

How to Create Dates for 15th and Last Day of Month in Excel

Authored by
stats writer

November 18, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Create Dates for 15th and Last Day of Month in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=95565>

In advanced spreadsheet management, particularly within finance, human resources, or operational scheduling, there is a frequent requirement to generate a precise list of dates that alternates between the 15th day and the last day of consecutive months. This pattern is essential for managing bi-monthly payments, generating payroll schedules, or tracking critical reporting deadlines. Generating this sequence manually across multiple years is prone to error and highly inefficient. Fortunately, Excel provides robust functions that allow us to automate this complex date creation process using a single, powerful formula.

Our objective is to construct a continuous list where each subsequent date is either the middle or the end of the month relative to the previous entry. Consider the structure of the desired output, which clearly illustrates this alternating pattern:

Contextualizing Bi-Monthly Date Generation in Excel

The need for dates falling specifically on the 15th and the last day of the month often arises in scenarios where processes are tied to semi-monthly cycles. These cycles require absolute precision, especially because the last day of the month varies (28, 29, 30, or 31 days). A static formula would fail to account for these variations, necessitating the use of specialized date functions designed to handle calendar anomalies, such as leap years and fluctuating month lengths.

To achieve this automation, we rely heavily on Excel's built-in date intelligence. The core challenge is creating a self-referencing mechanism where the formula inspects the previous date and decides whether the next date should be the end of the current month or the 15th of the following month. This decision tree is executed flawlessly by combining the conditional logic of the IF function with advanced date manipulation tools.

Before diving into the complex formula, it is critical to ensure that your starting date is correctly formatted and that you understand how Excel treats dates internally--as sequential serial numbers. This numerical representation is what allows arithmetic operations, like adding 15 days, to function accurately within the spreadsheet environment.

Prerequisite: Initiating the Date Sequence

The first step in generating any date sequence involves establishing a clear starting point. This initial date serves as the anchor for the entire series, allowing the subsequent formulas to reference it and continue the pattern indefinitely.

For our specific example, we must manually input the desired launch date. Since our goal is to alternate between the 15th and the last day, starting with either of these two options is ideal. We will choose the 15th day of a specific month to begin the sequence. We will type in the starting

date, ensuring it is recognized by Excel as a valid date format. For consistency, we will use January 15, 2023.

We will type in **1/15/2023** into cell **A2**. It is important that the cell reference in the formula (A2) matches the location of this initial input. Ensure that the cell formatting is set to Date format to display the output correctly, though the underlying formula works purely with serial values.

	A	B	C	D	E	F
1	Date					
2	1/15/2023					
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						

Implementing the Dynamic Date Formula

Once the starting date is established in cell **A2**, we can implement the self-referencing, dynamic formula into the next cell in the sequence, which is **A3**. This single formula must contain all the logic necessary to determine if the date in **A2** is the 15th or the last day of the month, and subsequently calculate the correct next date.

The formula we use integrates three crucial Excel functions: IF, DAY, and EOMONTH function. We type the following syntax directly into cell **A3**:

```
=IF(DAY(A2)<16,EOMONTH(A2,0),EOMONTH(A2,0)+15)
```

This formula is highly efficient because it handles all the conditional branching and complex date math within one expression. After inputting this formula, simply press Enter to see the second date

in the sequence appear (which should be January 31, 2023, based on our starting date).

Visualizing the Automated Date Sequence

The true power of this technique is revealed when the formula is applied to subsequent rows. Once the formula is established in cell A3, we utilize the fill handle--the small square at the bottom-right corner of the cell--to copy the formula down column A. This action ensures that the relative references (A2 changes to A3, A3 changes to A4, and so on) update correctly, allowing the sequence to automatically calculate the alternating 15th and last days across months and years.

We will click and drag this formula down to populate the list as far as needed. For illustrative purposes, we will extend it to cover the first half of the year 2023:

	A	B	C	D	E	F	G	H
1	Date							
2	1/15/2023							
3	1/31/2023							
4	2/15/2023							
5	2/28/2023							
6	3/15/2023							
7	3/31/2023							
8	4/15/2023							
9	4/30/2023							
10	5/15/2023							
11	5/31/2023							
12	6/15/2023							
13	6/30/2023							
14								
15								
16								
17								
18								

As clearly demonstrated in the resulting list, we have successfully generated a continuous sequence that consistently provides the 15th and the last day of the month for each successive month from January through June of 2023. This method entirely eliminates the need for manual date calculation and management of varying month lengths.

Deconstructing the Advanced IF Logic

To fully appreciate the elegance of this solution, we must examine the intricate logic embedded within the primary formula. The core of the operation relies on the IF function, which acts as a switchboard, directing the calculation based on the condition of the previous cell.

Recall the formula structure:

```
=IF(DAY(A2)<16,EOMONTH(A2,0),EOMONTH(A2,0)+15)
```

The IF function evaluates a logical test and performs one of two actions: a result if the test is TRUE (the second argument), or a result if the test is FALSE (the third argument). In this case, the logical test is built around whether the previous date (A2) is before the 16th day of the month.

Logical Test: `DAY(A2)<16`. This checks if the day number of the date in A2 is less than 16. If A2 contains the 15th of the month (e.g., 1/15/2023), this test is TRUE. If A2 contains the 31st (e.g., 1/31/2023), this test is FALSE.

Value if TRUE: `EOMONTH(A2,0)`. If the previous date was the 15th (less than 16), the formula executes the TRUE condition, which calculates the last day of the current month.

Value if FALSE: `EOMONTH(A2,0)+15`. If the previous date was the last day of the month (not less than 16), the formula executes the FALSE condition. This calculation finds the end of the current month, and then adds 15 days to it, effectively jumping into the 15th day of the next month.

The Power of the EOMONTH Function in Date Calculation

The EOMONTH function is the cornerstone of this solution, as it is specifically designed to handle the irregularity of month lengths. Its primary purpose is to return the serial number for the last day of the month, either the month containing the start date or a month offset from it.

The syntax for EOMONTH is `EOMONTH(start_date, months)`.

The `start_date` is the reference point (e.g., cell A2).

The `months` argument specifies how many months into the future (positive number) or past (negative number) you want to calculate the end date.

In our formula, we use `EOMONTH(A2, 0)`. The `0` argument is crucial because it instructs Excel to find the last day of the month defined by the date in A2. This cleverly bypasses the difficulty of checking whether the month has 30, 31, 28, or 29 days, instantly providing the correct closing date for that cycle. This function inherently manages complexities like the leap year variation of February, ensuring the output is always accurate.

Analyzing the DAY Function and Conditional Branching

The DAY function serves as the critical mechanism for conditional branching. It extracts the day number (an integer between 1 and 31) from a given date serial number. By using `DAY(A2)`, the formula can effectively determine exactly where the previous date fell within its respective month.

The logical test `DAY(A2)<16` is the key to creating the alternating pattern:

Scenario 1: Previous Date is the 15th. If A2 is 1/15/2023, `DAY(A2)` returns 15. Since 15 is less than 16, the condition is TRUE. The formula then executes `EOMONTH(A2, 0)`, resulting in 1/31/2023 (the end of the current month).

Scenario 2: Previous Date is the Last Day. If A2 is 1/31/2023, `DAY(A2)` returns 31. Since 31 is NOT less than 16, the condition is FALSE. The formula executes `EOMONTH(A2, 0)+15`. Since `EOMONTH(A2, 0)` already finds the last day of January, adding 15 days pushes the date 15 days into February, resulting in 2/15/2023 (the 15th of the next month).

This construction ensures a seamless transition regardless of whether the previous date was the 15th or the last day, guaranteeing the list alternates correctly without ever requiring manual intervention.

Troubleshooting and Refining the Date List

While this formula is highly robust, users should be mindful of several practical considerations when deploying it across large datasets or in production environments.

First, **Date formatting** is crucial. Although the calculation relies on serial numbers, the visual output must be correctly formatted as a date to be useful. If your cells show five-digit numbers instead of dates, simply apply the appropriate short or long date format using the Number Format section of the Home tab.

Second, handling potential **errors**. If the starting cell (A2) is empty, or if an invalid entry is made, the subsequent formulas will return errors. It is often wise to wrap the formula in an additional IFERROR function to display a blank cell instead of an error message when the sequence extends beyond the required length or encounters a blank reference:

```
=IFERROR(IF(DAY(A2)<16,EOMONTH(A2,0),EOMONTH(A2,0)+15),"")
```

Third, **scalability**. Since the formula uses relative referencing, it is highly scalable. You can drag the formula down for hundreds of rows, generating years of bi-monthly dates accurately, including correct transitions across year boundaries (e.g., from December 31st to January 15th of the next year).

Conclusion and Further Applications

Mastering the combination of conditional logic (IF), day extraction (DAY), and month-end calculation (EOMONTH function) provides an incredibly efficient method for managing specialized date series in Excel. This technique is not limited solely to generating the 15th and last day; with minor modifications to the IF condition and the added value, you can adapt this structure for any recurring, non-standard date pattern required for financial modeling or administrative tasks.

This single-formula solution significantly enhances productivity and accuracy, ensuring that date-dependent processes, such as semi-monthly payroll runs or bi-weekly billing cycles, rely on mathematically sound and calendar-aware data. Users are strongly encouraged to review the official documentation for the EOMONTH function to explore its full capabilities in handling complex date manipulation scenarios.

ARABPSYCHOLOGY.COM