Learning Google Sheets: A Step-by-Step Guide to Calculating the Last Business Day of a Month

Authored by

Mohammed looti

November 12, 2025

Introduction to Calculating the Last Business Day in Google Sheets

Pinpointing the precise date of the last business day of any given month is a critical requirement across numerous professional disciplines, including detailed financial planning, rigorous project management, and accurate payroll processing. By conventional definition, a business day excludes weekends (Saturday and Sunday) and any officially recognized public holidays. While determining the final calendar day of the month is trivial, automatically calculating the actual last working day necessitates the strategic integration of several powerful date functions within Google Sheets. This sophisticated, automated approach is essential for maintaining accuracy, as it inherently accounts for dates that fall on non-working days.

Simple date arithmetic often proves insufficient because it fails to adjust when the month's final calendar day coincides with a non-working day, such as a Saturday or Sunday. For example, if a reporting period concludes on a Sunday, the legitimate last business day must, by definition, be the preceding Friday. Attempting to manually verify and adjust these dates across large, multi-year datasets is exceptionally time-consuming and dangerously susceptible to human error. Fortunately, Google Sheets provides specialized tools, primarily the **EOMONTH** and **WORKDAY** functions, which can be elegantly combined into a single, reliable formula capable of resolving this common chronological challenge with high efficiency.

The robust methodology we will explore relies on a two-step calculation process. It first uses the **EOMONTH** function to locate the first day immediately succeeding the end of the target month. Subsequently, it employs the **WORKDAY** function to step backward exactly one business day from that point. This systematic reversal is the most dependable method to guarantee that the resulting output date is consistently, and without exception, the final operational working day of the specified period. A thorough understanding of the mechanics underlying this combined formula is fundamental for anyone responsible for managing time-sensitive deadlines and data within the spreadsheet environment.

The Core Formula for Finding the Last Business Day

To precisely calculate the last business day of the month for any specified starting date, the following highly reliable formula must be deployed. This single line of code masterfully integrates the process of identifying the month's end and then applying standard business constraints. Although this example references the starting date provided in cell **A2**, the formula structure is entirely adaptable and can reference any cell containing a valid date entry.

=WORKDAY(EOMONTH(A2, 0)+1, -1)

The efficiency of this particular formula structure stems from its calculated sequence. It initiates the

process by leveraging the EOMONTH function to establish the reliable end boundary of the month. Immediately following this, the powerful WORKDAY function is employed to execute the necessary chronological adjustment. This carefully engineered combination ensures that the final output date will always represent the final working day of the month corresponding to the date found in cell **A2**, automatically excluding standard weekend days (Saturdays and Sundays).

It is essential to appreciate that despite its concise appearance, this formula encapsulates complex chronological logic. The calculation mechanism first determines the final calendar day of the month in question. Crucially, it then advances this date to the first day of the succeeding month before issuing the instruction to retreat exactly one business day. This systematic advancement and subsequent retreat guarantee success: if the last calendar day of the month happens to be a weekend, the formula seamlessly navigates this conflict and correctly returns the preceding Friday, thereby consistently meeting the strict definition of the last business day.

Step-by-Step Implementation and Verification

To fully grasp the practical utility of this essential date calculation, let us walk through a typical implementation scenario. Imagine we have a column populated with various dates, and our primary objective is to accurately ascertain the last business day for the respective month associated with each date. For this illustration, we assume the sample data is contained within Column A of our Google Sheets spreadsheet, as depicted below:

	A	B	C
1	**Date**		
2	1/25/2023		
3	2/27/2023		
4	3/14/2023		
5	4/1/2023		
6	5/28/2023		
7	6/15/2023		
8	7/22/2023		
9	8/28/2023		
10	9/4/2023		
11	10/30/2023		
12	11/16/2023		
13	12/25/2023		
14			
15			
16			

Our workflow dictates populating Column B with the calculated last business day corresponding to every date listed in Column A. We initiate this process by carefully inputting the core formula into cell **B2**, which aligns with the first date in our dataset (cell **A2**, 1/25/2023). This crucial initialization step executes the calculation for January 2023. Following entry, we utilize the powerful fill handle feature inherent in Google Sheets: by selecting cell **B2** and dragging the formula downward through the remaining rows in Column B, the sheet automatically adjusts the cell reference (A2 becomes A3, A4, and so forth), enabling instantaneous calculation across the entire range of data.

=WORKDAY(EOMONTH(A2, 0)+1, -1)

Upon successful application of the formula across all required rows, the completed spreadsheet clearly presents the calculated last business days adjacent to the original date inputs:

B2 =WORKDAY(EOMONTH(A2, 0)+1, -1)

	A	B	C
1	**Date**	**Last Business Day of Month**	
2	1/25/2023	1/31/2023	
3	2/27/2023	2/28/2023	
4	3/14/2023	3/31/2023	
5	4/1/2023	4/28/2023	
6	5/28/2023	5/31/2023	
7	6/15/2023	6/30/2023	
8	7/22/2023	7/31/2023	
9	8/28/2023	8/31/2023	
10	9/4/2023	9/29/2023	
11	10/30/2023	10/31/2023	
12	11/16/2023	11/30/2023	
13	12/25/2023	12/29/2023	
14			
15			
16			

We can verify the accuracy by examining the first entry: the date **1/25/2023** falls within January 2023. Consulting a standard calendar for that month confirms that the last calendar day, Tuesday the 31st, was indeed a working day. Therefore, the formula is expected to return 1/31/2023.

As anticipated, cell B2 correctly displays **1/31/2023**. This validation process confirms that the underlying methodology operates flawlessly, regardless of whether the last calendar day of the month is a standard weekday or if it necessitates the logical backtracking mechanism to correctly identify the preceding Friday as the true final business day.

Deconstructing the Formula: EOMONTH and WORKDAY Functions

To fully appreciate the efficacy and reliability of this technique, it is imperative to dissect and understand the intricate interaction between the nested functions. Let us recall the standard formula structure used to determine the last business day based on a date in cell **A2**:

=WORKDAY(EOMONTH(A2, 0)+1, -1)

The computation begins internally with the EOMONTH function. Its syntax is defined as `EOMONTH(start_date, months)`. By assigning the `months` argument a value of **0**, the function reliably returns the last calendar day of the month specified by the `start_date` (A2). For instance, using 1/25/2023 as the start date, `EOMONTH(A2, 0)` yields 1/31/2023. This outcome is crucial because it precisely establishes the boundary, or end point, of the period we are currently analyzing.

The subsequent logical step involves the addition of **+1** to the result generated by **EOMONTH**. This seemingly minor arithmetic operation represents the key pivot point that enables the entire formula to function correctly. By incrementing the last day of the current month by one day, the calculation effectively transitions to the **first calendar day of the subsequent month**. Continuing our earlier example, adding one day to 1/31/2023 results in 2/1/2023. This calculated date--the first day of the next period--is systematically passed forward to serve as the `start_date` argument for the encompassing **WORKDAY** function.

Finally, the outer WORKDAY function takes command, utilizing the syntax `WORKDAY(start_date, num_days)`. We feed it the calculated first day of the next month as its starting point. Critically, we set the `num_days` argument to **-1**. This negative one instruction directs the **WORKDAY** function to search backward for exactly one business day from the provided start date. Because the starting point is the 1st of the next month, counting backward one business day automatically and accurately locates the last business day of the previous month, successfully bypassing any non-working days (weekends) that may have occurred at the end of the preceding month.

Advanced Usage: Incorporating Custom Holidays and Weekends

While the standard **WORKDAY** function is highly effective for environments operating on a typical Monday-through-Friday schedule, many organizations must account for specific, observed holidays

or alternative weekend arrangements. For these more intricate scenarios, Google Sheets provides the more versatile **WORKDAY.INTL** function. This function shares the core logic of **WORKDAY** but introduces optional arguments that allow users to define custom weekend patterns and specify a dynamic input range for company-observed holidays.

The syntax for this advanced function is `WORKDAY.INTL(start_date, num_days, , )`. To modify our robust original formula to incorporate a list of internal holidays--for example, if they are stored in a named range such as **Holidays!A1:A20**--we only need to introduce the two additional optional parameters.

=WORKDAY.INTL(EOMONTH(A2, 0)+1, -1, 1, Holidays!A1:A20)

In this advanced application, the argument **1** is used to explicitly specify the standard weekend structure (Saturday and Sunday, where 1 means Saturday/Sunday). The final argument, `Holidays!A1:A20`, is a direct reference to a designated column or range containing the dates of all non-standard, observed company holidays. By incorporating this dedicated range, the **WORKDAY.INTL** function achieves the necessary intelligence to skip these specific dates during its calculation, ensuring that the resulting last business day is truly a day when the organization is fully operational. This level of meticulous detail is absolutely paramount for enterprises that depend on precise date calculations for regulatory compliance and official financial reporting.

Conclusion and Best Practices

The nested methodology leveraging **EOMONTH** and WORKDAY functions provides the definitive, scalable, and reliable solution for automatically pinpointing the last working day of any month within Google Sheets. The strategy of calculating the first day of the subsequent month and then stepping backward one business day guarantees that the result consistently and accurately handles all standard weekend conflicts and non-working periods.

When applying these formulas, a critical best practice involves ensuring that all input cells (such as **A2**) are consistently and correctly formatted as dates. While Google Sheets generally manages date conversions well, inconsistent formatting can lead to unpredictable errors, especially since these functions rely on the underlying serial date values for calculation. Furthermore, for any scenario requiring the exclusion of organizational or localized holidays, transitioning to the **WORKDAY.INTL** function is highly recommended to ensure compliance with specific operational calendars.

Mastering this specific formula represents a significant step toward maximizing efficiency in managing time-series data. It completely bypasses the need for manual, error-prone calendar checks and establishes a robust, scalable solution for handling large, dynamic datasets where

critical financial deadlines, project milestones, or reporting periods are intrinsically tied to the last business day of the month.

Additional Resources for Google Sheets Date Functions

For users seeking to further expand their knowledge and utilization of date and time calculations within Google Sheets, the following related resources offer excellent avenues for exploration:

How to Calculate the Number of Working Days Between Two Dates.
Using the NETWORKDAYS Function for Project Timelines.
Advanced Filtering Techniques Based on Date Ranges.