Author & Presenter
Devendra Prasad

Scope of Training & Discussion

- Excel Background and Object Model
- Cell Referencing (absolute / relative), Ranges & Named Range
- Validation Options and cell drop down
- 3 other Useful Function (match, address, indirect)
- VBA Programming
    - Record a Macro (relative code)
    - Debug a macro
    - Variables & Data Types
    - Functions and Procedures
    - Loops
    - UI Design
    - File Handling
    - Printing
    - increase processing performance
- Tricks for Spark Lines / Special Symbols
- Dynamic Pivot / Dynamic Charts
- OLE Concept (Architecture Bits)
    - Interacting with Office Apps
    - Web scrapping
    - Connecting to Database
    - Connecting to Unix Environment
    - Connecting to Web Services
    - Connecting to REST APIs

https://books.google.co.in/books?id=2I6bHP3WOIC&pg=PA63&hl=en#v=onepage&q&f=fa

Excel Background and Object Model

- **Microsoft Excel** is a spreadsheet developed by Microsoft for Microsoft Windows, Mac OS X, and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. It has been a very widely applied spreadsheet for these platforms, especially since version 5 in 1993, and it has replaced Lotus 1-2-3 as the industry standard for spreadsheet

- Similar to large collection of 2d array consisting of rows and columns

- Google has also developed google spreadsheet program over the web where a programmer can leverage the **Javascript** skills to generate awesome web page and cool web application reporting framework but that limitation in terms of exposting critical data/database over the http internet

- Used majorly for reporting on data that would range from simple to complex. Data can be mathematical or statistical

- VBA(dialect of Visual Basic) opens the world of programming to Naïve Users as well experts

- VBA was removed from Mac Excel 2008, as the developers did not believe that a timely release would allow porting the VBA engine natively to Mac OS X. VBA was restored in the next version, Mac Excel 2011, although the build lacks support for ActiveX objects, impacting some high level developer tools

- Macro Recorder here is great feature which the naïve users leverage to learn automation

- VBA code interacts with the spreadsheet through the Excel *Object Model*, a vocabulary identifying spreadsheet objects, and a set of supplied functions or *methods* that enable reading and writing to the spreadsheet and interaction with its users (for example, through custom toolbars or *command bars* and *message boxes*). User-created VBA subroutines execute these actions and operate like macros generated using the macro recorder, but are more flexible and efficient

- Excel users can access external data sources

- Excel can accept data in real time through several programming interfaces, which allow it to communicate with many data sources such as Bloomberg and Reuters (through addins such as Power Plus Pro)

- DDE : "Dynamic Data Exchange" uses the message passing mechanism in Windows to allow data to flow between Excel and other applications

- Alternatively, Microsoft Query provides ODBC-based browsing within Microsoft Excel

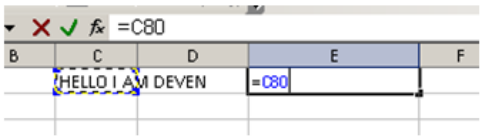
- There are APIs available through which excel interactions can be done e.g. Java POI, ExcelPackage, PHPExcel, Excel Service(.Net) allows programmers to leverage excel capabilities over server side programming
- Many vendors develop pluggin with in excel to give abstracted view of ther products e.g. smart view for essbase cube, SAS, Power Plus Pro, RBS Charts Addin

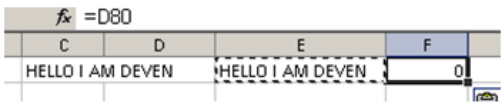
Cell Referencing

Relative Referencing

Pointing to a cell **without $ sign** means its relative and copying the formula cell to a new cell will change formula and it will behave relative based on the target cell

That's excel's default behavior

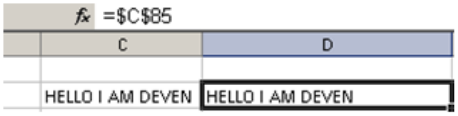


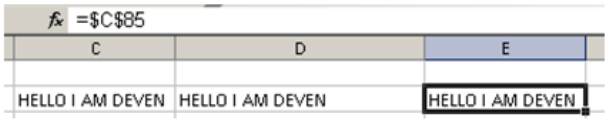


Absolute Referencing

Pointing to a cell **with $ sign** means its absolute and copying the formula cell to a new cell will not change formula and it will behave absolutely based on the source cell

That's excel's default behavior



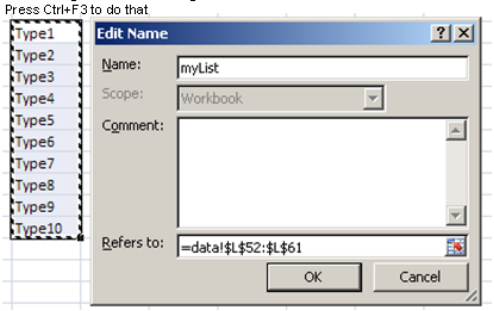


Validation and In cell dropdown

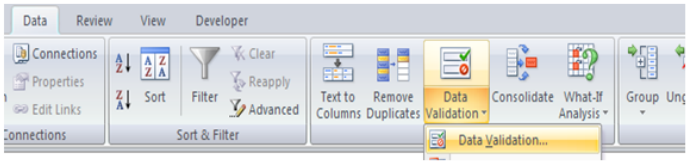
Get the name range manager by pressing ctrl+f3 and create a named range

In cell drop down can point to named range from any worksheet to any worksheet

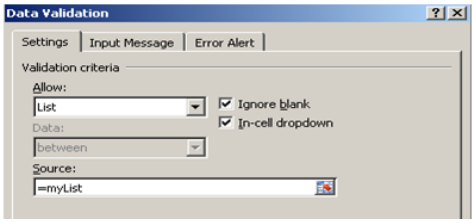
Select Range in excel and assign a name to it.
Press Ctrl+F3 to do that



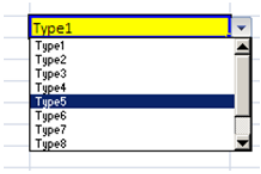
To create in cell dropdown, goto below menu



Select any cell where you want the drop down to appear, Choose list and assign the named range in source of data



Check your dropdown



3 Other Useful Functions

	010	020	030	040	050
010	7629.51	6310.00	936.17	2500.49	5738.16
020	8703.26	9638.54	1008.49	8201.39	1745.70
030	9667.14	4143.69	2539.20	2424.04	4893.54
040	4351.35	7791.66	2593.25	4200.85	8222.87
050	2988.79	6020.44	5580.31	2373.22	4342.45
060	2605.11	2070.59	539.77	1985.64	5295.59
070	1375.28	8073.79	9697.73	2885.38	6590.90

=51+MATCH(D62,C52:C58,0)

MATCH(lookup_value, **lookup_array**, [match_type])

=ADDRESS(E64,E65,,,"data")

ADDRESS(**row_num**, column_num, [abs_num], [a1], [sheet_text])

use of match address,indirect

	020
070	8073.79

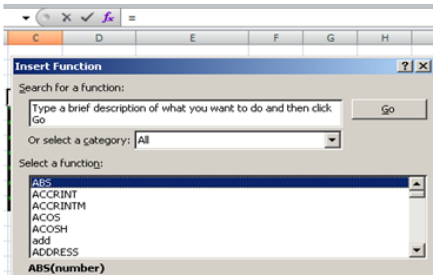
Match Row	58
Match Col	5
Address	data!E58
Indirect	8073.79

=INDIRECT(E66)

INDIRECT(**ref_text**, [a1])

MATCH

- Its different than vlookup / hlookup. Faster than these function as we just get the row / column reference which is INETEGER on selected range
- Unlike vlookup / hlookup, it doesn't depend on having primary key in first column in selected range or dataset
- It makes the use of point to point index like search
- If parameters given for lookup in a column, it will return row number
- If parameters given for lookup in a row, it will be return column number



ADDRESS

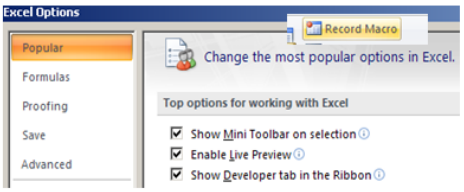
- once we get the row and column reference, we can get the address in excel recognized format with use of ADDRESS function or we can concatenate the two and make the cell address

INDIRECT

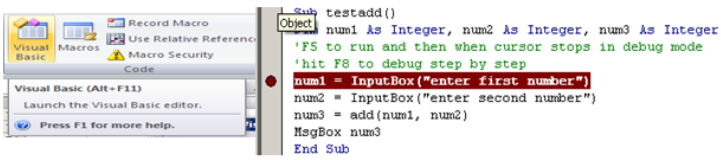
- once we have the cell address, I can get the value placed over that cell

VBA Programming

Record a Macro



Debug a Macro



Variables and Data Types

VBA's Built-in Data Types

Data Type	Bytes Used	Range of Values
Boolean	2	true or False
Integer	2	-32,768 to 32767
Long	4	-2,147,483,648 to 2,147,483,647
Single	4	-3.402823E38 to 1.401298E45
Double (negative)	8	-1.79769313486232E308 to -4.94065645841247E-324
Double (positive)	8	4.94065645841247E-324 to 1.79769313486232E308
Currency	8	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Date	8	1/1/100 to 12/31/9999
String	1 per character	Varies according to the number of characters
Object	4	Any defined object

Absolute and Relative Macro Programming

```
Sub relativeCheck()
Dim rng As Range
Set rng = Selection
rng.Value = "test"
End Sub
```

```
Sub absolueCheck()
Dim rng As Range
Set rng = Selection
Sheet1.Range(rng.Address).Value = "test"
End Sub
```

Loops (for, for each while)

```
Sub testForLoop()
For i = 1 To 1000
    Sheet1.Range("A" & i) = "Counter Value: " & i
Next
End Sub
```

```
Sub testForEach()
Dim rng As Range
'For Each rng In Sheet1.Range("Named Range Variable")
For Each rng In Sheet1.Range(Selection.Address)
    rng.Value = "For Each Test"
Next
End Sub
```

File Handling

```
Open path For Output As #1
For i = 1 To 100
    Print #1, data_marks.Range("a" & i) & "|" & data_marks.Range("b" & i)
Next
'Print #1, "my test file"
Close #1
```

```
Sub testReadFile()
cntr = 0
Open path For Input As #1
For i = 1 To 100
    cntr = cntr + 1
    Line Input #1, varStr
    Sheet1.Range("A" & cntr) = varStr
Next
End Sub
```

```
FileCopy src, dest
Name src As dest
Kill src
```

Increase Processing Performance

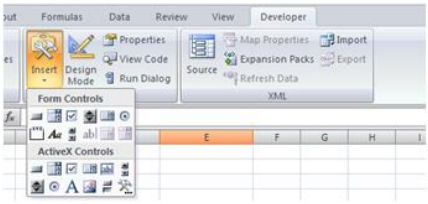
```
Sub testForPerformance()
On Error GoTo trap
    Application.ScreenUpdating = False
    Application.DisplayAlerts = False
    Application.Calculation = xlCalculationManual

    '--------------------------------
        'your macro code goes here....
    '--------------------------------

    Application.ScreenUpdating = True
    Application.DisplayAlerts = True
    Application.Calculation = xlCalculationAutomatic
Exit Sub
trap:
MsgBox Err.Description
    Application.ScreenUpdating = True
    Application.DisplayAlerts = True
    Application.Calculation = xlCalculationAutomatic
End Sub
```

Printing

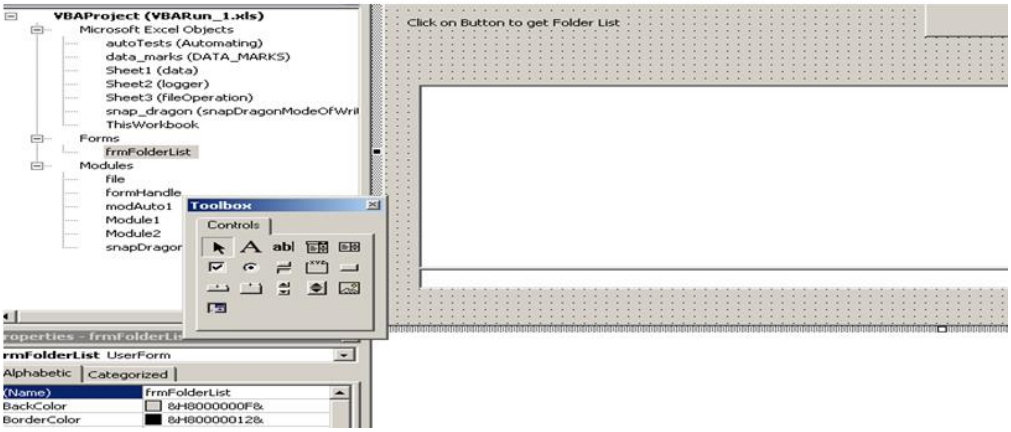
```
Sub getPrints()
    'ExecuteExcel4Macro "PRINT(1,,,1,,false,,,,,,2,,""Microsoft XPS Document Writer on Ne01:"",,TRUE,,FALSE)"
    'autoTests.Range("A1:D11").PrintOut Copies:=1, Preview:=False, ActivePrinter:="Microsoft XPS Document Writer
    autoTests.Range("A1:D11").PrintOut Copies:=1, Preview:=False, ActivePrinter:="\\delms00118\hp hpac"
End Sub
```

UI Design (form design + Sheet Design)

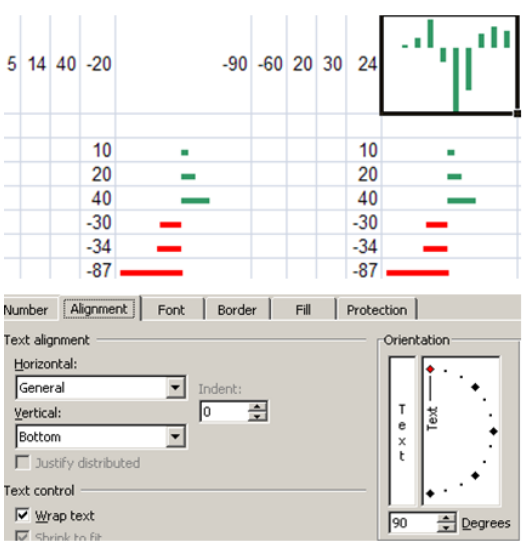
In excel sheet

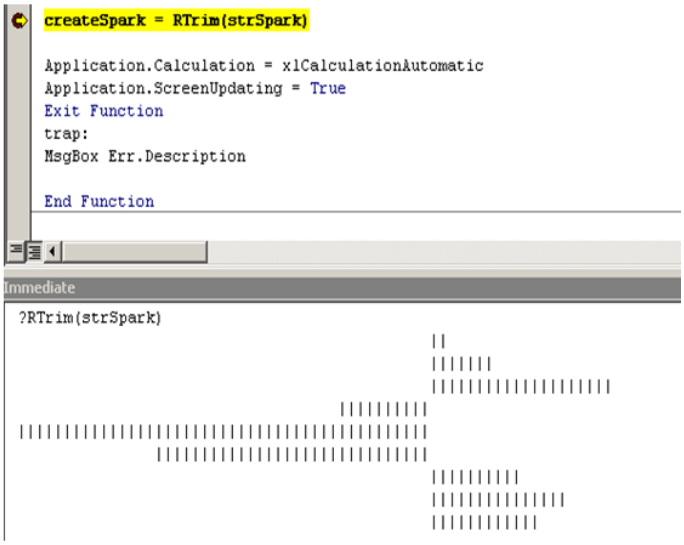


In VBA form model



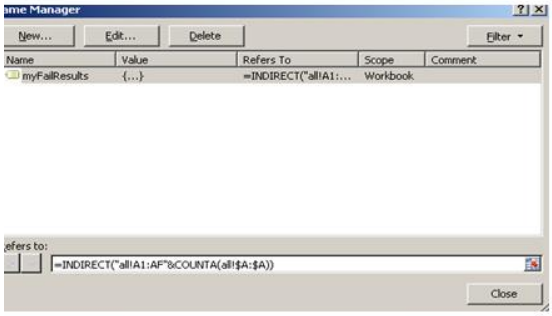
Spark Lines and Symbols

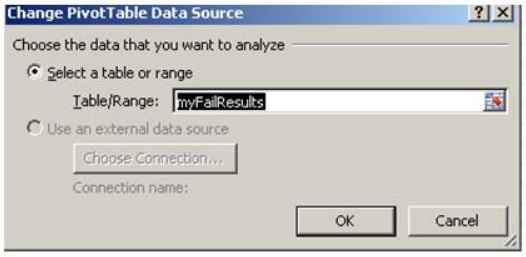




Wingdings 2 Font	Aharoni Font
✓	P

Dynamic Pivot

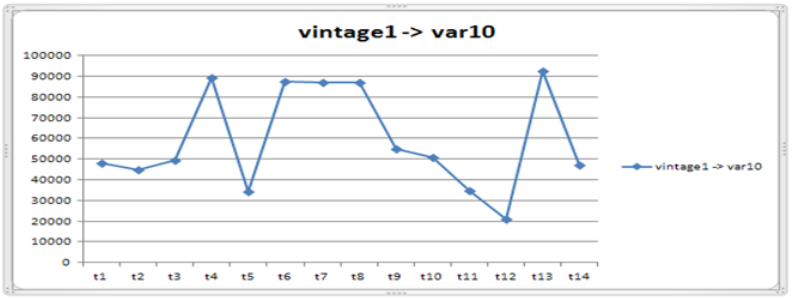




Dynamic Chart

SELECTION CRITERIA
vintage1
var10

Apply Changes

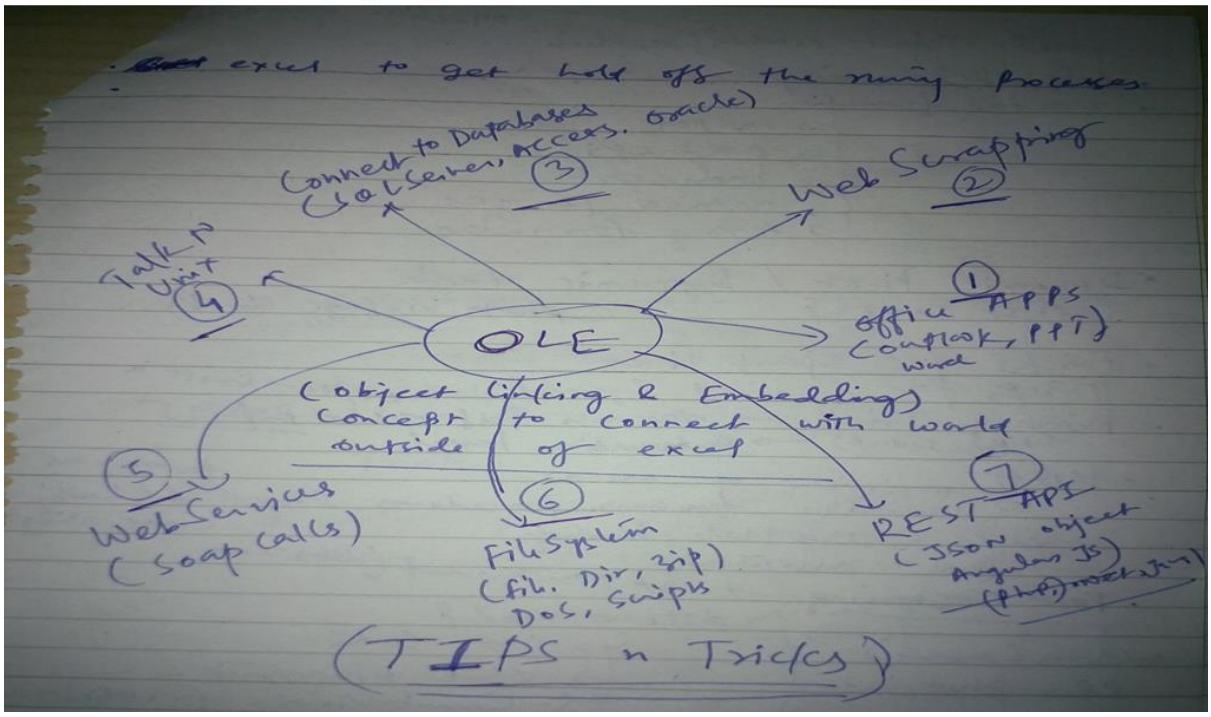


	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14
var10	47857.89	44664.84	49528.36	89472.54	34058.58	87590.25	87122.09	87117.75	54672.93	50741.24	34695.13	20948.87	92390.79	47177.42

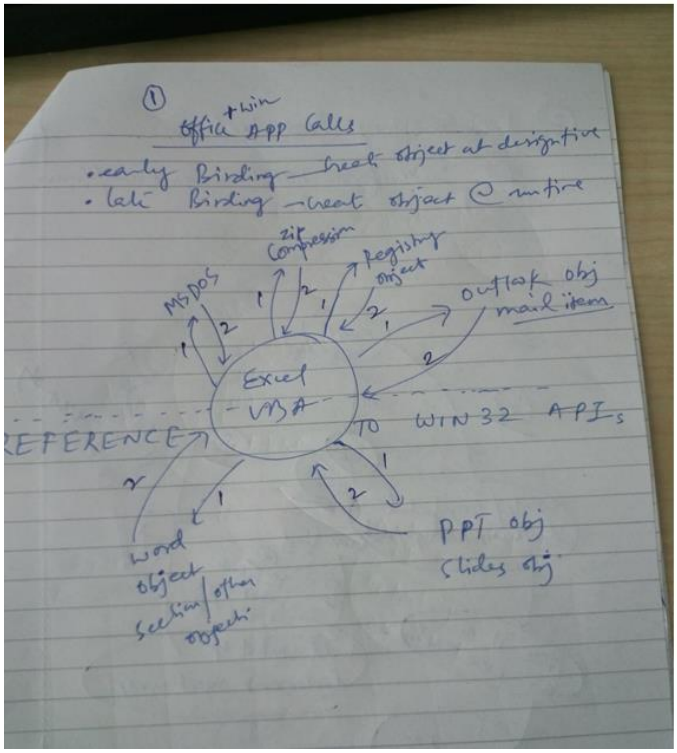
```
    'loop into the time scales in vintage sheet and plot the timeseries and data
    'clean old dumped values as well
    dbOccupiedCells = WorksheetFunction.CountA(dashBoard.Range("20:20")) + 1
    dbOccupiedAddress = Replace(Application.Evaluate("=address(1," & dbOccupiedCells & ")"), "$1", "")
    dashBoard.Range("B20:" & dbOccupiedAddress & "21").ClearContents
    For cnt = startCol To myCount
        dashBoard.Cells(timeScaleRowInDb, cnt) = sh.Cells(myTimeRow, cnt)
        dashBoard.Cells(dataRowInDb, cnt) = sh.Cells(myDataRow, cnt)
        DoEvents
    Next
```

```
Dim chrt As Chart, ser As Series
Dim dbOccupiedCells%, dbOccupiedAddress$
    Set chrt = dashBoard.ChartObjects("Chart 1").Chart
    dbOccupiedCells = WorksheetFunction.CountA(dashBoard.Range("20:20")) + 1
    dbOccupiedAddress = Replace(Application.Evaluate("=address(1," & dbOccupiedCells & ")"), "$1", "")

'     For cnt = 1 To chrt.SeriesCollection.Count
```

OLE (Object Linking & Embedding)



OLE – Office Apps




```
Sub sendOutlookEmail()
Dim OutApp As Object
Dim mail  As Object
Set OutApp = CreateObject("Outlook.Application")
Set mail = OutApp.CreateItem(0)
        With mail
            .To = "prasadf"
            .CC = ""
            .BCC = ""
            .Subject = "DP Mailer"
            .Body = "Hi there"
            .Attachments.add ThisWorkbook.path & "\DATA_MARKS.txt"
            .Send
        End With
Set OutMail = Nothing
Set OutApp = Nothing
End Sub
```

Prasad, Devendra, M&IB DP Mailer

Date: Older

DP Mailer

Prasad, Devendra, M&IB

Follow up
Completed on 09 December 2015 09:46.

To: Prasad, Devendra, M&IB

Attachments: DATA_MARKS.txt (5 KB)

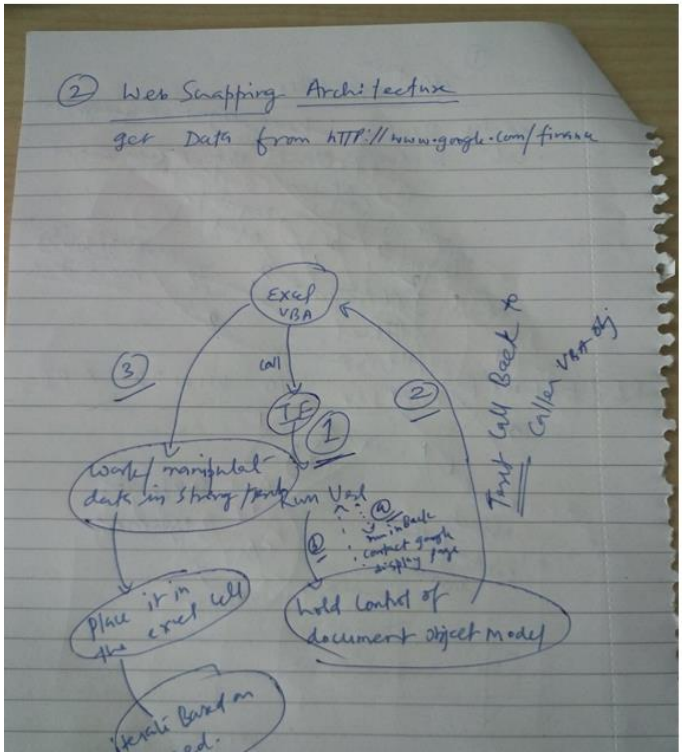
Hi there

Other Apps connectivity

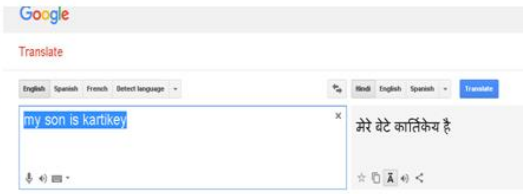
```
patn = patn & "\DATA_MARKS.txt"
Shell "notepad.exe " & path, vbNormalFocus
Shell "winword.exe " & path, vbNormalFocus
'Shell "excel.exe " & path, vbNormalFocus
'Shell "iexplore.exe " & path, vbNormalFocus
'Shell "d:\batTest.bat", vbNormalFocus
```

```
Sub checkZipFileCount()
Dim cnt%
Set sh = CreateObject("Shell.Application")
Set ZipFile = sh.Namespace(CVar("C:\DEVEN\shared\CRSE_2015120312020520_OKGKTYC.ZIP"))
For Each fileInZip In ZipFile.Items
    If LCase(fileInZip) Like LCase("*.DAT") Then
        cnt = cnt + 1
    End If
```

OLE – Web Scrapping



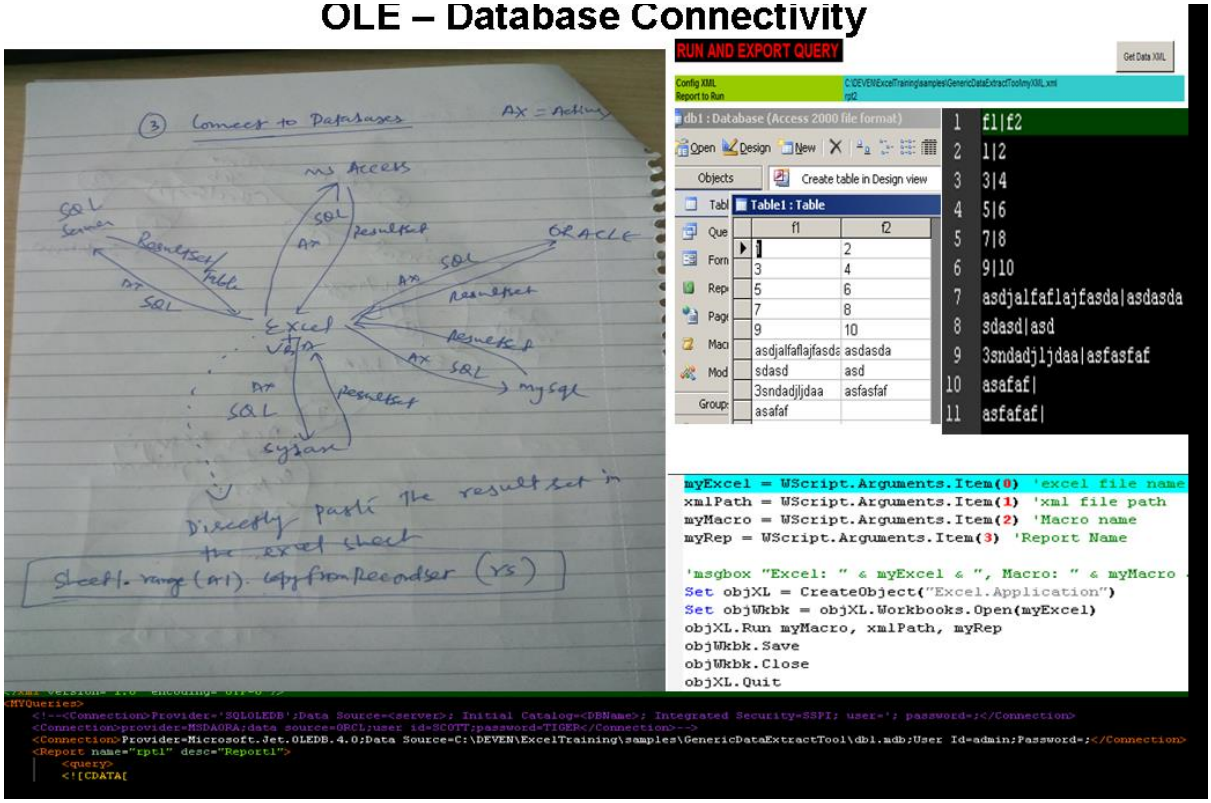
Text	From	To	Formula
my son is kartikey	en	hi	मेरे बेटे कार्तिकेय है
hello friend	en	hi	हैलो दोस्त
I am jon who are you	en	az	Mən Jon am siz olan
that is great	en	la	, quae magna
does this work?	en	af	werk dit?
I sure hope so	en	sk	Dúfam, že jo
Hey I am talking to myself	en	eu	Hey ari naiz neure
well isn't that odd	en	pl	dobrze, nie jest to, że kurs
ok I will stop now	en	tr	Tamam ben şimdi duracak



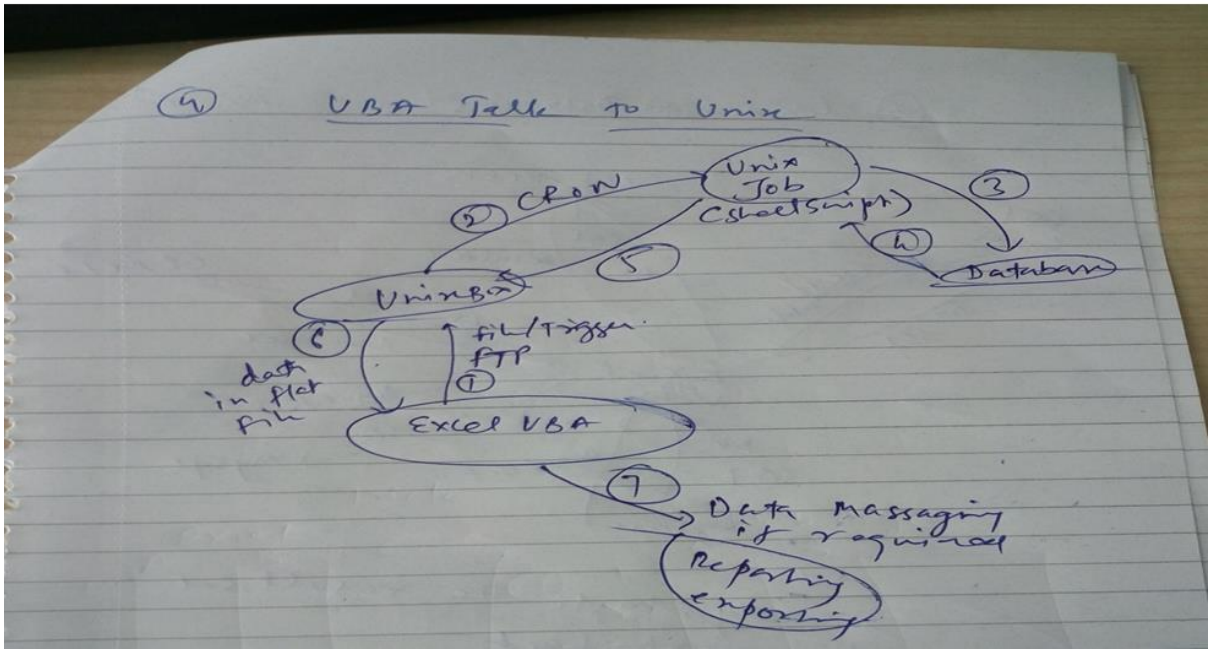
goog	GOOG	Alphabet Inc	763.25	-3.56	chr	-0.46	531.60B	NASDAQ	304466804484872	GOOG
bidu	BIDU	Baidu Inc (ADR)	208.54	+0.24	chg	0.12	71.99B	NASDAQ	700029	BIDU
fb	FB	Facebook Inc	105.61	-0.57	chr	-0.54	300.28B	NASDAQ	296878244325128	FB
twtr	TWTR	Twitter Inc	24.46	-0.56	chr	-2.24	17.09B	NYSE	32086821185414	TWTR

```
'make the url to call back
WebPage = "https://www.google.com/finance?q=" & tick
Application.StatusBar = WebPage
'make implicit call to IE or Chrome or firefox browser
ie.Navigate WebPage
'wait until pages load in the window
Application.Wait (Now + TimeValue("0:00:3"))
Do Until ie.ReadyState = 4
DoEvents
Application.Wait (Now + TimeValue("0:00:1"))
```

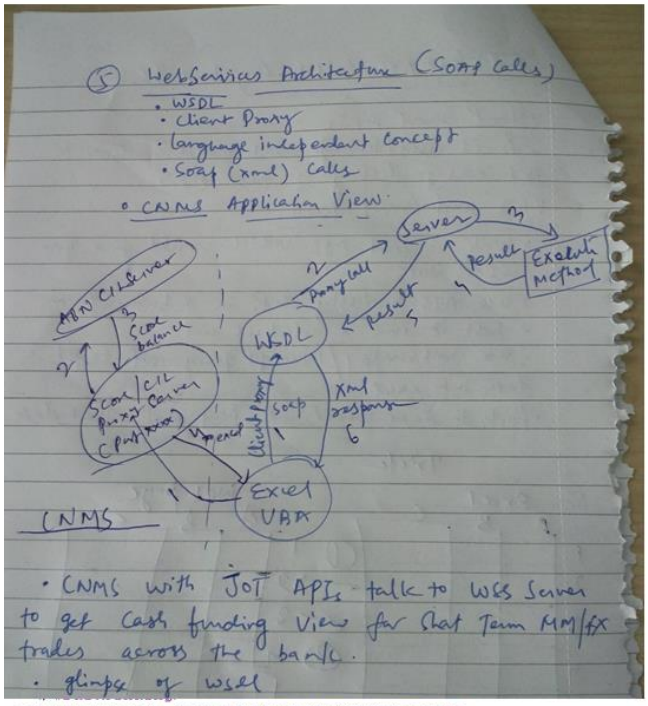
OLE – Database Connectivity



OLE - FTPs


VBA Talk To Unix
Unix Job (Shell Script)
CRON
Database
Unix Box
FTP
data in flat file
Excel VBA
Data massaging if required
Reporting exporting

OLE – SOAP(Simple Object Access Protocol) over http



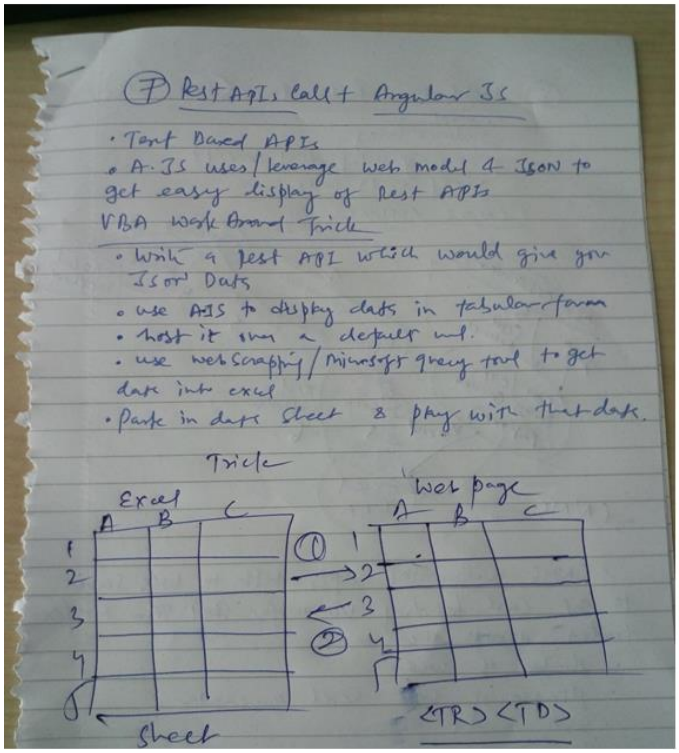

```
▼<s:element name="AuthenticateUser">
 ▼<s:complexType>
  ▼<s:sequence>
     <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string"/>
     <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string"/>
     <s:element minOccurs="0" maxOccurs="1" name="permission" type="s:string"/>
   </s:sequence>
  </s:complexType>
 </s:element>
```

```
Sub getDataFromWebService()
On Error GoTo trap
    Dim ws As New MSSOAPLib30.SoapClient30, soapWebServiceURL As String, fResult As Variant
    'soapWebServiceURL = "http://localhost/WebsiteProject/Service.asmx?WSDL"
    'soapWebServiceURL = "c:\TaxonNameService.wsdl"
    soapWebServiceURL = "c:\CNMSAuthenticationService.wsdl"
    ws.MSSoapInit soapWebServiceURL 'initialize SoapClient Object (URL of WSDL of the web service)
    fResult = ws.AuthenticateUser("prasadf", "Rnov2015", "WSS-Debit-Credit")
    Set ws = Nothing
    Sheet1.Range("A9") = "Web Response back from RBS SSO API @ " & Now
    Sheet1.Range("A10") = fResult
Exit Sub
trap:
MsgBox Err.Description
End Sub
```

Web Response back from RBS SSO API @ 10/12/2015 11:59:25
Xzpb/KpDD2mYKA/C8LY+Am0p6Y+dkXsiNw/s6kFt5o8yl4GoZF/cFdNl1nuEpwxs6Xdwvl

```
▼<wsdl:service name="AuthenticationService">
 ▼<wsdl:port name="AuthenticationServiceSoap" binding="tns:AuthenticationServiceSoap">
    <soap:address location="https://webservices.fm.rbsgrp.net/SecurityServices/v3/AuthenticationService.asmx"/>
```

OLE – REST APIs



file:///C:/jsonText.json

Apps Urls KTK

"jsonArr":{" ":{"----":[," "}

```
Sub testJSON()
On Error GoTo trap
    Dim sURL As String, sHTML As String, sAllPosts As String
    Dim oHttp As Object
    Dim blWSExists As Boolean
    Set oHttp = CreateObject("MSXML2.XMLHTTP") 'late binding call
    'sURL = "http://"
    sURL = "file:///C:/jsonText.json"
    oHttp.Open "POST", sURL, False
    oHttp.setRequestHeader "Content-type", "application/json"
    oHttp.setRequestHeader "Accept", "application/json"
    oHttp.Send '(mType = OPEN_SYSTEM_TRADE & systemOwnerId = 10)
    sHTML = oHttp.ResponseText
    Sheet1.Range("A1").Value = sHTML
Exit Sub
trap:
MsgBox Err.Description
End Sub
```

JSON is array like structure which is both more compact and (in my view) more readable - in transmission it can be "faster" simply because less data is transferred.
XML has more data text to represent a simple node
XML schema can be simple to very complex but JSON is just text based object notation with String or Number data type only

```
     "header": "SVG Viewer",
     "items": [
          {"id": "Open"},
          {"id": "OpenNew", "label": "Open New"},
          null,
          {"id": "ZoomIn", "label": "Zoom In"},
          {"id": "ZoomOut", "label": "Zoom Out"},
          {"id": "OriginalView", "label": "Original View"},
          null,
          {"id": "Quality"},
          {"id": "Pause"},
          {"id": "Mute"},
          null,
          {"id": "Find", "label": "Find..."},
          {"id": "FindAgain", "label": "Find Again"},
          {"id": "Copy"},
          {"id": "CopyAgain", "label": "Copy Again"},
          {"id": "CopySVG", "label": "Copy SVG"},
          {"id": "ViewSVG", "label": "View SVG"},
          {"id": "ViewSource", "label": "View Source"},
          {"id": "SaveAs", "label": "Save As"},
          null,
          {"id": "Help"},
          {"id": "About", "label": "About Adobe CVG Viewer..."}
     ]
}}
```

The same message expressed as XML:

```
<menu>
     <header>Adobe SVG Viewer</header>
     <item action="Open" id="Open">Open</item>
     <item action="OpenNew" id="OpenNew">Open New</item>
     <separator/>
     <item action="ZoomIn" id="ZoomIn">Zoom In</item>
     <item action="ZoomOut" id="ZoomOut">Zoom Out</item>
     <item action="OriginalView" id="OriginalView">Original View</item>
     <separator/>
     <item action="Quality" id="Quality">Quality</item>
     <item action="Pause" id="Pause">Pause</item>
     <item action="Mute" id="Mute">Mute</item>
     <separator/>
     <item action="Find" id="Find">Find...</item>
     <item action="FindAgain" id="FindAgain">Find Again</item>
```