

# How to Convert Dates to Text in Power BI: A 2-Step Guide

Authored by  
**stats writer**

January 13, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Convert Dates to Text in Power BI: A 2-Step Guide*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125997>

Converting data types within Power BI is a fundamental skill for data modeling and visualization, ensuring that your data behaves correctly within reports. While dates are typically stored in a specialized date format for calculation purposes, converting a date field into a text string is often necessary for presentation layers, custom concatenation, or when exporting fixed-format data. This transformation allows users to control the exact appearance of the date, moving beyond standard regional settings to create precise, customized outputs.

The process of converting a date to text can be approached through two distinct methodologies, each tailored to different requirements within the Power BI environment. The first method involves straightforward manipulation using the user interface within the data modeling or query views, offering a quick solution for basic type changes. The second, more powerful method utilizes DAX (Data Analysis Expressions) functions, such as `FORMAT` or `CONVERT`, which provide granular control over the output format and are essential for creating calculated columns or measures that rely on specific text representations of dates. Understanding when and how to apply these two approaches is key to effective data preparation.

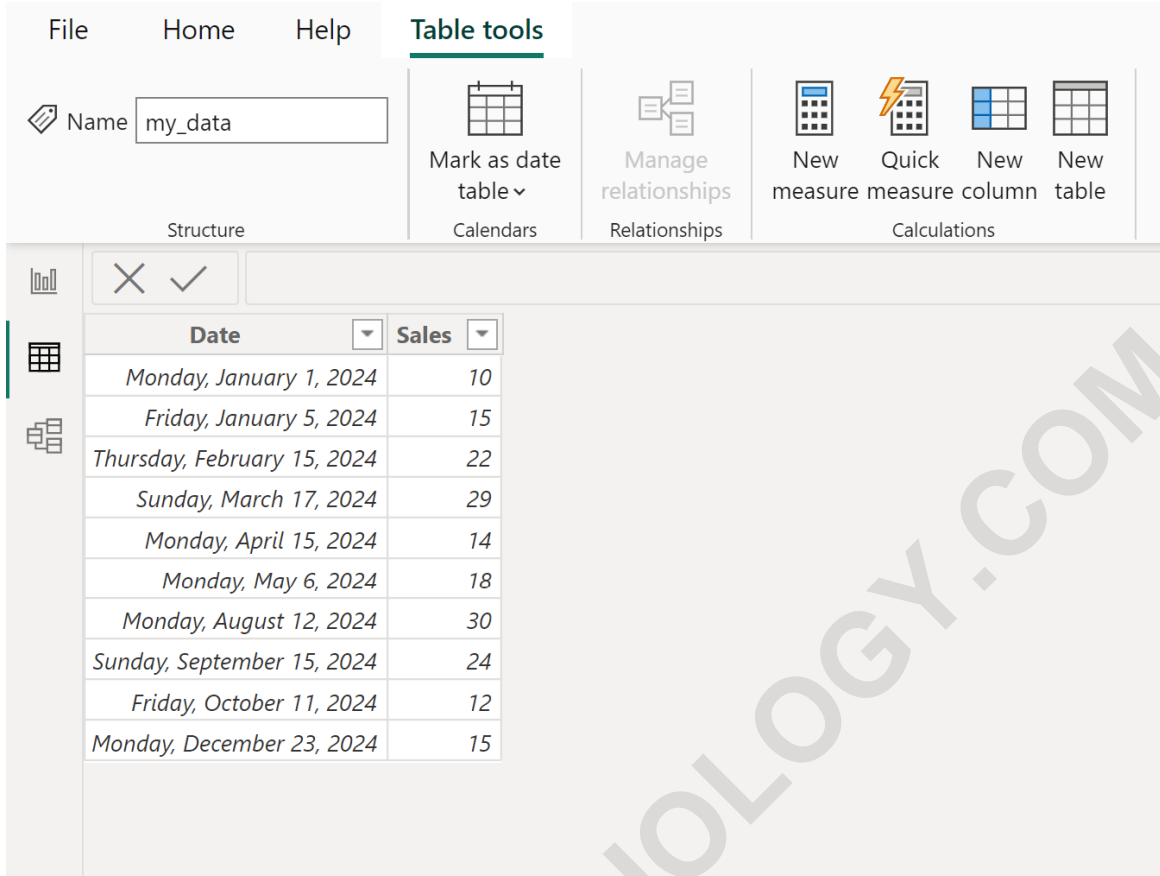
Historically, methods often relied on the DAX `FORMAT` function, which is highly flexible, allowing you to specify the exact date output format (e.g., "MMMM d, yyyy"). The lesser-used `TEXT` function (which is effectively an alias for `FORMAT` in many contexts) or the `CONVERT` function offer alternative pathways. However, for sheer speed and simplicity, especially when no custom formatting is needed, the UI approach is often preferred. Both methods are valid and applicable whether working in the Power Query Editor or directly in the DAX formula bar for calculated entities.

To summarize the core techniques, there are two primary ways to efficiently transform a date column into a text column within your Power BI data model:

**Method 1: Use the Data Type Dropdown Menu in Column Tools.** This method is non-destructive and changes the column's underlying data type representation in the model.

**Method 2: Write a Custom DAX Function.** This method creates a new calculated column, preserving the original date column while providing a textual representation.

We will demonstrate how to implement these strategies using a sample dataset. Consider the following simple table containing dates that we wish to transform:



The screenshot shows the Power BI ribbon with the 'Table tools' tab selected. The ribbon is divided into four main sections: Structure, Calendars, Relationships, and Calculations. The 'Calculations' section is currently active, displaying icons for 'New measure', 'Quick measure', 'New column', and 'New table'. Below the ribbon, a data table is visible with two columns: 'Date' and 'Sales'. The 'Date' column contains various dates, and the 'Sales' column contains numerical values.

Date	Sales
Monday, January 1, 2024	10
Friday, January 5, 2024	15
Thursday, February 15, 2024	22
Sunday, March 17, 2024	29
Monday, April 15, 2024	14
Monday, May 6, 2024	18
Monday, August 12, 2024	30
Sunday, September 15, 2024	24
Friday, October 11, 2024	12
Monday, December 23, 2024	15

## Method 1: Quick Conversion Using the Data Type Dropdown Menu

The most direct way to change the representation of a column is through the Data Type dropdown menu available in the modeling ribbon. This method is ideal when you simply need the column to be treated as a text field for visualization purposes and do not require specific, custom date formatting (like converting '01/15/2023' into 'January 15th, 2023'). Instead, it relies on Power BI's default internal conversion rules, which usually result in the standard short date format for the locale. This approach is fast, intuitive, and does not require writing any code, making it highly accessible for users who are new to DAX.

This process is applied directly to the column within the Data View or Report View after the data has been loaded from the Power Query Editor. When you execute this conversion, Power BI immediately updates the data model metadata, signaling that the values in this column should now be interpreted as text strings rather than date objects. While simple, it is important to remember that converting a column to text permanently removes the ability to use that specific column for date-based calculations, such as calculating the number of days between two dates or aggregating data based on date hierarchy.

If you are aiming for highly customized output, such as ensuring all dates are displayed as

YYYYMMDD without separators, the UI conversion might fall short. It is designed for generic data type changes. For complex formatting needs, the second method utilizing [DAX](#) is the recommended path, as it allows for precise format strings to be applied during the transformation process. However, if the goal is merely to treat the date column as categorical text, the UI modification is the quickest route.

## Step-by-Step Guide: UI Conversion via Column Tools

Suppose our objective is to convert the existing **Date** column in the sample table from its current date format (marked by the calendar icon) to a text format (marked by the 'ABC' icon). This procedure takes place entirely within the main [Power BI](#) Desktop interface under the Data View:

Navigate to the **Data View** where your tables are displayed.

Select the column header of the **Date** column you intend to modify.

Upon selecting the column, the **Column tools** tab will automatically become active in the ribbon at the top of the interface.

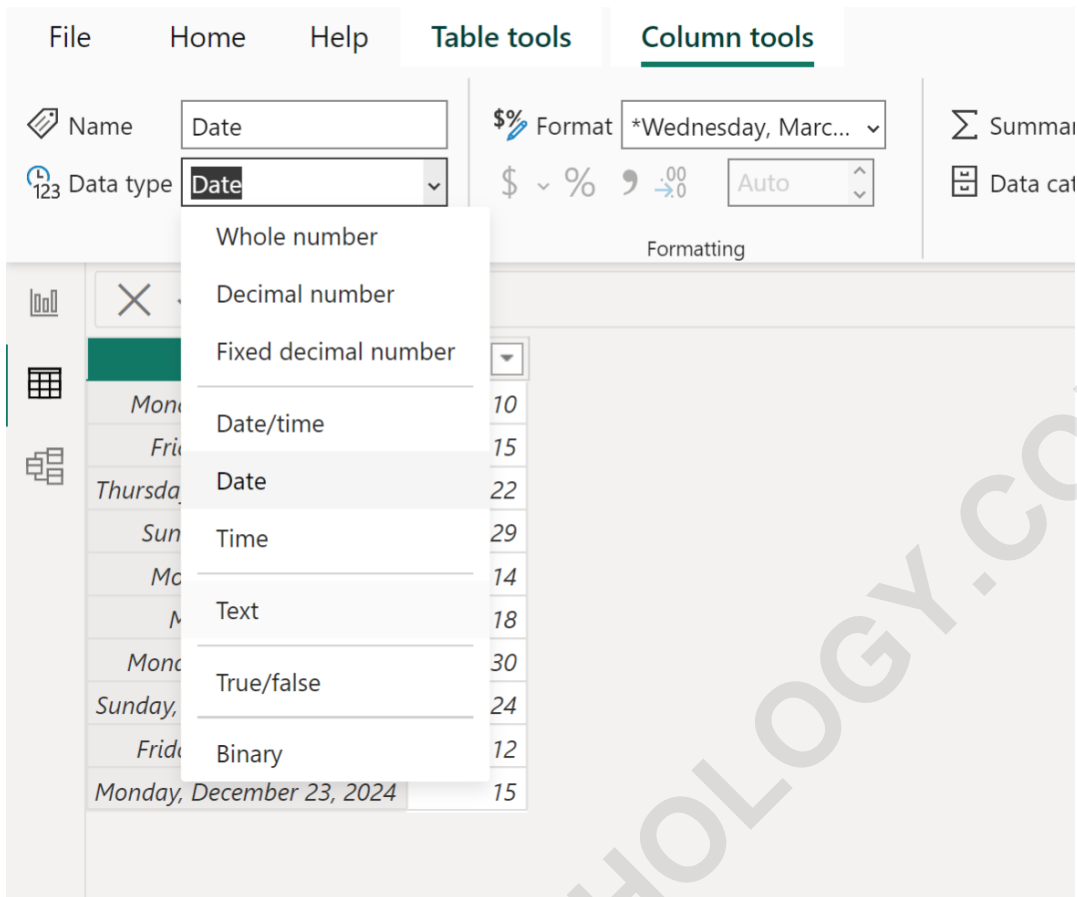
Within the **Column tools**, locate the **Formatting** section.

Click on the **Data type** dropdown menu. This menu currently displays "Date" or "Date/Time".

From the list of available types, select **Text**.

Upon clicking **Text**, a prompt may appear warning you about the potential loss of data integrity or calculation capability, as you are changing the underlying data type. Confirm the change to apply the transformation. The visual indicator next to the column name will switch from the date icon to the text icon (ABC).

The visual depiction of this step is shown below, illustrating the selection of the Text type from the dropdown menu:



Once confirmed, the column is transformed into a text field. The resulting table will reflect this change, making the date values appear as basic strings, ready for use in text-based visualizations or concatenations:

The screenshot displays the Power BI ribbon with the 'Column tools' tab selected. Under the 'Structure' section, the 'Name' property is 'Date' and the 'Data type' is 'Text'. Under the 'Formatting' section, the 'Format' property is 'Text'. Below the ribbon, a table is shown with two columns: 'Date' and 'Sales'. The 'Date' column contains the following values: 1/1/2024, 1/5/2024, 2/15/2024, 3/17/2024, 4/15/2024, 5/6/2024, 8/12/2024, 9/15/2024, 10/11/2024, and 12/23/2024. The 'Sales' column contains the following values: 10, 15, 22, 29, 14, 18, 30, 24, 12, and 15. A large watermark 'ARABPSYCHOLOGY.COM' is overlaid diagonally across the image.

## Method 2: Leveraging DAX for Custom and Calculated Text Fields

When the requirement is to maintain the original date column for mathematical purposes while simultaneously needing a text version for display, the use of DAX functions is mandatory. This approach involves creating a **New Column** within the data model. This calculated column is distinct from the source column, meaning the original date field retains its date data type and remains available for time intelligence functions and other date-specific calculations. This practice aligns perfectly with best practices in data modeling, ensuring that raw data types are preserved.

There are two primary DAX functions typically employed for this conversion: `CONVERT` and `FORMAT`. The `CONVERT` function is generally used for explicit type casting, turning one data type into another specified type, such as transforming a Date to a String. The FORMAT function, however, is significantly more versatile for dates, as it allows the user to define a precise format string (e.g., "dd/MM/yyyy", "Qtr 1 YYYY") which dictates the exact textual representation of the date value. Choosing between them depends entirely on whether a standard conversion is sufficient or if complex custom formatting is required.

For scenarios where complex formatting is not a concern, the `CONVERT` function provides a clean,

readable formula to achieve the conversion. It is particularly useful for quickly generating a text column that follows the default system short date format. Conversely, the power of `FORMAT` lies in its ability to handle nuanced requirements, such as outputting dates in a specific cultural or reporting standard that differs from the default locale settings configured in the Power BI Desktop environment. Both functions reside within the **Table Tools** ribbon when creating a new calculated column.

## Implementing the CONVERT Function for Basic Text Output

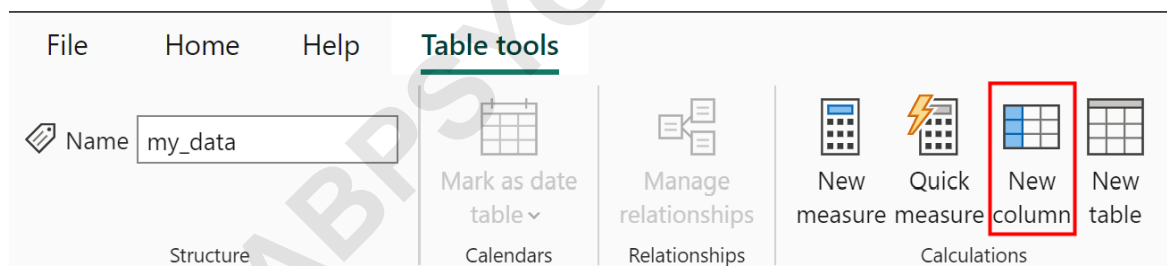
To implement the transformation using DAX, we must first establish a new column where the textual output will reside. This ensures the integrity of our existing data model remains intact. The following steps guide the user through the process of adding a new calculated column and applying the CONVERT function:

Ensure you are in the **Data View** of Power BI Desktop.

Locate the **Table tools** tab in the ribbon.

Click the icon labeled **New column**. This action opens the formula bar, waiting for a DAX expression.

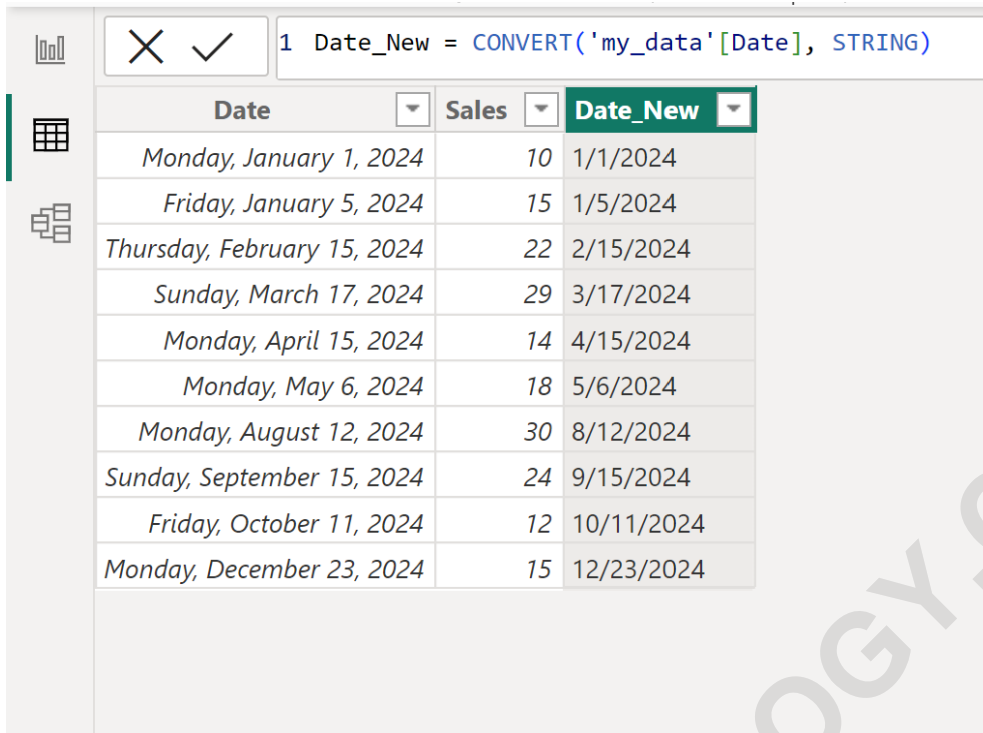
The interface change after clicking **New column** is critical, as it prepares the environment for the DAX formula input:



Then type the following formula into the formula bar:

```
Date_New = CONVERT('my_data', STRING)
```

This expression creates a new column named **Date\_New** that shows each of the corresponding date values from the **Date** column represented as text values in the format mm/dd/yyyy:



Date	Sales	Date_New
Monday, January 1, 2024	10	1/1/2024
Friday, January 5, 2024	15	1/5/2024
Thursday, February 15, 2024	22	2/15/2024
Sunday, March 17, 2024	29	3/17/2024
Monday, April 15, 2024	14	4/15/2024
Monday, May 6, 2024	18	5/6/2024
Monday, August 12, 2024	30	8/12/2024
Sunday, September 15, 2024	24	9/15/2024
Friday, October 11, 2024	12	10/11/2024
Monday, December 23, 2024	15	12/23/2024

It is worth noting that while `CONVERT` is straightforward for simple type casting, if your objective is to ensure the date always appears in a format like "YYYY-MM-DD", you must turn to the highly flexible `FORMAT` function, which we will explore next, as `CONVERT` does not allow format string specification.

### Advanced Technique: Using the `FORMAT` Function for Specific Date Styles

For professional reporting, defaulting to the system's short date format is often insufficient. Report consumers may require dates to be represented in verbose formats, financial formats (e.g., showing only the month and year), or specific ISO standards. This level of customization is achieved through the `FORMAT` function in DAX. The `FORMAT` function takes the date value and a format string as arguments, returning the result as a text string based on the specified pattern.

The basic syntax for the `FORMAT` function is `FORMAT(Value, Format_String)`. The power lies in the `Format_String`, which uses specific placeholders:

"DD" or "dd": Day of the month (with leading zero).

"MM" or "mm": Month number (with leading zero).

"MMM": Abbreviated month name (e.g., Jan).

"MMMM": Full month name (e.g., January).

"YYYY" or "yyyy": Four-digit year.

For example, to convert the date into a textual representation reading "Month Name, Year (e.g., March, 2024)", the following formula would be used in a new calculated column:

```
Date_Formatted_Text = FORMAT('my_data', "MMMM, yyyy")
```

Using `FORMAT` ensures that regardless of the regional settings of the [Power BI Desktop](#) instance or the service, the textual output remains consistent, which is crucial for cross-regional data consumption and adherence to strict corporate styling guides. This function should be the preferred method whenever visual precision is a priority over simple type conversion.

## Choosing the Right Method: UI vs. DAX

Deciding between changing the [data type](#) via the UI and creating a calculated column using DAX hinges on two factors: the need for customization and the need to preserve the original date column's calculation capabilities. The UI method is best viewed as a quick, destructive transformation--it alters the source column's type permanently. This is acceptable if the original date column is no longer needed for time intelligence or numeric sorting.

Conversely, the DAX approach, whether using `CONVERT` or `FORMAT`, is non-destructive. It adds an entirely new column to the model, ensuring that the source date column remains available for all computational tasks. While this increases the overall column count in your model, the benefit of preserved data integrity and enhanced reporting flexibility almost always outweighs the minor storage increase. It is generally considered best practice in Power BI modeling to preserve base data types and use DAX calculated columns for derived or presentation-specific transformations.

## Conclusion and Further Reading

Successfully converting dates to text in Power BI involves choosing the appropriate tool for the job. For fast, non-customized conversions, the Column Tools UI dropdown is effective. For scenarios requiring precise formatting or the preservation of the original date field for analysis, mastering the `FORMAT` and `CONVERT` functions in [DAX](#) is essential. These techniques ensure data is correctly prepared for both analysis and elegant presentation within your reports.

**Note:** You can find the complete documentation for the [CONVERT function](#) in DAX on the official Microsoft documentation pages, detailing all compatible data types and syntax requirements.

The following tutorials explain how to perform other common tasks in Power BI: