

Excel that convert SF datetimes to Excel format and vice versa

```
Public Function ConvertSFDateTime(ctl As Range) As Variant

Dim strSFDateTime As String, strSFDatePortion As String, strSFTimePortion As String, strCtl As String

Dim sglCharT As Single, sglCharZ As Single, sglTimeLen As Single

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False
Application.EnableEvents = False
Application.DisplayStatusBar = False

strCtl = CStr(ctl.Value2)

If strCtl <> vbNullString Then

    ' +1 added, so that it returns the rightmost part of the string starting at the first
    character after "T"
    sglCharT = InStr(1, strCtl, "T") + 1
    sglCharZ = InStr(1, strCtl, "Z")
    sglTimeLen = sglCharZ - sglCharT - 1

    strSFDatePortion = Left(strCtl, 10)

    strSFTimePortion = Mid(strCtl, sglCharT, sglTimeLen)

    ConvertSFDateTime = Format(DateValue(strSFDatePortion) + TimeValue(Replace(strSFTimePortion,
    ".00", "")), _
    "yyyy-mm-dd hh:mm:ss")
```

```

Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True
Application.EnableEvents = True
Application.DisplayStatusBar = True

End If

End Function

Public Function ConvertToSFDateTime( cel As Range) As String

Dim strCel As String, strDatePart As String, strTimePart As String

Dim dateCel As Date

Dim sglFractionalLocation As Single

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False
Application.EnableEvents = False
Application.DisplayStatusBar = False

' VBA won't handle fractional seconds, finds the character number in the string where a period
occurs
sglFractionalLocation = InStr(1, cel, ".")

' Strips out the fractional second
If sglFractionalLocation > 0 Then
    strCel = Left(cel, sglFractionalLocation - 1)
Else
    strCel = cel
End If

If IsDate(strCel) Then
    strDatePart = DateValue(strCel)
    ' Determines if time component exists in the value of cel
    If DateValue(strCel) = CDate(strCel) Then
        ConvertToSFDateTime = Format(strCel, "yyyy-mm-dd")
    Else
        ConvertToSFDateTime = Format(strCel, "yyyy-mm-ddThh:mm:ssZ")
    End If
End If

```

```

    End If
End If

Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True
Application.EnableEvents = True
Application.DisplayStatusBar = True

End Function

Public Function ConcatenateMult(rngConcatenateCells As Range, bAddSingleSpace As Boolean, _
Optional strDelimiter As String, Optional strWrapCellValue As String)

Dim rngCel As Range, rngLastCel As Range

Dim bCompareCells As Boolean, bWrapCellValue As Boolean

Dim lngLastRow As Long, lngLastCol As Long

Dim strWrapTrue As String, strWrapFalse As String

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False
Application.EnableEvents = False
Application.DisplayStatusBar = False

' Sets the bounds for the last row and column from the rngConcatenateCells argument
lngLastRow = rngConcatenateCells.Row + rngConcatenateCells.Rows.Count - 1
lngLastCol = rngConcatenateCells.Column + rngConcatenateCells.Columns.Count - 1

Set rngLastCel = Cells(lngLastRow, lngLastCol)

' Loops through each cell in rngConcatenateCells
' If bAddSingleSpace is True, adds a single space in-between concatenated cells
' If optional strDelimiter argument is provided, adds delimiter between each concatenated
value
' If optional strWrapCellValue argument is provided, wraps each concatenated cell with that
delmiiter
For Each rngCel In rngConcatenateCells

```

```

bCompareCells = (rngCel.Address = rngLastCel.Address)

' Sets bWrapCellValue to true and applies strWrapCellValue,
' only if the length of the argument is greater than 0 characters
'
' NOTE: [Uses CStr(Replace(rngCel, "'", "'")) to add the escape "\"" character
' before all apostrophes, so errors don't occur with SQL queries
'
' Change this back to CStr(rngCel) if you need to concatenate values without the escape]
If Len(strWrapCellValue) > 0 Then
    bWrapCellValue = True
    strWrapTrue = strWrapCellValue & CStr(Replace(rngCel, "'", "'")) & strWrapCellValue
Else
    bWrapCellValue = False
    strWrapFalse = CStr(Replace(rngCel, "'", "'"))
End If

' bCompareCells determines if the cell's address is equal to the last cell's address in
the range
Select Case bCompareCells
    ' If true, the strDelimiter and strWrapCellValue arguments are not applied; function
ends
    Case Is = True
        If bWrapCellValue Then ConcatenateMult = ConcatenateMult & strWrapTrue & " "
        If Not bWrapCellValue Then ConcatenateMult = ConcatenateMult & strWrapFalse & " "

    ' If false, continues to concatenate cells
    Case Is = False
        Select Case Len(strDelimiter)
            Case Is = 0

                ' Adds a single space in between values
                If bAddSingleSpace Then
                    If bWrapCellValue Then ConcatenateMult = ConcatenateMult & strWrapTrue
& " "
                    If Not bWrapCellValue Then ConcatenateMult = ConcatenateMult &
strWrapFalse & " "
                Else
                    If bWrapCellValue Then ConcatenateMult = ConcatenateMult & strWrapTrue
                    If Not bWrapCellValue Then ConcatenateMult = ConcatenateMult &
strWrapFalse

```

```

        End If
    Case Else
        If bAddSingleSpace Then
            If bWrapCellValue Then ConcatenateMult = ConcatenateMult & strWrapTrue
& strDelimiter & " "
            If Not bWrapCellValue Then ConcatenateMult = ConcatenateMult &
strWrapFalse & strDelimiter & " "
        Else
            If bWrapCellValue Then ConcatenateMult = ConcatenateMult & strWrapTrue
& strDelimiter
            If Not bWrapCellValue Then ConcatenateMult = ConcatenateMult &
strWrapFalse & strDelimiter
        End If
    End Select
End Select
Next rngCel

' Hardcoded parentheses added for SFDC conversions -
' Apex Data Loader, Force.com Explorer, and other SOQL query tools require format of values
' in an IN clause as:
'
' ('[value1]', '[value2]', '[value3]', ..., '[value(n)]')
'
' (e.g. Select Id, Name FROM Contact WHERE Account.Name IN ('Kimberly-Clark', 'IBM'))
ConcatenateMult = "(" & Trim(ConcatenateMult) & ")"

Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True
Application.EnableEvents = True
Application.DisplayStatusBar = True

End Function

```

Revision #1

Created 14 August 2019 10:45:15 by Windyo

Updated 14 August 2019 10:46:41 by Windyo