**Add working days but avoid holidays**

[SteveH](https://www.sqlservercentral.com/author/steve-14359), 2019-07-10 (first published: 2019-06-26)

FROM: <https://www.sqlservercentral.com/scripts/add-working-days-but-avoid-holidays-3>

Very often there is a need to generate a date by adding (or subtracting) a number of working days. A process might have a lead time of N days and so a date which is N working days **after** another is required. Or perhaps an activity has to happen on a given date and a pre-activity date which is M working days **before** that has to be generated.

Straightforward enough if only weekends have to be considered - but what happens when statutory holidays or other non-working days get in the way?

The following Scalar Valued Function allows the user to generate days which are a number of working days either before or after a given date. It uses a table called tblHoliday in which each record represents a single non-working day. The table has an ID field called HolidayID and the date of the non-working day is held in a field called HolidayDate.

The Function will work however day-of-the-week-numbering has been set up since it finds the DW (day of the Week) values for Saturday and Sunday before performing a simple step-through-and-count loop which ignores weekends and holidays. This can easily be altered for countries where weekends are on other days of the week.

(The earlier version of this function would not work for holidays recorded as a series of contiguous dates. - This version does.).

CREATE FUNCTION [dbo].[fns\_AddWorkingDays]

(

@StartDate datetime,

@N INT

)

RETURNS datetime

AS

BEGIN

-- This ensures that however the server is configured for dates

-- the function will know the DATEPART(DW values for Saturday

-- and Sunday

declare @SaturdayDW int

declare @SundayDW int

set @SaturdayDW = DATEPART(DW,CONVERT(datetime,'2019 January 5')) -- A Saturday

set @SundayDW = DATEPART(DW,CONVERT(datetime,'2019 January 6')) -- A Sunday

-----------------------------------------------------------------

-- If @N is zero then reduce the date by 1

-- and try adding one day

if @N=0

begin

set @N=1

set @StartDate=DATEADD(DAY,-1,@StartDate)

end

----------------------------------------------------------------

-- If @N GTE 0 then increment dates while counting

-- If @N LT 0 then decrement dates while counting

declare @increment int

if @n>=0 set @increment = 1 else set @increment = -1

----------------------------------------------------------------

declare @CountDays int

set @CountDays=0

declare @LoopDate datetime

set @LoopDate = @StartDate

while @CountDays<ABS(@N)

begin

set @LoopDate=DATEADD(DAY,@increment,@LoopDate)

while exists(select HolidayID from tblHoliday where HolidayDate=@LoopDate)

or DATEPART(DW,@LoopDate)= @SaturdayDW

or DATEPART(DW,@LoopDate)= @SundayDW

begin

set @LoopDate=DATEADD(DAY,@increment,@LoopDate)

end

set @CountDays=@CountDays+1

end

return @LoopDate

END