The Workbook object

introduction to the Workbook

This chapter will focus on the Workbook object. There are only a handful of things that are done at the Workbook level, but those few items are quite important. The main things to be done with a Workbook are:

- Create, open, close, and save Workbooks
- Manage references to Workbooks
- Change certain properties of the Workbook (TODO: are there any?)
- Access certain properties of the Workbook (e.g. Path)

This will be a quick chapter since the items below are fairly straight forward. Having said that, this chapter will be quite relevant if you are working through a complex workflow that involves creating temporary or new Workbooks to store data or analysis. Being able to create and work with Workbooks will give you the confidence to fire up a new Workbook for a one-off analysis instead of polluting the existing Workbook with a one-off Worksheet or other data dump.

understanding the Workbook Object Model

The Workbook is an object that serves two main purposes:

- Provide a foothold to a number of other more useful functions (e.g. Sheets)
- Provide a reference to the underlying data within a spreadsheet while working through a workflow

For the first point, the goal of the Workbook is to actually use its properties to do some task. For the latter point, the Workbook is simply a container that holds data which is necessary to interact with while creating a workflow. To be honest, there is very little of use within the Workbook object that is not a reference to some other object. Typically, the main tasks to actually be done with the Workbook are to Open, Save, and Close them. That is, you move away from the Workbook object as quickly as you can because you just need a reference.

working with Workbook references

There are a couple of ways to obtain a reference to a Workbook that are useful:

- ActiveWorkbook - refers to the Workbook that has focus
- ThisWorkbook - refers to the Workbook which contains the code that is executing
- Workbooks.Open() - will open a Workbook and return a reference
- Workbooks(index) - will grab a reference to the currently opened Workbook
- Workbooks.Add() - will create a new blank Workbook or a Workbook according to a supplied template

I find that all of those approaches are used equally across my code. The one exception might be ThisWorkbook which I typically avoid. In reality, I should probably use it more because I find myself going to some length to maintain a reference to a Workbook while opening or creating Workbooks.

For Workbooks, the biggest thing to be aware of that there are a number of unqualified references that exist within VBA that are a part of the ActiveWorkbook. Those include:

- Worksheets and Sheets
- Names?

These unqualified references can really bite you when you are expecting it. The problem with unqualified references is that they work great initially, before the workflow becomes complex. They will then silently fail later when you start creating new Workbooks and otherwise changing the focus or active Workbook. The problem is that nearly all of the unqualified references apply to the ActiveWorkbook. Working with Workbooks is the one task that will often change the focus of Excel regardless of how you create things.

useful properties of the Workbook

Although I have railed against the Workbook object, there are a handful of things that it can do:

- Reference `Names` which contains all of the global named ranges

- Others?
- Charts?

Worksheets vs. Sheets

WHen working with Worksheets, there are a pair of objects which will provide access to the underlying Sheets. They are different in how they handle Charts which are visible as a Worksheet. The rule is: Sheets will return the Charts, whereas Worksheets will only return the list of objects which are actually Worksheets. If you do not use Charts as Worksheets, then you will never notice a difference between these two objects. The one thing you will notice is that the ActiveWorksheet will not be of type Worksheet which means that you can never get Intellisense on one of the most useful objects.