Project Controls Expo 2016

Creating automations that gather, treat and present Project information ensuring its more efficient control

Pedro Manuel Barreto de Oliveira, Efacec Power Solutions, Portugal
pboliveira@efacec.com, +351 936064045

Subject Category: General Project Controls

Abstract

Microsoft has a programming language – the VBA – which is common to its programs (among which: Project, Access, Excel, PowerPoint and Outlook) and allows to automate tasks in each of them and to control any of those programs from any other.

VBA also has the capability of exchanging information with sites and servers on the web.

All this makes VBA a great programming language to automate long and complex procedures that gather, treat and present information relevant to a project.

Many non-MS programs allow to export information to Excel, so it can also be incorporated to the analysis performed by VBA.

This paper shows what can be done by VBA in order to automate the repetitive processes that present indicators of the behaviour of the projects and of each of their systems. It shows how this can be performed in a consistent way and with minimum human intervention, guaranteeing that these processes are not neglected in the busiest stages of the projects, when is most important to keep control of all projects aspects.

1 Brief author profile

Pedro Oliveira has a University Degree in Materials Physics and another in Electronics and Telecommunications Engineering and PMP Certification.

He started his career in 1989 as High School Teacher – he taught Mathematics, Physics and Chemistry for over 8 years. He then worked at Optimus - a Mobile Network Operator - for 11 years, and since 2009 he works at Efacec Power Solutions.

He works in PM since 2006, as Leader of: PM Group, Projects Implementation Department or Operational Planning Support.

He developed several MS tools and Android apps and was a speaker in 22 conferences on Telecommunications and Project Control.

2 Introduction

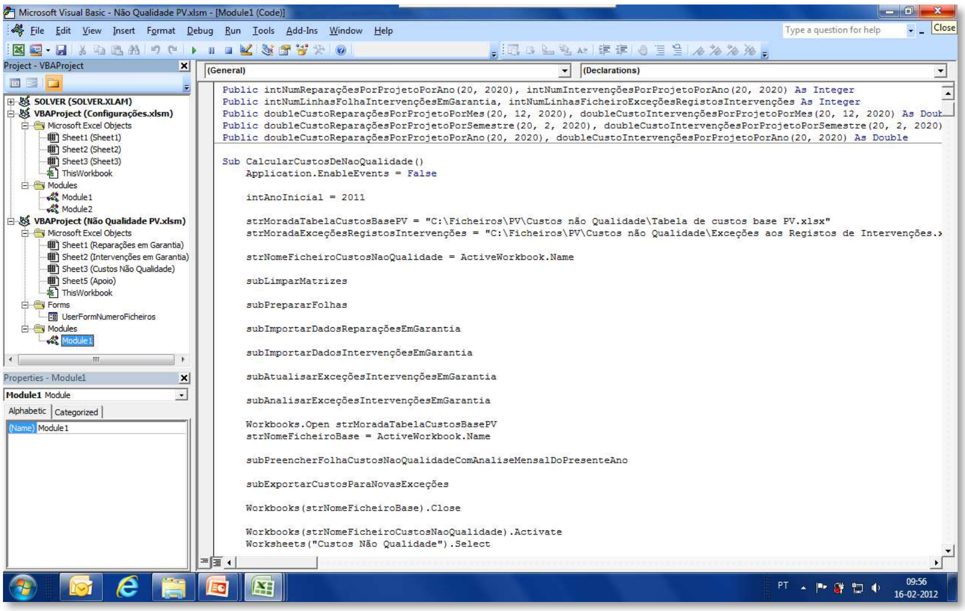
To effectively control a project, we need clear, accurate, consistent, relevant and in time information, but several factors hinder this:

- The dissemination of information by various tools, some of them proprietary and focused on particular systems, that do not communicate with the ERP
- Human errors in manual importation processes
- The amount of time to gather, treat and present the information, that typically prevents from having updated information and leads to the disregard of this process in the busiest and most challenging stages of the projects

The scope of this paper is to show how to create a sustainable process of repeatedly retrieving, treating and presenting the information from a set of tools in a quick, automatic and coherent way, that can be adapted to each case, using the programming language VBA.

3 VBA – Visual Basic for Applications

VBA is a programming language shared by several MS programs, among which: Project, Access, Excel, PowerPoint and Outlook. It is closely related to Visual Basic, but only runs code within a host program, rather than creating standalone applications.



```
Public intNumReparaçõesPorProjetoPorAno(20, 2020), intNumIntervençõesPorProjetoPorAno(20, 2020) As Integer
Public intNumLinhasFolhaIntervençõesEmGarantia, intNumLinhasFicheiroExceçõesRegistosIntervenções As Integer
Public doubleCustoReparaçõesPorProjetoPorMes(20, 12, 2020), doubleCustoIntervençõesPorProjetoPorMes(20, 12, 2020) As Douk
Public doubleCustoReparaçõesPorProjetoPorSemestre(20, 2, 2020), doubleCustoIntervençõesPorProjetoPorSemestre(20, 2, 2020)
Public doubleCustoReparaçõesPorProjetoPorAno(20, 2020), doubleCustoIntervençõesPorProjetoPorAno(20, 2020) As Double

Sub CalcularCustosDeNaoQualidade()
    Application.EnableEvents = False

    intAnoInicial = 2011

    strMoradaTabelaCustosBasePV = "C:\Ficheiros\PV\Custos não Qualidade\Tabela de custos base PV.xlsx"
    strMoradaExceçõesRegistosIntervenções = "C:\Ficheiros\PV\Custos não Qualidade\Exceções aos Registos de Intervenções.x

    strNomeFicheiroCustosNaoQualidade = ActiveWorkbook.Name

    subLimparMatrizes

    subPrepararFolhas

    subImportarDadosReparaçõesEmGarantia

    subImportarDadosIntervençõesEmGarantia

    subAtualisarExceçõesIntervençõesEmGarantia

    subAnalisarExceçõesIntervençõesEmGarantia

    Workbooks.Open strMoradaTabelaCustosBasePV
    strNomeFicheiroBase = ActiveWorkbook.Name

    subPreencherFolhaCustosNaoQualidadeComAnaliseMensalDoPresenteAno

    subExportarCustosParaNovasExceções

    Workbooks(strNomeFicheiroBase).Close

    Workbooks(strNomeFicheiroCustosNaoQualidade).Activate
    Worksheets("Custos Não Qualidade").Select
```

Fig. 1. VBA environment

VBA programming pieces are named macros and they can be used to control many aspects of those MS programs, including manipulating user interface features, such as menus and toolbars, and working with custom user forms or dialog boxes.

VBA can be used to perform just about everything we can perform manually in those MS programs and even tasks not possible to perform any other way, but probably the most important is that it can perform tasks which would be quite tedious or difficult to perform manually.

Beyond setting a sequence of actions in the program where it is embedded, a macro can open or create a file in any of the other four programs and act on it, performing the typical action of that other program.

A VBA macro can also exchange information with internet sites using Internet Explorer and use FTP to exchange information with servers.

It is therefore a very powerful tool and the core of this paper.

4 Gathering the information

If the information is in a Project, Access or Excel file, VBA can open that file and retrieve the relevant information. This can be performed even if the macro is embedded in a file from a distinct program (e.g., we can create a macro inside an Excel file that gets information from a Project file) – in this case, the macro will open an object of the distinct program and open the corresponding file, in order to get the desired data. The file with the data to be imported can be located in the same computer as the file with the macro, in a share or housed in a server – as long as the user has access to it, it can be automatically accessed and the corresponding information can be retrieved.

If the information is in a HTTP site (e.g., web tools, price quotes or exchange rates), it can also be automatically retrieved as long as the user has access to that site – if a login is needed, a manual action won't be needed as long as the user is comfortable in placing the login information in the VBA code.
There are two ways to get the information from a HTTP site using VBA:

- Import the complete web site to an Excel sheet and identify the cells that hold the desired values
- Identify the web site fields whose values we want to import, by analysing its HTML code, and instruct VBA to get their values

If the information is in a different program (ERP or other proprietary tools), typically VBA cannot access it directly. Here, a manual action is needed prior to running the VBA macro – the user should export the information from those tools to Excel files (most of the times these programs allow this type of data export). After this process is performed, the generated files can be automatically treated as referred above.

It is fundamental that the information to be fetched by a macro respects the structure the macro expects it to have. Examples of structures that should not change in Excel files to be imported are:

- the name of a sheet or its position within the file
- the name of a column or its position within the sheet
- the row where information starts
- the type of data contained in a cell

Macros should be prepared to some degree of innovation in the files they import, but it's impossible to cope with all the creativity users can have – macros need to make some assumptions on this matter.
In this respect, SW tools are far more trustworthy than their users: the more automated the processes are, the more we can trust the information they provide is consistent.
Another benefit of SW automation is the autonomy it provides: the information is always available when it's needed and one doesn't need to wait for someone to have the time and the will to provide it.

5 Treating the information

After gathering the information, the VBA macro can automatically treat it in all sorts of ways the user would do manually. The information can be verified, cross analysed, compiled, summarized and placed in arrays, tables or charts.
SQL commands can be launched on the Access DBs, to get the information quickly.

Time, cost and resources data can be taken from Project files.
Tables, charts, pivot tables and pivot charts can be automatically generated in Excel.

One advantage of this process when compared to the manual use of the programs is the time: all actions defined in the macro are performed sequentially without any loss of time. Another advantage is that the process is performed always in the same way, guaranteeing that two separate iterations over the same information have exactly the same output and that the output is consistent in time.

Despite sharing a programming language, each MS program has specificities in that language, as VBA must address the specific needs of each program and work with specific building blocks.

Project building blocks are:

- the project file
- the task
- the resource
- time scaled values

e.g., to change the name of the 23rd task of a project file we can write:

Project1.Tasks(23).Name = “Project Controls Expo 2016”

Access building blocks are:

- the database
- the table
- the recordset
- the record
- the field
- the report

e.g., to update the *conference* field of each record of a table whose year field has the value 2016 we can write:

DoCmd.RunSQL "update tableConferences set conference = ‘Project Controls Expo 2016' where year = 2016"

Excel building blocks are:

- the workbook
- the worksheet
- the cell
- other (chart, table, pivot table,…)

e.g., to change the value of the cell in the 4th row and 5th column of Sheet1 we can write:

Workbooks(Workbook1).Worksheets(“Sheet1”).Cells(4, 5).Value = “Project Controls Expo 2016”

6 Presenting the information

The information can be presented in Excel files with tables and charts automatically created by the VBA macro, but other, more suited for this purpose, possibilities exist.

The first one is of course a Presentation in PowerPoint. The VBA macro can:

- Create slides
- Add or change the text in slides
- Import Excel tables and charts to a slide

A PowerPoint template can be created and its information updated in every iteration of the macro. The presentation file could be sent to a email distribution list or placed in a share to which the target audience has access – both processes can be implemented within the macro and performed during its execution.

PowerPoint building blocks are:

- the presentation
- the slide
- the shape (picture, chart, text box,…)

e.g., to change the text inside the 2nd text box in slide 10 we can write:

Presentatios1.Slides(10).Shapes(2).TextFrame.TextRange.Text = “Project Controls Expo 2016”

Another possibility is to automatically update a HTTP site. An example is a Sharepoint site within an organization whose information (tables, charts or text) would automatically be updated in every iteration of the macro. The target audience would know the site address and could regularly visit it. An automatic email could be sent by the macro to that audience, informing of the site update.

The third possibility is to update a file hosted on a server, which is uploaded by a mobile device (tablet or smartphone) application, updating its charts and tables. This solution involves the creation of the app to be distributed to the target audience and allow an easier access to the information.

Any of these solutions guarantees a consistent presentation of the information – from one iteration to the other the information should change, but not its presentation and disposition.

7 Conclusion

It is important to have the project relevant information available and updated in order to control it and take the right actions in the right moment. VBA can be used to automatically exchange information with multiple parties:

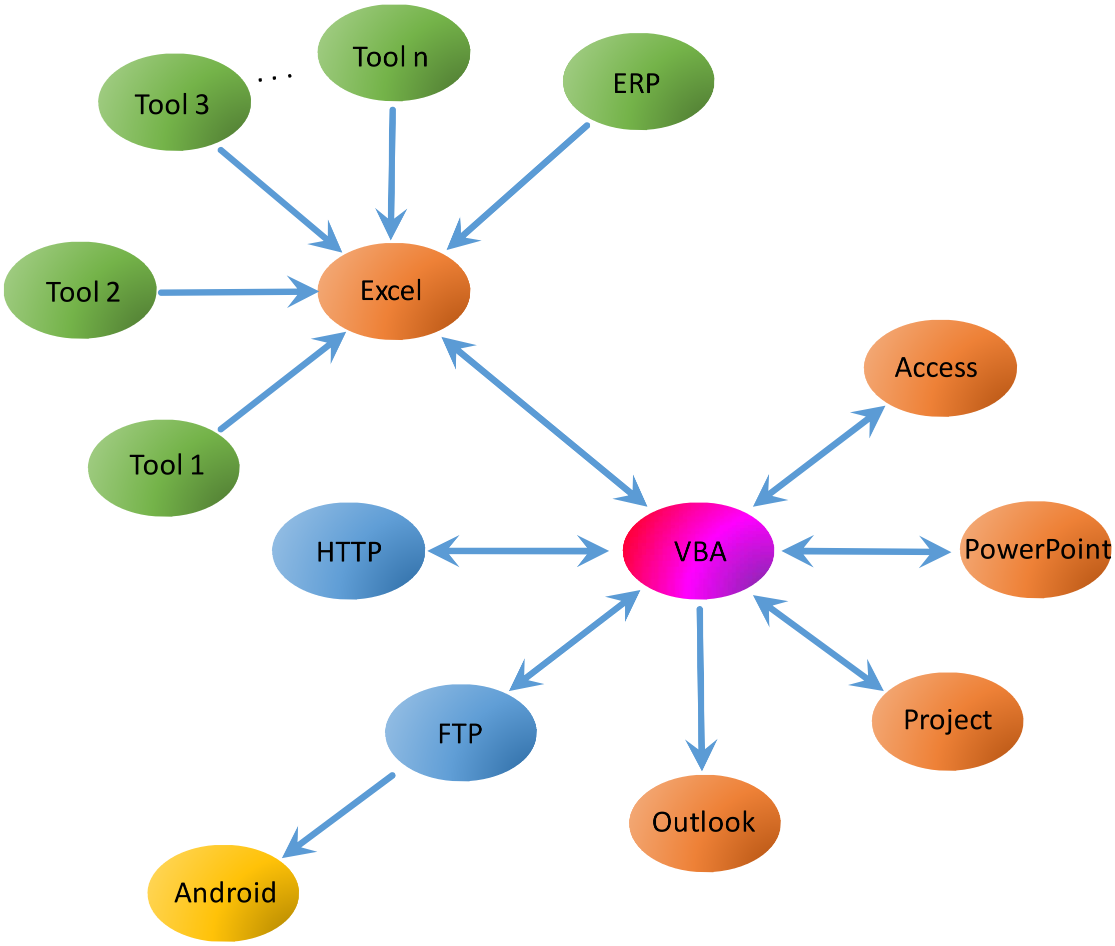


Fig. 2. VBA Data Interactions referred in this paper

VBA macros can perform most of the action a user would follow to update and present the information with several advantages:

- They do it in less time
- They need minimum human intervention
- They always follow the same procedure, creating a consistent output
- They do not commit mistakes

The information can automatically be presented to the target audience in one or several ways:

- An Excel file
- A PowerPoint presentation file
- An updated site
- Updated information in a mobile device app

The macros that will iterate and update the information can be launched by the user by clicking a button, but they can also be automatically launched by programs such as Windows Task Scheduler, avoiding the need of any human action for each iteration. With such an implementation, the information is updated with the desired frequency without allocating any human resources time.

The human resources allocated to a project are typically limited and in the busiest and most challenging stages of the projects are usually allocated to implementation tasks, leaving no time to collect and analyse data about the performance of the project. It makes sense to invest time in the initial stage of a project, preparing an automatic process to perform those activities later.

Instructing a macro to perform one such analysis is usually more time consuming than "manually" performing the analysis once, but if we are to repeatedly perform it, we end up saving time and achieving a more coherent, sustainable and error free output.

VBA comes free with the corresponding MS programs and can be considered as a way of using them – probably the most advanced way.
It is quite intuitive and easy to start being used as MS Excel and Project convert our actions into code, inviting us to see how the programming works.

Estimates point to a number of VBA developers around the globe in the order of the hundreds of thousands and it is easy to find VBA development freelancers in the net practicing fair prices.

8 Acronyms

App – Application
DB - Database
ERP - Enterprise Resource Planning
FTP – File Transfer Protocol
HTML - HyperText Markup Language
HTTP - Hypertext Transfer Protocol
MS - Microsoft
PM – Project Management
PMP – Project Management Professional
SW - Software
VBA - Visual Basic for Applications