonly used function in Excel is the **SUM** function. With SUM, you can quickly and easily add a range of data. However, Excel's **SUBTOTAL** function is sometimes preferable and is often an Excel best practice. For example, when working with data ranges that exhibit the following three characteristics, you should probably use SUBTOTAL instead of SUM to add your data.

- If you perform intermediate summations on your data in the same column, SUBTOTAL is a better tool because it decreases the risk you might double-count data.

- SUBTOTAL is the preferred function if your data contains hidden rows because it provides flexibility regarding summing hidden rows.

- If you need to filter your data, use SUBTOTAL to ensure that the summation includes only non-filtered rows.

Examples of each of these scenarios follow.

SUBTOTAL vs. SUM on Intermediate Calculations

Consider the data presented in **Figure 102** and notice that Total Assets' values differ for columns D and E.

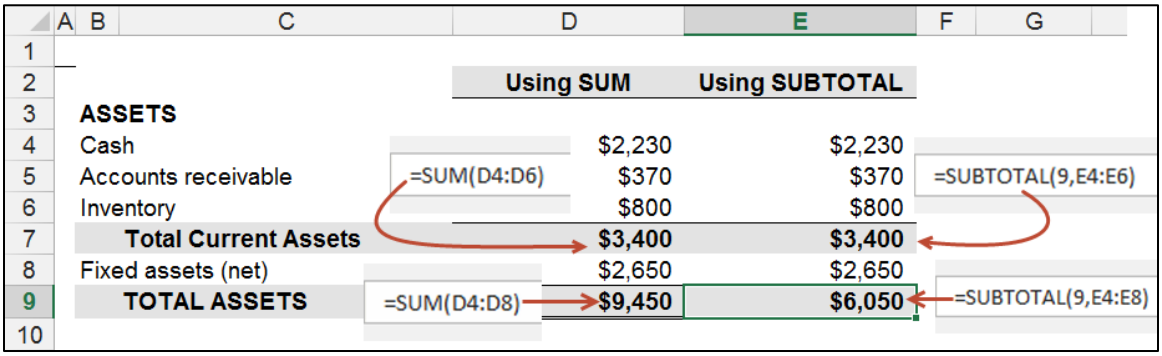


Figure 102 - SUBTOTAL vs. SUM on Intermediate Calculations

The difference in *Total Assets* in Figure 102 is attributable to using the SUM function in column D and the SUBTOTAL function in column E.

A SUM function totals all cells in the specified range. Therefore, in cell D9, the SUM function includes all the "raw" data in cells D4 through D8, as well as the previous summation of *Total Current Assets* in cell D7. In other words, the value for Total Current Assets is "double-counted" in the summation of *Total Assets*.

In column E, the SUBTOTAL functions, by definition, exclude any cells in the range that contain SUBTOTAL functions. Therefore, although the SUBTOTAL function in cell E9 refers to the cells E4 through E8, it ignores the value generated by the previous SUBTOTAL function in cell E7. Accordingly, using SUBTOTAL functions instead of SUM functions reduces the likelihood of including values from intermediate calculations.

You might be puzzled about the syntax if you have not used the SUBTOTAL function before. The correct syntax for a formula containing SUBTOTAL is:

=SUBTOTAL(function_num, ref1,[ref2],…)

where

- *Function_num* is a required value ranging from 1 to 11 or 101 to 111, and
- *Ref1, Ref2, etc.,* refer to the cells or ranges of cells you want to include in the calculation. You must enter at least one reference and may enter as many as 255 references.

The *function_num* argument instructs SUBTOTAL about the type of calculation you would like to perform. More specifically, SUBTOTAL can perform eleven different calculations, as listed in **Table 5**.

FUNCTION_NUM (INCLUDES HIDDEN VALUES)	FUNCTION_NUM (IGNORES HIDDEN VALUES)	FUNCTION
1	101	AVERAGE
2	102	COUNT
3	103	COUNTA
4	104	MAX
5	105	MIN
6	106	PRODUCT
7	107	STDEV
8	108	STDEVP
9	109	SUM
10	110	VAR
11	111	VARP

Table 5 - Function_Num Arguments Associated with the SUBTOTAL Function

Based on the function_num arguments in **Table 5**, if you want SUBTOTAL to perform a summation, you insert argument "9" or "109" into the formula; likewise, if you wish SUBTOTAL to calculate an average, you insert argument "1" or "101."

SUBTOTAL vs. SUM on Hidden Rows

The primary difference between the two series of function_num arguments associated with SUBTOTAL is how they operate on values in hidden rows in the references on which SUBTOTAL is acting. To illustrate, consider the example shown in **Figure 103**; specifically, notice that a user has hidden row 8 from the worksheet.

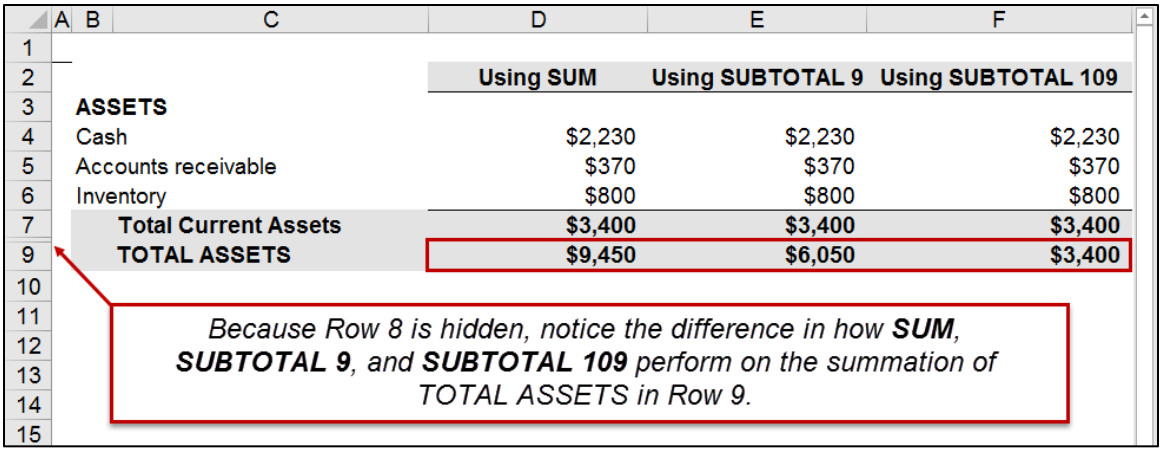


Figure 103 - SUM vs. SUBTOTAL 9 vs. SUBTOTAL 109

The formula in cell F9 in Figure 103 is **=SUBTOTAL(109, F4:F8)**. Using the "109" function_num argument causes the SUBTOTAL function to ignore values in hidden rows.

SUBTOTAL vs. SUM on Filtered Data

Another prime example of when you should probably use SUBTOTAL instead of SUM is when you or another user filters the data used in a formula. Excel continues to include the rows that do not meet the filter criteria if you use a SUM function on filtered data. On the other hand, SUBTOTAL excludes the values from the filtered rows when calculating.

Figure 104 illustrates the differences in calculations on filtered data when using SUM and SUBTOTAL. The three formulas in cells E433 through E435 are as follows:

- E433: *=SUM(E2:E432)*,
- E434: *=SUBTOTAL(9,E2:E432)*, and
- E435: *=SUBTOTAL(109,E2:E432)*.

Despite the similarities among these three formulas, SUM and SUBTOTAL yield dramatically different results when applied to filtered data, indicating that using SUBTOTAL is an Excel best practice.

	A	B	C	D	E	F
1	Date	Check Number	Vendor	Account	Amount	
427	Dec-18	2579	RSI, Inc.	Supplies	599.85	
428	Dec-18	2580	Farmer's Produce	Food	1,018.10	
429	Dec-18	2581	Farmer's Produce	Food	809.82	
430	Dec-18	2582	Farmer's Produce	Food	1,394.15	
431	Dec-18	2583	Fresh Meats, Inc.	Food	976.07	
432						
433					$ 778,025.19	<<< SUM Function
434					$ 778,025.19	<<< SUBTOTAL 9 Function
435					$ 778,025.19	<<< SUBTOTAL 109 Function

Figure 104 - SUM and SUBTOTAL Functions on Filtered Data

Securing Excel Data And Workbooks

Chapter 4

Once you build an Excel workbook, you should take steps to protect that workbook's data, formulas, and structure. Securing your spreadsheets is necessary to prevent unauthorized changes, whether accidental or intentional, and to keep confidential information protected. This chapter discusses best practice techniques for securing your Excel spreadsheets.

Learning Objectives

Upon completing this chapter, you should be able to:

- Manage Excel's security settings in the Trust Center;
- List and describe six methods for securing Excel workbooks; and
- Apply digital signatures to workbooks.

Securing Excel in the Trust Center

With the Excel 2007 release, Microsoft centralized all the security and privacy settings for Excel workbooks in the **Trust Center**. From the Trust Center, you can 1) activate or deactivate ActiveX controls, 2) manage add-ins, 3) enable or disable macros, 4) turn security alerts on the Message Bar on or off, 5) remove or view trusted publishers, 6) block or unblock external content, 7) create, remove, or change trusted file locations, and manage privacy options. Excel's default settings will likely suit most users' security and privacy needs.

The Trust Center is accessible from the **File** tab of the Ribbon by selecting **Options, Trust Center,** and **Trust Center Settings**. For example, **Figure 105** illustrates the **Trust Center** for **Macro Settings**.

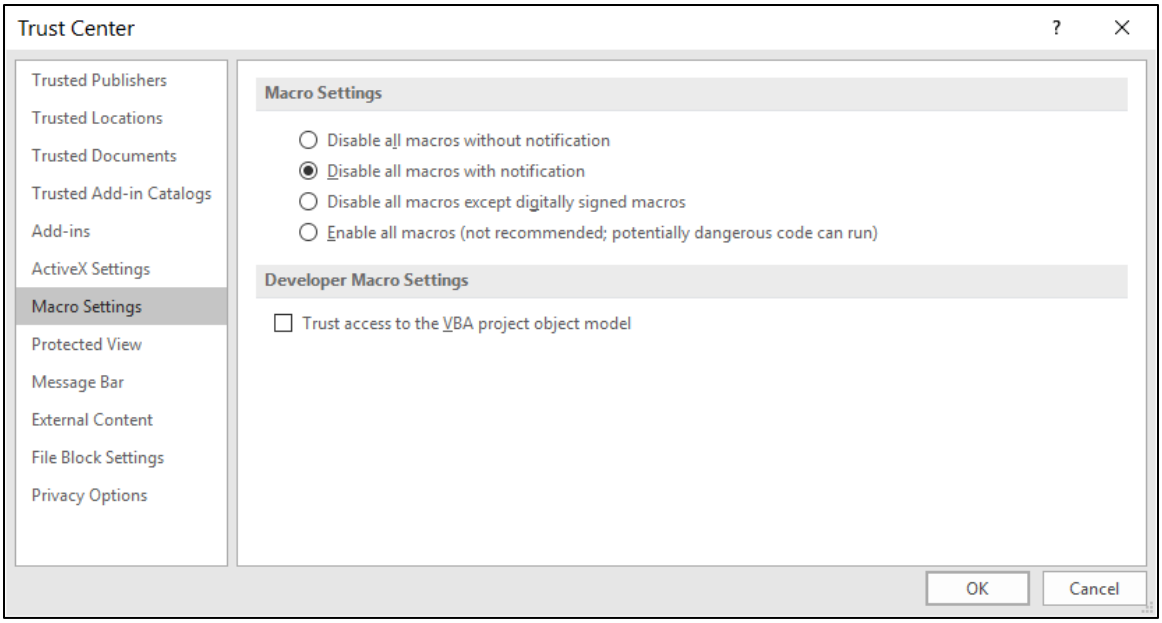


Figure 105 - Accessing the Trust Center to Modify Security and Privacy Settings

Note that by default, Excel disables macros *with notification*. However, whenever you open a workbook that contains a macro, you can choose whether to enable the macro, as shown in **Figure 106**.

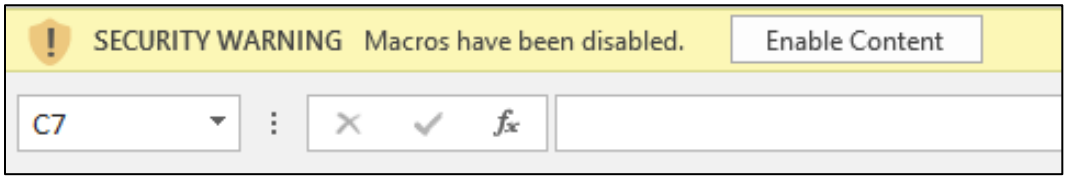


Figure 106 - Enabling Macros in a Workbook

Securing Documents

All capabilities for directly securing workbooks are accessible from the **File** tab of the Ribbon. First, click the **File** tab, then select **Info**. Upon doing so, you will see five buttons: 1) **Protect Workbook**, 2) **Inspect Workbook**, 3) **Version History**, 4) **Manage Workbook**, and 5) **Browser View Options**. For example, after clicking the **Protect Workbook** button, you may 1) force a workbook to always open in read-only mode, 2) encrypt a workbook with a password, 3) protect the current worksheet, 4) protect the workbook for structure, 5) restrict access using **Information Rights Management** (IRM), 6) add a digital signature to a workbook, and 7) mark a workbook as final.

Clicking the **Check for Issues** button enables you to 1) inspect the workbook's properties, 2) check the workbook's accessibility for those who might find it challenging to read, and 3) check compatibility with prior versions of Excel. Clicking the **Version History** button allows you to

manage previous workbook versions. Choosing **Manage Workbook** lets you recover and manage unsaved workbooks. Finally, selecting **Browser View Options** lets you pick what others can see when viewing the workbook on the Web.

Excel provides several levels of protection for workbooks and workbook content. From an accountant's perspective, these protection options are essential because they provide the tools to prevent unauthorized changes to financial statements, transaction journals, and other significant documents. They also restrict unauthorized access to these documents. The following list summarizes some of the levels of protection offered by Excel.

1. **Always open Read-Only** – This setting forces Excel to open the workbook in read-only mode. In this mode, you must opt in to editing.

2. **Encrypted workbook with password protection** – Excel encrypts the workbook with 128-bit AES encryption, and Excel requires a password to open the workbook.

3. **Restrict Access Using Information Rights Management (IRM) protection** – Excel encrypts the workbook with 128-bit AES encryption. If you use this feature, you must authenticate to a Rights Management Server to use IRM to open, modify, copy, print, or distribute the workbook.

4. **Unencrypted workbook with modification protection** – The workbook is not encrypted, and anyone may open it. However, Excel protects the workbook from modification unless you enter the password to modify it when you open the workbook. The workbook opens in read-only mode if you do not enter the password.

5. **Workbook structure protection** – The workbook is unencrypted; anyone may open it. However, Excel protects some elements of the workbook from modification. You may enable workbook protection with or without a password.

6. **User-specified worksheet element protection** – The workbook is not encrypted; anyone may open it. However, Excel protects user-specified worksheet elements from modification on a worksheet-by-worksheet basis. You may enable worksheet protection with or without a password.

You can use the workbook and worksheet protection options in tandem. For example, to protect a workbook from unauthorized access and unwanted modification, you may secure specific worksheet elements and then encrypt the workbook with a password.

Encrypting Documents

Excel allows you to *encrypt* and password-protect workbooks. Click **File, Info, Protect Workbook,** and select **Encrypt with Password**, as shown in **Figure 107**.

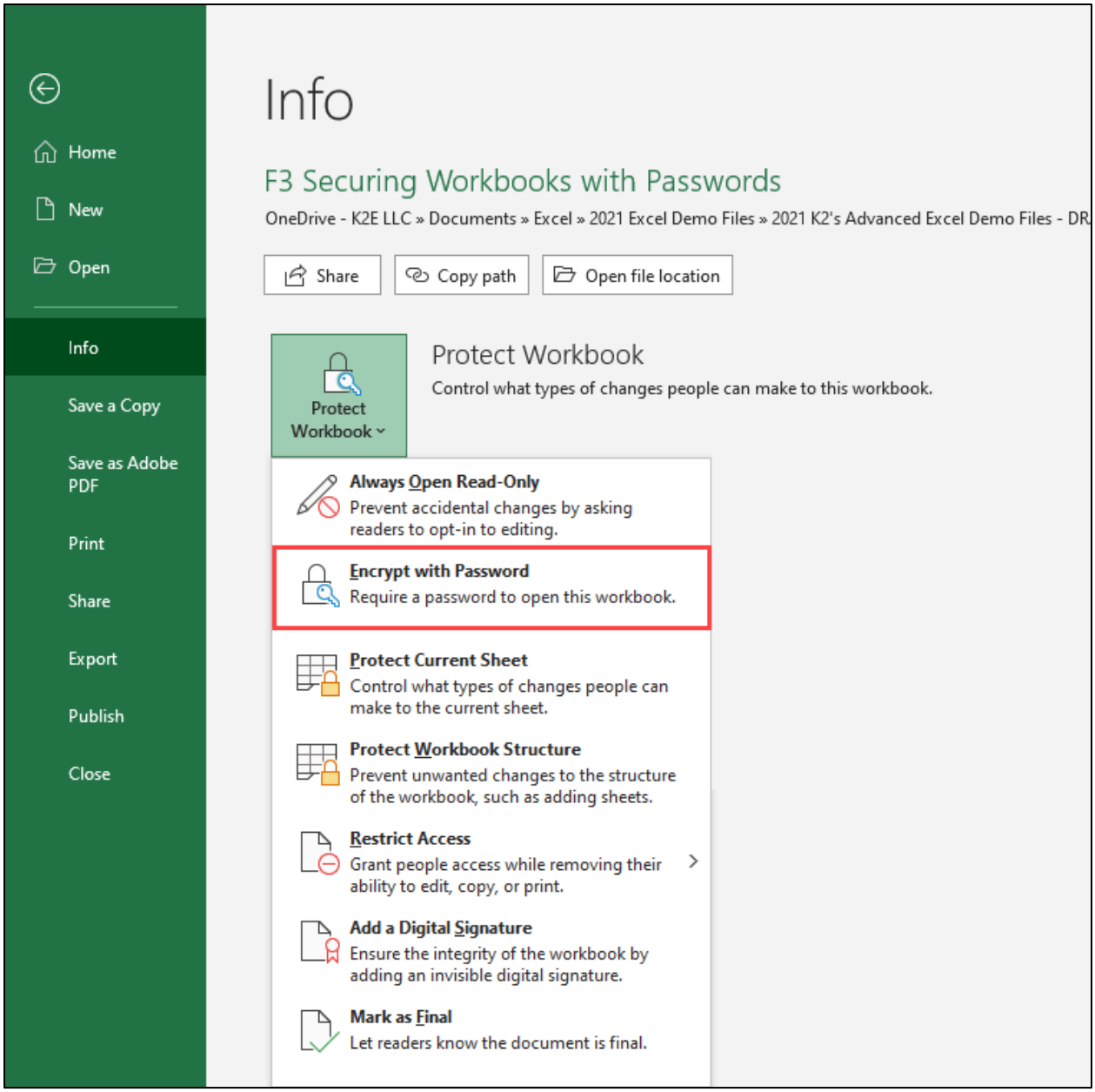


Figure 107 - Encrypting a Workbook

Type a strong password in the **Encrypt Document** dialog box and click **OK**. Confirm the password and click **OK** again to complete the process. By default, Excel encrypts workbooks using 128-bit AES encryption. Therefore, these passwords are not likely to be "cracked." Of course, you should store the password in a secure place, such as a password manager.

Protecting A Worksheet

You protect entire workbooks or individual worksheets within a workbook at your discretion. To secure individual sheets, ***unlock*** any cells or objects that users can change. Select the cells to

unlock, right-click, and choose **Format Cells** on the context-sensitive menu. On the **Protection** tab of the **Format Cells** dialog box, uncheck the **Locked** option. Then, select **Protect Sheet** on the Ribbon's **Review** tab to open the **Protect Sheet** dialog box, as shown in **Figure 108**. In the **Allow all users of this worksheet to** list, select the objects upon which users can edit. Next, enter a password and click **OK**. Confirm the password and then click **OK** again to protect the sheet. The password prevents users who do not know it from unprotecting the worksheet. Repeat these steps for every worksheet that you must protect.

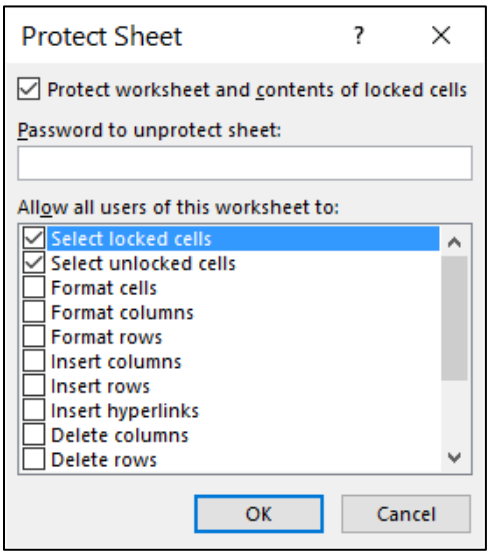


Figure 108 - Protecting an Individual Worksheet

Select which actions you will allow on the current worksheet in the Protect Sheet dialog box. **Table 6**, adapted from Excel's Help system, describes all the options in the Protect Sheet dialog box.

CLEAR THIS CHECK BOX	TO PREVENT USERS FROM
Select locked cells	Moving the pointer to cells for which the **Locked** check box is selected on the **Protection** tab of the **Format Cells** dialog box. By default, users can select locked cells.
Select unlocked cells	Moving the pointer to cells for which the **Locked** check box is cleared on the **Protection** tab of the **Format Cells** dialog box. Users can select unlocked cells and press the TAB key to move between them on a protected worksheet.
Format cells	Changing any of the options in the **Format Cells** or **Conditional Formatting** dialog boxes. Suppose you applied conditional formats before you protected the worksheet. If so, the formatting changes when a user enters a value that satisfies a different condition.

CLEAR THIS CHECK BOX	TO PREVENT USERS FROM
Format columns	Using any column formatting commands, including changing column width or hiding columns (**Home** tab, **Cells** group, **Format** button).
Format rows	Using any row formatting commands, including changing row height or hiding rows (**Home** tab, **Cells** group, **Format** button).
Insert columns	Inserting columns.
Insert rows	Inserting rows.
Insert hyperlinks	Inserting new hyperlinks, even in unlocked cells.
Delete columns	Deleting columns. If **Delete columns** is protected and **Insert columns** is not, a user can insert columns that they cannot delete.
Delete rows	Deleting rows. If **Delete rows** is protected and **Insert rows** is not, a user can insert rows that they cannot delete.
Sort	Using any commands to sort data (**Data** tab, **Sort & Filter** group). Users cannot sort ranges that contain locked cells on a protected worksheet, regardless of this setting.
Use AutoFilter	Using the drop-down arrows to change the filter on ranges when AutoFilters are applied. Regardless of this setting, users cannot apply or remove AutoFilters on a protected worksheet.
Use PivotTable reports	Formatting, changing the layout, refreshing, modifying PivotTable reports, or creating new reports.
Edit objects	Doing any of the following: • Making changes to graphic objects, including maps, embedded charts, shapes, text boxes, and controls that you did not unlock before protecting the worksheet. • Making any changes, such as formatting, to an embedded chart. The chart continues to be updated when you change its source data. • Adding or editing comments.

CLEAR THIS CHECK BOX	TO PREVENT USERS FROM
Edit scenarios	Viewing scenarios that you have hidden, making changes to scenarios that you have prevented changes to, and deleting these scenarios. Users can change the values in unprotected cells and add new scenarios.

Table 6 - Protect Sheet Options

After selecting the options you want to allow on the current worksheet, enter a password in the Password to unprotect sheet box, then click **OK** to begin protecting the worksheet. Choose **Unprotect Sheet** from the Ribbon's **Formulas** tab to remove the protection settings if necessary. Then, enter the worksheet's password when prompted.

Limiting Access To Specific Ranges In A Worksheet

Consider **Excel's Allow Users to Edit Ranges** feature if you want a more detailed, customizable set of protection options. With this feature, you can select ranges in a worksheet and grant specific users the right to edit them. Your computer must run Windows and be connected to a domain to use this technique without relying on additional passwords. Otherwise, you can use passwords to enable this feature.

To enable this feature with passwords, follow these six steps.

1. Begin by selecting the worksheet you want to protect with this technique.

2. Select the cell(s) you want to allow another user to edit. Ensure that the cell(s) are locked.

3. On the Ribbon's **Review** tab, select **Allow Users to Edit Ranges** to open the **Allow Users to Edit Ranges** dialog box, pictured in **Figure 109**.

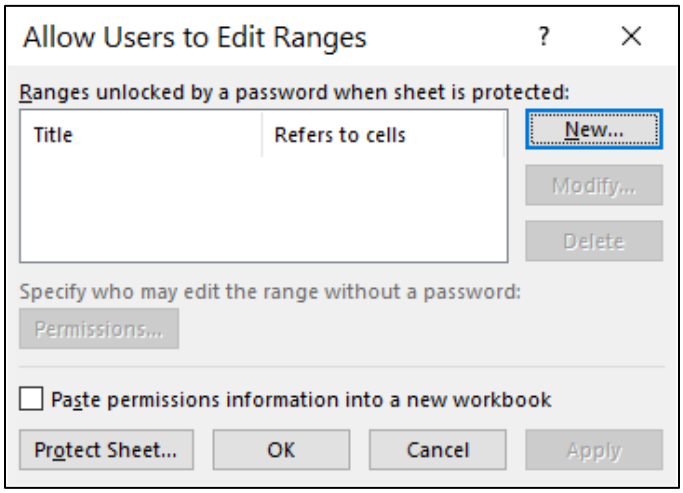


Figure 109 - Allow Users to Edit Ranges Dialog Box

4. In the **Allow Users to Edit Ranges** dialog box, choose **New** to open the **New Range** dialog box. In the New Range dialog box, enter a name in the **Title** box, the range of cells on which you will allow edits in the **Refers to cells** box, and a password in the **Range password** box, as shown in **Figure 110**.

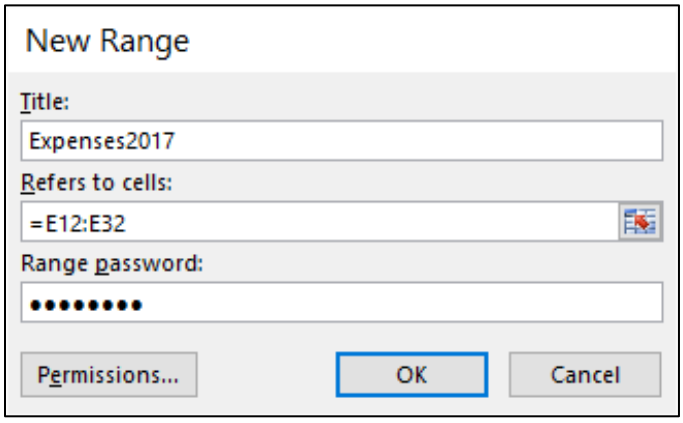


Figure 110 - Adding a Range for Editing

5. Click **OK** to accept your changes and return to the Allow Users to Edit Ranges dialog box.

6. In the Allow Users to Edit Ranges dialog box, click **Protect Sheet** to open the **Protect Sheet dialog box** shown in Figure 108. Select the elements you want to make available for edits in the **Protect Sheet** dialog box. Then, add an optional password and click **OK** to save your changes.

After completing the six steps outlined above, any user who attempts to perform any action on the protected range must first enter the correct password to validate to Excel that they have the right to edit that range. Otherwise, Excel blocks cell editing.

If you connect your computer to a domain, you do not need to use passwords to take advantage of this feature. Beginning with step 4 above, you can click the **Permissions** box instead of adding passwords and choose which users can edit the range without a password, as shown in **Figure 111**.

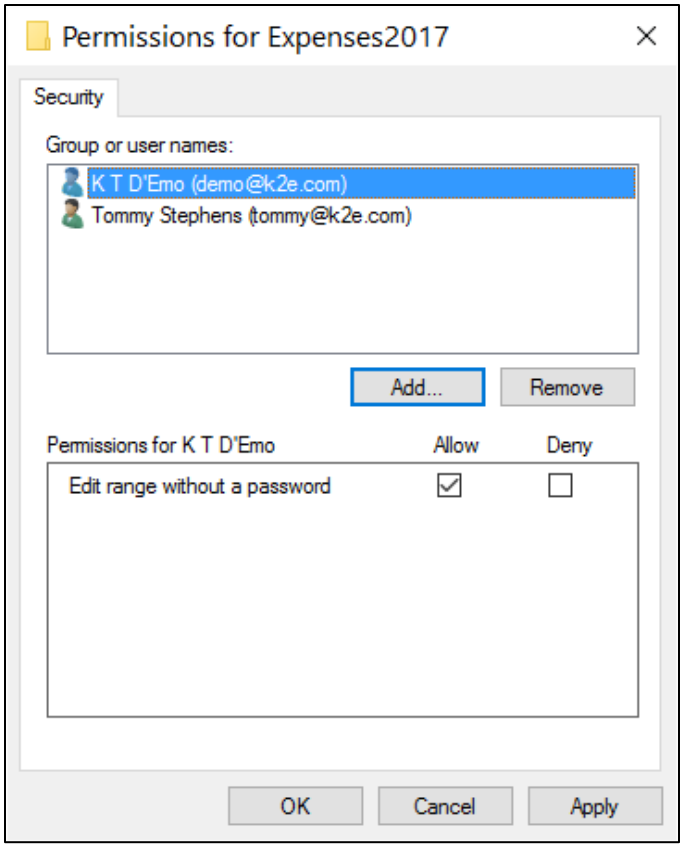


Figure 111 - Adding a User Permission to an Excel Worksheet

This technique does not preclude the use of passwords; it only means that they are optional. If desired, you can add user permissions and require passwords for even more robust security. Additionally, you can add user groups such as *Sales*, *Executive*, and *Manufacturing* instead of individual users, assuming those groups already exist in the domain.

Adding Protection to A Workbook

In addition to protecting the contents of a worksheet, you can also secure a workbook's structure. This protection can lock the workbook window's position and prevent users from changing the workbook's overall layout. As with worksheet protection, you can enhance workbook protection with an optional password. Again, this password is independent of the encryption and read-only passwords discussed earlier.

To protect a workbook, select **Protect Workbook, Protect Structure and Windows** on the Ribbon's **Review** tab to open the dialog box shown in **Figure 112**.

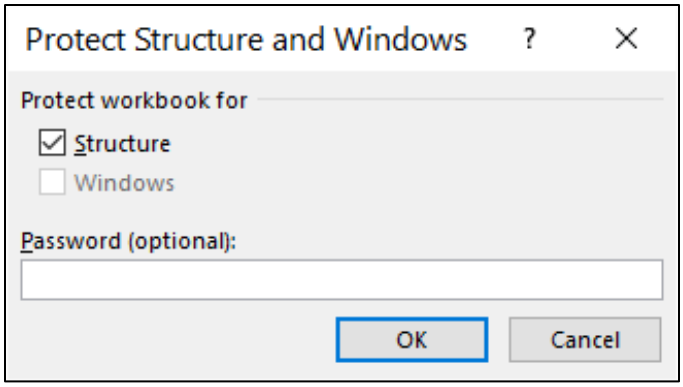


Figure 112 - Protecting a Workbook in Excel

Select the options desired, enter a password, and click **OK**. Confirm the password and click **OK** again. Note that checking **Structure** *prevents* users from taking the following actions:

- Viewing hidden worksheets;
- Moving, deleting, hiding, or changing the names of worksheets;
- Inserting new worksheets or chart sheets;
- Moving or copying worksheets to another workbook;
- Displaying the source data for a cell in the data area or showing report filter pages on separate worksheets in PivotTable reports;
- Creating a scenario summary report for scenarios; and
- Working with the analysis tools from the Analysis ToolPak, which places results on a new worksheet.

> Note that using passwords to restrict modifying worksheets or workbooks does *not* encrypt the workbooks. Passwords that restrict changes (such as formatting and editing) are weak and can be easily broken by online password-cracking services.

Adding Signatures to Documents

In addition to restricting changes to documents, Excel facilitates paperless processes by enabling digital signing. There are two ways to add signatures to an Excel workbook: 1) the signature can be a digital signature that is not visible in a workbook, or 2) the document can contain a **Signature Line** that allows users to add a typed or holographic signature to a workbook.

Adding a Digital Signature to a Document

You can insert digital signatures into documents to establish the document's authenticity, integrity, and origin. For example, a CFO could indicate that she has reviewed and approved the quarterly financial statements by digitally signing. Unlike an Office signature line, a digital signature is not visible within the document's contents. Still, recipients can determine whether the document was digitally signed by viewing its digital signature. Alternatively, they can look for the **Signatures** icon on the status bar at the bottom of the screen. Clicking on the icon opens the **Signatures** task pane, as shown in **Figure 113**.

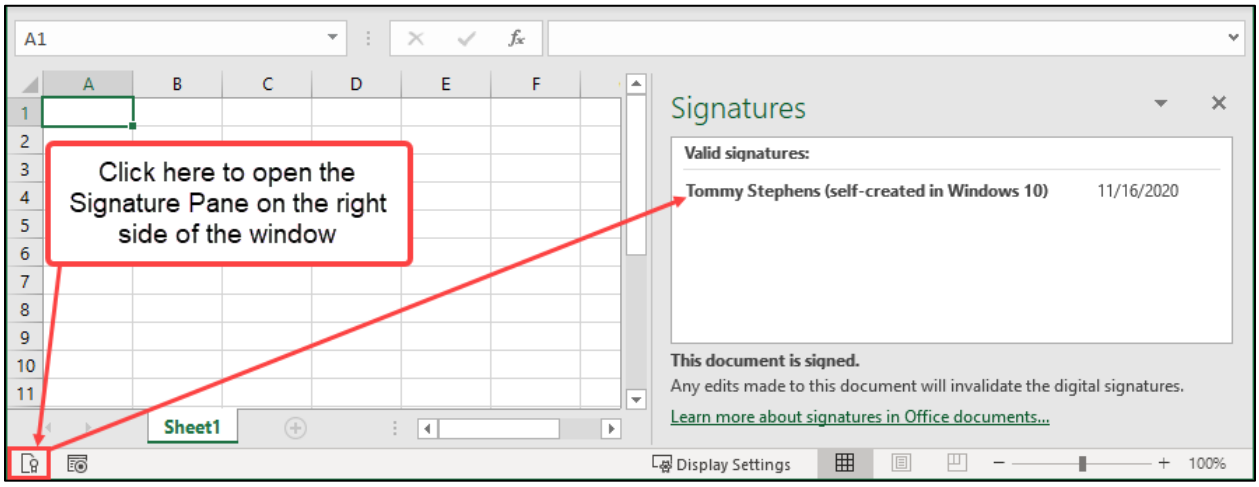


Figure 113 - Signatures Icon and Pane in Excel

To add an invisible digital signature to a document, click **File**, **Info**, **Protect Workbook**, and **Add a Digital Signature** to open the **Sign** dialog box shown in **Figure 114**. Next, enter the purpose of signing the document and click **Sign** to add your digital signature.

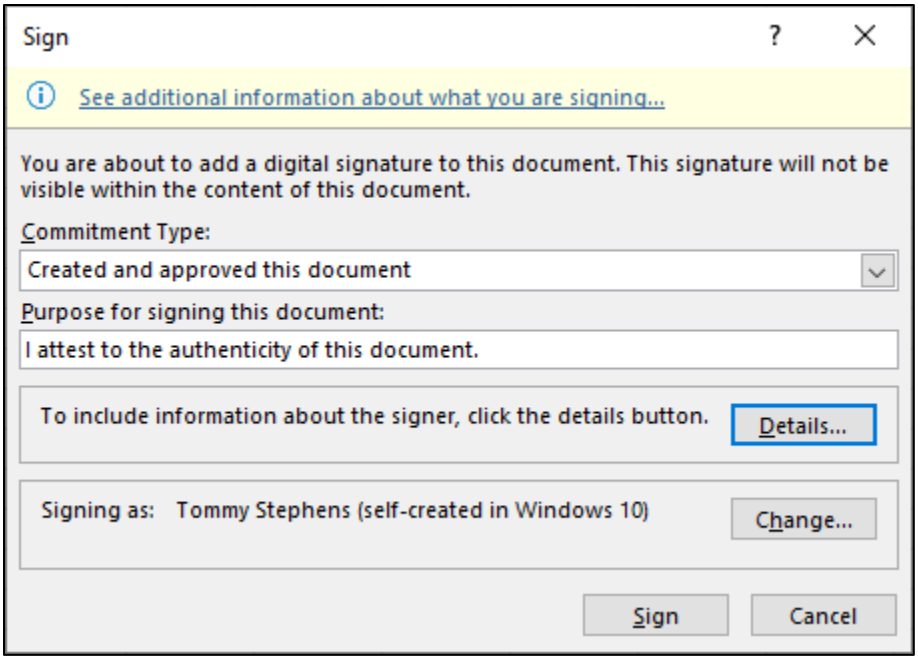


Figure 114 - Inserting a Digital Signature

Once signed, no one can change a signed document without deleting the digital signature. To remove a digital signature, open the **Signature** task pane, click the drop-down arrow next to the signature to remove, and click **Remove Signature**. Then, confirm the removal of the signature.

Inserting a Signature Line into a Document

You can insert **Signature Lines** into documents to indicate that a document is to be signed or that someone has already signed it. Once inserted, a signature line appears, as shown in **Figure 115**.

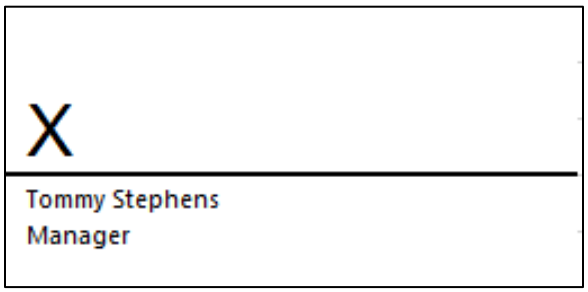


Figure 115 - Sample Signature Line

To insert a signature line into a document, select **Signature Line** from the Ribbon's **Insert** tab of the Ribbon to open the **Signature Setup** dialog box, as shown in **Figure 116**. Then, enter the appropriate information and click **OK** to add the signature line to the document.

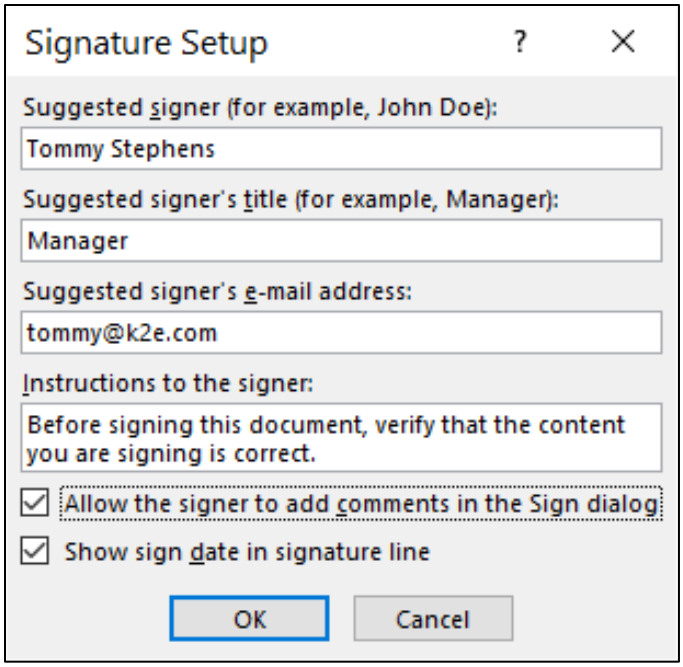


Figure 116 - Signature Setup Dialog Box

The document's intended signer adds a signature by right-clicking on the signature line and selecting **Sign** from the context-sensitive menu, as shown in **Figure 117**.

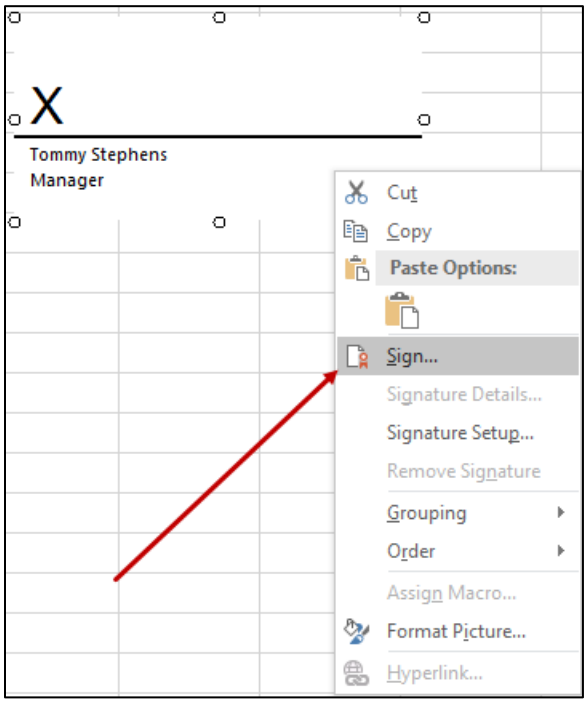


Figure 117 - Right-click on a Signature Line to Sign a Document

Signers can type in their names or select an image of their signature, enter the purpose for signing the document, and click **Sign,** as shown in **Figure 118**.

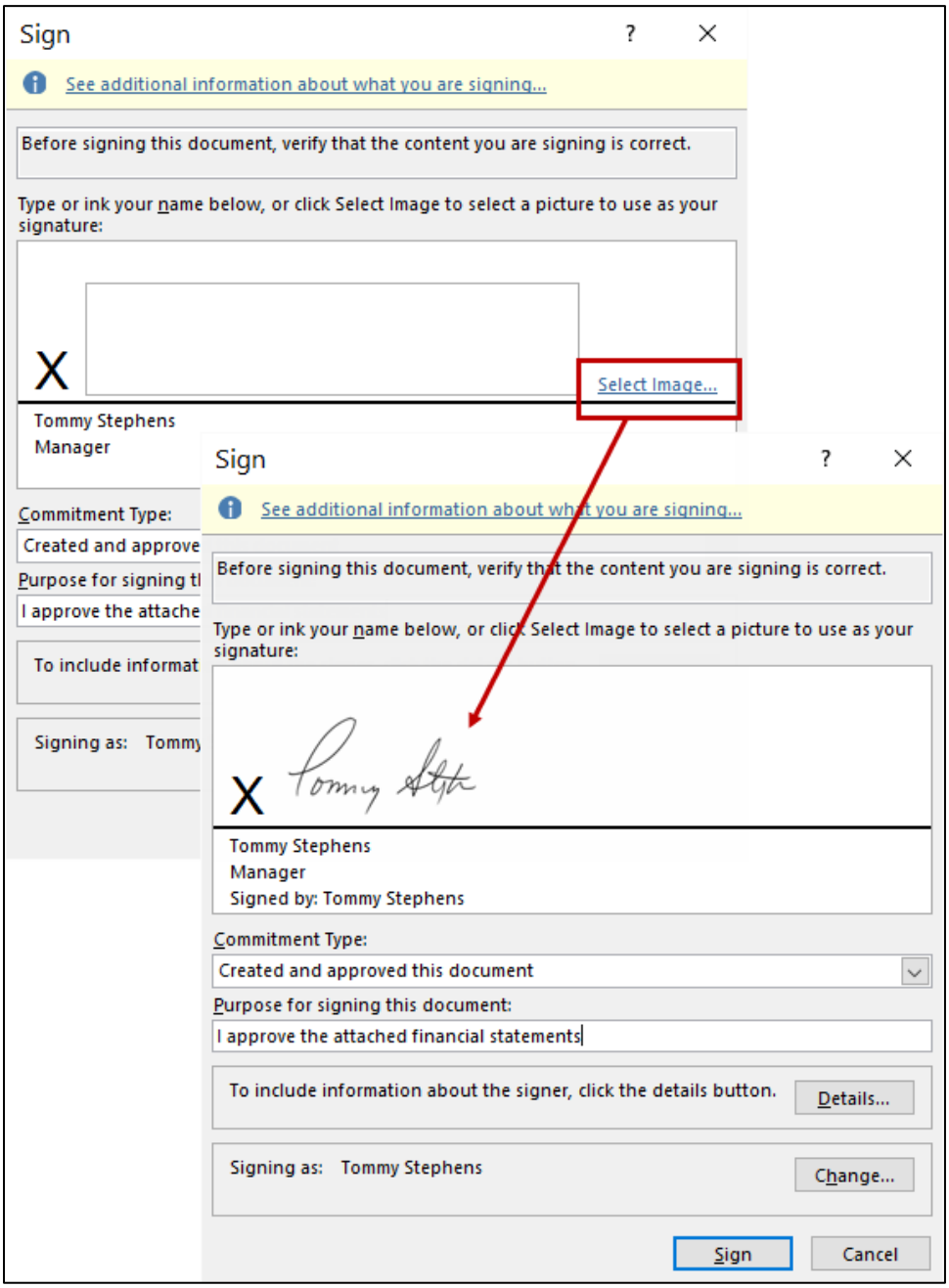


Figure 118 - Signing a Workbook with a Signature Image

Again, by using Signature Lines, responsible officials could indicate their approval of financial statements (or other sensitive documents) before releasing the records to others.

Removing Hidden Data

You can remove hidden data using the **Document Inspector** feature. The Document Inspector examines a document and removes hidden metadata, personal information, or content otherwise saved with the document. Examples of data that the Document Inspector removes include comments, versions, tracked changes, annotations, properties, and hidden text. From a security and confidentiality standpoint, these data types could be harmful if exposed. Therefore, remove this type of data before distributing the document.

To take advantage of this feature, click **File**, **Info**, **Check for Issues**, and **Inspect Document** to open the **Document Inspector** dialog box shown in **Figure 119**. In the **Document Inspector**, indicate the types of data for which the Document Inspector should look and then click **Inspect** to begin finding and removing the hidden data.

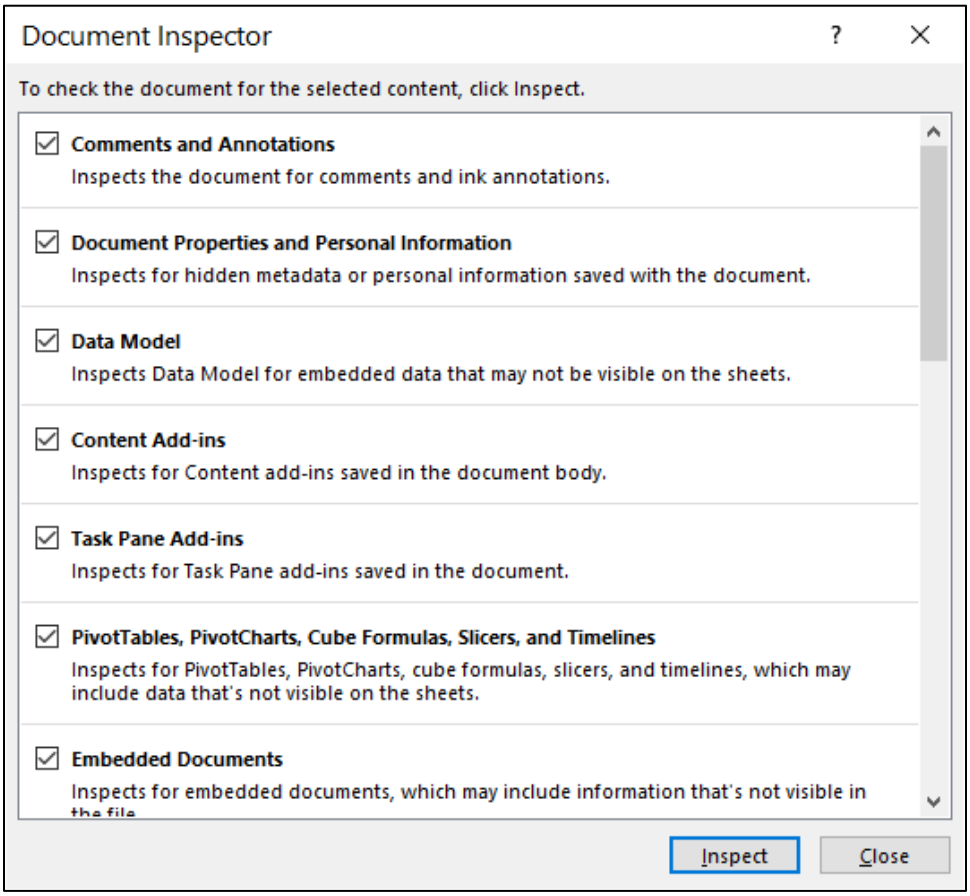


Figure 119 - Document Inspector Window

The Document Inspector displays the inspection results in a separate window, as shown in **Figure 120**. You can choose to remove the data found or leave it in the document.

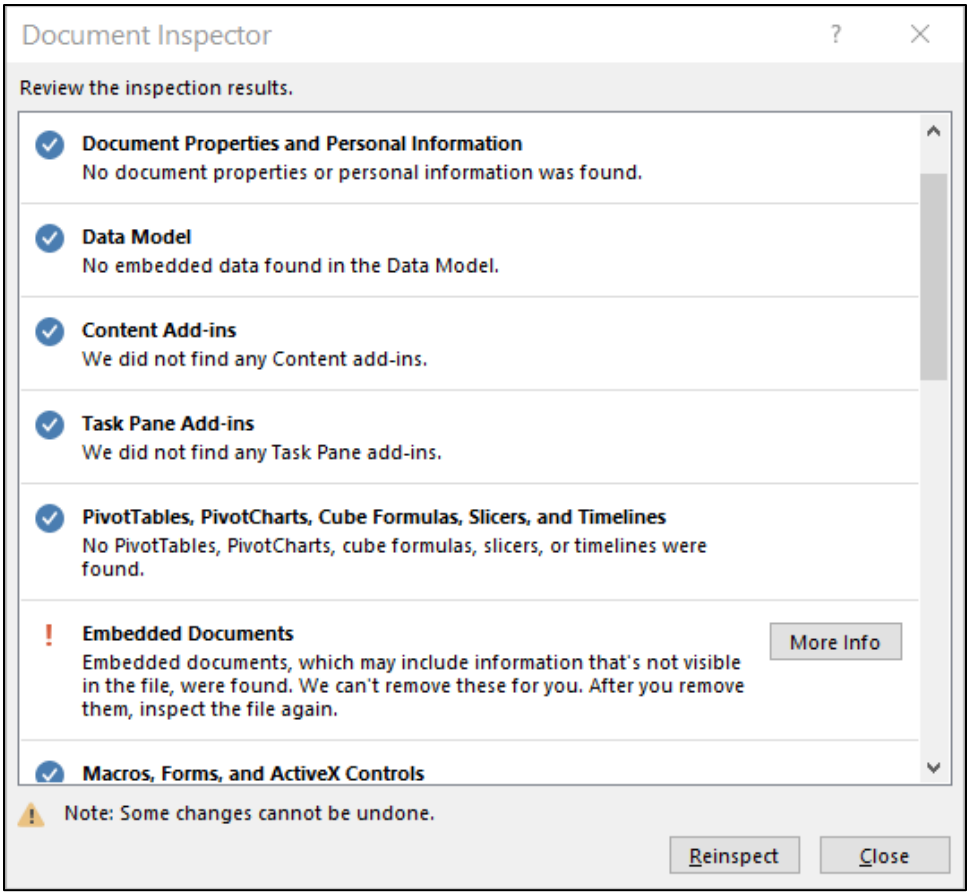


Figure 120 - Results of Document Inspector Examination

When Excel documents undergo numerous revisions and iterations before being "finalized," the Document Inspector provides much-needed functionality to prevent document history from being retained in the workbook.

Preventing Changes to Documents

One of Excel's more significant protection features is marking a document as ***Final***. When you indicate a spreadsheet is "Final," Excel disallows future changes as long as it remains in that status. While **Mark as Final** is a welcome feature, you should rely on it only to prevent accidental or unintentional changes to a document. Therefore, don't view Mark as Final as a "security" feature because users can easily remove the "final" status from the file and edit the document.

To add the *final* status to a document, click **File**, **Info**, **Protect Workbook**, and **Mark as Final**, as shown in **Figure 121**. This action makes the document read-only and prevents changes if it remains in this state. To remove the protection from a document, repeat the steps outlined, and the document will return to an editable format.

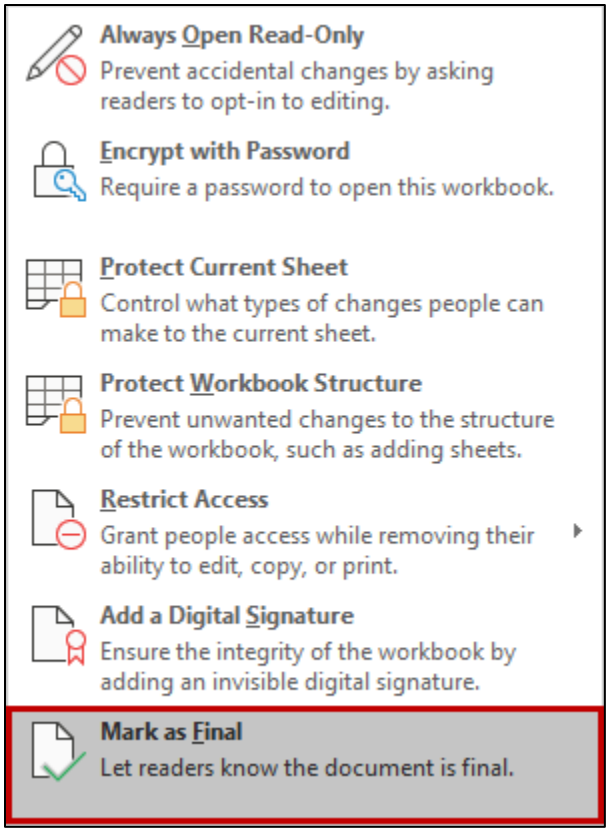


Figure 121 - Marking a Document as Final

?	Which of Excel's security tools are you most likely to use when attempting to secure sensitive information in your workbooks?

Best Practices For Collaborating And Sharing In Excel

Chapter 5

Sharing workbooks with others to obtain their input, comments, and expertise is a common need. Frequently, preparing multi-department budgets, generating production schedules, and building financial statements require collaboration. In this chapter, you will learn best practices for collaborating and sharing in Excel.

Learning Objectives

Upon completing this chapter, you should be able to:

- List multiple methods for collaborating on an Excel document;
- Differentiate between Excel's Co-Authoring and Shared Workbook features; and
- Identify some of the relative advantages of using SharePoint, SharePoint Online, and Teams as the backbone of collaboration efforts.

Sending Workbooks Via Email

Although not necessarily the best option, email is the most common way to collaborate. You can send workbooks as email attachments to other users for their input, revision, or review. Excel makes this easier by automating the process. Select **File**, **Share**, **Email**, and **Send as Attachment** to email a workbook to other users**,** as shown in **Figure 122**. You can also send workbooks as PDF or XPS attachments or via an Internet-based fax service.

Of course, using email as the backbone of collaboration efforts is fraught with drawbacks. First is the security issue: email is notoriously insecure for sending data. Therefore, you must take appropriate precautions to secure the email's contents before sending it. Examples of such precautions include:

- Using an encrypted email service,
- Encrypting the Excel workbook with a password, and
- E-mailing a link to the workbook after storing it in a secure cloud-based portal.

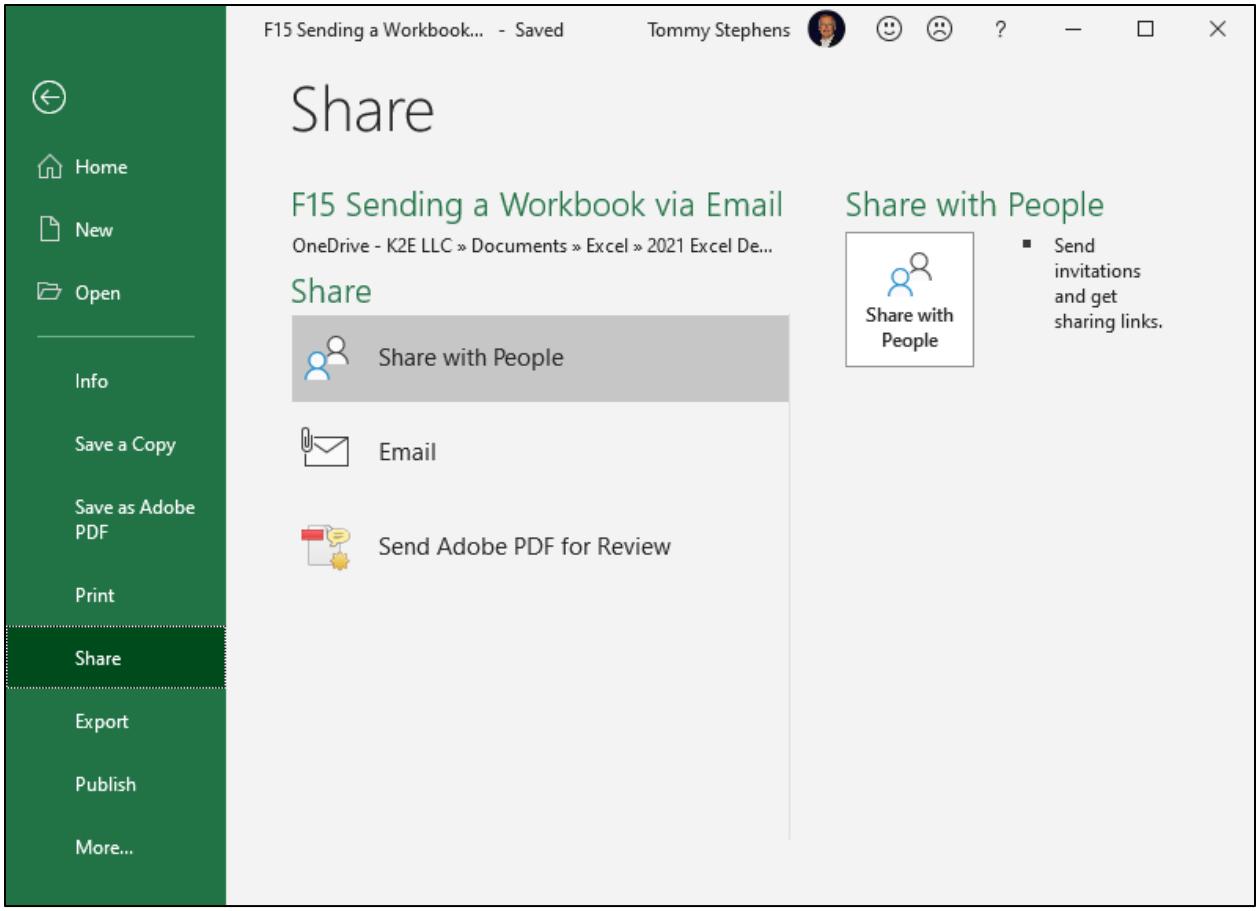


Figure 122 - Sending a Workbook as an Attachment

Co-Authoring Excel Workbooks

Additionally, you may be able to send a *link* to the document instead of the document itself. This feature depends on which version of Excel you use and whether you store your documents in a cloud-based repository such as **OneDrive for Business**. You can send a link by choosing the **Share with People** option shown in Figure 122. Sending a link can facilitate real-time collaboration because all users will simultaneously access and edit the same workbook.

Excel's co-authoring feature allows multiple users to access and edit Excel workbooks simultaneously, using the Internet as the "backbone" of the process. In addition to co-authoring in Excel, you can co-author in Word, PowerPoint, and OneNote. With co-authoring, any user with appropriate rights can access and edit documents, and these edits can be performed collaboratively and in real time. Thus, the author of an Excel workbook used to calculate a company's tax accrual could share it with its external accountant in public practice. Then, the author and the external accountant could review and edit the workbook in real time.

You must have a Microsoft 365 subscription to set up co-authoring for a workbook. You must also store your document in a cloud-based location such as OneDrive, OneDrive for Business, or

SharePoint Online (discussed later in this chapter). After you do so, click **Share** in the upper-right corner of the workbook, or use the **Share with People** option described previously. Next, you can enter the **Link Settings** for the collaboration and send the document links to all invited parties. **Figure 123** illustrates that process.

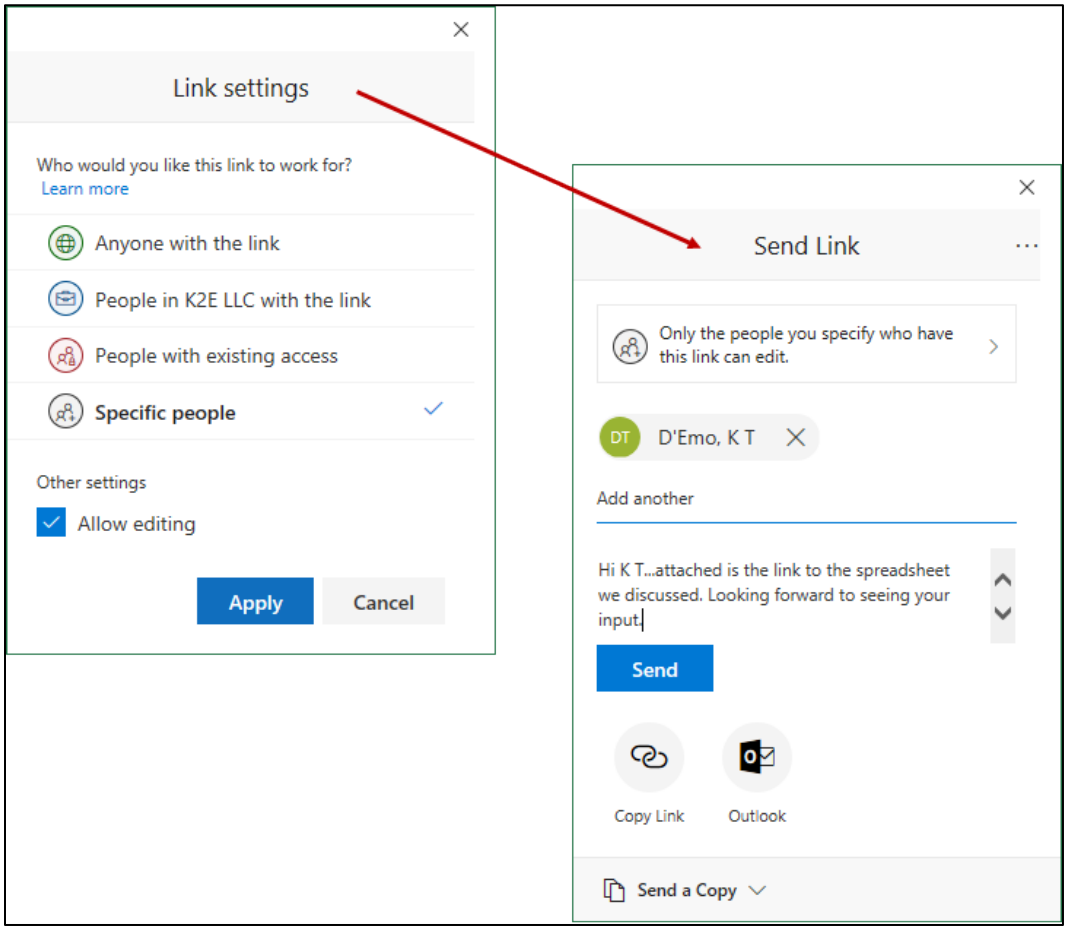


Figure 123 - Setting Up an Excel Workbook for Co-Authoring

Once you share the document, you and other users can access and edit it simultaneously. When collaborating using this technique, others' changes to the workbook appear almost instantly. Note that other users do not need Excel installed on their devices to access and edit the workbook. If they do not have Excel installed on their device, the workbook will open in **Excel Online**, a web-based version of Excel.

A common question regarding co-authoring is, “What happens when multiple users attempt to edit the same cell simultaneously?” In this case, Excel uses “cell locking” to allow only one person access to a cell at any time. For example, if Sara clicks a cell to edit it, Juan cannot select that same cell until Sara moves to another cell. The last change made to a cell is the one that saves when the workbook saves, whether through a manual save or Excel’s **AutoSave** feature. If you store the workbook in OneDrive, OneDrive for Business, or SharePoint Online, you can access previous versions and “roll back” the workbook to a prior version.

Shared Workbooks, A Legacy Feature

Shared workbook functionality allows multiple users to view and make changes to a workbook concurrently when it is located in an accessible network location. For example, if your team is working on a project that requires input from several staff members, sharing the project workbook lets them work on it simultaneously. Compare that to the commonly used approach of exclusively using a workbook. While one staff member uses a workbook, others cannot access or edit it. Hence, work on the project slows because access to the project document is restricted when a staff member is working on it.

Before proceeding, you should know that Microsoft has tagged Shared Workbooks as a "legacy" feature in Excel. Following is a passage from Microsoft's website addressing this issue.

> *If you're using the latest version of Excel and you're an Office 365 subscriber, you may find that buttons related to the "Shared Workbook" feature are no longer on the Review tab. We've removed them because there's a much easier and better way to share workbooks, and that's with co-authoring.*

Another page on Microsoft's site adds the following.

> *Shared Workbooks is an older feature that allows you to collaborate on a workbook with multiple people. This feature has many limitations, and has been replaced by co-authoring.*

Given the two statements above, we encourage caution before investing substantial resources in implementing shared workbooks extensively across an organization. Furthermore, once considered state-of-the-art, this feature's limitations are significant, particularly when compared to co-authoring.

In the following example, the CFO for GTM Printing has created a workbook for his office managers to enter budget updates for the third and fourth quarters of the current year. The workbook has a separate sheet for each office and has been saved in a shared folder on a server in the home office, accessible via remote access software. The CFO wants to share the workbook so managers can enter their budget updates, even if another manager already uses it. From the CFO's vantage point, periodic updates to the budget are essential so that managers can routinely compare actual to budgeted results.

To share a workbook, open the workbook to share. From the **Review** tab of the Ribbon, select **Share Workbook**. In the **Share Workbook** dialog box, check **Allow changes by more than one user at the same time,** as shown in **Figure 124**. Enter the days to maintain the change history on the **Advanced** tab. The default is thirty days. Enter an arbitrarily large number, such as 3650 days, to keep the changes indefinitely.

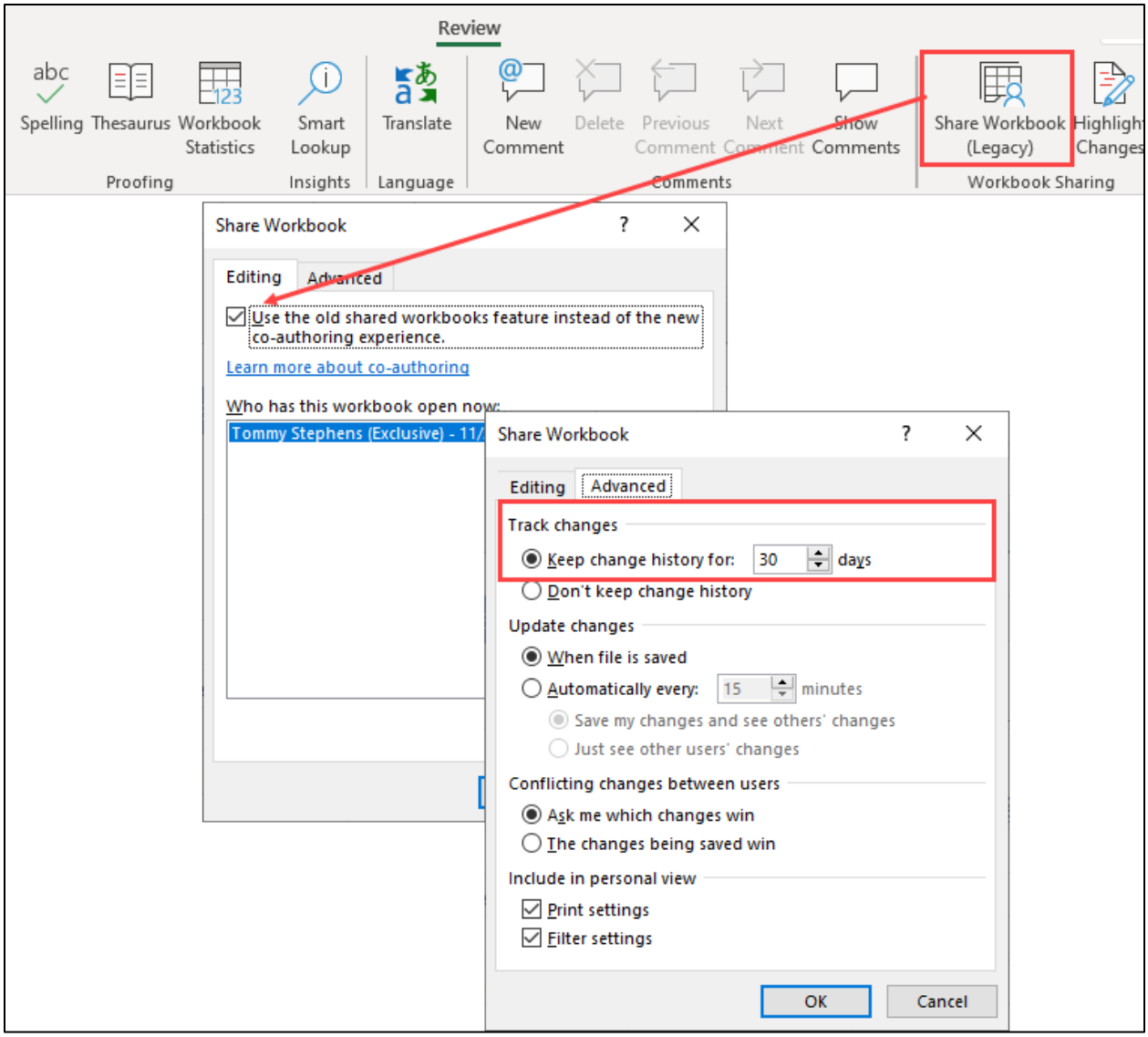


Figure 124 - Sharing a Workbook Using the Legacy Technique

After a workbook is shared, multiple users can view and modify it concurrently, and Excel automatically maintains a history of all changes. The change history tracks what changed, who changed it, and its date and time. The **Keep change history** setting on the **Advanced** tab of the **Shared Workbook** dialog box determines how long the change history is maintained. Unfortunately, not all Excel functionality is available in a shared workbook. **Table 7** lists the restricted functionality in a shared workbook and some possible workarounds.

SHARED WORKBOOK FUNCTIONALITY	
Users Cannot	**Possible Workaround**
Create an Excel table	

Insert or delete blocks of cells	Entire rows and columns can be inserted.
Delete worksheets	
Merge cells or split merged cells	
Add or change conditional formats	Existing formats continue to appear as cell values change.
Add or change data validation	Cells continue to be validated when you type new values.
Create or change charts or PivotCharts	Existing charts and reports can be viewed.
Insert or change pictures or objects	Existing pictures and objects can be viewed.
Insert or change hyperlinks	Existing hyperlinks continue to work.
Use drawing tools	Existing drawings and graphics can be viewed.
Assign, change, or remove passwords	Existing passwords remain in effect.
Protect or unprotect worksheets or the workbook	Existing protection remains in effect.
Create, change, or view scenarios	
Group or outline data	Existing outlines can be used.
Insert automatic subtotals	Existing subtotals can be viewed.
Create tables	Existing tables can be viewed.
Create or change PivotTable reports	Existing reports can be viewed.
Write, record, change, view, or assign macros	Existing macros that do not access unavailable features can be run, and macros can be recorded and stored in another workbook.
Change or delete array formulas	Existing array formulas continue to calculate correctly.
Use a data form to add new data	An existing data form can be used to find a record.
Work with XML data	

Table 7 – Shared Workbook Functionality

Note that functionality incorporated into a workbook when shared is still functional in the shared workbook with few exceptions. You can build a fully functional shared workbook by anticipating users' needs and adding functionality ***before*** sharing the workbook. For example, if you anticipate needing a PivotTable in a shared workbook, create the PivotTable before sharing the workbook. If necessary, you can enable all restricted functionality by un-sharing a workbook. However, unsharing a workbook will cause Excel to delete any previously recorded workbook history.

To edit a shared workbook, open and modify it like any other workbook. Excel maintains the change history in the background. By default, each user has a ***personal view*** of a shared workbook. Personal views preserve filter and print settings for each user. To use the original print

and filter settings, open the shared workbook. From the **Review** tab of the Ribbon, select **Share Workbook**. On the **Advanced** tab, uncheck **Print settings** and **Filter settings** in the **Include in personal view** section and click **OK**.

Whenever a user saves a shared workbook, any changes made by other users during a concurrent session are reflected in the workbook. If multiple users make changes, the changes are color-coded by user. Hovering the cursor over a changed cell identifies the change made, the change's date and time, and who made the change, as shown in **Figure 125**. Changes by other users remain highlighted until someone saves the shared workbook again.

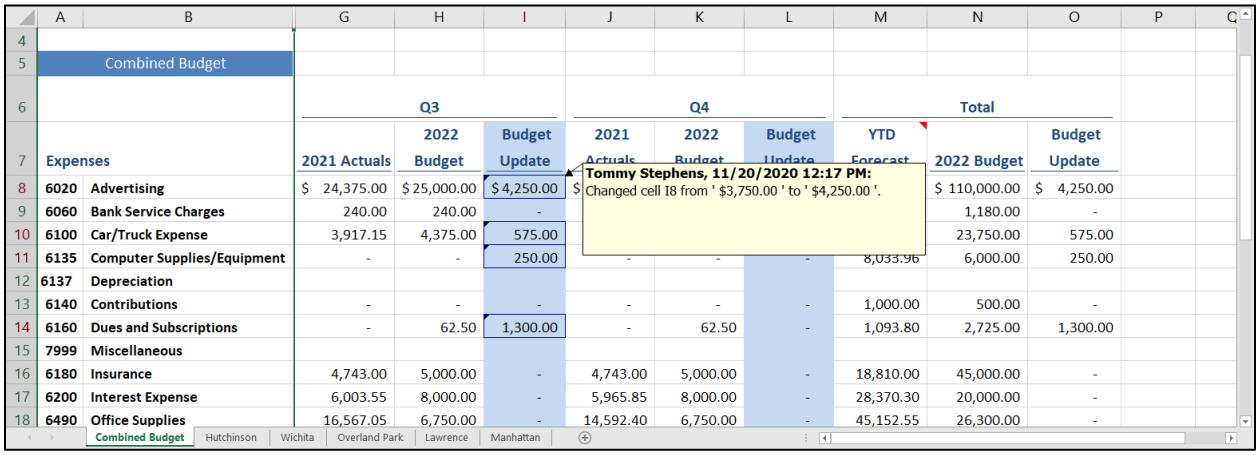


Figure 125 - Changes Identified by Date, Time, and User

If more than one user changes to the same cell or cells during a concurrent session, Excel will prompt the second and subsequent users who attempt to save their changes to resolve the identified conflicts, as shown in **Figure 126**.

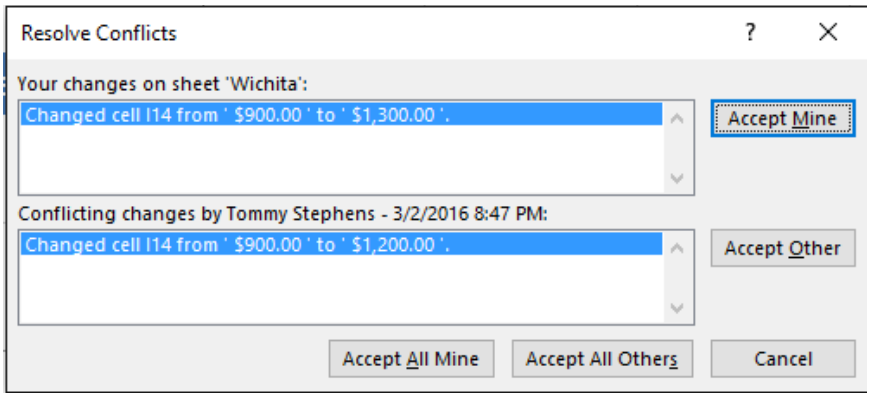


Figure 126 - Resolving Conflicts in a Concurrent Session

Users can accept their own changes or those saved by others. Excel records *all* saved changes in the change history, regardless of whether a change was accepted or rejected during conflict resolution. This action allows a shared workbook *owner* – the user who shared the workbook first – to accept or reject changes during a review process. The owner maintains all rights to manage the workbook. For example, an owner can remove users from a shared workbook, accept or reject conflicting changes made by multiple users, and un-share a protected workbook.

Excel tracks all changes from when the owner shares the workbook until the owner unshares the workbook. Upon unsharing the workbook, Excel deletes all change history from the document. To show tracked changes in a shared workbook, select **Track Changes** and **Highlight Changes** from the Ribbon's **Review** tab, as shown in **Figure 127**.

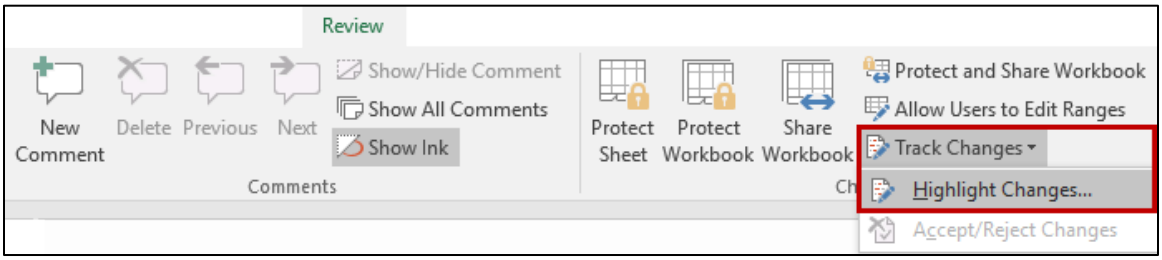


Figure 127 - Tracking Changes in a Shared Workbook

In the **Highlight Changes** dialog box, check **Highlight changes on screen** and/or **List changes on a new sheet** and click **OK**, as shown in **Figure 128**.

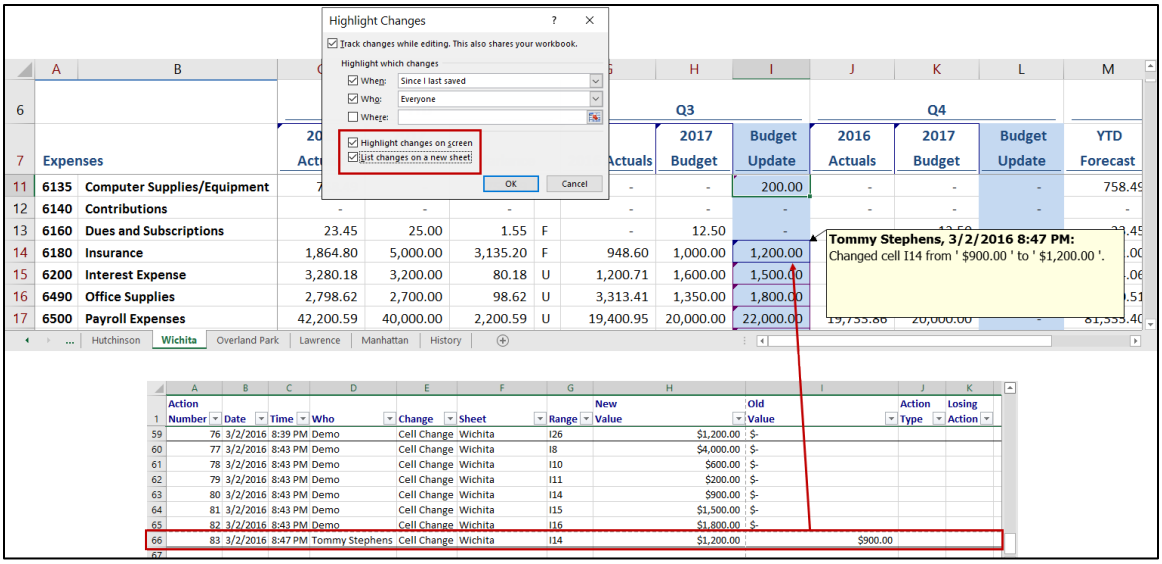


Figure 128 - Displaying Changes on Screen and a Separate Worksheet

Excel surrounds all cells containing changes that meet the criteria set in the **Highlight Changes** dialog box with a navy-blue border and inserts a navy-blue arrow point in the upper left corner, as shown in Figure 128. Hovering the cursor over a highlighted cell exposes a pop-up containing the user's name who made the change. Additionally, Excel displays the date and time of the change, along with its nature. If the number of changes is extensive, or you made changes over a large range in a worksheet or multiple sheets, creating a new worksheet that lists the changes is a better alternative. Excel will create a new worksheet named **History** that contains a chronological list of all changes that meet the selection criteria set in the **Highlight Changes** dialog box. Note that Excel does not save a **History** sheet when you save a shared workbook. However, you can generate an up-to-date history sheet at any time.

You may *accept* or *reject* changes at any time. Further, you can repeat that process as often as necessary. All saved changes are available for a reviewing user to accept or reject. In other words, if three users – *Tommy*, *Debbie*, and *Mac* – changed a single cell, all three values would be available for acceptance during the review process, even though the amount entered by Tommy was the value currently stored in the cell. Further, rejecting a change in one review does not preclude accepting the exact change in a subsequent review.

To accept or reject changes made to a shared workbook, select **Track Changes, Accept/Reject Changes** from the Ribbon's **Review** tab. Next, enter any desired criteria in the **Select Changes to Accept or Reject** dialog box and click **OK**, as shown in **Figure 129**. Excel will step through the workbook, stopping at each cell that contains a change matching the selection criteria. Excel displays the Accept or Reject Changes dialog box at each stop to let users accept or reject previously saved changes.

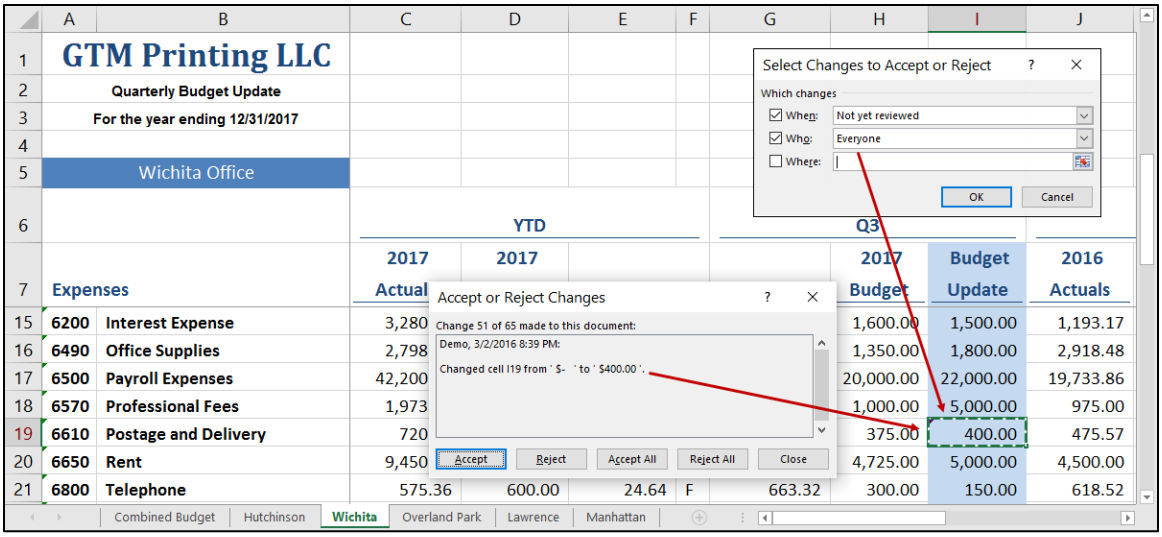


Figure 129 - Accepting or Rejecting Changes

To stop sharing a workbook, select **Share Workbook** on the **Review** tab of the Ribbon. Then, uncheck **Allow changes by more than one user at the same time** on the **Editing** tab of the **Share Workbook** dialog box and click **OK**. *Caution*! When you un-share a workbook, it is un-shared, and Excel deletes the change history. If the change history is important, create an up-to-date **History** worksheet and print it or copy it to a new workbook. After un-sharing the workbook, you can copy the history sheet back into it. However, you must rename the worksheet because *History* is a reserved sheet name. You can use the **Protect and Share** workbook feature to prevent unauthorized unsharing of a workbook.

Shared workbooks primarily enable multiple users to collaborate on tasks or projects. However, you can choose to share a workbook solely to enable change tracking. In other words, a single user may share a workbook so that Excel automatically records all changes made on a large project. A user may not intend to allow other users to change the project workbook, but has enabled sharing only for change tracking. If the user documents all material changes to the workbook, the workbook becomes self-documenting. In the example shown in **Figure 130**Note how Excel tracked the change. Also, see how the user's note explains the purpose of the change. This functionality is helpful in capital budgeting, business planning, and countless other Excel-based tasks.

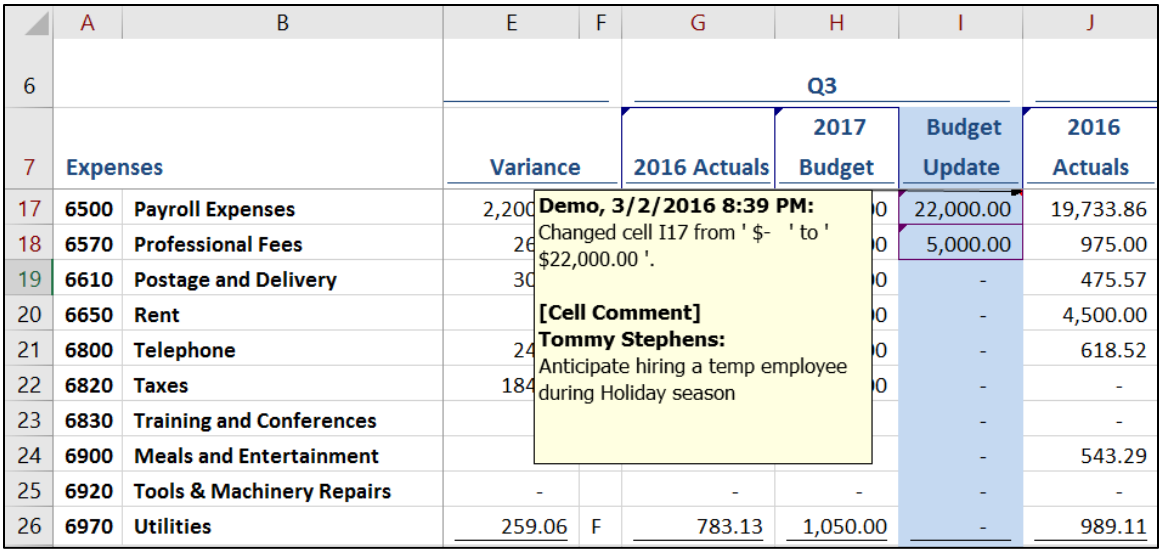


Figure 130 - Using Comments to Explain the Worksheet Changes

Merging Workbooks

As discussed in the previous section, shared workbooks allow multiple users to change a workbook simultaneously. Typically, a shared workbook is stored in a network location accessible by all expected users. Individual users access the workbook over the network and make any necessary changes. However, what if not all users have access to the network? You can also use shared workbooks to merge multiple users' changes into a single "master" workbook. In this case, you will provide all users with a copy of the shared workbook, each with a unique name. Each user will work independently on their copy of the shared workbook and then email or otherwise send it to the "master" user when they complete their edits. The master user will move all returned workbooks into a single folder, open the original workbook, and merge the changes from the returned workbooks into the original in a single automated step. The process is fully automated, with no cutting, pasting, or rebuilding formulas necessary. Nor is there a need to reformat data. Imagine the time savings for organizations using this method to consolidate departmental financial results, rather than the traditional approach of copying and pasting data from multiple workbooks into a single workbook.

You must share the workbook ***before*** making copies and distributing them to others. You cannot merge workbooks that are not copies of a shared workbook. Further, each workbook copy must have a unique name. Including users' names in the name of their shared workbook is an easy way to create unique names. For example, if you intend to distribute copies of a shared workbook named *Project 2021-01* to multiple users, the workbooks' names could be *Project 2021-01-Juan*, *Project 2021-01-Valerie*, etc.

The functionality of shared workbooks – change tracking, conflict resolution, accepting and rejecting changes – is available in a merged workbook. You will address any conflicting entries when merging the workbook copies. However, you should seek to minimize conflicts among users by constructing the master workbook to reduce the opportunities for contradictory entries. For example, you can place the input required of individual users on separate worksheets.

To use shared workbooks for independent user input, do the following.

1. Share a workbook, as discussed earlier in this chapter.

2. Then, replicate the shared workbook to create a copy for each collaborator. Make sure the master workbook remains unchanged.

3. Give each shared workbook a unique name and then distribute them to individual users.

4. After users make their changes, they must return the changed workbooks to the owner.

Additionally, you must add the **Compare and Merge Workbooks** command to the Ribbon or Quick Access Toolbar. After adding the tool to either location, you can merge the changed workbooks into the master workbook.

1. Move all the returned workbooks into the same folder with the original, unchanged shared workbook. Then, open the original workbook.

2. Click the **Compare and Merge Workbooks** icon on the QAT or Ribbon to open the **Select Files to Merge into Current Workbook** dialog box. You can use **CTRL + Click** to highlight all workbooks to be merged and click **OK**, as shown in **Figure 131**.

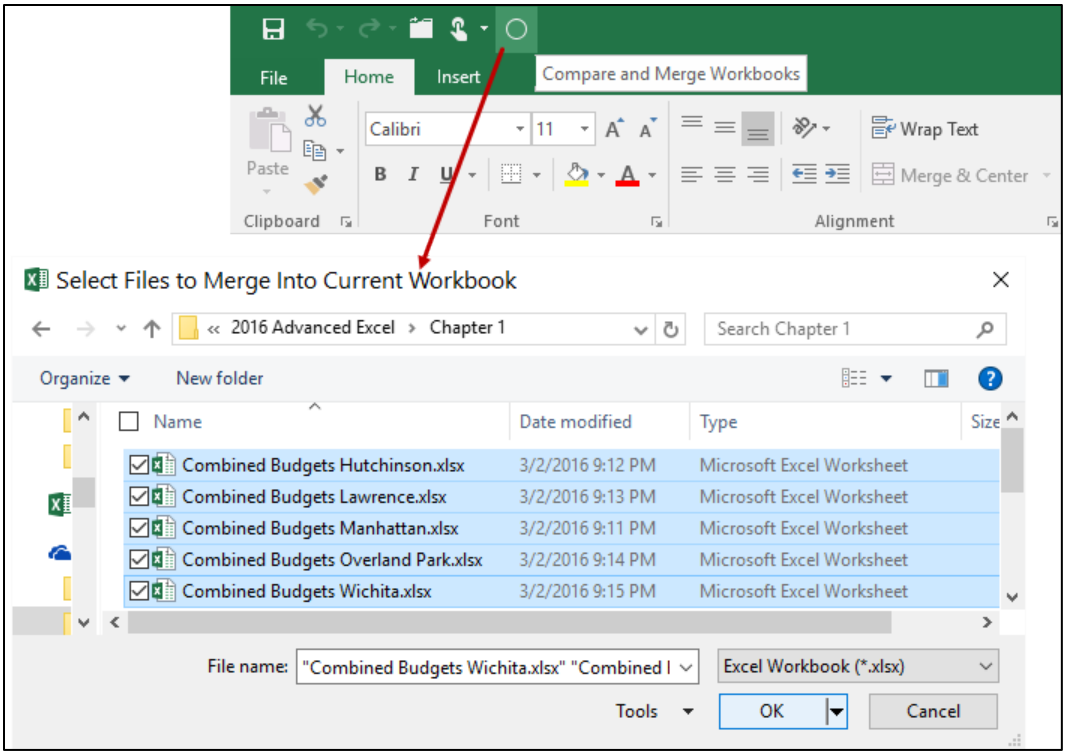
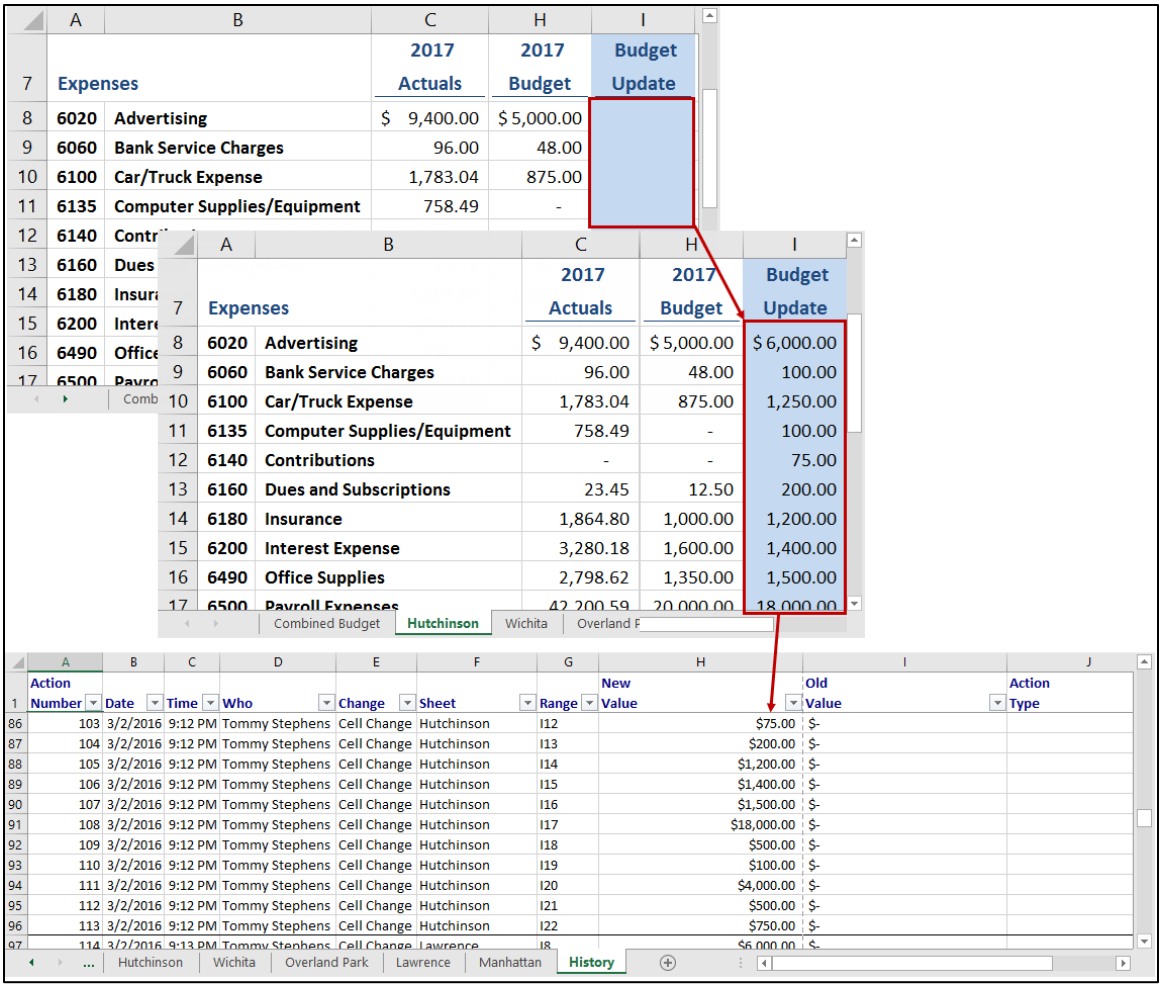


Figure 131 - Selecting Workbooks to Merge into the Original Shared Workbook

That simple automated process combined all the changes in the users' workbooks. Additionally, Excel created a complete change history in the original workbook from which you can see every edit, as shown in **Figure 132**. You can review the workbook and accept or reject each edit. You can also resolve conflicts caused by multiple users changing the same cell's contents. The processes to accomplish these tasks are the same as those for ordinary shared workbooks.



Expenses			2017 Actuals	2017 Budget	Budget Update
8	6020	Advertising	$ 9,400.00	$ 5,000.00	$ 6,000.00
9	6060	Bank Service Charges	96.00	48.00	100.00
10	6100	Car/Truck Expense	1,783.04	875.00	1,250.00
11	6135	Computer Supplies/Equipment	758.49	-	100.00
12	6140	Contributions	-	-	75.00
13	6160	Dues and Subscriptions	23.45	12.50	200.00
14	6180	Insurance	1,864.80	1,000.00	1,200.00
15	6200	Interest Expense	3,280.18	1,600.00	1,400.00
16	6490	Office Supplies	2,798.62	1,350.00	1,500.00
17	6500	Payroll Expenses	42,200.59	20,000.00	18,000.00

Combined Budget | Hutchinson | Wichita | Overland Park

Action Number	Date	Time	Who	Change	Sheet	Range	New Value	Old Value	Action Type
103	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I12	$75.00	$-	
104	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I13	$200.00	$-	
105	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I14	$1,200.00	$-	
106	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I15	$1,400.00	$-	
107	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I16	$1,500.00	$-	
108	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I17	$18,000.00	$-	
109	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I18	$500.00	$-	
110	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I19	$100.00	$-	
111	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I20	$4,000.00	$-	
112	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I21	$500.00	$-	
113	3/2/2016	9:12 PM	Tommy Stephens	Cell Change	Hutchinson	I22	$750.00	$-	
114	3/2/2016	9:13 PM	Tommy Stephens	Cell Change	Lawrence	I8	$6,000.00	$-	

Hutchinson | Wichita | Overland Park | Lawrence | Manhattan | History

Figure 132 - Merging Workbooks Produces a Combined Workbook with Change History

You can merge one, some, or all workbooks in a single- or multiple-merge process. Further, you can combine the same workbook *multiple* times; however, Excel incorporates into the master workbook only the changes made since the last merge. Subsequent merges will not duplicate data previously imported into the master workbook. This trait is beneficial in processes that span multiple weeks or months, such as the annual budget process. A master user, such as a budget administrator, can merge changes as they occur during the budget process.

Collaborating Using SharePoint Server, SharePoint Online, and Teams

SharePoint Server, **SharePoint Online**, and **Teams** provide three additional opportunities to collaborate on Excel and other Microsoft Office documents. If you have access to one or more of these tools – and most Microsoft 365/Office 365 subscribers will – then you have terrific options for collaborating with other users – both inside and outside your organization – to get better results in less time.

Publishing to SharePoint Server and SharePoint Online

From a practical perspective, the most significant difference between SharePoint Server and SharePoint Online is that SharePoint Server is a local resource, and SharePoint Online is Cloud-based. Functionally, the two platforms are otherwise nearly identical. Excel can use either SharePoint platform to facilitate collaboration using the co-authoring experience discussed previously. To take advantage of this capability, save your workbook to your SharePoint environment. Thereafter, multiple users can access the workbook simultaneously and co-author the document.

You can co-author with the Excel Web App only if everyone uses the Excel Web App to access the workbook in this environment. If anyone uses the desktop client application of Excel to access the workbook, co-authoring in the Excel Web App is disabled for that workbook while it is open in the client application.

Given the similarity of this process to that previously discussed when saving the workbook in OneDrive, you might ask, "*What is an advantage of using SharePoint for Excel co-authoring*?" SharePoint provides several unavailable options if you choose to co-author through OneDrive.

1. **Permissions**. With SharePoint, you can establish customized permissions at the user level to control how users can access and edit documents. These permissions can be granular if necessary, including providing access at the site, library, folder, or document level.

2. **Versioning**. SharePoint lets you create versions of your documents, so you can refer to a prior version if necessary while reviewing a workbook for completeness and accuracy.

3. **Number of versions**. If you enable versions in SharePoint, you can also establish the number of versions to retain. This feature helps manage storage space used by large documents with many revisions.

4. **Check out**. Finally, SharePoint supports document check-out. When you check out a workbook, SharePoint locks it for editing until you check it back into the library.

Co-Authoring Using Microsoft Teams

The newest and perhaps best form of collaborating involves using **Microsoft Teams** as the platform for co-authoring. Microsoft created Teams around the idea that the most significant needs of information workers are communication and collaboration. In other words, these capabilities are not an afterthought; instead, Microsoft incorporated them into the fundamental design of Teams.

The Teams platform takes advantage of the functionality provided by SharePoint Online to facilitate co-authoring, but in a more user-friendly environment. More to the point, Microsoft enables co-authoring by default in Teams. Thus, if you open a workbook stored in Teams and other users have the same workbook open, all users can edit the workbook and see others'

changes in real time. Notably, as shown in **Figure 133**, when you open a workbook in Teams, it opens in the **Excel Web App**. However, you can click **Open in Desktop App** if you require access to features not available in the Web App.

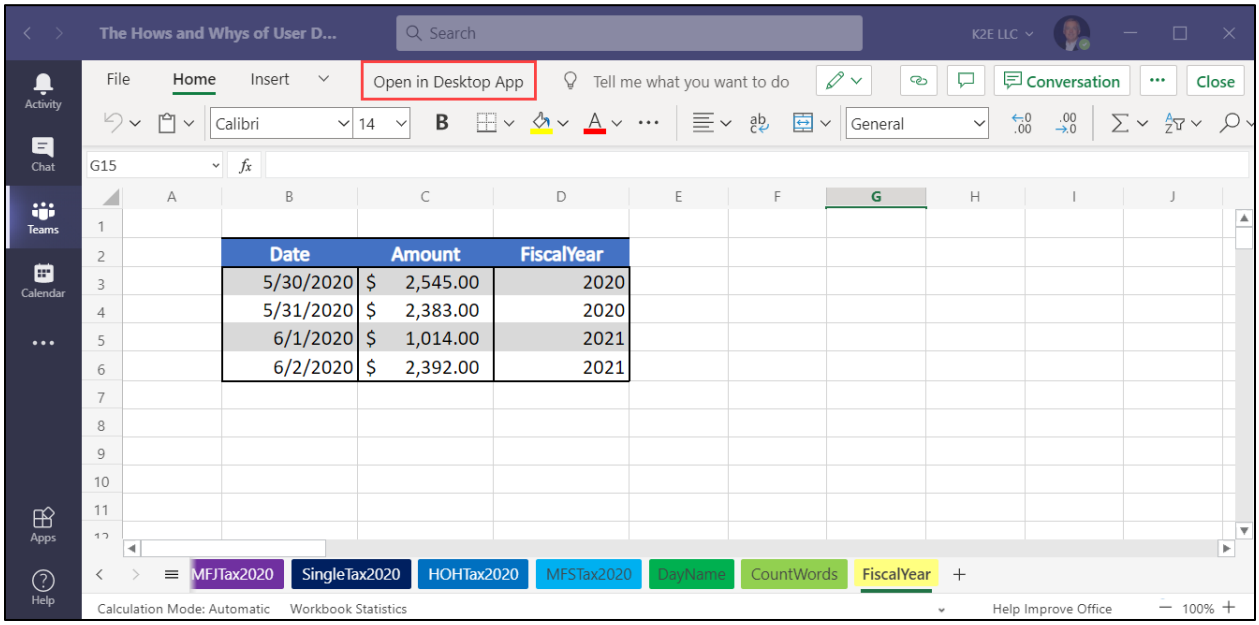


Figure 133 - Opening a Workbook in Teams Using the Excel Web App

As more users take advantage of the Teams platform, expect significant increases in collaboration due to its ease of use.

?

Suppose you must collaborate on a critical Excel document with a team member. What are the relative merits and limitations of using each collaboration method below?

- Sharing the document via email?
- Co-authoring the document via OneDrive or OneDrive for Business?
- Saving the document to SharePoint?
- Using Excel's native Shared Workbook feature?
- Storing the workbook in Teams?

Reporting And Communicating With Excel

Chapter 6

In almost all cases, you will ultimately communicate the results of an Excel project to other stakeholders. The process for doing so might be a "grid-style" report, a chart, or some interactive object such as a chart. This chapter discusses best practices for reporting and communicating your results.

Learning Objectives

Upon completing this chapter, you should be able to:

- List and describe three methods for consolidating data for reporting purposes;
- Create Custom Views in Excel and use them to speed printing processes; and
- Integrate Excel data into Word and PowerPoint documents.

Consolidating Data in Excel

Many Excel users face a specific challenge when consolidating spreadsheet data from multiple divisions, departments, and profit centers. Numerous options exist for this task, including using formulas, the Excel Data Consolidate tool, and creating consolidated PivotTables.

Consolidating Via Formulas

Likely the most common way to consolidate Excel data, formulas offer a straightforward method for combining results from individual spreadsheets into a single spreadsheet that provides totals. Yet, even when using formulas to consolidate data, you can take multiple approaches, including "plus, plus, plus" formulas and SUM functions.

Consolidating Data with "Plus, Plus, Plus" Formulas

Perhaps the simplest of all the approaches is the "plus, plus, plus" formula. With this approach, you build formulas like the one highlighted in **Figure 134** to consolidate your data.

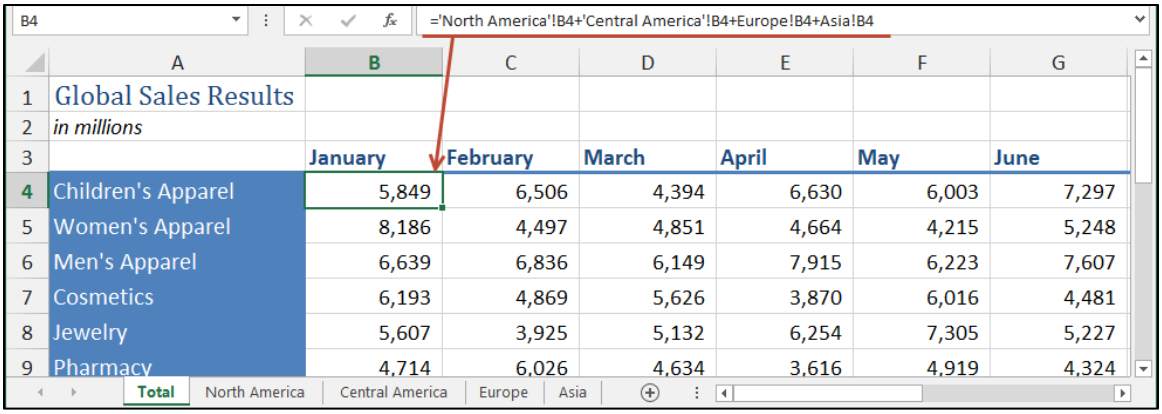


Figure 134 - Sample "Plus, Plus, Plus" Formula

The consolidating formula's primary advantages include its simplicity and transparency, as virtually all Excel users will instantly understand what it accomplishes. Another benefit of this approach is that the formula refers to individual cells. Therefore, if any of the cell's locations change – such as when inserting or deleting columns or rows – the formula updates to reflect the source data's new location.

Using SUM Functions to Consolidate Data

A second approach to consolidating data is to use a SUM function in a formula, as shown in **Figure 135**. As with the previous method, this approach offers simplicity and provides construction speed. More specifically, you can likely build this formula faster than a "plus, plus, plus" formula, particularly when consolidating many data sources.

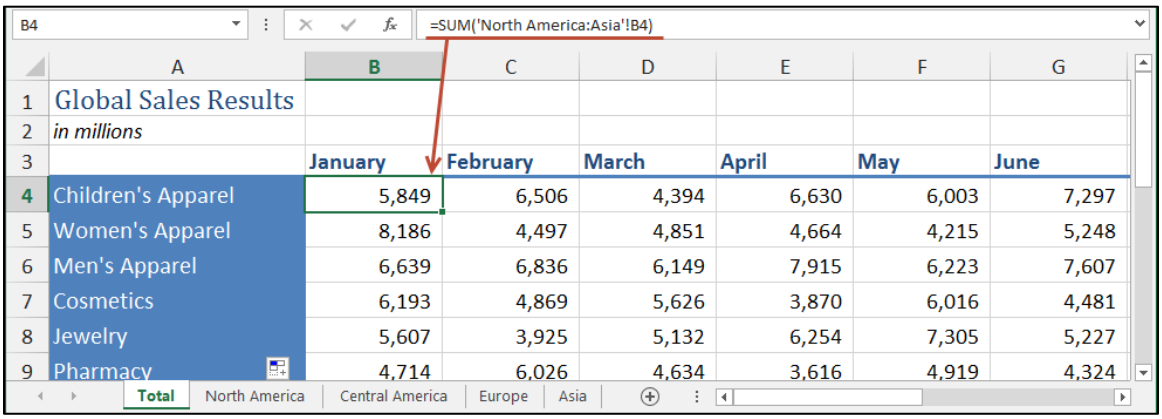


Figure 135 - Using a SUM Function to Consolidate Data

The primary disadvantage of this approach is that it refers to a range of cells, and if the location of one or more of those cells changes, the formula cannot adapt to the new locations. Thus, if you use this approach, you should ensure that the data locations you are consolidating with the SUM function remain fixed.

Creating SUM Formulas Via Wildcard Characters

Using the SUM function to consolidate data, you can speed up building your formulas in many cases by using a wildcard character such as an asterisk (*). For example, instead of typing the formula **=SUM('North America:Asia'!B4)**, you could instead type **=SUM('*'!B4)**. Upon pressing **Enter**, Excel converts your entry to **=SUM('North America:Asia'!B4).**

Using Excel's Data Consolidate Feature

Instead of using formulas to consolidate data, a better approach might be Excel's **Data Consolidate** feature. With Data Consolidate, Excel automates data consolidation. Further, Data Consolidate does not require that you arrange your data symmetrically; even if your data is asymmetrical, it can still accurately consolidate results.

Data Consolidate offers two approaches to consolidating results: Consolidating by Category and Consolidating by Position. If you know your data is and will remain symmetrical, you can use Consolidate by Position; however, if your data is or may become asymmetrical, you should use Consolidate by Category.

Consolidating by Position

To consolidate data residing in the same relative position in all the consolidating ranges, choose **Data Consolidate** from the Ribbon's **Data** tab to open the **Consolidate** dialog box, pictured in **Figure 136**.

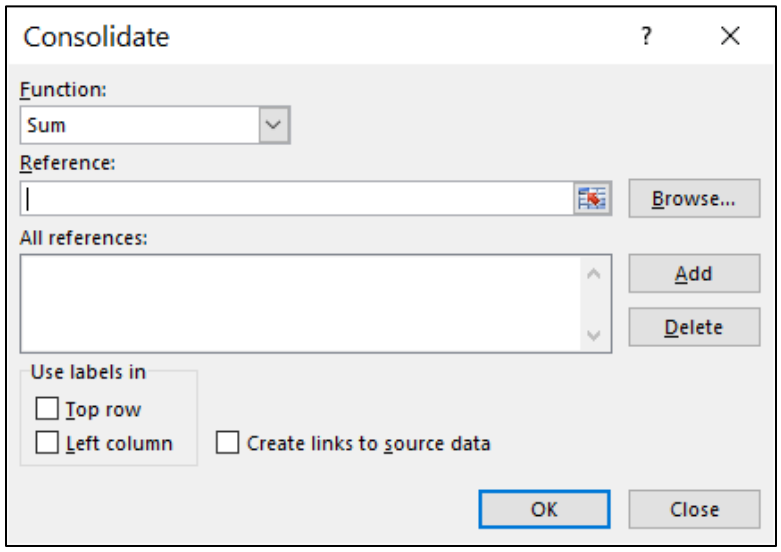


Figure 136 - Consolidate Dialog Box

In the Consolidate dialog box, choose the **Function** you wish to perform. **Sum** is the default, but Data Consolidate can perform ten other functions also.

1. Average
2. Count
3. Count Numbers
4. Max
5. Min
6. Product
7. Standard Deviation
8. Standard Deviation of the Population
9. Variance
10. Variance of the Population

In the **Reference** section of the Consolidate dialog box, enter or select the first range of data you want to include in the consolidation. Do not include column headers or row labels. Upon completing this entry, click **Add**. Then, repeat this process for all other consolidating ranges, including ranges in other workbooks. If you want your consolidation to update in real time as the source data changes, check the box next to "Create links to source data." Once you have entered all items in the Consolidate dialog box, it should resemble that shown in **Figure 137.**

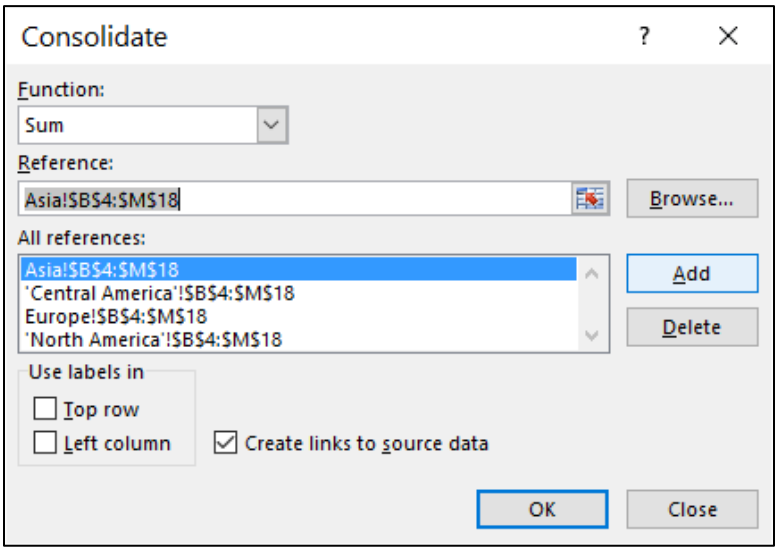


Figure 137 - Completed Consolidated Dialog Box

Click **OK** to complete the process, and Excel inserts the consolidated results into your worksheet at the selected cell's position. When you choose Data Consolidate, Excel inserts rows into the current worksheet and builds links back to the source data. It also automatically inserts SUM functions into the current worksheet to consolidate the data from the linked rows. Finally, it uses Excel's **Grouping** function to collapse the rows so that they are hidden from view but are available for you to drill down and see the details. **Figure 138** illustrates the results of a Consolidation by Position.

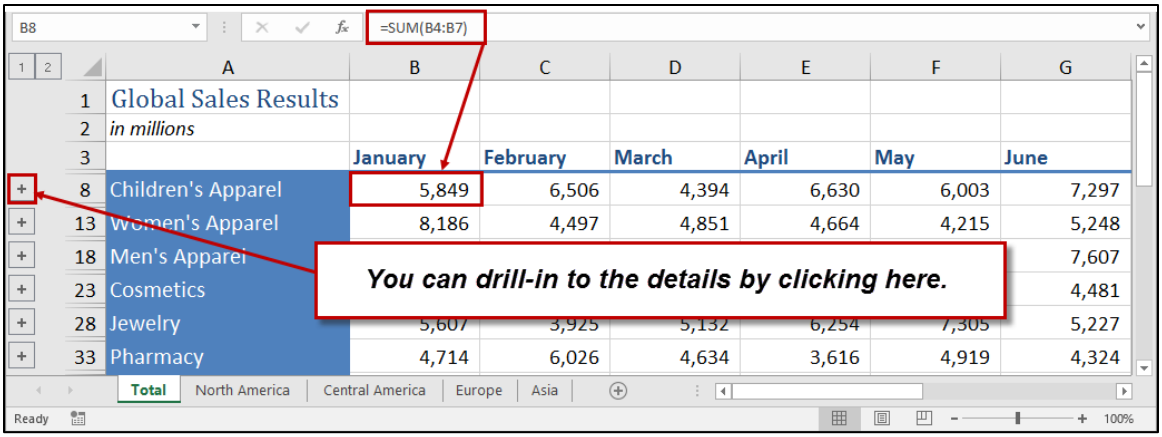


Figure 138 - Completed Consolidate by Position Illustration

Consolidating by Category

If your data is asymmetrical or could become so, you should use **Consolidate by Category** to complete the consolidation task. Consolidate by Category operates much like Consolidate by Position, except it examines the column headers and row labels of your consolidation ranges and creates a consolidating entry for every unique intersection of a column header and a row label. Accordingly, you must include column headers and row labels when entering your consolidating ranges in the Consolidate dialog box.

To demonstrate Consolidating by Category, assume the same data as the previous example. Also, assume that not all regions have identical product lines. In that case, the consolidating data ranges for each region will differ, necessitating consolidating by Category. To complete the consolidation, follow the same steps you used when consolidating by Position, but include the row labels and column headers in the consolidating ranges. In the Consolidate dialog box, check the boxes next to **Use labels in Top Row** and **Left column.** Your Consolidate dialog box should look like that pictured in **Figure 139.**

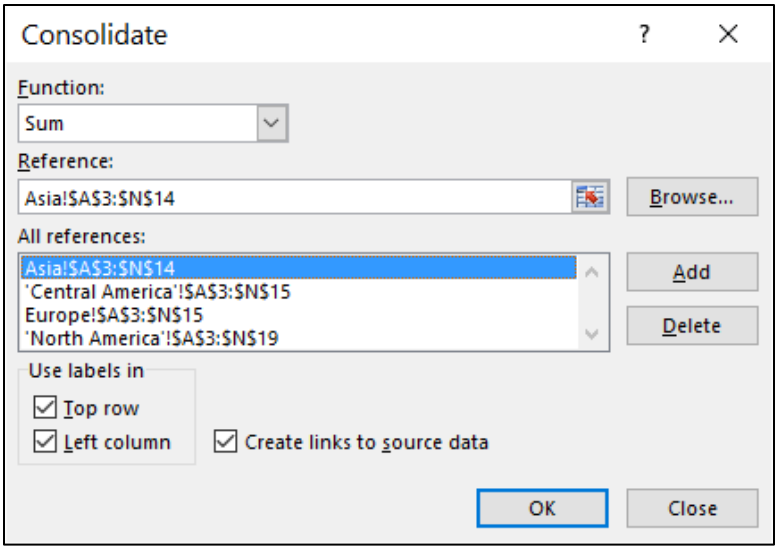


Figure 139 - Consolidating by Category

Notice in Figure 139 that the four consolidating data ranges do not contain the same number of rows. This result is because the consolidating ranges are, in fact, asymmetrical, yet the consolidation produced an accurate report, as demonstrated in **Figure 140**.

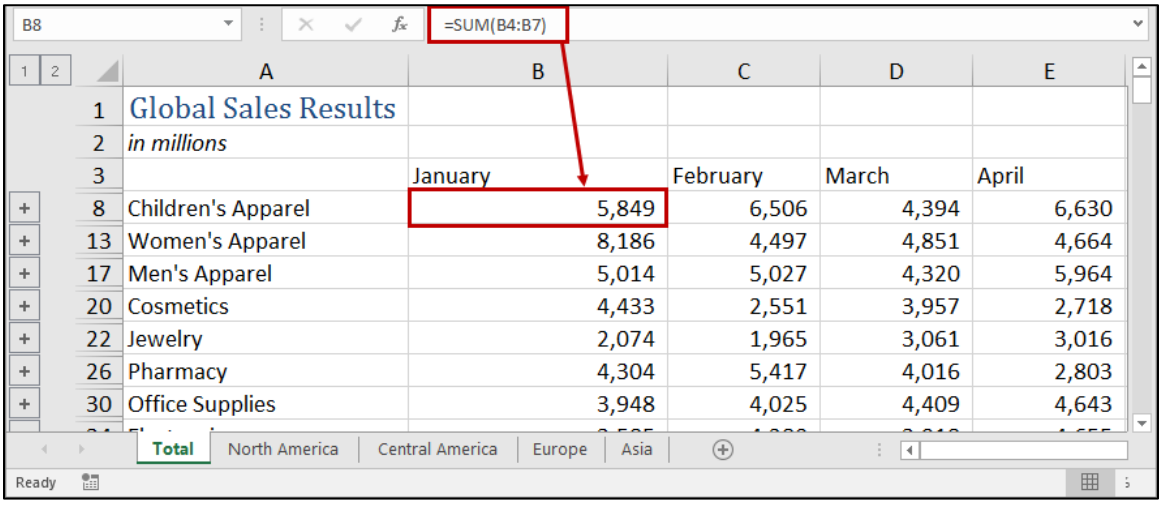


Figure 140 - Completed Consolidate by Category Illustration

Data Consolidate – by Position or Category – represents an Excel best practice because it allows you to summarize multiple data ranges quickly. In addition, though Data Consolidate uses formulas, it constructs them for you, ensuring the integrity of your consolidated results.

Creating Consolidation PivotTables

The last tool to consider when consolidating data – and perhaps best – is **Consolidation PivotTables**. Even for those who do not ordinarily use Excel PivotTables, Consolidation

PivotTables are potent tools for consolidating and summarizing data.[3] PivotTables summarize the data in a computationally correct fashion and present options for grouping data and adding calculations that might prove problematic using other techniques.

To build consolidating PivotTables in Excel, you must customize your user interface to add the **PivotTable and PivotChart Wizard** to the Quick Access Toolbar or the Ribbon. Upon completing this task, you are ready to begin your consolidation.

To build a consolidating PivotTable, click the **PivotTable and PivotChart Wizard** and choose **Multiple consolidation ranges** from the dialog box shown in **Figure 141**. Click **Next** to continue.

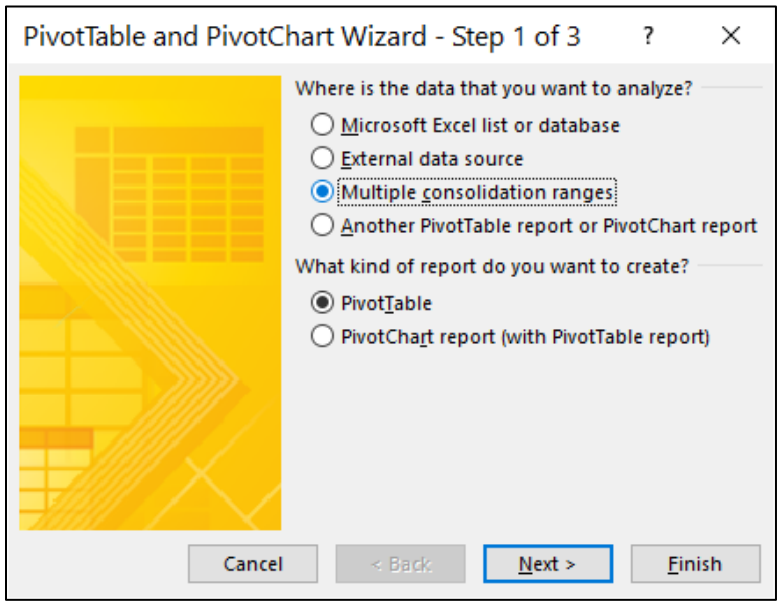


Figure 141 - Step 1 of Creating a Consolidation PivotTable

In the next dialog box, choose **I will create the page fields** as shown in **Figure 142**. **Page fields** in PivotTables are simply filters that act upon the entire report.

[3] A full discussion of PivotTables is beyond the scope of this course. For more information on PivotTables, consider "Excel PivotTables for Accountants." You can find out more about this course at https://www.k2e.com/seminars/excel-pivottables.

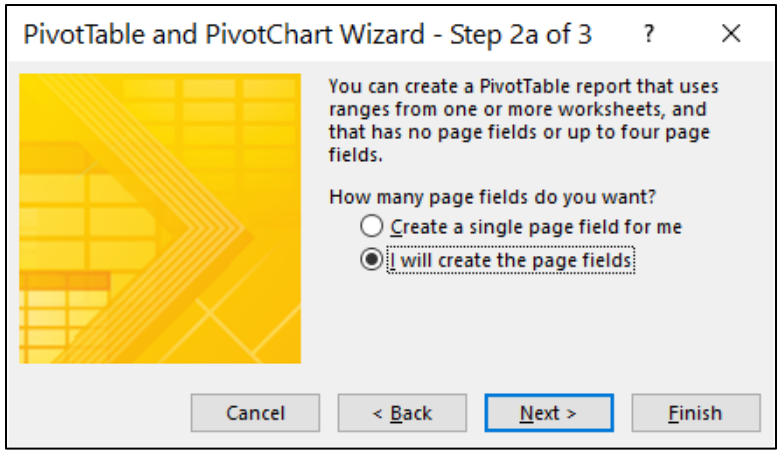


Figure 142 - Step 2a of Creating a Consolidation PivotTable

For each data range you wish to consolidate, enter the range into the dialog box presented in **Figure 143,** click **Add,** and indicate you want a single page field. Then, enter the name for your page field. Repeat this process for each range and click **Next** when you have entered the data for all data ranges. As with Data Consolidation, notice that Consolidation PivotTables do not require symmetrical data ranges.

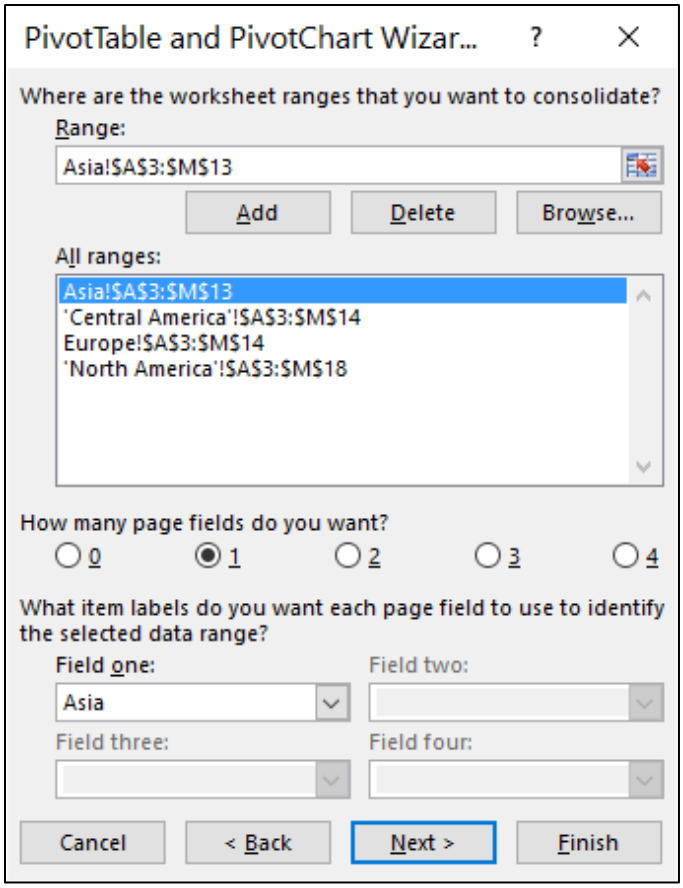


Figure 143 - Entering Details about Consolidating Ranges

Finally, in Step 3 of the PivotTable and PivotChart Wizard, indicate that you want your consolidated PivotTable on a new worksheet and click **Finish**, as shown in **Figure 144**.

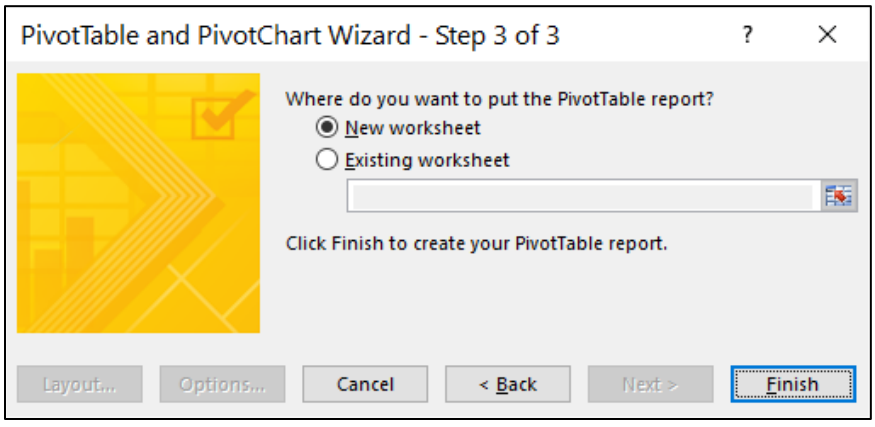


Figure 144 - Completing the Consolidation PivotTable

Your PivotTable will appear on a new worksheet, as shown in **Figure 145**.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Page1	(All)												
2														
3	**Sum of Value**	**Column Labels**												
4	**Row Labels**	**January**	**February**	**March**	**April**	**May**	**June**	**July**	**August**	**September**	**October**	**November**	**December**	**Grand Total**
5	Automotive	3854	4873	2236	4781	5072	4660	3412	4344	4919	3426	5406	2054	49037
6	Children's Apparel	5849	6506	4394	6630	6003	7297	7411	4966	4924	4404	4536	5907	68827
7	Cosmetics	4433	2551	3957	2718	4086	3028	3152	4220	3455	3703	4065	3475	42843
8	Electronics	3585	4280	3918	4655	3060	5151	4876	2868	3735	4299	3687	4838	48952
9	Food	4949	5655	5291	3606	5655	4783	5394	5182	5807	5061	4734	2787	58904
10	Furniture	6523	5910	4788	5530	4380	6086	5213	6705	5757	4741	4813	4573	65019
11	Household Items	5280	6465	3694	5216	4038	6041	4651	5679	4941	4896	4063	6359	61323
12	Jewelry	2074	1965	3061	3016	3141	1519	3597	2651	3815	1835	2760	1317	30751
13	Men's Apparel	5014	5027	4320	5964	4861	6419	5792	2842	3461	5416	5019	5249	59384
14	Office Supplies	3948	4025	4409	4643	3293	2910	3621	4366	4291	5499	4759	2519	48283
15	Pharmacy	4304	5417	4016	2803	4199	3442	5110	4046	4867	6826	6377	4639	56046
16	Photography	5557	6376	4230	5079	5143	6040	5894	5185	4874	5178	4657	7337	65550
17	Sporting Goods	4531	4113	5390	3209	4561	4732	4245	3649	6134	2190	2527	4573	49854
18	Toys & Video Games	4352	5482	6671	6485	5889	4292	6614	3340	4679	3728	7571	5194	64297
19	Women's Apparel	8186	4497	4851	4664	4215	5248	6262	5685	5906	7868	6409	4181	67972
20	**Grand Total**	**72439**	**73142**	**65226**	**68999**	**67596**	**71648**	**75244**	**65728**	**71565**	**69070**	**71383**	**65002**	**837042**

Figure 145 - Completed Consolidation PivotTable

After generating the initial PivotTable, you can format it and use other Excel and PivotTable techniques to develop a more polished report, such as the one in **Figure 146**.

	A	B	C	D	E	F
1	Region	(All)				
2						
3	Sales by Product Line by Region					
4		1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Grand Total
5						
6	**Apparel**					
7	Children's Apparel	16,749	19,930	17,301	14,847	68,827
8	Men's Apparel	14,361	17,244	12,095	15,684	59,384
9	Women's Apparel	17,534	14,127	17,853	18,458	67,972
10	**Apparel Total**	**48,644**	**51,301**	**47,249**	**48,989**	**196,183**
11	**Electronics**					
12	Electronics	11,783	12,866	11,479	12,824	48,952
13	Photography	16,163	16,262	15,953	17,172	65,550
14	Toys & Video Games	16,505	16,666	14,633	16,493	64,297
15	**Electronics Total**	**44,451**	**45,794**	**42,065**	**46,489**	**178,799**
16	**Food**					
17	Food	15,895	14,044	16,383	12,582	58,904
18	**Food Total**	**15,895**	**14,044**	**16,383**	**12,582**	**58,904**
19	**Household**					
20	Furniture	17,221	15,996	17,675	14,127	65,019
21	Household Items	15,439	15,295	15,271	15,318	61,323
22	**Household Total**	**32,660**	**31,291**	**32,946**	**29,445**	**126,342**
23	**Other**					
24	Automotive	10,963	14,513	12,675	10,886	49,037
25	Jewelry	7,100	7,676	10,063	5,912	30,751
26	Office Supplies	12,382	10,846	12,278	12,777	48,283
27	Sporting Goods	14,034	12,502	14,028	9,290	49,854
28	**Other Total**	**44,479**	**45,537**	**49,044**	**38,865**	**177,925**
29	**Pharmacy & Cosmetics**					
30	Cosmetics	10,941	9,832	10,827	11,243	42,843
31	Pharmacy	13,737	10,444	14,023	17,842	56,046
32	**Pharmacy & Cosmetics Total**	**24,678**	**20,276**	**24,850**	**29,085**	**98,889**
33	**Grand Total**	**210,807**	**208,243**	**212,537**	**205,455**	**837,042**

Figure 146 - Formatted Consolidation PivotTable

?	Which consolidation technique discussed in this chapter will you most likely use to consolidate multiple data ranges? Why?

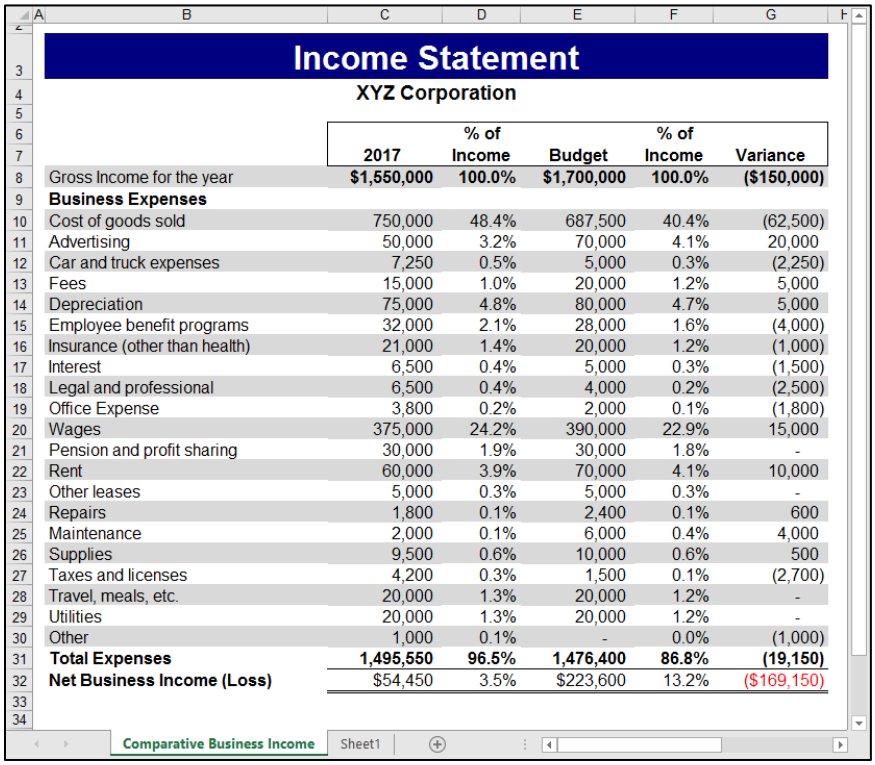
Streamlining Printing Processes with Custom Views

You can create a **Custom View** in Excel to save a series of settings, enabling you to retrieve them quickly. These settings can include hidden rows, hidden columns, and filters. Also, Custom Views can save print settings, including margins, page orientation, headers, and footers. For example, suppose you have an Excel spreadsheet from which you create various reports by hiding and unhiding columns and rows, applying filters, and editing print settings. In that case, you can streamline these processes with Custom Views.

To illustrate, consider the Income Statement pictured in **Figure 147**. Further, suppose that you need the ability to print this report in three different ways:

1. As shown, with columns A through G visible;
2. Without the budget or variance data, hiding columns E through G; and
3. With columns D through G hidden and displaying only the actual dollars in column C.

You can save three different custom views to facilitate switching between the different versions of the report.



Income Statement

XYZ Corporation

	2017	% of Income	Budget	% of Income	Variance
Gross Income for the year	**$1,550,000**	**100.0%**	**$1,700,000**	**100.0%**	**($150,000)**
Business Expenses					
Cost of goods sold	750,000	48.4%	687,500	40.4%	(62,500)
Advertising	50,000	3.2%	70,000	4.1%	20,000
Car and truck expenses	7,250	0.5%	5,000	0.3%	(2,250)
Fees	15,000	1.0%	20,000	1.2%	5,000
Depreciation	75,000	4.8%	80,000	4.7%	5,000
Employee benefit programs	32,000	2.1%	28,000	1.6%	(4,000)
Insurance (other than health)	21,000	1.4%	20,000	1.2%	(1,000)
Interest	6,500	0.4%	5,000	0.3%	(1,500)
Legal and professional	6,500	0.4%	4,000	0.2%	(2,500)
Office Expense	3,800	0.2%	2,000	0.1%	(1,800)
Wages	375,000	24.2%	390,000	22.9%	15,000
Pension and profit sharing	30,000	1.9%	30,000	1.8%	-
Rent	60,000	3.9%	70,000	4.1%	10,000
Other leases	5,000	0.3%	5,000	0.3%	-
Repairs	1,800	0.1%	2,400	0.1%	600
Maintenance	2,000	0.1%	6,000	0.4%	4,000
Supplies	9,500	0.6%	10,000	0.6%	500
Taxes and licenses	4,200	0.3%	1,500	0.1%	(2,700)
Travel, meals, etc.	20,000	1.3%	20,000	1.2%	-
Utilities	20,000	1.3%	20,000	1.2%	-
Other	1,000	0.1%	-	0.0%	(1,000)
Total Expenses	**1,495,550**	**96.5%**	**1,476,400**	**86.8%**	**(19,150)**
Net Business Income (Loss)	$54,450	3.5%	$223,600	13.2%	($169,150)

Comparative Business Income | Sheet1

Figure 147 - Sample Income Statement for Custom Views

To build the Custom View that displays columns A through G, make those columns visible. Next, establish all desired print settings from the **Page Layout** tab of the Ribbon. Then, choose **Custom Views** to open the Custom Views dialog box from the **View** tab of the Ribbon. Next, click **Add** to open the **Add View** dialog box in the Custom Views dialog box pictured in **Figure 148**. Next, in the **Name** box, enter the name for this Custom View and check the boxes next to **Print settings** and **Hidden rows, columns, and filter settings**. Finally, click **OK** to complete the process.

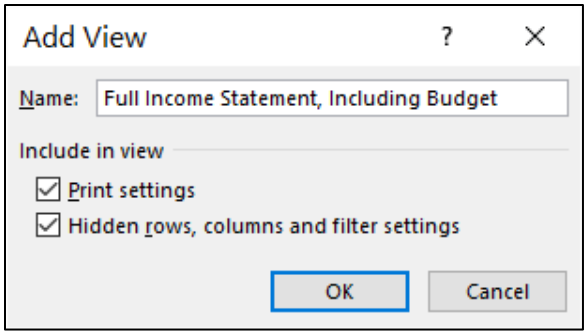


Figure 148 - Adding a Custom View

To create the Custom View that hides the budget and variance data, hide columns E, F, and G in Excel. Then, repeat the abovementioned process to create a Custom View to preserve this setting.

Lastly, to create the Custom View that only displays the dollars associated with the income statement, hide columns D, E, F, and G. Repeat the steps outlined above to create a Custom View to preserve these settings.

To recall a previously saved Custom View, choose **Custom Views from the View tab of the Ribbon**. Excel then lists all the Custom Views in the workbook, as shown in **Figure 149**.

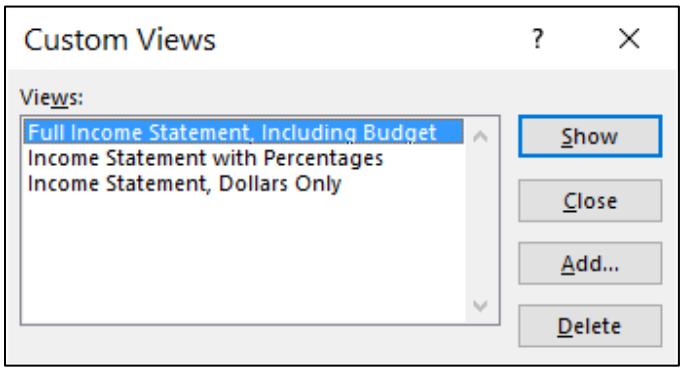


Figure 149 - Selecting a Custom View

Select the **Custom View** you would like to retrieve and click **Show**. Immediately upon doing so, Excel restores the Custom View, including all print settings associated with the Custom View.

Using Custom Views to automate switching the configurations represents an Excel best practice for those who view or print a spreadsheet in different layouts and designs. It eliminates the need to change settings manually each time you need to change the report's layout. Accordingly, you save time and ensure that the desired settings appear in the spreadsheet without exception.

?	How could you incorporate Excel's Custom Views feature to streamline the process of applying filters, hiding rows and columns, and producing reports? Before participating in this course, were you familiar with this feature?

Integrating Excel Data into Word and PowerPoint Documents

Accounting and financial professionals often wish to integrate Excel data like financial statements into Word documents and PowerPoint presentations. Because many techniques exist for achieving this objective, you should first decide your ultimate goal for the integrated data. Specifically, do you need to link the data dynamically to the other application, or is a static copy-and-paste function suitable?

Linking Excel Reports into Word Documents

As a best practice, when you need to link Excel reports into Word documents, you should insert the Excel data as a picture in the Word document. Doing so reduces the probability that a user will change the Word document's Excel-based data purposefully or accidentally.

To link Excel reports into Word documents, copy the data from Excel. Next, choose **Paste Special**, and **Paste Link** as a **Bitmap**, as shown in **Figure 150**.

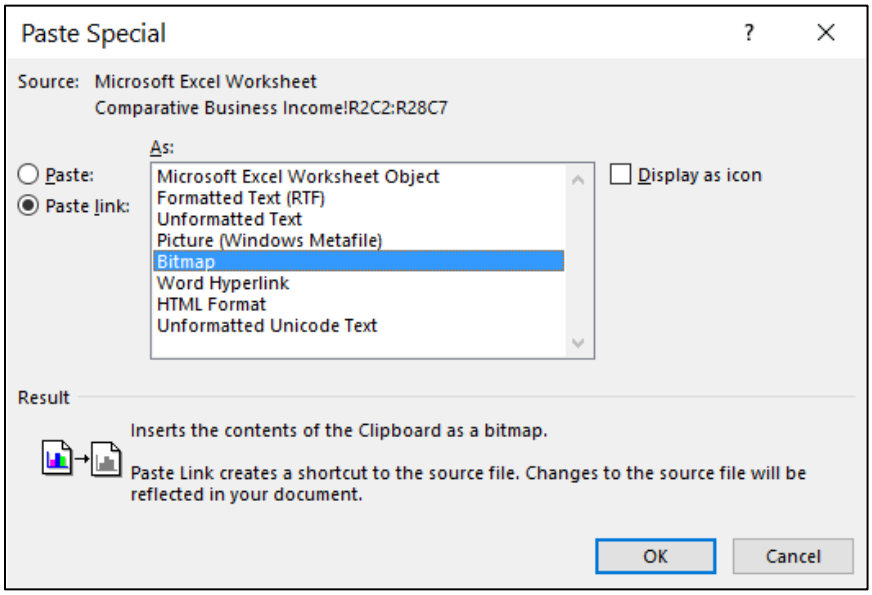


Figure 150 - Pasting into Word as a Linked Bitmap Image

Pasting the data into Word in this fashion creates a graphic image you cannot alter in Word. However, as the source data for the image changes in Excel, the Word document's image updates. Additionally, as you change formats for the Excel data, the formats also update the Word document.

If a static copy of the data is all you require, use the same technique for linking, but select **Paste** instead of **Paste Link** in Figure 150.

Linking Excel Data and Charts to PowerPoint

If you need to link Excel data onto a PowerPoint slide, doing so will be similar to the process described above for linking into a Word document. First, copy the data you wish to link into PowerPoint. Then, navigate to the desired PowerPoint slide and choose **Paste Special**, **Paste link** as a **Microsoft Excel Worksheet Object**, as shown in **Figure 151**.

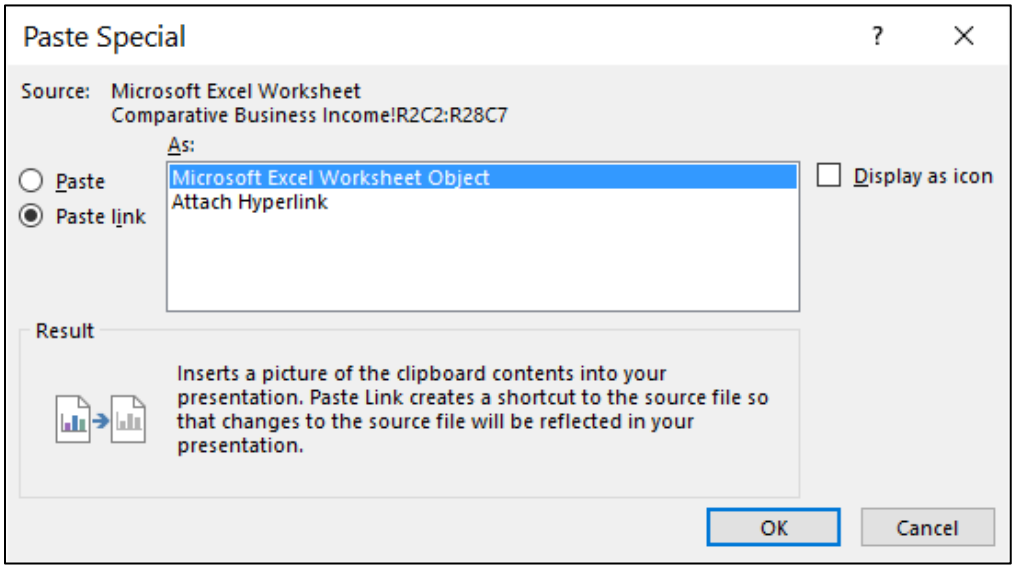


Figure 151 - Linking Excel Data onto a PowerPoint Slide

This action will link the data onto the slide and maintain the data's appearance as it exists in Excel. If you do not need to link the data to PowerPoint, you can choose one of the **Paste** options instead.

Notwithstanding the above, if you wish to link Excel data into a PowerPoint chart object, doing so is slightly different. Rather than linking the *Excel data*, you will need to build the chart in Excel and then link the *chart* into PowerPoint instead. To complete this process, begin by creating the chart in Excel. Next, copy the chart and navigate to the PowerPoint slide to which you want to link the chart. Next, choose **Paste Special**, **Paste Link** as a **Microsoft Excel Chart Object**. Finally, click **OK** to complete the process, as shown in **Figure 152**.

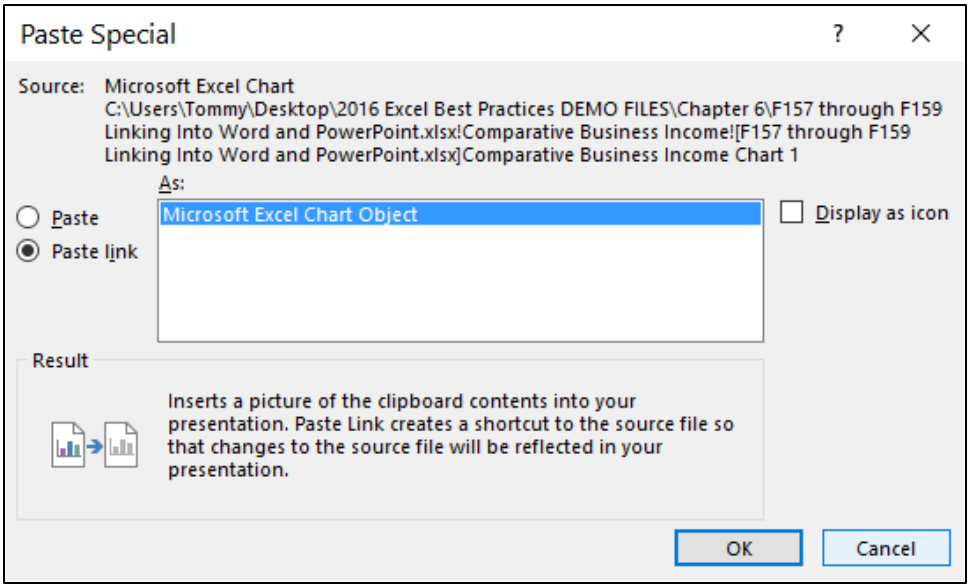


Figure 152 - Linking a Chart from Excel into PowerPoint

Upon completing this process, when a data change in Excel causes the chart to change, the linked version of the chart in PowerPoint will also update. Following these best practices will ensure that your Excel data and charts appear as desired in Word and PowerPoint.