

CORTEX/CPM

Complex Project Management
Sydney-Bangkok-Singapore

Macro for streaming MS-Project Gantt chart to a spreadsheet

User Notes v4.0

Luis Jimenez

jimenezl@cortexcpm.com

April 2020

1. Introduction

This macro creates a simple MS-Excel spreadsheet representation of a Gantt chart created in MS-Project.

There are many occasions in which project team members either do not have a license to use MS-Project or they are not fully familiar with the tool and therefore an alternative representation is required.

This tool is not intended to replace a properly constructed project schedule. Rather, the purpose is to facilitate the visualization and discussion of project tasks in a different presentation format.

This SOFTWARE PRODUCT is provided by THE PROVIDER "as is" and "with all faults." THE PROVIDER makes no representations or warranties of any kind concerning the safety, suitability, lack of viruses, inaccuracies, typographical errors, or other harmful components of this SOFTWARE PRODUCT. There are inherent dangers in the use of any software, and you are solely responsible for determining whether this SOFTWARE PRODUCT is compatible with your equipment and other software installed on your equipment. You are also solely responsible for the protection of your equipment and backup of your data, and THE PROVIDER will not be liable for any damages you may suffer in connection with using, modifying, or distributing this SOFTWARE PRODUCT.

2. Functional Specification

The macro is written in Visual Basic and it has the following functional characteristics:

1. Time scale is in days
2. The following fields are exported:
 - Task Name
 - WBS ID
 - Start Date
 - Finish Date
 - Duration
 - Resources
 - % Complete
3. Critical tasks are shown as red bars and non-critical tasks are shown as blue bars
4. Tasks duration is shown in whole days
5. Zero duration tasks (e.g. milestones) are shown as 1-day activities but in black color.
6. The start date for the display in the spreadsheet is the one declared in the 'Project Information' dialog box.

Project Information for 'Sample Project v4'

Start date: Wed 1/04/20 Current date: Tue 31/03/20

Finish date: Fri 24/04/20 Status date: NA

Schedule from: Project Start Date Calendar: Standard

All tasks begin as soon as possible. Priority: 500

Enterprise Custom Fields

Department:

Custom Field Name	Value
-------------------	-------

Help Statistics... OK Cancel

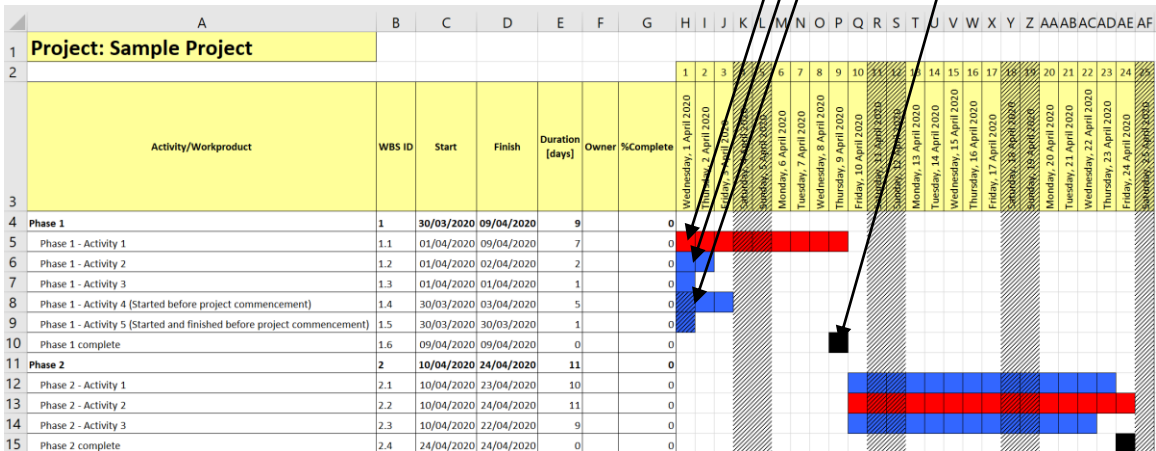
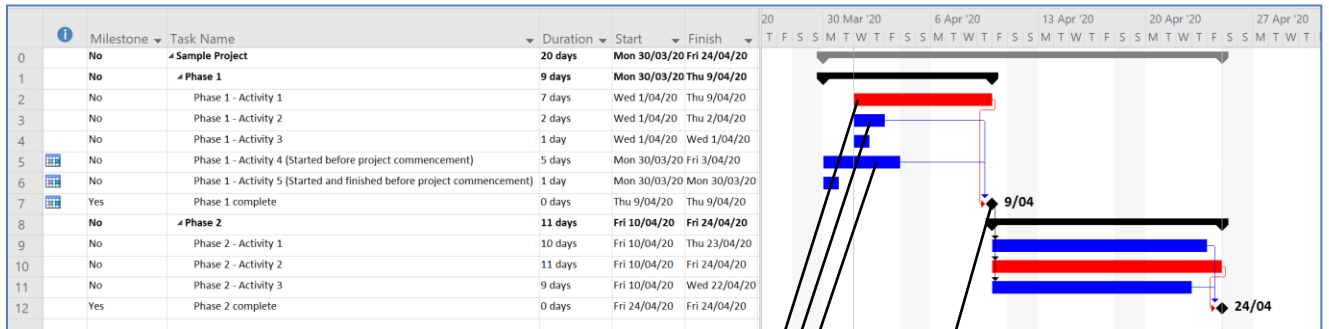
7. Tasks started before the project start date but not yet completed, are truncated to show only the section of the bar within the project start and end date. First day is shaded to signal that the task has an early start.
8. Tasks started and completed before the project start, are shown as a single shaded cell on the first date of the project.
9. The macro can handle projects with a duration of up to 730 days (2 years)

10. Only two level of indentation are shown (summary and operational). For additional information on the hierarchical structure please refer to the WBS ID column.

3. Example

The following diagram shows a MS-Project Gantt chart and its representation after streaming into an Excel spreadsheet.

Sample Project – MSP presentation

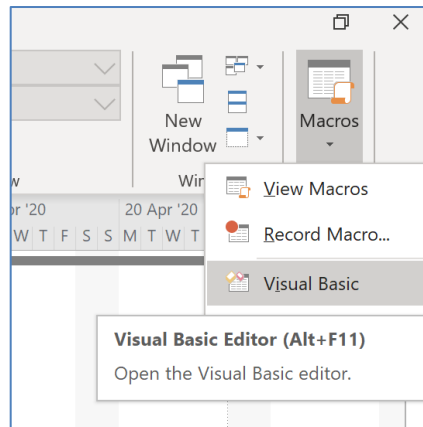


Sample Project – View in Excel format after running the macro

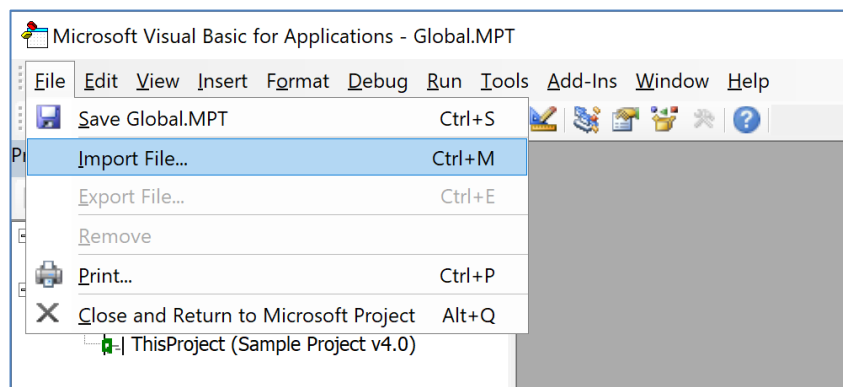
4. Installing of the Macro

1. Import the macro into your project file

The first step in the installation of the macro is to import the .bas file into your project file. For this you have to first open the Visual Basic Editor as shown in the following figure:

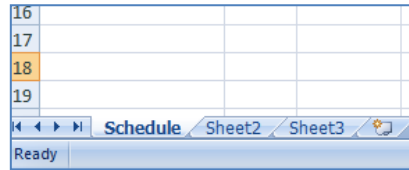


And then select the *Import* option from the Visual Basic menu to import the .bas file

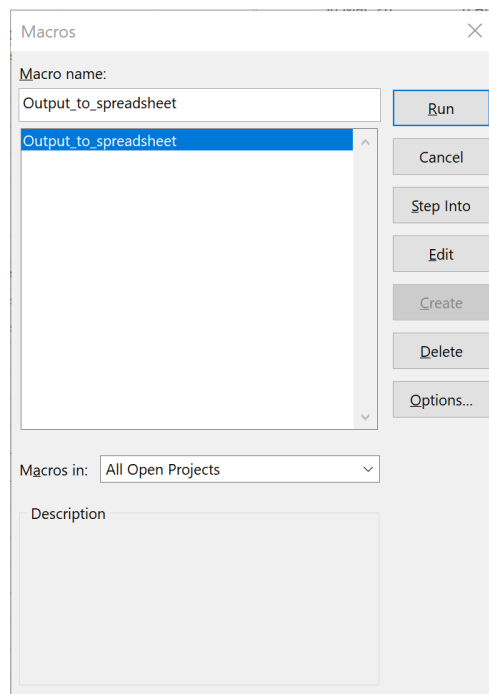


5. Running the macro

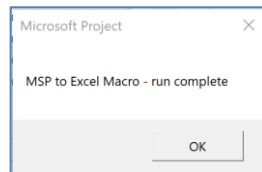
1. Start MS-Excel
2. Open or create a spreadsheet to receive the data.
3. Rename one of the tabs to 'Schedule' (without quotes).



4. Make sure this is the active spreadsheet
5. Go to Macros in the project file and select 'Output-to-spreadsheet' to run.



6. Wait till the 'MSP to Excel Macro - run complete' message appears.

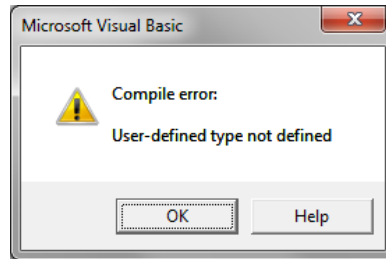


7. Verify results in the spreadsheet

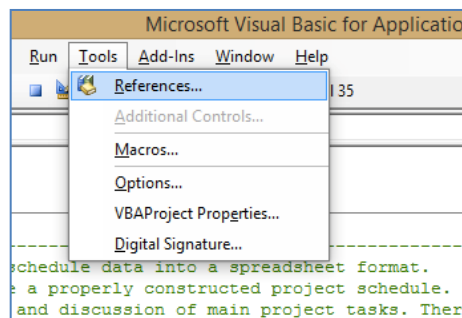
Note: if you wish to re-run the macro you must first delete all contents from the 'Schedule' sheet.

6. Potential Problem: Library Selection

This macro uses some MS library calls and therefore it requires the proper library to be selected. If you do not select the proper library you may get the following message:

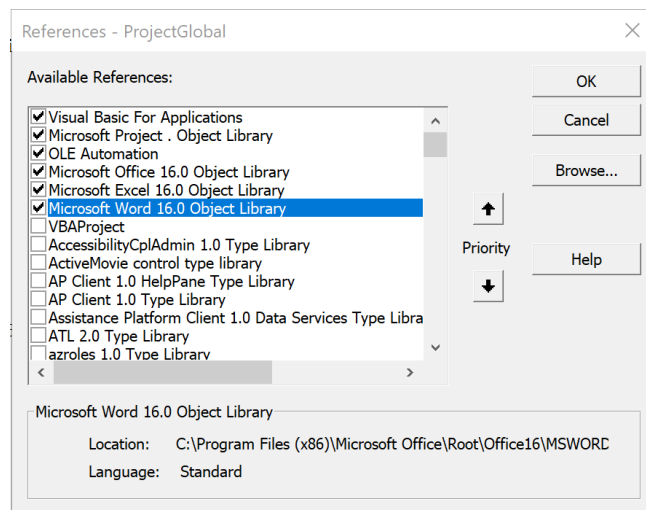


Step 1. Select the *References...* item from the *Tools* menu in Visual basic window in MS-Project



Step 2. Select the proper library for:

- Visual Basic
- Microsoft Project
- Microsoft Office
- Microsoft Excel
- Microsoft Word



7. Source Code

```
Option Explicit
Const MAX_DAYS = 1000
Const NUMBER_OF_HEADERS = 7
```

```
'-----
'This macro streams critical schedule data into a spreadsheet format.
'It is not intended to replace a properly constructed project schedule. Rather the purpose is to
'facilitate the visualisation and discussion of main project tasks. There are many occasions in which
'project team either do not have a license to use MS-Project or they are not fully familiar with the tool
```

```
'This SOFTWARE PRODUCT is provided by THE PROVIDER "as is" and "with all faults." THE PROVIDER makes
'no representations or warranties of any kind concerning the safety, suitability, lack of viruses, inaccuracies,
'typographical errors, or other harmful components of this SOFTWARE PRODUCT. There are inherent dangers in the
'use of any software, and you are solely responsible for determining whether this SOFTWARE PRODUCT is compatible
'with your equipment and other software installed on your equipment. You are also solely responsible for the
'protection of your equipment and backup of your data, and THE PROVIDER will not be liable for any damages you
'may suffer in connection with using, modifying, or distributing this SOFTWARE PRODUCT.'
```

```
'Version 4.0 1-04-2020
```

```
'-----
Sub Output_to_spreadsheet()
```

```
Dim ThisProject As Project
Dim DetailedTask As Task
Dim xlDateCell As Excel.Range
Dim xlDayNumberCell As Excel.Range
Dim StartDateOffset As Integer
Dim FinishDateOffset As Integer
Dim TaskColor As Long
Dim HeaderColor As Long
Dim NumberOfHeaders As Integer
Dim NumberOfColumns As Integer
Dim NumberOfRows As Integer
Dim TaskDuration As Integer
Dim xlBarRange As Excel.Range
Dim xlDateRange As Excel.Range
Set ThisProject = ActiveProject
Dim xlApp As Excel.Application
Dim xlBook As Excel.Workbook
Dim xlSheet As Excel.Worksheet
Dim xlRange As Excel.Range
Dim i As Integer
Dim EarlyTask As Boolean
```

```
'-----
'setup output spreadsheet
'-----
```

```
On Error GoTo 0
On Error Resume Next
Set xlApp = GetObject(, "Excel.application")
If Err.Number > 0 Then
    MsgBox ("Excel must be running. Run terminated")
    Exit Sub
End If
```

```
Set xlBook = xlApp.ActiveWorkbook
If xlBook Is Nothing Then
    MsgBox ("No active workbook. Run terminated")
    Exit Sub
End If
```

```
On Error GoTo 0
```



```

Set xlSheet = xlApp.ActiveSheet
On Error Resume Next
xlBook.Worksheets("Schedule").Activate
If Err.Number > 0 Then
    MsgBox ("No Schedule sheet. Run terminated")
    Exit Sub
End If

On Error GoTo 0
Set xlSheet = xlApp.ActiveSheet
On Error Resume Next

'-----
'Set up spreadsheet headings
'-----
HeaderColor = Word.wdColorLightYellow

Set xlRange = xlApp.ActiveSheet.Range("A3:A3")

With xlRange
    .Value = "Activity/Workproduct"
    .Font.Bold = True
    .Interior.Color = HeaderColor
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlBottom).Weight = xlThin
    .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

Set xlRange = xlRange.Offset(0, 1)
With xlRange
    .Value = "WBS ID"
    .Font.Bold = True
    .Interior.Color = HeaderColor
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
    .Columns.NumberFormat = "@"
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlBottom).Weight = xlThin
    .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

Set xlRange = xlRange.Offset(0, 1)
With xlRange
    .Value = "Start"
    .Font.Bold = True
    .Interior.Color = HeaderColor
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlBottom).Weight = xlThin
    .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

Set xlRange = xlRange.Offset(0, 1)
With xlRange
    .Value = "Finish"
    .Font.Bold = True
    .Interior.Color = HeaderColor
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlBottom).Weight = xlThin
    .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

Set xlRange = xlRange.Offset(0, 1)
With xlRange
    .Value = "Duration [days]"
    .Font.Bold = True
    .Interior.Color = HeaderColor
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic

```

```

        .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlBottom).Weight = xlThin
        .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
    End With

```

```

Set xlRange = xlRange.Offset(0, 1)
With xlRange
    .Value = "Owner"
    .Font.Bold = True
    .Interior.Color = HeaderColor
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlBottom).Weight = xlThin
    .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

Set xlRange = xlRange.Offset(0, 1)
With xlRange
    .Value = "%Complete"
    .Font.Bold = True
    .Interior.Color = HeaderColor
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlBottom).Weight = xlThin
    .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

NumberOfHeaders = 7

```

```

NumberOfColumns = 7

```

```

'-----
'Set date headings
'-----

```

```

Dim StartDate As Date
Dim ProjectDurationInDays As Integer

```

```

StartDate = ThisProject.ProjectStart

```

```

ProjectDurationInDays = DateDiff("d", ThisProject.ProjectStart, ThisProject.ProjectFinish)

```

```
ProjectDurationInDays = ProjectDurationInDays + 2
```

```
i = 0
```

```
Set xlDayNumberCell = xlApp.ActiveSheet.Range("A2:A2")
```

```
Set xlDayNumberCell = xlDayNumberCell.Offset(0, NumberOfHeaders - 1)
```

```
Dim ThisDate As Date
```

```
Dim ThisDay As Day
```

```
Dim WeekDayNumber As Integer
```

```
Do While i < ProjectDurationInDays
```

```
    Set xlRange = xlRange.Offset(0, 1)
```

```
    ThisDate = StartDate + i
```

```
    WeekDayNumber = Weekday(ThisDate)
```

```
    With xlRange
```

```
        .Value = StartDate + i
```

```
        .HorizontalAlignment = xlGeneral
```

```
        .VerticalAlignment = xlBottom
```

```
        .WrapText = False
```

```
        .Orientation = 90
```

```
        .AddIndent = False
```

```
        .IndentLevel = 0
```

```
        .ShrinkToFit = False
```

```
        .ReadingOrder = xlContext
```

```
        .MergeCells = False
```

```
        .Columns.AutoFit
```

```
        .Interior.Color = HeaderColor
```

```
        .NumberFormat = "[$-C09]dddd, d mmmm yyyy;@"
```

```
        .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
```

```
        .Borders(Excel.Constants.xlRight).Weight = xlThin
```

```
        .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
```

```
        .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
```

```
        .Borders(Excel.Constants.xlTop).Weight = xlThin
```

```
        .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
```

```
        .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
```

```
        .Borders(Excel.Constants.xlBottom).Weight = xlThin
```

```
        .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
```

```
    End With
```

```
    i = i + 1
```

```
Set xlDayNumberCell = xlDayNumberCell.Offset(0, 1)
```

```
With xlDayNumberCell
```

```
    .Value = i
```

```
    .HorizontalAlignment = xlCenter
```

```
    .VerticalAlignment = xlCenter
```

```
    .WrapText = False
```

```
    .AddIndent = False
```

```
    .IndentLevel = 0
```

```
    .ShrinkToFit = False
```

```
    .ReadingOrder = xlContext
```

```
    .MergeCells = False
```

```
    .Interior.Color = HeaderColor
```

```
    .Borders(Excel.Constants.xlLeft).LineStyle = xlContinuous
```

```
    .Borders(Excel.Constants.xlLeft).Weight = xlThin
```

```
    .Borders(Excel.Constants.xlLeft).ColorIndex = xlAutomatic
```

```
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
```

```
    .Borders(Excel.Constants.xlRight).Weight = xlThin
```

```
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
```

```
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
```

```
    .Borders(Excel.Constants.xlTop).Weight = xlThin
```

```
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
```

```
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
```

```

        .Borders(Excel.Constants.xlBottom).Weight = xlThin
        .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
    End With

    NumberOfColumns = NumberOfColumns + 1

Loop

'-----
'Output the project data
'-----
Dim WBSAsString As String

Set xlRange = xlApp.ActiveSheet.Range("a3:a3")

Set xlBarRange = xlApp.ActiveSheet.Range("a3:a3")

NumberOfRows = 3

For Each DetailedTask In ThisProject.Tasks

    Set xlRange = xlRange.Offset(1, 0)

    NumberOfRows = NumberOfRows + 1

    '-----
    'Task name
    '-----

    xlRange.Value = DetailedTask.Name
    If DetailedTask.Summary = True Then
        xlRange.Font.Bold = True
    Else
        xlRange.IndentLevel = 1
    End If

    With xlRange
        .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlRight).Weight = xlThin
        .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
        .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlTop).Weight = xlThin
        .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
        .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlBottom).Weight = xlThin
        .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
    End With

    '-----
    'WBS ID
    '-----

    Set xlRange = xlRange.Offset(0, 1)

    WBSAsString = DetailedTask.WBS

    xlRange.Value = " " & WBSAsString

    If DetailedTask.Summary = True Then
        xlRange.Font.Bold = True
    End If

    With xlRange
        .HorizontalAlignment = xlLeft
        .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlRight).Weight = xlThin
        .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    End With

```

```

.Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
.Borders(Excel.Constants.xlTop).Weight = xlThin
.Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
.Borders(Excel.Constants.xlBottom).Weight = xlThin
.Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

'-----
'Start date
'-----

```

```

Set xlRange = xlRange.Offset(0, 1)
xlRange.Value = DetailedTask.Start
xlRange.NumberFormat = "dd/mm/yyyy;@"

```

```

If DetailedTask.Summary = True Then
    xlRange.Font.Bold = True
End If

```

```

With xlRange
.Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
.Borders(Excel.Constants.xlRight).Weight = xlThin
.Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
.Borders(Excel.Constants.xlTop).Weight = xlThin
.Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
.Borders(Excel.Constants.xlBottom).Weight = xlThin
.Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

'-----
'Finish date
'-----

```

```

Set xlRange = xlRange.Offset(0, 1)
xlRange.Value = DetailedTask.Finish
xlRange.NumberFormat = "dd/mm/yyyy;@"

```

```

If DetailedTask.Summary = True Then
    xlRange.Font.Bold = True
End If

```

```

With xlRange
.Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
.Borders(Excel.Constants.xlRight).Weight = xlThin
.Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
.Borders(Excel.Constants.xlTop).Weight = xlThin
.Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
.Borders(Excel.Constants.xlBottom).Weight = xlThin
.Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

'-----
'Duration
'-----

```

```

Set xlRange = xlRange.Offset(0, 1)
xlRange.Value = DetailedTask.Duration / (60 * 8)
If DetailedTask.Summary = True Then
    xlRange.Font.Bold = True
End If

```

```

With xlRange
.Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
.Borders(Excel.Constants.xlRight).Weight = xlThin
.Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic

```

```

        .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlTop).Weight = xlThin
        .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
        .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlBottom).Weight = xlThin
        .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
    End With

```

```

'-----
'Resources
'-----

```

```

Set xlRange = xlRange.Offset(0, 1)
xlRange.Value = DetailedTask.ResourceNames
If DetailedTask.Summary = True Then
    xlRange.Font.Bold = True
End If

```

```

With xlRange
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlBottom).Weight = xlThin
    .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

'-----
'Percentage complete
'-----

```

```

Set xlRange = xlRange.Offset(0, 1)
xlRange.Value = DetailedTask.PercentComplete
If DetailedTask.Summary = True Then
    xlRange.Font.Bold = True
End If

```

```

With xlRange
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlRight).Weight = xlThin
    .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlTop).Weight = xlThin
    .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlBottom).Weight = xlThin
    .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

'-----
'move the bar range
'-----

```

```

Set xlBarRange = xlBarRange.Offset(1, 0)

```

```

'-----
'draw a bar for each task which is not a summary
'-----

```

```

EarlyTask = False

```

```

If DetailedTask.Summary = False Then

```

```

'-----

```

```

'search for the column offset for the start date
'-----
Set xlDateCell = xlApp.ActiveSheet.Range("a3:a3")

'if the start date of the task is earlier than the start of the project, then
'equates the the offset to the number of headers
If DateValue(DetailedTask.Start) < DateValue(ThisProject.ProjectStart) Then
    StartDateOffset = NUMBER_OF_HEADERS
    EarlyTask = True

Else

    StartDateOffset = 0 'NUMBER_OF_HEADERS '0
    Do While StartDateOffset <= (ProjectDurationInDays + NUMBER_OF_HEADERS) And (DateValue(xlDateCell.Value) <>
DateValue(DetailedTask.Start))
        Set xlDateCell = xlDateCell.Offset(0, 1)
        StartDateOffset = StartDateOffset + 1
    Loop

End If
'-----
'search for the column offset for the finish date
'-----
'if the start date of the task is earlier than the start of the project, then
'equates the the offset to the number of headers

If DateValue(DetailedTask.Finish) < DateValue(ThisProject.ProjectStart) Then
    FinishDateOffset = NUMBER_OF_HEADERS
    EarlyTask = True
Else
    Set xlDateCell = xlApp.ActiveSheet.Range("a3:a3")
    FinishDateOffset = 0 'NUMBER_OF_HEADERS '0
    Do While FinishDateOffset <= (ProjectDurationInDays + NUMBER_OF_HEADERS) And (DateValue(xlDateCell.Value) <>
DateValue(DetailedTask.Finish))
        Set xlDateCell = xlDateCell.Offset(0, 1)
        FinishDateOffset = FinishDateOffset + 1
    Loop
End If
TaskDuration = FinishDateOffset - StartDateOffset

'-----
'color selection for the task bar. Blue is normal, Red for critical tasks
'-----

If DetailedTask.Critical = True Then
    TaskColor = Word.wdColorRed
Else
    TaskColor = Word.wdColorLightBlue
End If

'-----
'Change the color if it is a milestone
'-----

'
    If TaskDuration = 0 Then
    If DetailedTask.Milestone = True Then

        ' TaskColor = Word.wdColorDarkY 'ellow
        TaskColor = Word.wdColorBlack
    End If

Set xlDateCell = xlApp.ActiveSheet.Range("a3:a3")
Set xlDateCell = xlDateCell.Offset(0, StartDateOffset)

Set xlBarRange = xlBarRange.Offset(0, StartDateOffset)
xlBarRange.Interior.Color = TaskColor

With xlBarRange
    .Borders(Excel.Constants.xlLeft).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlLeft).Weight = xlThin

```



```

        .Borders(Excel.Constants.xlLeft).ColorIndex = xlAutomatic
        .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlRight).Weight = xlThin
        .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
        .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlTop).Weight = xlThin
        .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
        .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlBottom).Weight = xlThin
        .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
    End With

    'Shade if early task

    If EarlyTask = True Then
        With xlBarRange.Interior
            .Pattern = xlLightUp
            .PatternColorIndex = xlAutomatic
        End With
    End If

    '-----
    'Shade if week end
    '-----

    WeekDayNumber = Weekday(xlDateCell.Value)

    If WeekDayNumber = 1 Or WeekDayNumber = 7 Then
        With xlBarRange.Interior
            .Pattern = xlLightUp
            .PatternColorIndex = xlAutomatic
        End With
    End If

    Do While TaskDuration > 0
        Set xlBarRange = xlBarRange.Offset(0, 1)
        xlBarRange.Font.Bold = True
        xlBarRange.Interior.Color = TaskColor

        Set xlDateCell = xlDateCell.Offset(0, 1)

        With xlBarRange
            .Borders(Excel.Constants.xlLeft).LineStyle = xlContinuous
            .Borders(Excel.Constants.xlLeft).Weight = xlThin
            .Borders(Excel.Constants.xlLeft).ColorIndex = xlAutomatic
            .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
            .Borders(Excel.Constants.xlRight).Weight = xlThin
            .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
            .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
            .Borders(Excel.Constants.xlTop).Weight = xlThin
            .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
            .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
            .Borders(Excel.Constants.xlBottom).Weight = xlThin
            .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
        End With

        TaskDuration = TaskDuration - 1
    Loop

    Set xlBarRange = xlBarRange.Offset(0, -1 * FinishDateOffset)
    Set xlDateCell = xlDateCell.Offset(0, -1 * FinishDateOffset)

    End If

skip1:

```

Skip2:

```
'-----
'Go back to column A
'-----
Set xlRange = xlRange.Offset(0, -1 * (NumberOfHeaders - 1))

Next DetailedTask

'-----
'Shade the weekend cells/columns
'-----

Set xlDateCell = xlApp.ActiveSheet.Range("a3:a3")
Set xlDateCell = xlDateCell.Offset(0, NumberOfHeaders)

Set xlBarRange = xlApp.ActiveSheet.Range("a3:a3")
Set xlBarRange = xlBarRange.Offset(0, NumberOfHeaders)

'Traverse all date headers/columns
Dim j As Integer

For j = NumberOfHeaders To (NumberOfColumns - 1)

'if it is saturday or sunday then shade the cell

    WeekDayNumber = Weekday(xlDateCell.Value)

    If WeekDayNumber = 1 Or WeekDayNumber = 7 Then
        With xlDateCell.Interior
            .Pattern = xlLightUp
            .PatternColorIndex = xlAutomatic
        End With

        Set xlBarRange = xlApp.ActiveSheet.Range("a2:a2")
        Set xlBarRange = xlBarRange.Offset(0, j)

        For i = 1 To NumberOfRows - 1
            With xlBarRange.Interior
                .Pattern = xlLightUp
                .PatternColorIndex = xlAutomatic
            End With

            Set xlBarRange = xlBarRange.Offset(1, 0)
        Next i
    End If

    Set xlDateCell = xlDateCell.Offset(0, 1)
Next j

'-----
'Format column properties such as width
'-----

Set xlBarRange = xlApp.ActiveSheet.Range("a:a")

For j = 1 To NumberOfColumns

    With xlBarRange
        .Columns.AutoFit
        .Font.Size = 8
    End With

    Set xlBarRange = xlBarRange.Offset(0, 1)
Next j

'-----
'Draw borders around last cells
```

```

'-----
'Set xlBarRange = xlApp.ActiveSheet.Range("a1:a1")
'
'
'For j = 1 To NumberOfColumns
'    With xlBarRange
'        .Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
'        .Borders(Excel.Constants.xlTop).Weight = xlThin
'        .Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
'    End With
'
'    Set xlBarRange = xlBarRange.Offset(0, 1)
'Next j

Set xlBarRange = xlApp.ActiveSheet.Range("a1:a1")

Set xlBarRange = xlBarRange.Offset(NumberOfRows - 1, 0)

For j = 1 To NumberOfColumns
    With xlBarRange
        .Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlBottom).Weight = xlThin
        .Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
    End With

    Set xlBarRange = xlBarRange.Offset(0, 1)
Next j

Set xlBarRange = xlApp.ActiveSheet.Range("a2:a2")

Set xlBarRange = xlBarRange.Offset(0, NumberOfColumns - 1)

For i = 1 To NumberOfRows - 1
    With xlBarRange
        .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
        .Borders(Excel.Constants.xlRight).Weight = xlThin
        .Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
    End With
    Set xlBarRange = xlBarRange.Offset(1, 0)
Next i

'-----
'Project title
'-----

Set xlRange = xlApp.ActiveSheet.Range("a1:a1")
With xlRange
    .Value = "Project: " & ThisProject.ProjectSummaryTask.Name
    .Font.Bold = True
    .Interior.Color = HeaderColor 'xlAutomatic
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
    .Font.Size = 16
    .Borders(Excel.Constants.xlLeft).LineStyle = xlContinuous
    .Borders(Excel.Constants.xlLeft).Weight = xlThin
    .Borders(Excel.Constants.xlLeft).ColorIndex = xlAutomatic
    .Borders(Excel.Constants.xlRight).LineStyle = xlContinuous

```

```

.Borders(Excel.Constants.xlRight).Weight = xlThin
.Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
.Borders(Excel.Constants.xlTop).Weight = xlThin
.Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
.Borders(Excel.Constants.xlBottom).Weight = xlThin
.Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

'-----
File Name
'-----

```

```

Set xlRange = xlRange.Offset(NumberOfRows + 1, 0)
With xlRange
.Value = "Source file path/name: " & ThisProject.FullName

```

```

.Font.Bold = True
.Interior.Color = xlAutomatic
.HorizontalAlignment = xlLeft
.VerticalAlignment = xlCenter
.WrapText = True
.Orientation = 0
.AddIndent = False
.IndentLevel = 0
.ShrinkToFit = False
.ReadingOrder = xlContext
.MergeCells = False
.Font.Size = 8
.Borders(Excel.Constants.xlLeft).LineStyle = xlContinuous
.Borders(Excel.Constants.xlLeft).Weight = xlThin
.Borders(Excel.Constants.xlLeft).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlRight).LineStyle = xlContinuous
.Borders(Excel.Constants.xlRight).Weight = xlThin
.Borders(Excel.Constants.xlRight).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlTop).LineStyle = xlContinuous
.Borders(Excel.Constants.xlTop).Weight = xlThin
.Borders(Excel.Constants.xlTop).ColorIndex = xlAutomatic
.Borders(Excel.Constants.xlBottom).LineStyle = xlContinuous
.Borders(Excel.Constants.xlBottom).Weight = xlThin
.Borders(Excel.Constants.xlBottom).ColorIndex = xlAutomatic
End With

```

```

MsgBox ("MSP to Excel Macro - run complete")

```

```

End Sub

```