

How to Find the Last Business Day of the Month in Google Sheets

Authored by
mohammed loot

January 9, 2026

RECOMMENDED CITATION

mohammed loot (2026). *How to Find the Last Business Day of the Month in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125170>

The Challenge of Date Calculations in Spreadsheets

Calculating the exact last business day of a given month is a common requirement in financial reporting, project management, and payroll processing. While one could employ complex nested IF function logic combined with the WEEKDAY function to check for weekends, Google Sheets provides a far more streamlined solution. By combining the power of the **EOMONTH** and **WORKDAY** functions, we can create a simple, elegant formula that reliably identifies the required date.

This tutorial will guide you through the professional method for identifying the final working day of the month, ensuring that you can accurately manage time-sensitive data within your spreadsheets. We will detail the specific formula structure and then provide a practical, step-by-step example demonstrating its application across a column of dates.

Prerequisites: Understanding Key Google Sheets Functions

Before diving into the solution, it is essential to understand the roles played by the two primary functions we will utilize: **EOMONTH** and **WORKDAY**. These functions are designed specifically for handling date arithmetic, especially when standard weekdays or working periods are involved. Mastering these components ensures you can adapt the formula to various scenarios.

The EOMONTH function (End of Month) is crucial as it calculates the date that is a specified number of months before or after another date. Its syntax is straightforward: `EOMONTH(start_date, months)`. By setting the `months` argument to 0, the function efficiently returns the very last calendar day of the month corresponding to the `start_date`, saving us from manually determining 30, 31, 28, or 29 days.

The WORKDAY function is equally vital. This function calculates the date after a specified number of working days, excluding weekends (Saturday and Sunday). Its syntax is `WORKDAY(start_date, num_days, [holidays])`. Critically, **WORKDAY** can also move backward in time if the `num_days` argument is negative. This backward calculation capability is what allows us to reliably pinpoint the final working day of the target month.

The Core Formula for Finding the Last Business Day

To accurately determine the final working day, we combine the previously discussed functions into a concise and powerful formula. This method avoids the complexity of manual weekend checks and holiday exceptions (although holiday exceptions can be added if needed, as discussed later).

You can use the following refined formula to find the **last business day** of the month for any given date reference in Google Sheets:

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

This particular construction, when referencing cell **A2** (which contains the starting date), will return the calculated **last business day** of that specific month. The logic behind adding **+1** and subsequently subtracting **-1** business day ensures absolute accuracy, even when the last calendar day of the month falls on a non-working day.

We will now walk through a practical implementation to illustrate how this formula operates within a dynamic dataset, providing clarity on its use in real-world scenarios.

Example: How to Find Last Business Day of Month in Google Sheets

Consider a scenario where you are managing project deadlines or payroll submissions, and you have a column listing various dates throughout the year. Your goal is to associate each entry with the **last business day** of the month it belongs to.

Suppose we have the following column of dates located in column A of our Google Sheets spreadsheet:

	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 objective is to populate column B with the corresponding final working day for the month

associated with each date listed in column A. This is a common requirement for auditing or scheduling purposes where absolute cutoff dates are necessary.

To initiate this process, we will input the required formula into cell **B2**. This calculation will specifically find the last business day of the month corresponding to the date found in cell **A2** (January 25, 2023):

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

Once entered, the formula calculates the initial result. The next step involves applying this calculation efficiently to the rest of the dataset. Instead of manually entering the formula for every row, we leverage Google Sheets' functionality to automate the process.

Applying the Formula Across the Dataset

After successfully generating the result in cell B2, we utilize the standard spreadsheet method of formula propagation. We click on the bottom-right corner of cell B2--known as the fill handle--and drag this formula down through the remaining cells in column B. This action automatically adjusts the cell reference (A2 changes to A3, A4, and so on) for each subsequent calculation.

After clicking and dragging the formula down, column B will be populated with the calculated results, presenting a clear summary of the final working days for each respective month:

B2 $\text{=WORKDAY}(\text{EOMONTH}(\text{A2}, 0)+1, -1)$

	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			

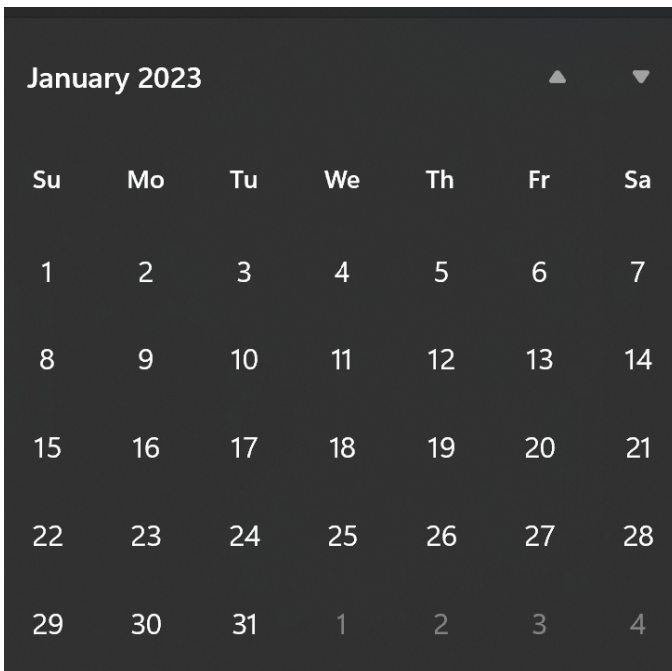
As illustrated in the resulting table, column B now accurately displays the **last business day** for the month that the date in column A belongs to. This streamlined process demonstrates the efficiency of using combined functions for complex date manipulations.

Validation and Verification of Results

It is important to verify that the generated results align with actual calendar dates, particularly when dealing with critical payroll or financial data. Let's examine the first entry in our dataset for validation purposes.

The initial date in the **Date** column of **1/25/2023** belongs to the month and year of January 2023. We need to confirm that the calculated result in cell B2 (which should be 1/31/2023) is indeed the final working day of that month, excluding weekends.

If we consult a standard calendar for January 2023, we observe the following structure near the end of the month. January 28th and 29th fall on a Saturday and Sunday, respectively. Consequently, the final day of the month, Tuesday, January 31st, becomes the last available business day:



January 2023						
Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Based on this calendar verification, we confirm that our formula correctly returns **1/31/2023**. Thus, our formula accurately calculates the date.

Deconstructing the Formula: How WORKDAY and EOMONTH Interact

Understanding the internal mechanism of the formula is key to adapting it for future use. Recall the formula applied to find the last business day for the date in cell **A2**:

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

The magic occurs through nested functionality, processed by Google Sheets from the innermost parentheses outward. The formula execution can be broken down into three distinct, logical steps:

Identifying the Next Month's Start: The internal component, `EOMONTH(A2, 0)`, first determines the last calendar day of the month corresponding to the date in A2. By adding `+1` to this result, the formula effectively shifts the date to the **first calendar day of the subsequent month**. This creates a reliable starting point for the final calculation step.

Initiating Backward Calculation: The outer WORKDAY function then takes over, using this first day of the next month as its `start_date`.

Final Business Day Retrieval: Crucially, the `num_days` argument is set to `-1`. This instructs the WORKDAY function to calculate the date corresponding to one business day prior to the starting point. Since the starting point is the first day of the next month, moving backward one business day infallibly lands us on the last business day of the **current** month, skipping any intervening

weekends.

This sequential logic ensures that the calculation is robust, highly efficient, and consistently accurate across all date values in the column.

Expanding the Solution: Incorporating Holidays

While the basic `WORKDAY(EOMONTH(...))` formula successfully handles standard weekends, many professional environments require the exclusion of official public holidays as well. The [WORKDAY function](#) is designed to accommodate this through its optional third argument.

To include holiday exceptions, you must first create a separate range or column within your spreadsheet containing a list of all official holiday dates (e.g., cell range D2:D20). You then adjust the core formula slightly to reference this list:

```
=WORKDAY(EOMONTH(A2, 0)+1, -1, D2:D20)
```

By specifying the holiday range, the function will not only skip Saturdays and Sundays but also skip any date listed in that range, thereby ensuring the returned date is the true final working business day, even if a holiday falls on the 30th or 31st of the month.

Related Google Sheets Tutorials

Mastering date manipulation in spreadsheets opens up numerous possibilities for advanced analysis and reporting. The ability to precisely manage dates relative to working periods is foundational. For those seeking to further enhance their proficiency in [Google Sheets](#), exploring related functions can prove highly beneficial.

The following tutorials explain how to perform other common tasks in Google Sheets:

How to calculate the number of working days between two dates.

Using the **NETWORKDAYS** function for holiday-inclusive calculations.

Generating dynamic date ranges based on month and year inputs.