



Using Date and Date/Time in Formulas

Salesforce, Spring '17



CONTENTS

Using Date and Date/Time Values in Formulas	1
Sample Date Formulas	4

USING DATE AND DATE/TIME VALUES IN FORMULAS

Date formulas are useful for managing payment deadlines, contract ages, or any other features of your organization that are time or date dependent.

Two data types are used for working with dates: Date and Date/Time. Most values that are used when working with dates are of the Date data type, which store the year, month, and day. Some fields, such as `CreatedDate`, are Date/Time fields, meaning they not only store a date value, but also a time value (stored in GMT but displayed in the users' time zone). Date and Date/Time fields are formatted in the user's locale when viewed in reports and record detail pages.

You can use operations like addition and subtraction on Date and Date/Time values to calculate a future date or elapsed time between two dates. If you subtract one date from another, for example, the resulting value will be the difference between the two initial values in days (Number data type). The same operation between two Date/Time values returns a decimal value indicating the difference in number of days, hours, and minutes.

For example, if the difference between two Date/Time values is 5.52, that means the two values are separated by five days, 12 hours (0.5 of a day), and 28 minutes (0.02 of a day). You can also add numeric values to Dates and Date/Times. For example, the operation `TODAY () + 3` returns three days after today's date. For more information and examples of working with dates, see the list of [Sample Date Formulas](#).

Throughout the examples, the variables `date` and `date/time` are used in place of actual Date and Date/Time fields or values.

Keep in mind that complex date functions tend to compile to a larger size than text or number formula functions, so you might run into issues with formula compile size. See [Tips for Reducing Formula Size](#) for help with this problem.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

TODAY() and NOW()

The `TODAY ()` function returns the current day, month, and year as a Date data type. This function is useful for formulas where you are concerned with how many days have passed since a previous date, the date of a certain number of days in the future, or if you just want to display the current date.

The `NOW ()` function returns the Date/Time value of the current moment. It's useful when you are concerned with specific times of day as well as the date.

For details on how to convert between Date values and Date/Time values, see [Converting Between Date/Time and Date](#).

The DATE() Function

The `DATE ()` function returns a Date value, given a year, month, and day. Numerical Y/M/D values and the `YEAR ()`, `MONTH ()`, and `DAY ()` functions are valid parameters for `DATE ()`. For example `DATE (2013, 6, 1)` returns June 6, 2013. Similarly, `DATE (YEAR (TODAY ()), MONTH (TODAY ()) + 3, 1)` returns the Date value of the first day three months from today in the current year, assuming the date is valid (for example, the month falls between 1 and 12).

If the inputted Y/M/D values result in an invalid date, the `DATE ()` function returns an error, so error checking is an important part of working with Date values. You can read about methods for handling invalid dates in [Sample Date Formulas](#).

Converting Between Date/Time and Date

Date and Date/Time aren't interchangeable data types, so when you want to perform operations between Date and Date/Time values, you need to convert the values so they are both the same type. Some functions (such as `YEAR()`, `MONTH()`, and `DAY()`) also only work on Date values, so Date/Time values must be converted first.

Use the `DATEVALUE (date/time)` function to return the Date value of a Date/Time. For example, to get the year from a Date/Time, use `YEAR (DATEVALUE (date/time))`.

You can convert a Date value to a Date/Time using the `DATETIMEVALUE (date)` function. The time will be set to 12:00 a.m. in Greenwich Mean Time (GMT), and then converted to the time zone of the user viewing the record when it's displayed. For a user located in San Francisco, `DATETIMEVALUE (TODAY ())` returns 5:00 p.m. on the previous day (during Daylight Saving Time) rather than 12:00 a.m. of the current day. See [A Note About Date/Time and Time Zones](#) for more information.

Converting Between Date and Text

If you want to include a date as part of a string, wrap the Date value in the `TEXT ()` function to convert it to text. For example, if you want to return today's date as text, use:

```
"Today's date is " & TEXT ( TODAY () )
```

This returns the date in the format "YYYY-MM-DD" rather than in the locale-dependent format. You can change the format by extracting the day, month, and year from the date first and then recombining them in the format you want. For example:

```
"Today's date is " & TEXT ( MONTH ( date ) ) & "/" & TEXT ( DAY ( date ) ) & "/" & TEXT ( YEAR ( date ) )
```

You can also convert text to a Date so you can use the string value with your other Date fields and formulas. You'll want your text to be formatted as "YYYY-MM-DD". Use this formula to return the Date value:

```
DATEVALUE ( "YYYY-MM-DD" )
```

Converting Between Date/Time and Text

You can include Date/Time values in a string using the `TEXT ()` function, but you need to be careful of time zones. For example, consider this formula:

```
"The current date and time is " & TEXT ( NOW () )
```

In this formula, `NOW ()` is offset to GMT. Normally, `NOW ()` would be converted to the user's time zone when viewed, but because it's been converted to text, the conversion won't happen. So if you execute this formula on August 1st at 5:00 PM in San Francisco time (GMT-7), the result is "The current date and time is 2013-08-02 00:00:00Z".

When you convert a Date/Time to text, a "Z" is included at the end to indicate GMT. `TEXT (date/time)` returns "Z" if the field is blank. So if the Date/Time value you're working with might be blank, check for this before converting to text:

```
IF (
  ISBLANK ( date/time ),
  "",
  TEXT ( date/time )
)
```

To convert a string to a Date/Time value, use `DATETIMEVALUE()` passing in a string in the format "YYYY-MM-DD HH:MM:SS". This method returns the Date/Time value in GMT.

A Note About Date/Time and Time Zones

Date and Date/Time values are stored in GMT. When a record is saved, field values are adjusted from the user's time zone to GMT, and then adjusted back to the viewer's time zone when displayed in record detail pages and reports. With Date conversions this doesn't pose a problem, since converting a Date/Time to a Date results in the same Date value.

When working with Date/Time fields and values, however, the conversion is always done in GMT, not the user's time zone. Subtracting a standard Date/Time field from another isn't a problem because both fields are in the same time zone. When one of the values in the calculation is a conversion from a Text or Date value to a Date/Time value, however, the results are different.

Let's say a San Francisco user enters a value of 12:00 AM on August 2, 2013 in a custom Date/Time field called `Date_Time_c`. This value is stored as 2013-08-02 07:00:00Z, because the time difference in Pacific Daylight Time is GMT-7. At 12:00 p.m. PDT on August 1st, the user views the record and the following formula is run:

```
Date_Time_c - NOW()
```

In the calculation, `NOW()` is 2013-08-01 19:00:00Z, and then subtracted from 2013-08-02 07:00:00Z, to return the expected result of 0.5 (12 hours).

Suppose that instead of `NOW()`, the formula converts the string "2013-08-01 12:00:00" to a Date/Time value:

```
Date_Time_c - DATETIMEVALUE( "2013-08-01 12:00:00" )
```

In this case, `DATETIMEVALUE("2013-08-01 12:00:00")` is 2013-08-01 12:00:00Z, and returns a result of 0.79167, or 19 hours.

There's no way to determine a user's time zone in a formula. If all of your users are in the same time zone, you can adjust the time zone difference by adding or subtracting the time difference between the users' time zone and GMT to your converted values. However, since time zones can be affected by Daylight Saving Time, and the start and end dates for DST are different each year, this is difficult to manage in a formula. We recommend using Apex for transactions that require converting between Date/Time values and Text or Date values.

SAMPLE DATE FORMULAS

Finding the Day, Month, or Year from a Date

Use the functions `DAY (date)`, `MONTH (date)`, and `YEAR (date)` to return their respective numerical values. Replace **date** with a value of type Date (e.g. `TODAY ()`).

To use these functions with Date/Time values, first convert them to a date with the `DATEVALUE ()` function. For example, `DAY (DATEVALUE (date/time))`.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Finding Out if a Year Is a Leap Year

This formula determines whether or not a year is a leap year. A year is only a leap year if it's divisible by 400, or if it's divisible by four but NOT by 100.

```
OR (
  MOD ( YEAR ( date ) , 400 ) = 0 ,
  AND (
    MOD ( YEAR ( date ) , 4 ) = 0 ,
    MOD ( YEAR ( date ) , 100 ) != 0
  )
)
```

Finding Which Quarter a Date Is In

For standard quarters, you can determine which quarter a date falls in using this formula. This formula returns the number of the quarter in which *date* falls (1–4) by dividing the current month by three (the number of months in each quarter) and taking the ceiling.

```
CEILING ( MONTH ( date ) / 3 )
```

The formula for shifted quarters is similar, but shifts the month of the date by the number of months between January and the first quarter of the fiscal year. The example below illustrates how you can find a date's quarter if Q1 starts in February instead of January.

```
CEILING ( ( MONTH ( date ) - 1 ) / 3 )
```

If you want to check whether a date is in the current quarter, add a check to compare the date's year and quarter with `TODAY ()`'s year and quarter.

```
AND (
  CEILING ( MONTH ( date ) / 3 ) = CEILING ( MONTH ( TODAY ( ) ) / 3 ) ,
  YEAR ( date ) = YEAR ( TODAY ( ) )
)
```


Finding the Week of the Year a Date Is In

To find the number of a date's week of the year, use this formula:

```
IF(
  CEILING( ( date - DATE( YEAR( date ), 1, 1) + 1) / 7) > 52,
  52,
  CEILING( ( date - DATE( YEAR( date ), 1, 1) + 1) / 7)
)
```

You can find the current week by determining how many days there have been in the current year and dividing that value by 7. The `IF()` statement ensures that the week number the formula returns doesn't exceed 52. So if the given date is December 31 of the given year, the formula returns 52, even though it's more than 52 weeks after the week of January.

Finding Whether Two Dates Are in the Same Month

To determine whether two Dates fall in the same month, say for a validation rule to determine whether an opportunity Close Date is in the current month, use this formula:

```
AND(
  MONTH( date_1 ) == MONTH( date_2 ),
  YEAR( date_1 ) == YEAR( date_2 )
)
```

Finding the Last Day of the Month

The easiest way to find the last day of a month is to find the first day of the next month and subtract a day.

```
IF(
  MONTH( date ) = 12,
  DATE( YEAR( date ), 12, 31 ),
  DATE( YEAR( date ), MONTH( date ) + 1, 1 ) - 1
)
```

Displaying the Month as a String Instead of a Number

To return the month as a text string instead of a number, use:

```
CASE (
  MONTH( date ),
  1, "January",
  2, "February",
  3, "March",
  4, "April",
  5, "May",
  6, "June",
  7, "July",
  8, "August",
  9, "September",
```

Sample Date Formulas

```
10, "October",  
11, "November",  
"December"  
)
```

If your organization uses multiple languages, you can replace the names of the month with a custom label:

```
CASE (  
  MONTH( date ),  
  1, $Label.Month_of_Year_1,  
  2, $Label.Month_of_Year_2,  
  3, $Label.Month_of_Year_3,  
  4, $Label.Month_of_Year_4,  
  5, $Label.Month_of_Year_5,  
  6, $Label.Month_of_Year_6,  
  7, $Label.Month_of_Year_7,  
  8, $Label.Month_of_Year_8,  
  9, $Label.Month_of_Year_9,  
  10, $Label.Month_of_Year_10,  
  11, $Label.Month_of_Year_11,  
  $Label.Month_of_Year_12  
)
```

Finding and Displaying the Day of the Week From a Date

To find the day of the week from a Date value, use a known Sunday (e.g. January 7, 1900) and subtract it from the date (e.g. TODAY ()) to get the difference in days. The MOD () function finds the remainder of this result when divided by 7 to give the numerical value of the day of the week between 0 (Sunday) and 6 (Saturday). The formula below finds the result and then returns the text name of that day.

```
CASE (  
  MOD( date - DATE( 1900, 1, 7 ), 7 ),  
  0, "Sunday",  
  1, "Monday",  
  2, "Tuesday",  
  3, "Wednesday",  
  4, "Thursday",  
  5, "Friday",  
  "Saturday"  
)
```

Note that this formula only works for dates after 01/07/1900. If you're working with older dates, use the same process with any Sunday prior to your earliest date (e.g. 01/05/1800).

You can also adjust this formula if your week starts on a different day. For example, if your week starts on Monday, you can use January 8, 1900 in your condition. The new formula looks like this:

```
CASE (  
  MOD( date - DATE( 1900, 1, 8 ), 7 ),  
  0, "Monday",  
  1, "Tuesday",  
  2, "Wednesday",  
  3, "Thursday",  
  4, "Friday",
```

Sample Date Formulas

```
5, "Saturday",  
"Sunday"  
)
```

Like the formula for getting the name of the month, if your organization uses multiple languages, you can replace the names of the day of the week with a variable like `$Label.Day_of_Week_1`, etc.

Finding the Next Day of the Week After a Date

To find the date of the next occurrence of a particular day of the week following a given `Date`, get the difference in the number of days of the week between a `date` and a `day_of_week`, a number 0–6 where 0 = Sunday and 6 = Saturday. By adding this difference to the current date, you'll find the date of the `day_of_week`. The `IF()` statement in this formula handles cases where the `day_of_week` is prior to the day of the week of the `date` value (e.g. `date` is a Thursday and `day_of_week` is a Monday) by adding 7 to the difference.

```
date + ( day_of_week - MOD( date - DATE( 1900, 1, 7 ), 7 ) )  
+  
IF(  
  MOD( date - DATE( 1900, 1, 7 ), 7 ) >= day_of_week,  
  7,  
  0  
)
```

You can substitute either a constant or another field in for the `day_of_week` value based on your needs.

Finding the Number of Days Between Two Dates

To find the number of days between two dates, `date_1` and `date_2`, subtract the earlier date from the later date: `date_1 - date_2`

You can alter this slightly if you want to determine a date a certain number of days in the past. For example, say you want a formula to return true if some date field is more than 30 days prior to the current date and false otherwise. This formula does just that:

```
TODAY() - 30 > date
```

Finding the Number of Business Days Between Two Dates

Calculating how many business days passed between two dates is slightly more complex than calculating total elapsed days. The basic strategy is to choose a reference Monday from the past and find out how many full weeks and any additional portion of a week have passed between the reference date and the date you're examining. These values are multiplied by five (for a five-day work week) and then the difference between them is taken to calculate business days.

```
(5 * ( FLOOR( ( date_1 - DATE( 1900, 1, 8 ) ) / 7 ) ) + MIN( 5, MOD( date_1 - DATE( 1900,  
1, 8 ), 7 ) ) )  
-  
(5 * ( FLOOR( ( date_2 - DATE( 1900, 1, 8 ) ) / 7 ) ) + MIN( 5, MOD( date_2 - DATE( 1900,  
1, 8 ), 7 ) ) )
```

In this formula, `date_1` is the more recent date and `date_2` is the earlier date. If your work week runs shorter or longer than five days, replace all fives in the formula with the length of your week.

Adding Days, Months, and Years to a Date

If you want to add a certain number of days to a date, add that number to the date directly. For example, to add five days to a date, the formula is **date** + 5.

Adding years to a date is fairly simple, but you do need to check that the future date is valid. That is, adding five years to February 29 (a leap year) results in an invalid date. The following formula adds *num_years* to *date* by checking if the date is February 29 and if the future date is not in a leap year. If these conditions hold true, the formula returns March 1 in the future year. Otherwise, the formula sets the Date to the same month and day *num_years* in the future.

```
IF (
  AND (
    MONTH( date ) = 2,
    DAY( date ) = 29,
    NOT (
      OR (
        MOD( YEAR( date ), 400 ) = 0,
        AND (
          MOD( YEAR( date ), 4 ) = 0,
          MOD( YEAR( date ), 100 ) != 0
        )
      )
    )
  ),
  DATE( YEAR( date ) + num_years, 3, 1 ),
  DATE( YEAR( date ) + num_years, MONTH( date ), DAY( date ) )
)
```

Adding months to a date is slightly more complicated as months vary in length and the cycle of months restart with each year. Therefore, a valid day in one month (January 31) might not be valid in another month (February 31). A simple solution is to approximate each month's length as 365/12 days:

```
date + ( ( 365 / 12 ) * Number_months )
```

While this formula is a good estimate, it doesn't return an exact date. For example, if you add two months to April 30 using this method, the formula will return June 29 instead of June 30. Returning an exact date depends on your organization's preference. For example, when you add one month to January 31, should it return February 28 (the last day of the next month) or March 2 (30 days after January 31)?

This formula does the following:

- Returns March 1 if the future month is a February and the day is greater than 28. This portion of the formula performs the same for both leap and non-leap years.
- Returns the first day of the next month if the future month is April, June, September, or November and the day is greater than 30.
- Otherwise, it returns the correct date in the future month.

This example formula adds two months to a given date. You can modify the conditions on this formula if you prefer different behaviors for dates at the end of the month.

```
DATE (
  YEAR( date ) + FLOOR( ( MONTH ( date ) + 2 - 1 ) / 12 ),
  MOD( MONTH ( date ) + 2 - 1 +
    IF( DAY ( date ) > CASE( MOD( MONTH( date ) + 2 - 1, 12 ) + 1,
      2, 28,
```

Sample Date Formulas

```
4, 30,  
6, 30,  
9, 30,  
11, 30,  
31 ), 1, 0 ), 12 ) + 1,  
IF( DAY( date ) > CASE( MOD( MONTH( date ) + 2 - 1, 12 ) + 1,  
2, 28,  
4, 30,  
6, 30,  
9, 30,  
11, 30,  
31 ),  
1, DAY( date )  
)  
)
```

If you're using these formulas for expiration dates, you might want to subtract a day from the return value to make sure that some action is completed *before* the calculated date.

Adding Business Days to a Date

This formula finds three business days from a given *date*.

```
CASE (  
  MOD( date - DATE( 1900, 1, 7 ), 7 ),  
  3, date + 2 + 3,  
  4, date + 2 + 3,  
  5, date + 2 + 3,  
  6, date + 1 + 3,  
  date + 3  
)
```

This formula finds the day of the week of the *date* field value. If the date is a Wednesday, Thursday, or Friday, the formula adds five calendar days (two weekend days, three weekdays) to the date to account for the weekend. If *date* is a Saturday, you need four additional calendar days. For any other day of the week (Sunday — Tuesday), simply add three days. You can easily modify this formula to add more or less business days. The tip for getting the day of the week might be useful if you need to adjust this formula.

Finding the Hour, Minute, or Second from a Date/Time

To get the hour, minute, and second from a Date/Time field as a numerical value, use the following formulas where *TZoffset* is the difference between the user's time zone and GMT. For hour in 24-hour format:

```
VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )
```

For hour in 12-hour format:

```
IF(  
  OR(  
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 0,  
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 12  
  ),  
  12,
```

Sample Date Formulas

```
VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )  
-  
IF(  
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,  
    0,  
    12  
)  
)
```

For minutes:

```
VALUE( MID( TEXT( date/time - TZoffset ), 15, 2 ) )
```

For seconds:

```
VALUE( MID( TEXT( date/time - TZoffset ), 18, 2 ) )
```

And, to get "AM" or "PM" as a string, use:

```
IF(  
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,  
    "AM",  
    "PM"  
)
```

To return the time as a string in "HH:MM:SS A/PM" format, use the following formula:

```
IF(  
    OR(  
        VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 0,  
        VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 12  
    ),  
    "12",  
    TEXT( VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )  
    -  
    IF(  
        VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,  
        0,  
        12  
    )  
    )  
)  
& ":" &  
MID( TEXT( date/time - TZoffset ), 15, 2 )  
& ":" &  
MID( TEXT( date/time - TZoffset ), 18, 2 )  
& " " &  
IF(  
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,  
    "AM",  
    "PM"  
)
```

When working with time in formula fields, you need to consider the time difference between your organization and GMT. See [A Note About Date/Time and Time Zones](#) on page 3 for help understanding the time zone offset used in this formula.

Finding the Elapsed Time Between Date/Times

To find the difference between two Date values as a number, subtract one from the other like so: **date_1** - **date_2** to return the difference in days.

Finding the elapsed time between two Date/Time values is slightly more complex. This formula converts the difference between two Date/Time values, **datetime_1** and **datetime_2**, to days, hours, and minutes.

```
IF(
  datetime_1 - datetime_2 > 0 ,
  TEXT( FLOOR( datetime_1 - datetime_2 ) ) & " days "
  & TEXT( FLOOR( MOD( (datetime_1 - datetime_2) * 24, 24 ) ) ) & " hours "
  & TEXT( ROUND( MOD( (datetime_1 - datetime_2) * 24 * 60, 60 ), 0 ) ) & " minutes",
  ""
)
```

Finding the Number of Business Hours Between Two Date/Times

The formula for finding business hours between two Date/Time values expands on the formula for finding elapsed business days. It works on the same principle of using a reference Date/Time, in this case 1/8/1900 at 16:00 GMT (9 a.m. PDT), and then finding your Dates' respective distances from that reference. The formula rounds the value it finds to the nearest hour and assumes an 8-hour, 9 a.m. – 5 p.m. work day.

```
ROUND( 8 * (
  ( 5 * FLOOR( ( DATEVALUE( date/time_1 ) - DATE( 1900, 1, 8 ) ) / 7 ) +
  MIN( 5,
    MOD( DATEVALUE( date/time_1 ) - DATE( 1900, 1, 8 ), 7 ) +
    MIN( 1, 24 / 8 * ( MOD( date/time_1 - DATETIMEVALUE( '1900-01-08 16:00:00' ), 1 ) )
  )
)
)
-
( 5 * FLOOR( ( DATEVALUE( date/time_2 ) - DATE( 1900, 1, 8 ) ) / 7 ) +
  MIN( 5,
    MOD( DATEVALUE( date/time_2 ) - DATE( 1996, 1, 1 ), 7 ) +
    MIN( 1, 24 / 8 * ( MOD( date/time_2 - DATETIMEVALUE( '1900-01-08 16:00:00' ), 1 ) )
  )
)
),
0 )
```

You can change the eights in the formula to account for a longer or shorter work day. If you live in a different time zone or your work day doesn't start at 9:00 a.m., change the reference time to the start of your work day in GMT. See [A Note About Date/Time and Time Zones](#) for more information.