

# Learn How to Convert UNIX Timestamps to Dates in Excel

Authored by  
**Mohammed loot**

November 14, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learn How to Convert UNIX Timestamps to Dates in Excel*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=781>

## Understanding the UNIX Timestamp

A [UNIX timestamp](#), often referred to as Unix time or Portable Operating System Interface (POSIX) time, is a standardized method for tracking time as a single, large integer. This number represents the total count of seconds that have elapsed since the designated starting point known as the [Epoch](#). This specific epoch is universally defined as January 1, 1970, at 00:00:00 Coordinated Universal Time ([UTC](#)), deliberately ignoring the occasional leap second. This standardized measurement system is widely adopted across global computer systems, operating systems, and databases because of its inherent simplicity and universality, allowing for straightforward storage, transmission, and comparison of temporal data across diverse platforms without the complex issues associated with local [time zones](#) or daylight saving variations.

Unlike conventional human-readable date and time formats--which involve complex units like months, days, years, hours, minutes, and seconds--a UNIX timestamp condenses all temporal data into one concise numerical value. For example, the integer 1648579200 represents the exact moment of March 29, 2022, at 00:00:00 UTC. While this numerical representation is mathematically precise and ideal for programmatic operations, it completely lacks immediate clarity for human interpretation and analysis. Consequently, when preparing data for reports, analysis, or general readability, converting these raw timestamps into a familiar date and time format becomes an essential step, especially when integrating data into powerful spreadsheet applications like [Microsoft Excel](#).

## The Fundamental Discrepancy: UNIX Time vs. Excel Serial Dates

The challenge in moving timestamp data into spreadsheets stems from the fundamentally different ways that UNIX systems and [Excel](#) manage and store time. As established, [UNIX timestamps](#) count seconds starting from January 1, 1970. In contrast, Excel employs a proprietary system where dates are stored as sequential serial numbers, where the date January 1, 1900, is internally represented by the number 1. Every subsequent day increases this serial count by one; thus, January 2, 1900, is 2, and so on. Furthermore, time of day is stored as a fractional component of that day, where midnight is 0 and noon is 0.5. This significant difference in base dates (1970 vs. 1900) and measurement units (seconds vs. days/fractions of a day) necessitates precise mathematical conversion formulas to ensure accurate data integration.

The primary issue arises because if a raw [UNIX timestamp](#) is simply pasted into an Excel cell, the application will treat it merely as a large, general numerical value, not a recognizable date or time. Without the proper conversion logic, this data is rendered functionally unusable for essential date-based operations such as filtering, sorting, or performing calculations within the spreadsheet environment. Therefore, bridging the conceptual gap between the two systems is critical; understanding the underlying principles of both the UNIX epoch and Excel's serial date system

allows us to construct effective conversion formulas that successfully transform abstract numerical data into meaningful and actionable date and time information.

## Method 1: Isolating the Date Component

For scenarios where the exact time of day is irrelevant and only the calendar date is needed (e.g., displaying 03/29/2022), you can convert a [UNIX timestamp](#) to a date-only format in [Excel](#) using a formula that carefully adjusts for both the unit difference and the epoch offset. This specific method ensures that any time component is ignored, resulting in a clean, day-level date. The core mathematical requirement is to convert the total seconds accumulated in the UNIX timestamp into a number of days, and subsequently add this result to the Excel serial number equivalent of the UNIX [Epoch](#).

The initial step of the conversion involves dividing the total seconds provided by the timestamp by the number of seconds contained within a single day. This constant is calculated as 60 seconds/minute multiplied by 60 minutes/hour multiplied by 24 hours/day, totaling **86400 seconds/day**. To ensure only the date remains, the [INT function](#) is applied to this division result, rounding the value down to the nearest whole number and effectively truncating any fractional part that represents the time. Finally, the [DATE function](#) is used to add the necessary offset: `DATE(1970,1,1)`. This function returns the specific serial number for January 1, 1970, thereby aligning the calculated UNIX days with Excel's 1900-based date system.

Here is the complete formula designed to convert a [UNIX timestamp](#) located in cell **A2** into a date-only format:

```
=INT(((A2/60)/60)/24)+DATE(1970,1,1)
```

## Step-by-Step Implementation: Date-Only Conversion

To illustrate the practical application of the date-only conversion method in [Excel](#), let us walk through a typical example. Assume you have a column of raw [UNIX timestamps](#) residing in column A, and your objective is to generate a corresponding column B displaying each of these numerical values as a conventional calendar date. Your initial spreadsheet setup should contain the raw timestamp data as shown in the introductory image below.

Start the process by navigating to cell **B2** and entering the provided conversion formula: `=INT(((A2/60)/60)/24)+DATE(1970,1,1)`. This formula efficiently takes the timestamp from cell A2, converts the total seconds into a number of full days since the UNIX [Epoch](#), and then applies the crucial offset to align it with Excel's date structure. After correctly typing the formula, press the Enter key to execute the calculation and view the initial converted date.

Once the formula is confirmed as correct in cell **B2**, you can swiftly apply it to the rest of your dataset. Select cell **B2**, and then locate the fill handle--the small square situated at the bottom-right corner of the cell. Click and drag this handle downwards through column B, ensuring it covers all rows that correspond to the timestamps in column A. This automatic action leverages Excel's relative referencing to populate column B with the converted dates for every timestamp, transforming the raw numerical data into easily digestible calendar dates, as demonstrated in the resulting image.

	A	B	C	D	E
1	<b>Timestamp</b>				
2	1648565400				
3	1589923203				
4	1602329845				
5	1389923293				
6	1492323890				
7	1289823238				
8	1298232377				
9	1478855549				
10	1238989823				
11					
12					
13					
14					
15					
16					
17					
18					

To convert each timestamp in column A into a date in column B, type the following formula into cell **B2**:

```
=INT(((A2/60)/60)/24)+DATE(1970,1,1)
```

Then, click and drag this formula down to each remaining cell in column B:

B2		=INT(((A2/60)/60)/24)+DATE(1970,1,1)				
	A	B	C	D	E	F
1	<b>Timestamp</b>	<b>Date</b>				
2	1648565400	3/29/2022				
3	1589923203	5/19/2020				
4	1602329845	10/10/2020				
5	1389923293	1/17/2014				
6	1492323890	4/16/2017				
7	1289823238	11/15/2010				
8	1298232377	2/20/2011				
9	1478855549	11/11/2016				
10	1238989823	4/6/2009				
11						
12						
13						
14						
15						

As demonstrated by the final output, column B successfully displays each [UNIX timestamp](#) from column A as a recognizable calendar date.

## Method 2: Converting to Full Date and Time

When the required level of detail extends beyond just the calendar date and demands the precise time of occurrence, converting a [UNIX timestamp](#) into a full date and time format (e.g., 03/29/2022 14:50:00) is essential. This method capitalizes on [Excel's](#) capability to store both date (integer part) and time (fractional part) within a single serial number. The formula used for this combined conversion is remarkably concise compared to the date-only version, but it requires a crucial final step: applying a custom [Number Format](#) to correctly reveal the time component hidden in the fraction.

The formula is simply `=A2/86400+25569`. The term `A2/86400` converts the UNIX timestamp, which is measured in seconds, into a total number of days, critically retaining the fractional decimal part that represents the hour, minute, and second. The constant **25569** is then added to this result. This specific number holds paramount importance as it represents the precise number of days separating Excel's base date (January 1, 1900) and the start of the UNIX [Epoch](#) (January 1, 1970). By incorporating this exact offset, the formula successfully aligns the calculated UNIX time with Excel's internal serial date system, generating a single, comprehensive serial number ready for formatting.

It is vital to recognize that immediately after applying this formula, the output cell will typically display a large numerical value, not the date and time. This occurs because Excel defaults to a general number format. To correctly visualize the complete date and time information, the user must explicitly apply a custom [Number Format](#), such as **mm/dd/yyyy hh:mm:ss**. This formatting step instructs Excel how to render the underlying serial number into an easily readable format that includes hours, minutes, and seconds, thereby completing the conversion process.

Here is the formula for full date and time conversion, assuming your [UNIX timestamp](#) is in cell **A2**:

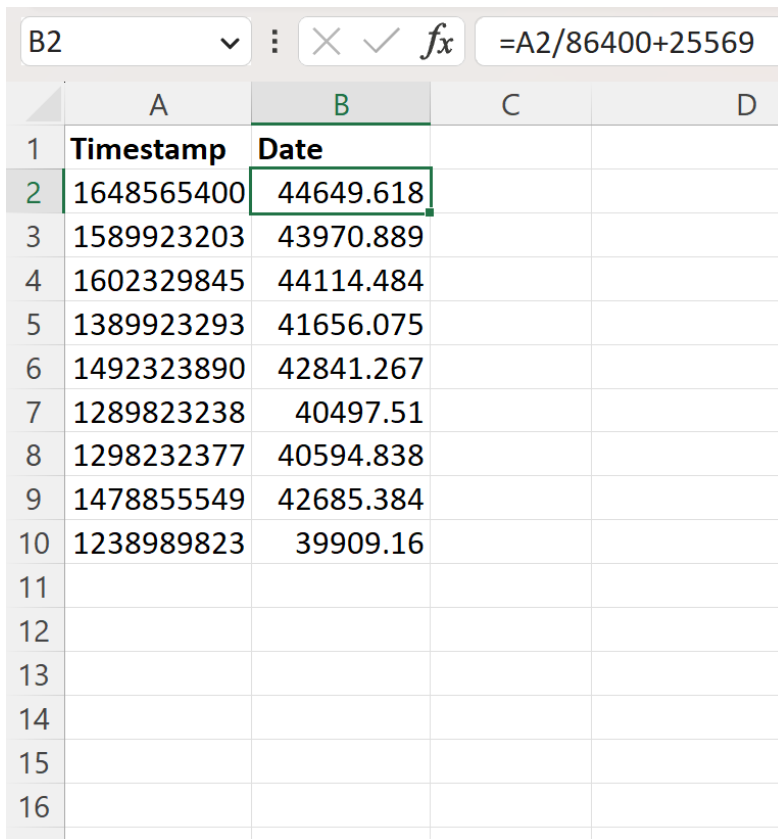
**=A2/86400+25569**

## Step-by-Step Implementation: Date and Time Conversion

To convert [UNIX timestamps](#) into a format that includes both the date and the precise time, follow these detailed steps within your [Excel](#) workbook. Assuming your source data is located in column A, the conversion results will be placed in column B.

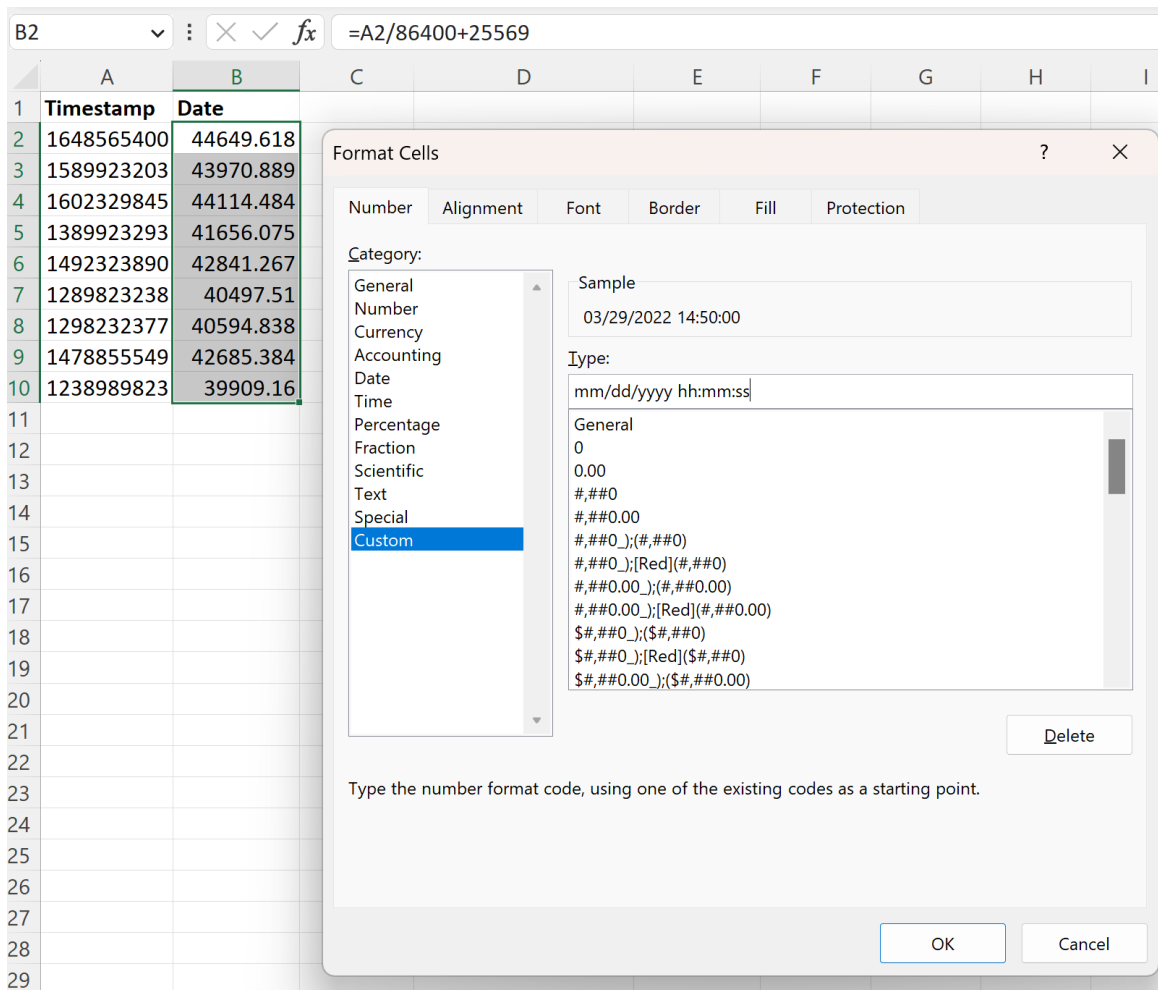
Begin by entering the streamlined formula **=A2/86400+25569** into cell **B2**. This formula executes the necessary arithmetic to transform the seconds-based timestamp from cell A2 into an Excel serial number that encompasses both date and time data. Press Enter, and as expected, cell B2 will initially display a large numerical value because the default cell format is General.

Next, apply this formula across your dataset. Select cell **B2** and click and drag the fill handle down through column B until it aligns with all timestamps in column A. This action populates column B with a series of numerical serial values, each representing a converted timestamp, as illustrated in the image provided below. The following step is crucial for visualization, as the data is currently only numeric.



	A	B	C	D
1	<b>Timestamp</b>	<b>Date</b>		
2	1648565400	44649.618		
3	1589923203	43970.889		
4	1602329845	44114.484		
5	1389923293	41656.075		
6	1492323890	42841.267		
7	1289823238	40497.51		
8	1298232377	40594.838		
9	1478855549	42685.384		
10	1238989823	39909.16		
11				
12				
13				
14				
15				
16				
17				

To make these numbers readable as dates and times, you must apply a custom [Number Format](#). Highlight the entire range of cells containing your converted values (e.g., **B2:B10**). Navigate to the **Home** tab on Excel's ribbon, and then click the small launcher arrow in the bottom-right corner of the **Number** group to open the "Format Cells" dialog box. Inside this window, select **Custom** from the Category list. In the **Type** field, manually enter the format string **mm/dd/yyyy hh:mm:ss**. This format tells Excel precisely how to interpret the underlying serial number and display the month, day, year, hour, minute, and second.



The screenshot shows an Excel spreadsheet with two columns: 'Timestamp' (Column A) and 'Date' (Column B). The data in Column A consists of UNIX timestamps, and the data in Column B consists of the corresponding dates. A 'Format Cells' dialog box is open, showing the 'Custom' category selected. The 'Type' field contains the format code 'mm/dd/yyyy hh:mm:ss'. The dialog box also shows a list of existing format codes and a 'Delete' button.

Timestamp	Date
1648565400	44649.618
1589923203	43970.889
1602329845	44114.484
1389923293	41656.075
1492323890	42841.267
1289823238	40497.51
1298232377	40594.838
1478855549	42685.384
1238989823	39909.16

After confirming the custom format and clicking **OK**, column B will immediately update, displaying each timestamp from column A as a complete date with hours, minutes, and seconds. This final formatting step renders the data fully accessible and ready for detailed time-based analysis and reporting.

	A	B	C	D
1	<b>Timestamp</b>	<b>Date</b>		
2	1648565400	03/29/2022 14:50:00		
3	1589923203	05/19/2020 21:20:03		
4	1602329845	10/10/2020 11:37:25		
5	1389923293	01/17/2014 01:48:13		
6	1492323890	04/16/2017 06:24:50		
7	1289823238	11/15/2010 12:13:58		
8	1298232377	02/20/2011 20:06:17		
9	1478855549	11/11/2016 09:12:29		
10	1238989823	04/06/2009 03:50:23		
11				
12				
13				
14				
15				
16				
17				
18				

## Crucial Considerations: Time Zones and Accuracy

When handling [UNIX timestamps](#) and converting them in [Excel](#), it is critically important to maintain a rigorous awareness of [time zones](#). By fundamental definition, a UNIX timestamp inherently represents a moment in time based on Coordinated Universal Time ([UTC](#)). This means the raw numerical value contains no information regarding local time zone offsets or daylight saving time adjustments. When the conversion formulas provided in this guide are executed in Excel, the resulting date and time values will also be in UTC.

If your specific analytical requirements demand that the displayed time be in a local time zone, you must introduce an additional mathematical adjustment to the Excel formula to account for that offset. For instance, if you are working in Eastern Standard Time (EST), which is UTC-5, you would need to subtract 5 hours from the converted UTC time. This is achieved by subtracting the fractional equivalent of 5 hours from the Excel serial number (e.g., adding ``-5/24`` to your formula). A failure to correctly factor in time zone differences can result in significant and misleading discrepancies in your data, particularly when precise local timing is a necessity for reporting or scheduling. Always verify whether the source system generating the UNIX timestamps is truly outputting pure UTC or if it has already applied a localized conversion before export.

Furthermore, the overall accuracy of the conversion is dependent on the integrity of the source [UNIX timestamp](#) itself. Users must ensure that the timestamps they are processing are valid and correctly formatted numerical values. Timestamps that are corrupted, truncated, or incorrectly generated by the source system will inevitably yield erroneous conversion results, irrespective of the correctness of the Excel formula. While Excel's date system is robust, it remains a strong practice to routinely spot-check converted dates against known reference values, especially for data points that occur close to midnight or those that potentially cross daylight saving boundaries in their original recording context.

## Deconstructing the Formulas: The Logic Behind the Conversion

A robust understanding of the arithmetic principles behind these [Excel](#) formulas is key to troubleshooting and adapting them for complex data scenarios. Both conversion methods rely on the core mathematical steps of converting the base unit from seconds into days, and then precisely adjusting for the difference in [UTC](#) epoch dates between UNIX time and Excel's internal serial system. The two critical constants used are 86400 (the number of seconds in a day) and 25569 (the number of days separating the two epochs).

In Method 1, the formula `=INT(((A2/60)/60)/24)+DATE(1970,1,1)` employs a meticulous division sequence: the total seconds in cell A2 are divided by 60 to get minutes, then by 60 again to get hours, and finally by 24 to get days. The [INT function](#) is deliberately applied to remove any fractional remainder, thereby ensuring only the calendar date remains. The `DATE(1970,1,1)` function is essential as it returns the exact serial number for January 1, 1970. By adding the calculated number of days to this specific Excel serial date, we effectively shift the base reference from the UNIX starting point to the Excel starting point, yielding the correct calendar date output.

Method 2, represented by the concise formula `=A2/86400+25569`, integrates the time component directly. Here, `A2/86400` converts the UNIX timestamp (in seconds) directly into a total number of days, including the vital fractional component that represents the time of day. The constant **25569** is the numerical difference in days between Excel's start date (January 1, 1900, which is day 1) and the UNIX Epoch (January 1, 1970). By adding this precise offset, the formula aligns the calculated UNIX days directly with Excel's serial date system. This produces a single serial number that, when correctly formatted using a custom [Number Format](#), displays both the date and the exact time.

## Conclusion and Further Data Mastery

The conversion of [UNIX timestamps](#) into easily readable dates and times in [Excel](#) is a fundamental skill for data professionals working with information sourced from various computer systems. The two distinct methods detailed here provide robust and reliable solutions, whether your analytical

requirements necessitate extracting only the calendar date or capturing the full, precise time information. By mastering the core differences between the UNIX and Excel date systems and accurately applying these formulas, you can seamlessly integrate complex timestamp data into your spreadsheets for effective analysis and insightful reporting.

Always remember the critical implications of [time zones](#); although [UNIX timestamps](#) are inherently [UTC](#), adjustments are often required for local display and interpretation. Ensure thorough verification of your source data and carefully consider the specific needs of your analysis to select the most appropriate conversion method and formatting. Proficiency in these techniques significantly elevates your data manipulation capabilities within Excel, transforming raw, abstract numbers into clear, actionable business intelligence.

## **Additional Resources for Advanced Excel Proficiency**

To further expand your skills in [Excel](#) and confidently address other common data manipulation and analytical challenges, we recommend exploring the following tutorials and resources. These guides can help you master a variety of functions and techniques that are invaluable for data analysis, detailed reporting, and automation within your spreadsheets.

[Understanding VLOOKUP and HLOOKUP Functions](#)

[Creating and Customizing Pivot Tables](#)

[Applying Conditional Formatting for Data Visualization](#)

[Implementing Data Validation Rules](#)

[Working with Text Functions \(LEFT, RIGHT, MID, CONCATENATE\)](#)