

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

Authored by
stats writer

January 25, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Find the Last Business Day of the Month in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127591>

Google Sheets: Find Last Business Day of Month

Google Sheets stands out as an exceptionally powerful and versatile spreadsheet tool, essential for modern data management, complex analysis, and collaborative efforts. Its extensive suite of built-in functions allows users to manipulate various data types, with date and time manipulation being particularly crucial for financial modeling, project scheduling, and payroll processing. Among the most common challenges faced by analysts and operational planners is the accurate calculation of specific dates relative to month-ends, especially when accounting for non-working days.

A frequently needed calculation is determining the last business day of any given month. This date is vital for numerous corporate activities, including invoicing cycles, financial reporting deadlines, and inventory cut-offs, as these events typically cannot occur on weekends or observed holidays. Manually calculating this date across hundreds of entries is highly inefficient and prone to error. Fortunately, Google Sheets offers a streamlined, dynamic solution leveraging powerful date functions, ensuring accuracy and efficiency in your operational planning.

Leveraging Date Functions in Google Sheets

The ability to automatically pinpoint the last weekday of a month is achieved primarily through the coordinated use of two specific Google Sheets functions: EOMONTH (End of Month) and WORKDAY. By combining these functions in a specific logical sequence, we can instruct the spreadsheet to identify the end of the calendar month and then step backward one working day. This methodology eliminates the need for complex nested IF statements or manual calendar checks, providing an elegant and robust formula that updates dynamically as the underlying source data changes.

Understanding the underlying logic is key to mastering this technique. We are essentially defining a known endpoint (the start of the **next** month) and then utilizing a function designed to navigate working days in reverse. This approach is highly reliable because the definition of a "business day" (typically Monday through Friday) is inherently standardized within the WORKDAY function, though it can be customized for specific scenarios, as we will explore later in the advanced considerations section.

Understanding the Concept of a Business Day

Before diving into the syntax, it is essential to clearly define what a business day entails within the context of a spreadsheet environment. Generally, a business day, or working day, is defined as any day that is not a weekend day (Saturday or Sunday) and often excludes nationally recognized holidays. For standard calculations in Google Sheets using the basic WORKDAY function, the

definition strictly excludes only Saturdays and Sundays.

This distinction is crucial because the goal of finding the "last business day" is not merely to find the last calendar day of the month, but to find the last day where standard business operations would occur. For instance, if a month ends on a Sunday, the last business day would be the preceding Friday. If it ends on a Friday, that Friday is the last business day. The formula must be intelligent enough to handle these automatic shifts, regardless of whether the month has 30 or 31 days, or whether it is a leap year.

For organizations that observe specific holidays (e.g., Christmas, Thanksgiving), the standard WORKDAY function might require an extension, utilizing its optional third argument, or by switching to the more advanced **WORKDAY.INTL** function. However, for the foundational, most common use case--simply avoiding weekends--the core formula we introduce is perfectly suited and highly efficient.

The Core Formula: Identifying the Last Business Day of the Month

The powerful calculation needed to determine the last business day relies on a combination of functions nested within one another. This nested approach ensures that the calculation is executed sequentially, deriving the necessary inputs for the final function call.

The complete formula required to find the last business day of the month associated with a date stored in cell **A2** is presented below:

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

This compact formula efficiently encapsulates the three primary logical steps required for the calculation: defining the month, moving to the boundary, and calculating backward. It is a highly optimized string that analysts frequently rely upon for various date-related reporting tasks within the spreadsheet tool environment. The result of this specific formula will always return a valid workday, excluding any Saturday or Sunday that might fall on the month end.

Step-by-Step Breakdown of the Formula Components

To truly appreciate the elegance of this solution, we must dissect the formula piece by piece, starting from the innermost function and working our way outward.

EOMONTH(A2, 0): The EOMONTH function is designed to return the last day of a month that is a specified number of months away from a starting date. By setting the second argument (the offset) to **0**, the function simply returns the last calendar day of the month that the date in cell A2 belongs to. If A2 contains 1/15/2023, this function returns 1/31/2023. This establishes the absolute calendar

boundary.

EOMONTH(A2, 0) + 1: By adding **1** to the result of EOMONTH, we shift the date reference exactly one day into the future. Continuing the previous example, if EOMONTH returned 1/31/2023, adding 1 results in 2/1/2023. This is the crucial logical pivot point: we are establishing the **first day of the next month** as our starting point for the final calculation.

WORKDAY(Start_Date, -1): The WORKDAY function calculates a future or past date that is a specific number of working days away from a start date. By using the output of step 2 (the first day of the next month) as the **Start_Date** and providing **-1** as the number of working days, we instruct the function to count backward exactly one working day. Counting backward one workday from 2/1/2023 will necessarily land us on the last business day of January 2023, bypassing any weekend days that might have occurred at the end of January.

This combination ensures that regardless of where the calendar month ends, the final output will always be the last non-weekend day of that month. This mechanism is highly robust and avoids complex conditional logic that would otherwise be necessary to check if the end of the month falls on a Saturday or Sunday.

Practical Application: A Detailed Google Sheets Example

To demonstrate the functionality of this sophisticated formula, let us consider a scenario where we have a dataset containing various dates, and we require the corresponding last business day for reporting purposes.

Suppose we have the following column of operational or transactional dates in column A of our Google Sheets worksheet:

	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 calculated last business day corresponding to each date in Column A. This process is straightforward and relies on entering the formula once and then efficiently propagating it down the column.

We begin by placing our formula into cell **B2**, referencing the date in **A2**:

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

Once the formula is entered and confirmed in cell B2, Google Sheets will instantly calculate the required date. To apply this logic to the entire dataset, we simply use the autofill handle (the small square box in the bottom-right corner of cell B2) and drag the formula down the length of the data in Column A. This action automatically adjusts the cell reference (A2 becomes A3, A4, and so on) for each subsequent row, calculating the last business day for every listed date.

B2 ∇ | fx =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			

As depicted in the resultant image, Column B now holds the precise last business day for the respective month of the date in Column A. This automated calculation saves substantial time when managing large datasets requiring periodic scheduling.

Analyzing and Verifying the Results

It is crucial to verify the output of complex date functions to ensure the formula is performing as intended, especially when dealing with month transitions. Let's examine the first entry in our example: the date **1/25/2023**.

This date belongs to January 2023. Using a standard calendar reference, we must determine the last day of that month and check if it is a weekend. January 2023 ended on Tuesday, the 31st. Since Tuesday is a standard business day, the calculation should return January 31, 2023.



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

As confirmed by the output in cell B2, the formula correctly returns **1/31/2023**. Now, consider a case where the month ends on a Saturday, for instance, September 2023. September 30, 2023, was a Saturday. The formula calculates the first day of the next month (October 1st) and counts back one workday. Since October 1st was a Sunday, counting back one workday lands on Friday, September 29, 2023. The reliability of this formula lies in its ability to dynamically handle these weekend shifts without manual intervention.

This detailed verification process confirms that the nested use of EOMONTH and WORKDAY provides an accurate and validated method for operational date calculation within the spreadsheet tool environment.

Advanced Considerations: Excluding Holidays and Custom Weekends

While the basic formula is highly effective for excluding standard Saturday and Sunday weekends, many businesses observe statutory or company-specific holidays that also need to be excluded from the definition of a business day. For scenarios where holidays must be factored in, we must transition from the basic WORKDAY function to its international counterpart: **WORKDAY.INTL**.

The **WORKDAY.INTL** function offers greater flexibility by allowing the user to specify custom weekend parameters (e.g., Friday/Saturday weekends) and, critically, accept a list of holiday dates. To incorporate holidays, you would typically list all non-working holidays in a dedicated range within your spreadsheet (e.g., F1:F20) and include that range as the fourth argument in the WORKDAY.INTL function.

The advanced formula structure, incorporating a holiday range called **Holidays**, would look like this:

=WORKDAY.INTL(EOMONTH(A2, 0)+1, -1, 1, Holidays)

In this advanced version, the argument **1** specifies the standard weekend pattern (Saturday/Sunday). By employing **WORKDAY.INTL**, the system checks the starting point (the first day of the next month) and counts back one business day, ensuring it skips Saturdays, Sundays, and any dates explicitly listed in the designated holiday range. This provides the highest level of accuracy for complex financial or manufacturing schedules.

Conclusion: Streamlining Financial and Operational Planning

The ability to quickly and accurately determine the last business day of the month is a fundamental requirement for efficient financial and operational workflows. By mastering the nested combination of the EOMONTH and WORKDAY functions, users of Google Sheets gain a powerful tool for automating deadline calculations, optimizing payroll schedules, and ensuring timely reporting.

This single, elegant formula eliminates the repetitive manual task of cross-referencing calendars and significantly reduces the potential for human error in date-sensitive operations. Whether you are performing simple monthly reconciliations or developing complex financial models, integrating this technique into your spreadsheet tool arsenal will undoubtedly enhance your productivity and data integrity.

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