

## IELM 231 IT for Logistics and Manufacturing

### Lab 8-9. Linking Visual Basic with MS Excel spreadsheets

#### Background:

Many companies often store data in Excel sheets, and sometimes in Databases (e.g. Access, SQL server, etc.) It is quite common that we need to collect some data from the DB to assist in making some type of management decisions. Since Excel is a popular spreadsheet program, we will learn how to combine the flexibility of a programming language with the classical data store (DB) and Excel.

#### Objective:

In this lab, we will learn the basics of linking VB with a DB (Access) and generate an Excel file. Further, the VB program will control the contents of the Excel file, including cell shapes, colors, data as well as mathematical formulas.

*The objective of this exercise* is to create an *EDD schedule* of some jobs in a Database, and use Excel to compute the machine utilization rate.

**Step 1.** In the earlier labs, you had created a DB table to store Jobs data. Add one column to this table, called: ReleaseDate (of type Date).

For a given Job, the earliest date that it can be started is the ReleaseDate.

**Step 2.** Modify the web-page created in the earlier lab to include one additional Text box input for Release date. Save it as a new web page.

**Step 3.** Modify the CGI program you created earlier to for the earlier lab. **You only need the part** that will **add a job to the Database**. Link this CGI program to the web page of step 2.

**Step 4.** Write a VB program that will:

1. Get all the jobs in Jobs Table, sorted by Due date,
2. Create an Excel file in which your VB program will construct a Gantt chart of the EDD schedule as follows
  - (i) We will use *Row number 5*, with *Column A* having the text: "**Machine 1**"
  - (ii) All columns in the time horizon of the Gantt chart (in this lab, 200 days), have width= 5points
  - (iii) The sequence of the tasks is the EDD sequence
  - (iv) The VB program must find out the true start date of each job.
  - (v) The cells of *Row 5*, i.e. the Gantt chart, must be filled as follows:
    - Each job has a different color (back-ground color of the cell)
    - 1 cell = 1 unit of time; if job 3 has process time = 5, then it will occupy 5 cells.
    - If a cell has a job, it should have the job number (integer) as value of the cell.
    - If a cell has no job in it, it will have background color white, and white text with value= -1.
  - (vi) The *cell F5* has value "**Makespan**"; the *cell F6* should contain the makespan (time of completion of last job).

(vii) The *cell G5* has value: "[Utilization](#)"

(viii) The *cell G6* has a formula that computes the current machine utilization (= total time the machine is used / makespan)

NOTE: Your VB program must input the formula to compute the Makespan into this cell, and Excel will compute and display the actual value.

**Step 5.** Compile the VB program as a file called OutputExcelGanttChart.exe

NOTE: this program is executed from the host machine, not from the web interface.

---

Below is a list of some useful Excel functions that you can use in the VB program

***(a) Open a workbook, open a worksheet, and name the worksheet:***

```
Dim AppXls As Excel.Application
Dim ObjWb As Excel.Workbook
Dim ObjWs As Excel.Worksheet

Set AppXls = CreateObject("Excel.Application")
Set ObjWb = AppXls.Workbooks.Add
Set ObjWs = ObjWb.Worksheets.Add

ObjWb.SaveAs ("C:\TestCreate.xls")
```

***(b) How to put data in a cell, format the cell, etc.:***

***‘ Go to a cell:***

```
Range("C5").Select
```

***‘ Select a group of cells:***

```
Range("C3:D5").Select
Rows("1:10").Select ' selects rows 1=10
Columns("P:R").Select ' Selects columns P, Q and R.
```

***‘ Put text or a formula into a cell:***

```
Selection.value = "Some text value"
Selection.Formula= "=A1*B2"
```

***‘ Set the font, font size, and style for a cell***

```
Selection.Font.Name = "Times New Roman"
Selection.Font.Size = 10
Selection.Font.Bold = True
```

***‘ Set the cell color***

***‘ Method 1: use one of the 56 preset colors (index value = 1..56) for Excel***

```
ActiveCell.Interior.ColorIndex = 36
```

*‘ Method 2: actually set the RGB values for a color; R, G, B can 0..255*  
Range("A1:A6").Interior.Color = RGB(200,160,35)

*‘ Set width of a column, height of row (col of active cell/cells:*  
ActiveCell.ColumnWidth = 70  
Selection.RowHeight = 41.25

*‘ Setting the borders of cells or ranges:*  
Selection.Borders(xlDiagonalDown).LineStyle = xlNone  
Selection.Borders(xlDiagonalUp).LineStyle = xlNone  
Selection.Borders(xlEdgeBottom).LineStyle = xlNone  
Selection.Borders(xlEdgeLeft).LineStyle = xlNone  
Selection.Borders(xlEdgeRight).LineStyle = xlNone  
Selection.Borders(xlEdgeTop).LineStyle = xlNone  
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone  
Selection.Borders(xlInsideVertical).LineStyle = xlNone  
With Selection.Borders(xlEdgeLeft)  
    .LineStyle = xlContinuous  
    .Weight = xlThin  
    .ColorIndex = xlAutomatic  
End With  
With Selection.Borders(xlEdgeTop)  
    .LineStyle = xlContinuous  
    .Weight = xlThin  
    .ColorIndex = xlAutomatic  
End With  
With Selection.Borders(xlEdgeBottom)  
    .LineStyle = xlContinuous  
    .Weight = xlThin  
    .ColorIndex = xlAutomatic  
End With  
With Selection.Borders(xlEdgeRight)  
    .LineStyle = xlContinuous  
    .Weight = xlThin  
    .ColorIndex = xlAutomatic  
End With

---