

Extracting Time Values from Datetime in Excel: A Step-by-Step Guide

Authored by
Mohammed loot

November 10, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Extracting Time Values from Datetime in Excel: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15995>

Handling [datetime](#) data within spreadsheets often demands specific and controlled formatting adjustments. When performing data processing in [Excel](#), one of the most frequent requirements is the need to efficiently extract only the time component from a cell that contains both date and time information. This extraction process is critically important for specialized tasks such as calculating specific durations, comparing temporal timestamps without date interference, or meticulously preparing datasets for data visualization where the date element is entirely irrelevant to the analysis.

It is vital to recognize that simply altering a cell's numerical formatting in [Excel](#) only changes the visual appearance; the underlying [datetime](#) value remains intact and subject to future interpretation errors. To achieve a robust and permanent solution, we require a method that isolates and displays the time as a distinct, stable text string. The most effective and reliable technique available utilizes Excel's powerful built-in functions, specifically the versatile [TEXT function](#).

In the following guide, we will explore several precise formulas designed to convert a comprehensive [datetime](#) stamp into various desired time formats. This ensures you maintain absolute data integrity while achieving the exact visual output required for reporting or subsequent data processing. All methods described herein assume your source [datetime](#) value is stored in cell **A2**, though they are inherently flexible and easily adaptable to any cell reference within your worksheet.

Mastering Time Extraction Using the TEXT Function

The most straightforward and technically sound approach to converting a full [datetime](#) value into a dedicated time display in [Excel](#) is through the utilization of the [TEXT function](#). This powerful utility serves a singular, critical purpose: it accepts a numerical value, which in the context of Excel timekeeping is known as the [serial number](#), and meticulously transforms it into a customized text string based entirely on a specified format code. This capability is paramount because it allows us to effectively discard the date component (the integer part of the serial number) and dictate the exact presentation of the time portion, ensuring the result is stable, fixed, and non-calculable text.

Understanding the fundamental syntax of the [TEXT function](#) is key to its mastery. The structure is concisely defined as: **=TEXT(value, format_text)**. The "value" argument is simply the cell reference containing the original datetime stamp (e.g., A2). Conversely, the "format_text" argument is the specific sequence of characters enclosed in quotation marks that instructs Excel precisely how to render the time. By manipulating these codes--such as "H" for hour, "MM" for minute, and "SS" for second--we gain granular control over the output, allowing us to generate time displays tailored for various reporting and analytical needs.

We have detailed three distinct formulas below, designed to cover the most common time display requirements encountered in professional environments. Each formula utilizes the same core logic

of the [TEXT function](#) but employs a specialized format code to achieve unique results, whether that involves including standard [AM/PM](#) indicators or demanding high-precision displays that include seconds. These examples form the foundation for reliable time data manipulation within any complex spreadsheet.

Formula 1: 12-Hour Time Format with AM/PM Meridian Indicator

This specific formula should be utilized when reporting standards or user comprehension requires the time to be displayed in the universally recognized 12-hour format. This format enhances clarity by explicitly indicating the meridian, showing whether the recorded time falls before noon (AM) or after noon (PM). For example, a result might appear as "2:14 AM" or "5:30 PM." The format code **H:MM AM/PM** handles both the time conversion and the mandatory meridian indication automatically.

```
=TEXT(A2,"H:MM AM/PM")
```

Formula 2: 24-Hour Time Format (Technical and International Standard)

If your organizational workflow, technical specifications, or international collaboration demands the strict use of the 24-hour clock format--often referred to as military time--this formula is the solution. The 24-hour format provides a clean, unambiguous display that eliminates the need for the [AM/PM](#) indicator (e.g., 2:14 AM becomes 2:14, and 2:14 PM becomes 14:14). Utilizing the format code **H:MM** ensures strict compliance with technical time standards while efficiently isolating the time component from the full datetime stamp.

```
=TEXT(A2,"H:MM")
```

Formula 3: High-Precision Time Display Including Seconds

For specialized applications such as electronic data logging, scientific data capture, or any measurement where accuracy must extend to the second, it is necessary to extract this highest level of detail. By carefully including the appropriate format code components, specifically the seconds indicator (**:SS**), we can extract and display this crucial detail from the original [datetime](#) stamp. This results in a comprehensive time display such as "2:14:00."

```
=TEXT(A2,"H:MM:SS")
```

A Practical Demonstration of Time Conversion in Excel

To solidify the application and effectiveness of these formulas, we will now walk through a practical, real-world example using a standard [Excel](#) worksheet setup. Our primary objective is to take a column of raw [datetime](#) values and convert them into three separate, distinctly formatted time representations in adjacent columns, showcasing the flexibility of the [TEXT function](#).

We begin with a representative dataset located in Column A, where every cell contains a composite [datetime](#) stamp. For example, the initial data entry might be "9/1/2023 2:14:00 AM." It is essential to internalize that while this appears as standard text, [Excel](#) interprets this entire value as a single, large numerical value--a [serial number](#)--which the formulas will subsequently parse for the time component.

	A	B	C	D	E
1	Datetime				
2	1/1/2023 2:14 AM				
3	1/4/2023 5:19 AM				
4	2/12/2023 10:40 PM				
5	2/14/2023 8:59 AM				
6	3/1/2023 9:34 PM				
7	7/25/2023 10:13 PM				
8	10/26/2023 1:09 AM				
9	12/1/2023 12:45 PM				
10					
11					
12					
13					
14					
15					

The next critical step involves structuring our output columns. We designate columns B, C, and D for the three different time formats. We apply the specific conversion formulas into the second row of these columns (B2, C2, and D2), using cell **A2** as the input for all three conversions. This setup allows for a direct comparison of how the distinct format codes generate different time outputs from the exact same source data.

We implement the three formulas precisely as follows:

In cell **B2** (12-Hour AM/PM Format): **=TEXT(A2,"H:MM AM/PM")**

In cell **C2** (24-Hour Format): **=TEXT(A2,"H:MM")**

In cell **D2** (Seconds Included Format): **=TEXT(A2,"H:MM:SS")**

Once these initial formulas are established in the top row, we utilize [Excel's](#) highly efficient **fill handle** feature. By clicking and dragging the formulas down to cover the remaining rows in our dataset, the entire column of original [datetime](#) values is instantaneously processed. This bulk application is essential for efficient data management, reliably generating the desired time outputs across the entire range.

The resulting table clearly illustrates how each format code successfully isolates the time component, presenting it according to the specified requirements, ranging from the standard 12-hour display to the precise inclusion of seconds:

	A	B	C	D
1	Datetime	Time (AM/PM)	Time (no AM/PM)	Time (with seconds)
2	1/1/2023 2:14 AM	2:14 AM	2:14	2:14:00
3	1/4/2023 5:19 AM	5:19 AM	5:19	5:19:00
4	2/12/2023 10:40 PM	10:40 PM	22:40	22:40:00
5	2/14/2023 8:59 AM	8:59 AM	8:59	8:59:00
6	3/1/2023 9:34 PM	9:34 PM	21:34	21:34:00
7	7/25/2023 10:13 PM	10:13 PM	22:13	22:13:00
8	10/26/2023 1:09 AM	1:09 AM	1:09	1:09:00
9	12/1/2023 12:45 PM	12:45 PM	12:45	12:45:00
10				
11				
12				
13				
14				
15				
16				

Understanding Excel's Internal Timekeeping: The Serial Number System

To fully appreciate the sophistication and necessity of the [TEXT function](#) approach, it is paramount to understand the mathematical foundation upon which [Excel](#) constructs its date and time handling. Excel does not store these values as simple calendar entries; instead, it relies on the [serial number](#) system, which is the engine driving all chronological calculations within the program.

In this numerical framework, a complete [datetime](#) stamp is always represented by a single decimal number. The integer portion (the whole number before the decimal point) represents the date,

counting the number of days elapsed since January 1, 1900 (the standard starting date for Windows versions). Crucially, the fractional or decimal part of the number represents the time. This fraction is the mathematically precise proportion of a 24-hour day that has passed. For example, the fraction 0.5 exactly corresponds to noon (12 hours, or half a day), and 0.25 corresponds to 6:00 AM (a quarter of a day). Thus, a full datetime value like "9/1/2023 2:14:00 AM" is stored internally as a large number with a significant decimal component.

When we apply a formula such as the one used in cell **B2**, which targets only the time display:

```
=TEXT(A2,"H:MM AM/PM")
```

The [TEXT function](#) intelligently processes the entire [serial number](#) contained in **A2**. However, because the specific format code we provide--"H:MM [AM/PM](#)"--exclusively specifies time components, the function selectively focuses on and utilizes only the fractional part of the number. The large integer portion (the date) is effectively filtered out and ignored during the formatting process. The resulting output is guaranteed to be a **text string**, which prevents unexpected changes and ensures that the time display remains consistent, regardless of subsequent cell formatting operations applied elsewhere in the spreadsheet.

Comparing Methods: Why TEXT is Superior to Numerical Extraction

Although the [TEXT function](#) is the recommended industry standard for converting a [datetime](#) stamp into a text-based time display, it is instructive to briefly consider alternative approaches in [Excel](#), if only to highlight the TEXT function's superior efficiency.

One alternative involves using mathematical operations, often combined with functions like **MOD** or **INT**, to isolate the fractional part of the [serial number](#). For instance, the formula `=A2-INT(A2)` would indeed return the fractional component (the time value) as a raw numerical value. While the extraction is successful, this method presents a significant practical limitation: the result is an unformatted decimal (e.g., 0.093055556), not a legible time display. To make this result readable, you would then be required to manually apply a time format (e.g., H:MM) to the result cell, adding a secondary, manual step to the process.

The primary drawback of this two-step numerical method is that it separates the crucial extraction logic from the formatting logic. If the cell formatting is ever inadvertently removed or changed, the underlying numerical time value reverts immediately to a meaningless decimal fraction. In sharp contrast, the combined [TEXT function](#) approach executes both extraction and formatting simultaneously, returning a static, display-ready text string. This inherent stability and single-step operation make the TEXT function significantly more robust for rigorous reporting and data visualization preparation.

Final Considerations for Data Integrity and Calculation

A crucial final point for users is the implication of the [TEXT function](#) output. Because the function returns a **text string**, the resulting time is no longer recognized by Excel as a numerical value suitable for immediate arithmetic calculations. This means that if your subsequent workflow requires performing calculations--such as adding or subtracting time durations--on the extracted time, you must first convert the text string back into a numerical time value using dedicated functions like **VALUE** or **TIMEVALUE**.

However, for the vast majority of reporting, data presentation, and display purposes, generating a stable, consistently formatted text output is the precise objective. The TEXT function successfully achieves this, guaranteeing that the time always appears exactly as intended, bypassing the inherent unpredictability often associated with Excel's automatic date and time numerical formatting.

Additional Resources for Advanced Excel Mastery

Mastering the reliable conversion and manipulation of [datetime](#) data is a core skill for any professional performing advanced data analysis and reporting within [Excel](#). We highly recommend consulting the official Microsoft documentation for the [TEXT function](#) to review a comprehensive list of all available formatting codes. Expanding your knowledge of these codes allows for virtually infinite customization of your time and date outputs, enabling you to tailor data presentations precisely to any organizational or analytical requirement.

The following tutorials explain how to perform other common and essential operations in [Excel](#):