Field Marketing Analytics Request Tool Documentation

Administrators:

Jack Hobbs – Data Analyst – jack.hobbs@papamurphys.com

Jake Jirsa – Director of Marketing Analytics – jake.jirsa@papamurphys.com

CONTENTS

INTRODUCTION:

The Marketing team frequently receives requests from franchise owners for up-to-date information on the performance and health of their stores and DMAs. Often, the analytics required to adequately resolve these requests falls to the Performance Marketing team. Historically, these analytics have been performed on an Ad Hoc basis and expanding those reports to the system level required individual production of over 100 unique sets of Excel and PowerPoint files, one for each marketing area (DMA). For Performance Marketing, the responsibility of a standard set of quarterly marketing reports (swings) is a recent development only made possible through Visual Basic for Applications (VBA).

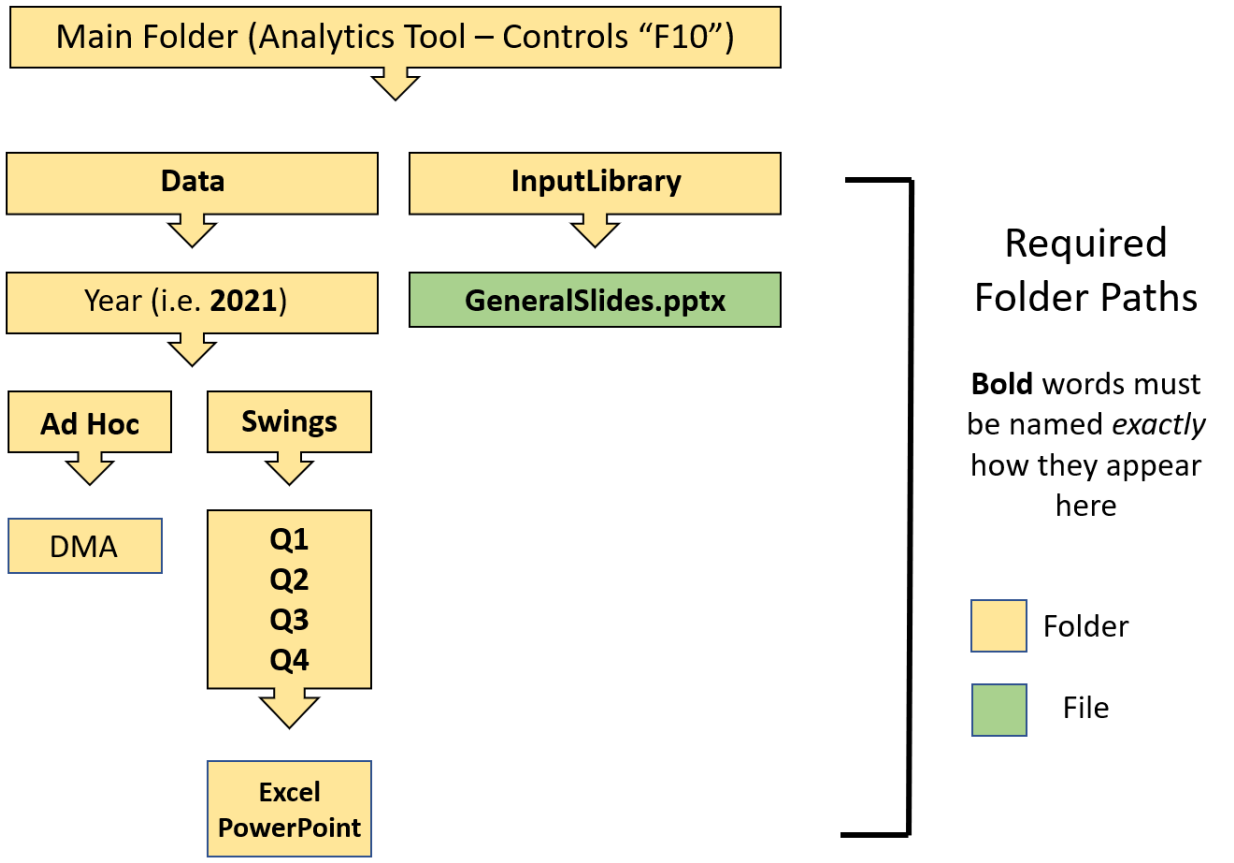
The justification for development of this tool is twofold:

1. Swings reports rely on the synthesis of several different charts into one PowerPoint and Excel file for each DMA. To perform this process by hand would take an analyst a week; with this tool it takes about an hour.
2. The analytics team was often fulfilling requests from the field in such a way that retooling it for other DMAs was time consuming. With an eye on expandability and modularity, we can integrate each analysis into a single tool where they can be accessed at any time, for any DMA. By updating just a few cells in Excel and a couple slides in PowerPoint, we can seamlessly integrate new and old analyses into swings reporting.

VBA is a code language that runs in Microsoft's Office ecosystem. It powers the macros that are prevalent in complex projects built in Excel, PowerPoint, and Word. VBA features unique functions and methods that specifically integrate into those programs and allow a programmer to automate significant workloads, especially projects that require production of new files and datasets.

FILE PATHING:

This tool relies on consistent folder pathing so changing certain aspects of the folder structure will break the program. If there is a need to move the main folder with all the input, output, and tool files, then this is possible by changing the outer path found in cell H23 in the "Controls" worksheet of the analysis tool Excel file. If there is a need to change how the folder is structured, this could feasibly be done via editing VBA code. I will cover this process further in the documentation. Otherwise, the internal structure of the main folder must retain these features and exact folder names:



One of the ways that the code was made more efficient was to produce the requested files and save them locally in a temporary folder. This temporary folder is copied and merged with the public folder that matches the year and quarter the user specifies. After the temporary folder is used and copied, it is automatically deleted by the code. This means that the user should never see this folder except during the process of running the analytics tool.

Testing hasn't yielded any problems, but there is a chance that errors occur due to permissions for creating, saving, and deleting folders. If this occurs, contact the administrator of the file who can use this documentation to bypass local saves and export files directly to the public drive. If the local folder is bypassed for direct saves, production of files for all DMAs will take approximately twice as long due to remote save times - about 80 minutes. If longer production time is a problem and local saving must be bypassed, execute the code at the support center where access speeds to the drive are much faster than via a remote connection.

UNDERSTANDING THE ANALYSIS TOOL:

The analysis tool houses many worksheets that fall into four categories:

- User input
- Analysis charts
- Parameter and meta-data for charts
- Data lookups that are used to populate charts

The following section will cover each worksheet.

CONTROLS

When the user first opens the analysis tool, they are greeted with the Controls sheet. On this page, there are many options that the user can change to configure the output from the tool. The user interacts and sets these choices with the data validation dropdown lists (DV) in the Excel range H16:H21. **If the user moves the cells that contain these DVs or edits the DVs in ways that are not accepted, it will break the tool.** Changes can be made; the tool natively supports adding DMAs via Excel. New charts should automatically update into the Chart List DV. Any other changes will need to be accounted for in the VBA code.

These options are:

DMA QUANTITY:

Sets the number of DMAs for which the tool produces Excel/PowerPoint files. All is mostly used for swings, whereas Selected is more useful for ad hoc analysis and testing slide object placement.

- All: every DMA in the Current DMA data validation list
- Selected: only the currently selected DMA

Source: list -> [All, Selected]

OUTPUT:

The tool is set up in such a way that the user can decide if they wish to produce DMA analytics files for Excel, PowerPoint, or both. Excel is the faster option of the two by far.

- Excel: produces only .xlsx files
- PowerPoint: produces only .ppt files
- Both: produces both .xlsx and .ppt files

Source: list -> [Excel, PowerPoint, Both]

CURRENT DMA:

A DV list of all DMAs in the system. Charts reference this DV and all fields in a chart are populated on a lookup of that value. DMAs can be added to this list via addition to General Lookups column A and updating the DV list range to include the new DMA (i.e. $A$2:$A$113 -> $A$2:$A$114).

- [DMA]: the name of a given DMA that is used for lookups and to name the file when it is saved

Source: list -> [='General Lookups'!$A$2:$A$113]

CHART LIST:

Controls what kinds of chart outputs are returned from the tool.

- Swings: produces a swings-ready list based on the charts represented in Parameters columns A/B
- All: produces output for a given DMA that contains **all** charts found in Parameters column E
- [other]: produces the specified chart - DV is automatically populated based on Parameters column E

Source: list -> [='Parameters'!$E$2:$E$200]

QUARTER:

User input determines in which Quarter folder the data will be saved.

- Ad Hoc: used specifically for one-off analysis on a given DMA
- Q1-Q4: mostly used for swings – useful for historical data

Source: list -> [Ad Hoc, Q1, Q2, Q3, Q4]

YEAR:

User input determines in which Year folder the data will be saved.

- 2021-2026: year of data in analysis

Source: list -> [2021,2022,2023,2024,2025,2026]

SLIDE FILE NAME:

User input determines which slide deck is populated with the selected slides.

- GeneralSlides.pptx: standard slide deck for swings reporting
- BlankFormatSlide.pptx: standard mostly blank slide for individual chart and all chart output

Source: list -> [GeneralSlides.pptx, BlankFormatSlide.pptx]

EXECUTE:

Pressing **Execute** will run the code using the options that the user set. It is assigned to the Execute macro.

CHARTS:

The charts worksheet is the source from which all output Excel and PowerPoint files are built. Each chart in the table draws its data from a corresponding worksheet detailed below. If the user is not looking to export, and instead just wishes to quickly verify some information about a DMA, this worksheet is a good point of reference. If the user changes the DMA but the charts do not update, simply go to the search bar in Excel and look up “Calculation Options” and set them to “Automatic”.

As of December 2021:

CHECK BY CHANNEL:

Check by Channel shows the current quarter average check for the EBT, Delivery, Online, and In-Store channels versus the respective averages from the prior quarter.

- Range: B3:D8
- Source: 2021 (Week)

DAY OF THE WEEK:

Day of the Week shows the year-to-date comparison of transactions for a DMA for 2021 vs 2019 and 2020 respectively. Despite being quite large and having many tables, the only one that is used for the Day of the Week slides and Excel worksheet is the table specified in the range below. That table compares the DMA to the PMI aggregate.

- Range: B22:G30
- Source: PU by DoW Merged

EBT QUARTER OVER QUARTER:

EBT Quarter over Quarter compares the DMA to the PMI aggregate for the proportion of transactions that were from EBT purchases. The table breaks down the information by quarter in percentage form.

- Range: B34:H40
- Source: EBT Quarter

LTO:

The Limited Time Offer chart provides DMA level mix performance for the previous quarter's LTO. The data is updated using Smartsheets media vote reports, filtered to the quarter's LTO. The chart for this data is unique in that it uses a Pivot Table rather than a reference driven custom table. Change in LTOs each quarter may require the user to adjust the parameters copy range (LTO 1).

- Range: B77:D93, G77:K83 (Base – 1 price point)
- Source: LTO Data

LOYALTY PENETRATION:

Loyalty Penetration is a dynamic set of tables that shows the percentage of loyalty penetration for a given DMA. The DMA is compared to the top 10 DMAs by loyalty penetration at the system level and the region level. Dynamic tables mean that the chart also contains store level data, which needs to be trimmed on the slides based on the size of the DMA. For instance, Seattle has around 100 stores whereas Birmingham has 6 stores. To include all 5 of the dynamic tables for Loyalty Penetration on Birmingham's slides would be redundant as only the first would be populated. Each dynamic table has space for 27 stores and deletion of excess tables is based on General Lookups column F: *LP Sizer*. The LP Sizer

value then informs the Parameters Drop Dynamic Slides table, which controls deletion of excess slides.

- Range: B44:Z73
- Source: LP Data, LP Tables

Note: LP Old Tables is simply a repository for LP Tables from prior quarters. The data is just values to save on computational overhead and storage for the workbook

PARAMETERS:

The Parameters worksheet controls all aspects of how charts are placed on Excel sheets and PowerPoint Slides. The VBA code loops through the fields on this worksheet and collects data that controls placement, text size, and excess slide deletion for copied charts. The worksheet is split into three sections: Main Swings Slides, All Slides, and Drop Dynamic Slides.

Each section specifies data types for its columns. These rules are **non-negotiable**. Putting string data in an integer column, for example, **will** break the code.

MAIN SLIDES:

Controls which charts populate the selected slide deck as well as the indexing of the slides.

- **Column A – Slide Number:** identifies the slide index onto which the corresponding chart will paste
  - Data Type: Integer
- **Column B – Slide Name:** the name of the chart assigned to the slide. Dynamic slides get unique numbers
  - Data Type: String

ALL SLIDES:

The All Slides section is the main point of control for placement of charts in the new DMA workbooks and slide decks. This is the section that a user would edit to manipulate where the charts are copied from, where they will appear on those files, text size, and identification for if a slide is dynamic.

- **Column D – Worksheet Name:** the name of the worksheet the chart should paste to in the new workbook. Multi-part dynamic slides paste to *the same* workbook, so have an **identical** value in this column.
  - Data Type: String

- **Column E – Slide Name:** the name of the slide that the chart will paste to in the new slide deck. Multi-part dynamic slides paste to *different* slides, so have a **unique** value in this column.
    - Data Type: String
- **Column F – Range:** the range to be copied from the Charts worksheet. Must be in the form of an Excel range.
    - Data Type: String
- **Column G – Horizontal Position:** the amount by which the chart is moved from the left edge of a slide to the right. Technically, this is the number of pixels to move the chart by. For our purposes, there are 72 pixels in an inch.
    - Data Type: Integer
- **Column H – Vertical Position:** the amount by which the chart is moved from the top edge of a slide to the bottom. Technically, this is the number of pixels to move the chart by. For our purposes, there are 72 pixels in an inch.
    - Data Type: Integer
- **Column I – Text Size:** sets the point font size for all text on the chart.
    - Data Type: Integer
- **Column J – Cell Height:** sets the row height in inches for all cells in the chart.
    - Data Type: Double
- **Column K- Cell Width:** sets the column width in inches for all cells in the chart.
    - Data Type: Double
- **Column L - Centered:** a binary marker to communicate if the chart should be centered on the slide. If turned on, this overrides vertical and horizontal translate positioning.
    - Data Type: Integer (0,1)
    - 1 = On, 0 = Off
- **Column M – Paste Cell:** the cell to which the chart is pasted in the new workbook. Must be in the form of an Excel range.
    - Data Type: String
- **Column N – Dynamic:** a binary marker to communicate if the chart is a dynamic range that may be deleted if not needed.
    - Data Type: Integer (0,1)
    - 1 = Yes, 0 = No

DROP DYNAMIC SLIDES:

This section communicates to the code which slides are scheduled for deletion based on information it receives from markers set in General Lookups such as LP Sizer.

Each dynamic chart will need to have its own column in this section between columns Q and Y. Any additional columns to this section beyond column Y will either require a changes to the VBA code that establishes the range it checks for slide names, or the user can append those columns onto the bottom of the existing data in columns P:Y. The VBA code will check down to the 50th row, so the second option is preferable.

Always populate this table with **only** one of the following:

- A string slide name that **exactly** matches the corresponding entry in column E
- No data, i.e. “”

GENERAL LOOKUPS:

The General Lookups worksheet is a reference sheet that controls many different aspects of the analysis tool workbook and analysis charts.

DMA INFORMATION:

This section is the reference point for all charts on DMA level information. This is the first stop when adding a new DMA to the analysis tool.

- **Column A – DMA:** the PMI name for each DMA. All slashes and whacks (\,/) have been replaced with dashes (-). The code should catch any deviation from this, but best practice is to strip bad characters before running the code.
    - Data Type: String
- **Column B – Region:** the PMI region for each DMA.
    - Data Type: String
- **Column C – DC:** the PMI distribution center for each DMA.
    - Data Type: String
- **Column D – Number of Stores:** a count of the stores in each DMA.
    - Data Type: Integer
- **Column E – Store Group:** the grouping for each DMA based on the number of stores. [1-2, 3-4, 5-10, 11-20, 20+]
    - Data Type: String
- **Column F – LP Sizer:** a function powered column that determines the sizer value for each DMA. There are 27 store spots in the loyalty penetration dynamic charts. The sizer function assigns the DMA based on its store count. 1-27 stores = 1, 28-54 stores = 2, 55-81 stores = 3, 82-108 stores = 4)
    - Data Type: Integer (Function)

MISCELLANEOUS:

This section has a variety of reference information.

- **Column Q - Slide File Name:** file names for the blank format slide and the main slides templates.
  - Data Type: String
  - GeneralSlides.pptx: standard slide deck for swings reporting
  - BlankFormatSlide.pptx: standard mostly blank slide for individual chart and all chart output

SLICERS

This tab controls data flow into the data tables that populate charts. Here, the user can select filters to ensure accurate data to the current quarter. Connections for the slicers can be determined by right clicking them and selecting *Report Connections*.

ADDING AN ANALYSIS CHART TO THE ANALYSIS TOOL:

CHART FORMATTING REQUIREMENTS:

For best results and to ensure no errors occur in the production of PowerPoint slides, there are some chart formatting guidelines that the user should adhere to:

- The chart **must** rely on a single DMA name reference cell for looking up its *current* DMA data.
  - This cell is connected to the DMA DV list.
- Ideally, the chart requires only **one** data worksheet for lookups.
  - The goal is to reduce the number of worksheets in the analysis tool.
- Aside from the current DMA cell, the chart must not rely on any further manipulations. Inclusion of additional DVs or other filters for chart population cannot be integrated.
- **No merged cells**
  - Instead, opt for centering across the selected cells as it is much more stable.
  - https://docs.microsoft.com/en-us/office/troubleshoot/excel/center-across-columns-button-disappear
- All text will be the same size, do not rely on different sized text.
- All columns will be the same width. This comes with a few considerations.
  - If you are including DMA names as values in a column of the chart, they are quite wide. Anticipate that a slide will only be able to be 5 or 6 columns wide for it to fit.

    - One column width can make charts look unbalanced; avoid including blank columns.
- All rows will be the same height. This comes with many of the same considerations as column width. Avoid multi-row height merged cells, but they shouldn't break anything.
- Avoid unnecessary additional information to aid in keeping workbook size down.
- **Test your charts using the "Missouri-Cape Girardeau MO - Paducah KY - Mt Verno" DMA, as this is the longest name in the system.**

PROCESS:

To begin, the user must first create the desired Excel chart and its corresponding data worksheet tab. It is advised to do this in a separate workbook to avoid unexpected changes to the main workbook. Lookups must point to the General Lookups tab in the main workbook. The user must adhere to the chart formatting rules to ensure chart compatibility.

1. Insert data tab(s) at the end of the analysis tool workbook
    a. Rename Pivots to something reasonable (i.e. LTO_CurrentYearTable)
    b. Add slicer controls to Slicers tab
2. Copy and paste chart at the bottom of the Charts worksheet
3. Add slide parameters to the Parameters worksheet
4. Update the GeneralSlides.pptx file if the new chart is to be included in that deck
5. Test slide placement and adjust parameters to achieve desired chart fit
    a. If the chart is not changing in response to reducing column widths or row heights, try reducing text size

POWERPOINT:

The PowerPoint slides are populated from selected charts onto a set of pre-made slides that the user specifies on the Controls worksheet. The default option, GeneralSlides.pptx, are used for swings reporting, but there is an option to build more default decks and add them to the Slide File Name DV on the Controls worksheet. To set a new default deck to be used for the PowerPoint export, replace cell Q2 in the General Lookups tab with the file name of the new default deck.

Slide decks should always be stored in the InputLibrary folder inside the main folder. **Never save over the default slide decks!** If edits need to be made, these files are edit-restricted to Performance Marketing administrators; these people are the primary contact point for edits.

VBA:

The primary intention of this section of documentation is to provide administrators with context to the code driving this tool. If an administrator needs to edit the codebase, this documentation should be an adequate guide. As with all programming, Google is the developer's best friend.

Visual Basic for Applications is a powerful tool for automating labor-intensive Word, Excel, and PowerPoint workflows. It is leveraged heavily in this analysis tool, but it is easy to edit the Excel sheet in ways that break the code. Fortunately, rectifying these edits is simple and back-ups are maintained by the administrators. VBA has been around for three decades: if a problem arises, someone on the internet has asked about it and gotten an answer.

If the developer is unfamiliar with the operation of a command, Microsoft maintains good VBA documentation: https://docs.microsoft.com/en-us/office/vba/api/overview/

To find and open the Visual Basic Editor: https://support.microsoft.com/en-us/office/find-help-on-using-the-visual-basic-editor-61404b99-84af-4aa3-b1ca-465bc4f45432

DEFINITIONS:

Some basic definitions, to aid the user in understanding the following documentation:

GENERAL:

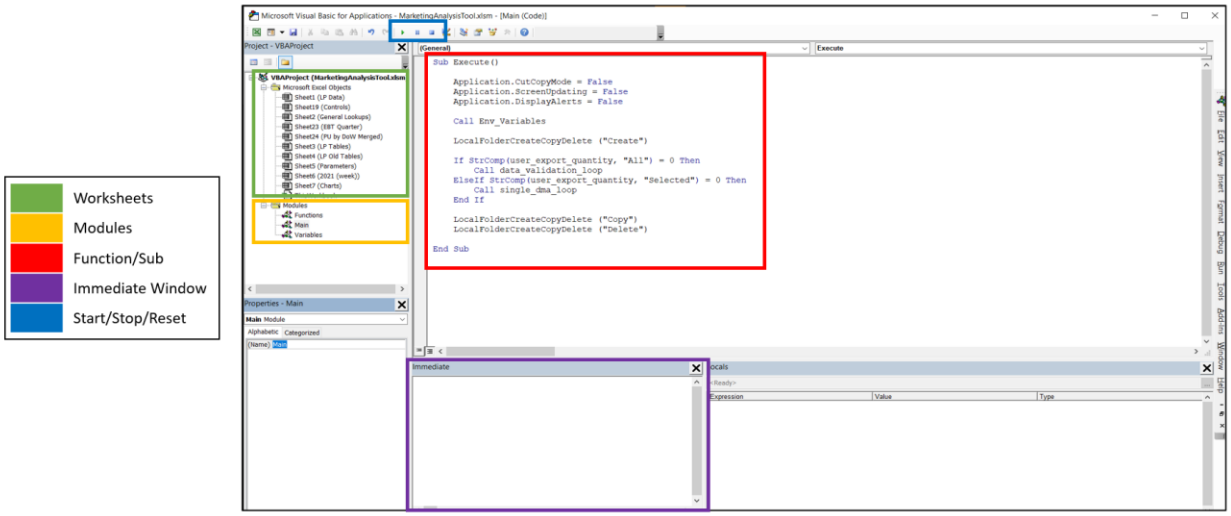
- **Module:** a place to store code, functions, variables
- **Function/Sub:** an object that can be called. When called, it executes code based on current variable values or based on values passed into it by the developer

VARIABLE TYPES:

- **String:** text data, placed in quotes.
- **Integer (Int):** numeric data, no decimals.
- **Double:** numeric data that allow decimals.
- **Workbook:** VBA variable type that represents a whole workbook.
- **Worksheet:** VBA variable type that represents a worksheet within a specified workbook.
- **Range:** VBA variable that represents an Excel range based on a given string (i.e. "A1:D4")

- **Object:** VBA general variable that can represent a PowerPoint slide, file object, an Office app, etc.
- **Collection:** VBA iterable storage container like a Python list.

NAVIGATING THE VBA ENVIRONMENT:



MODULE STRUCTURE:

The code for the analysis tool is divided into three modules. The developer should be able to make most necessary changes in the Variables module as this is where folder pathing is determined for save functions. For the most part, other functions should continue to work so long as the developer does not pass them incorrect variable types.

VARIABLES:

The Variables is divided into two sections: variable declarations, and variable assignment. VBA frequently requires the developer to establish what kind of data is attached to each variable. This is most often seen as ranges, strings, integers, doubles, and objects. By declaring them as **Public** they are accessible across the whole codebase, not just in their respective module. After declaration, a variable is then set to a specific value, usually by referencing an Excel range (e.g. Worksheet.Range(“A1”))

MAIN:

The Main module contains only one function, Execute. This is to isolate the primary function from the rest of the functions for readability and comprehension.

FUNCTIONS:

This module houses all functions necessary for the analytics tools. Broadly speaking, the module is divided into 4 parts: loop controls, general functions that handle in-application work and file overhead, Excel specific functions, and PowerPoint specific functions.

VARIABLES:

This section covers the definitions and uses of each variable declared in the analysis tool. Each variable will be defined in the following way:

- Category
  - Name
    - Data Type
    - Description
    - Reference sheet: cell **[If Applicable]**
    - [functions in which it is used]

WORKBOOK:

- **Master:**
  - Excel Workbook
  - Reference for the entire analysis tool workbook; used primarily to assign worksheets
  - Env_Variables, Excel_Open
- **new_workbook:**
  - Excel Workbook
  - Reference to the new workbook generated for Excel output
  - Single_DMA_Loop, Current_Chart_Loop, Excel_Open, Excel_Save

WORSKSHEETS:

- **ws_controls:**
  - Excel Worksheet
  - Reference to the Controls worksheet in the analysis tool
  - Env_Variables, Single_DMA_Loop
- **ws_charts:**
  - Excel Worksheet
  - Reference to the Charts worksheet in the analysis tool
  - Env_Variables, Current_Chart_Loop
- **ws_parameters:**
  - Excel Worksheet

    - Reference to the Parameters worksheet in the analysis tool
    - Env_Variables, Charts_Loop, Current_Chart_Loop
- **ws_lookups:**
    - Excel Worksheet
    - Reference to the General Lookups worksheet in the analysis tool
    - Env_Variables
- **ws_chart_worksheet:**
    - Excel Worksheet
    - Reference to the current chart worksheet in the new workbook when building Excel files
    - Current_Chart_Loop

USER SELECTIONS:

- **user_export_quantity:**
    - String
    - User selected value from the DMA Quantity DV in the Controls worksheet
    - Controls: H16
    - Env_Variables, Execute
- **user_export_type:**
    - String
    - User selected value from the Output DV in the Controls worksheet
    - Controls: H17
    - Env_Variables, Single_DMA_Loop, Current_Chart_Loop
- **user_chart_type:**
    - String
    - User selected value from the Chart List DV in the Controls worksheet
    - Controls: H19
    - Env_Variables, Charts_Loop, PowerPoint_Open
- **user_dma:**
    - Excel Range
    - References the currently selected DMA in the Current DMA DV; **controls chart data population**
    - Controls: H18
    - Env_Variables, Data_Validation_Loop, Single_DMA_Loop
- **user_dma_safe:**
    - String

    - References the currently selected DMA in the Current DMA DV with slashes replaced with dashes to make the file save safe; **controls chart data population**
    - Controls: H18
    - Env_Variables, Single_DMA_Loop, Local_Folder_Create_Copy_Delete, Excel_Save, PowerPoint_Save
    -
- **user_quarter:**
    - String
    - User selected value from the Quarter DV in the Controls worksheet
    - Controls: H20
    - Env_Variables, Single_DMA_Loop, Excel_Save, PowerPoint_Save,
- **user_year:**
    - String
    - User selected value from the Year DV in the Controls worksheet
    - Controls: H21
    - Env_Variables
- **user_save_type:**
    - String
    - User selected value from the Save Type DV in the Controls worksheet
    - Controls: H25
    - Env_Variables, Execute, Excel_Save, PowerPoint_Save

FILE PATH:

- **public_destination_folder:**
    - String
    - File path to the drive save location for exports from the analysis tool - ([Controls: H23]/user_year/user_quarter/)
    - Controls: H23
    - Env_Variables, Local_Folder_Create_Copy_Delete, Excel_Save, PowerPoint_Save
- **public_template_folder:**
    - String
    - File path to the drive location for templates - ([Controls: H23]/InputLibrary/)
    - Controls: H23
    - Env_Variables, Local_Folder_Create_Copy_Delete
- **local_destination_folder:**
    - String
    - File path to the temporary local folder for saves

    - Controls: H24
    - Env_Variables, Local_Folder_Create_Copy_Delete, Excel_Save, PowerPoint_Save
- **date_string:**
    - String
    - Current date with slashes replaced by dashes
    - Env_Variables, Excel_Save, PowerPoint_Save
- **adhoc_file_name:**
    - String
    - Name for ad hoc quarter export files; combination of date_string and user_chart_type
    - Env_Variables

SLIDES FILE PATHS:

- **main_slides_file_name:**
    - String
    - File path to the temporary local folder for saves
    - Controls: H24
    - Env_Variables
- **other_slides_file_name:**
    - String
    - File path to the temporary local folder for saves
    - Controls: H24
    - Env_Variables
- **user_slides_file_name:**
    - String
    - Logic for user selections for file path to the temporary local folder for saves
    - Env_Variables, Local_Folder_Create_Copy_Delete, PowerPoint_Open

LOCAL FOLDER CREATION:

- **fso:**
    - FileSystemObject - FileSystemObject object | Microsoft Docs
    - Object that enables VBA to interface with Windows for file/folder editing
    - Env_Variables, Local_Folder_Create_Copy_Delete

POWERPOINT:

- **powerpoint_app:**
    - Object

    - Reference object to open and close PowerPoint
    - PowerPoint_Open,
- **delete_index_collection:**
    - Collection
    - Iterable that contains slide indexes to control deletion of unneeded slides
    - Env_Variables, Single_DMA_Loop, Current_Chart_Loop
- **new_presentation:**
    - Object
    - Reference object for newly generated export PowerPoints
    - PowerPoint_Open, PowerPoint_Save, Current_Chart_Loop
- **main_slides_range:**
    - Range
    - Range of slide names that are to populate the export PowerPoint presentation for standard reporting
    - Parameters: B3:B50
    - Env_Variables
- **all_slides_range:**
    - Range
    - Range of all potential slide names that can populate the export PowerPoint presentation
    - Parameters: E3:E50
    - Env_Variables
- **selected_slides_count:**
    - Double
    - A count of the number of populated cells in the main_slides_range to limit the user_chart_type loop in the Charts_Loop function
    - Env_Variables, Charts_Loop
- **all_slides_count:**
    - Double
    - A count of the number of populated cells in the all_slides_range to limit the user_chart_type loop in the Charts_Loop function
    - Env_Variables, Charts_Loop, Current_Chart_Loop, PowerPoint_Open
- **ppt_slide:**
    - Object
    - Temporary reference to the slide for the current chart in the loop
    - Current_Chart_Loop
- **ppt_shape:**
    - Object

    - Temporary reference to the chart table object in the current ppt_slide
    - Current_Chart_Loop
- **ppt_index:**
    - Integer
    - Index number for the current ppt_slide in the deck
    - Parameters: A[loop_index_outer]
    - Current_Chart_Loop

LOOP CONTROLS:

- **loop_index_outer:**
    - Integer
    - Integer to demarcate the current chart in Charts_Loop for use in cell references
    - Charts_Loop, Current_Chart_Loop
- **loop_index_inner:**
    - Integer
    - Integer to iterate through all possible slides to compare against the current chart in Charts_Loop
    - Current_Chart_Loop
- **loop_current_chart:**
    - String
    - Name of the current chart in Charts_Loop to compare against all possible slides in the Current_Chart_Loop
    - Charts_Loop, Current_Chart_Loop

LOCAL:

- **worksheet_name:**
    - String
    - Non-unique name of the current chart in Charts_Loop to assign or locate the correct worksheet in the new_workbook to paste the chart values
    - Parameters: D[loop_inner_index]
    - Current_Chart_Loop
- **data_range:**
    - String
    - Charts worksheet range of the current chart in Charts_Loop
    - Parameters: F[loop_inner_index]
    - Current_Chart_Loop, Copy_Paste_Wait
- **horizontal_position:**

    - Integer
    - Numeric control (in inches) for left-right placement of ppt_shape on ppt_slide when centered = 0
    - Parameters: G[loop_inner_index]
    - Current_Chart_Loop
- **vertical_position:**
    - Integer
    - Numeric control (in inches) for up-down placement of ppt_shape on ppt_slide when centered = 0
    - Parameters: H[loop_inner_index]
    - Current_Chart_Loop
- **text_size:**
    - Integer
    - Numeric font point size for values in pasted charts
    - Parameters: I[loop_inner_index]
    - Current_Chart_Loop, PowerPoint_Object_Text_Size
- **cell_height:**
    - Double
    - Height of pasted cells in inches
    - Parameters: J[loop_inner_index]
    - Current_Chart_Loop, Copy_Paste_Wait, PowerPoint_Object_Text_Size
- **cell_width:**
    - Double
    - Width of pasted cells in inches
    - Parameters: K[loop_inner_index]
    - Current_Chart_Loop, Copy_Paste_Wait, PowerPoint_Object_Text_Size
- **centered:**
    - Integer (Binary)
    - Binary marker to control whether ppt_shape is centered on ppt_slide
    - Parameters: L[loop_inner_index]
    - Current_Chart_Loop, PowerPoint_Slide_Move
- **paste_cell:**
    - String
    - New_workbook range chart will paste to
    - Parameters: M[loop_inner_index]
    - Current_Chart_Loop, Copy_Paste_Wait

FUNCTION AND METHOD DOCUMENTATION:

This section covers the functions found in the analysis tool, divided by module.

MAIN:

EXECUTE:

Description:
Connected to the Execute button in the Controls worksheet, this function runs the correct subfunctions based on user choices in the data validation lists. First, it loads data from the workbook and creates a new local save folder if fast saving is enabled. Based on the DMA Quantity DV, the function either runs the Data_Validation_Loop or Single_DMA_Loop function. Finally, the local temporary folder is copied to the public destination folder and then deleted if fast saving is enabled.

Required Input Parameters:
None
Variables:
None
Subfunctions:
Env_Variables, Local_Folder_Create_Copy_Delete, Data_Validation_Loop, Single_DMA_Loop
Used In:
None

VARIABLES:

ENV_VARIABLES:

Description:
Sets correct values for many public variables, called once at the start of the Execute function.

Required Input Parameters:
None
Variables:
Master, ws_controls, ws_parameters, ws_charts, ws_lookups, user_dma, user_export_quanitity, user_export_type, user_chart_type, user_quarter, user_year, local_destination_folder, public_destination_folder, public_template_folder, date_string, adhoc_file_name, main_slides_file_name, other_slides_file_name,

user_slides_file_name, delete_index_collection_main_slides_range, all_slides_range, selected_slides_count, all_slides_count

Subfunctions:

None

Used In:

Execute

FUNCTIONS:

DATA_VALIDATION_LOOP:

Description:

The bulk of the logic for when DMA Quantity is set to *All*. Captures the Current DMA as an array, then iterates through this array from the first value. Excel is refreshed and recalculated on each loop. Error handling addresses a variety of errors that can arise. On each loop, the Single_DMA_Loop function is executed.

In effect, this function is a shell around the same function set that is run for DMA Quantity set to *Selected* that changes the Current DMA data validation list to populate charts with new data.

Required Input Parameters:

None

Variables:

data_validation_array, i (index), rows (index), user_dma

Subfunctions:

Excel_Recalc, Excel_Refresh, Single_DMA_Loop

Used In:

Execute

SINGLE_DMA_LOOP:

Description:

This function primarily contains logic for generating correct exports: Excel, PowerPoint, or Both. If the user chooses Ad Hoc as the Quarter DV selection, a new sub-folder named for the current DMA is created in the local folder if fast saving is enabled. The function first resets user_dma to the value in the Current DMA DV. Again, the code ensures no slashes are present in the DMA name, replacing them with dashes. Once the function determines the user's choice from the Output DV, it opens the correct applications using Excel_Open and PowerPoint_Open then runs the Charts_Loop function. After the charts have looped, the function deletes the unneeded Sheet1 from

new_workbook as well as unneeded slides based on the delete_index_collection. Finally, the Single_DMA_Loop function saves the exports to the local folder unless fast saving isn't enabled, in which case it is saved directly to the public folder.

Required Input Parameters:
None
Variables:
delete_index_collection
Subfunctions:
Excel_Open, PowerPoint_Open, Charts_Loop, Excel_Save, PowerPoint_Save
Used In:
Execute, Data_Validation_Loop

CHARTS_LOOP:

Description:
Another link in the logic chain to export populated files, Charts_Loop interprets user input in the Chart List DV to populate export files with slides specified. Based on the Chart List value, the function runs a loop that iterates from the third row of Parameters to the Nth row, where N = 3, selected_slides_count+2, or all_slides_count+2. Within that loop the function executes the Current_Chart_Loop. For N = 3, only the user selected slide is produced.

Required Input Parameters:
None
Variables:
user_dma, user_chart_type, loop_current_chart, loop_index_outer
Subfunctions:
Current_Chart_Loop
Used In:
Single_DMA_Loop

CURRENT_CHART_LOOP:

Description:
Current_Chart_Loop is the deepest layer of the export file production code, handling the assignment of parameters that control chart formatting and placement in the export files. The function loops through the names of all possible charts and compares them to the current name from Charts_Loop. On a match, the parameter variables are

reassigned. Following the assignment loop, the function populates the correct export files based on the user's Output DV selection.

Excel:

In the new_workbook, the function determines if there is a sheet named the same as parameter worksheet_name. If there is, the chart is pasted to paste_cell allowing the code to handle situations where there are charts that need several slides, but only one worksheet (i.e. Loyalty Penetration). If there isn't a matching worksheet, one is generated and the chart is pasted to paste_cell in that sheet. The pasted charts, thanks to the Copy_Paste_Wait function, are reformatted to the correct column widths, row heights, and text size.

PowerPoint:

In the new_presentation, the function grabs the index for the chart then sets ppt_slide to the correct slide from the template based on this index. The function loops through the Drop Dynamic Slides section from the Parameters worksheet. When the loop finds that the chart is dynamic, a marker is activated. When the marker is active, the ppt_slide object is added to delete_index_collection.

Following the slide deletion internal loop, the function pastes the current data range to ppt_slide. Finally, the shapes are passed to the PowerPoint_Object_Text_Size function where they are iterated through and each cell in the table is resized to the correct column width, row height, and text size. The shape is moved into the correct position on the slide.

Required Input Parameters:

None

Variables:

loop_index_inner, all_slides_count, inner_loop_slide, worksheet_name, data_range, horizontal_position, vertical_position, text_size, cell_height, cell_width, centered, paste_cell, user_export_type, ws_charts, ws_chart_worksheet, ppt_index, ppt_slide, ppt_shape, delete_dynamic_marker,

Subfunctions:

Excel_Worksheet_Exists, Copy_Paste_Wait, PowerPoint_Object_Text_Size, PowerPoint_Slide_Move, Excel_Refresh

Used In:

Charts_Loop

COPY_PASTE_WAIT:

Description:

This function accepts two input parameters, an object to be copied and a place to paste it. There is logic that determines if the paste location is an Excel worksheet or PowerPoint presentation. If it is a worksheet, the formats and values are copied and pasted to the paste_cell. Then, the column widths and heights are changed:

Column Width = cell_width * 10

Row Height = (cell_height * 10) + 15

When the detected object is a PowerPoint presentation slide, the copied chart is pasted as VBA DataType = 0 using a PasteSpecial method. This effectively pastes an editable Excel table into the slide.

Required Input Parameters:

copy_object: Object, usually an Excel range from the Charts worksheet

paste_object: Object, typically a slide or worksheet

Variables:

copy_object, paste_object, data_range

Subfunctions:

None

Used In:

Current_Chart_Loop

LOCAL_FOLDER_CREATE_COPY_DELETE:

Description:

To save the network overhead of saving each individual file as they are built, files are sent to a local folder that is then copied in its entirety to the public drive. This function accepts several strings that determine which operation of the function is executed. It sets fso as a FileSystemObject so that VBA can create and manipulate objects in the Windows File Explorer.

Create:

Using local_destination_folder as the file path for the method fso, if the folder exists it is deleted to provide a fresh space for new data. A new folder is created at the same path and the template slides from the Controls worksheet are copied to this new folder.

Ad Hoc:

When run, it creates a sub-folder named for the current DMA in the local folder.

Copy:

Deletes the template slide from the local folder then copies all the files over to the folder specified in the public_destination_folder path.
Delete:
Deletes the folder at the local_destination_folder path.

Required Input Parameters:
Operation: String, must be one of the four values listed above.
Variables:
fso, local_destination_folder, public_destination_folder, user_slides_file_name, user_dma
Subfunctions:
None
Used In:
Execute, Charts_Loop

EXCEL_OPEN:

Description:
Opens a new workbook for export.

Required Input Parameters:
None
Variables:
new_workbook
Subfunctions:
None
Used In:
Single_DMA_Loop

EXCEL_SAVE:

Description:
If fast saving is enabled, saves the new_workbook to the local folder then closes the document. If the user sets the Quarter DV to Ad Hoc, the file is placed into its respective DMA sub-folder. Otherwise, file is saved directly to the public folder.

Required Input Parameters:
None
Variables:
new_workbook, local_destination_folder, user_dma, adhoc_file_name

Subfunctions:
None
Used In:
Single_DMA_Loop

EXCEL_REFRESH:

Description:
VBA has a bad habit of "getting ahead of itself," leading to frustrating errors. As an example, a piece of code that copies an Excel table and then pastes it to another place can attempt to paste *before* it finishes copying data. This function essentially tells Excel to wait until it has completed its pending calculations.

Required Input Parameters:
None
Variables:
None
Subfunctions:
None
Used In:
Data_Validation_Loop, Current_Chart_Loop, Copy_Paste_Wait Excel_Save

EXCEL_RECALC:

Description:
Recalculates each worksheet in the workbook to populate charts with the correct DMA data.

Required Input Parameters:
None
Variables:
None
Subfunctions:
None
Used In:
Data_Validation_Loop

EXCEL_WORKSHEET_EXISTS:

Description:

Determines if a worksheet exists within the specified workbook. Returns false if none is found.

Required Input Parameters:
sheet_name: String, the name of the sheet that is being evaluated
wb: Workbook, the workbook object to evaluate
Variables:
sheet, wb, sheet_name
Subfunctions:
None
Used In:
Current_Chart_Loop

POWERPOINT_OPEN:

Description:
Opens the PowerPoint app and a new presentation from the user specified template file path, which is minimized to the tray. If the Chart Type DV is set to All, it opens the blank template slide and extends it to the number of slides needed to capture all the data.

Required Input Parameters:
None
Variables:
powerpoint_app, new_presentation, all_slides_count
Subfunctions:
None
Used In:
Single_DMA_Loop

POWERPOINT_SAVE:

Description:
If fast saving is enabled, saves new_presentation to the correct local folder location based on whether the Quarter DV is set to Ad Hoc or not. If it is set to Ad Hoc, the presentation is saved to the DMA specific sub-folder. Otherwise, file is saved directly to the public folder.

Required Input Parameters:
None
Variables:

New_presentation, local_destination_folder, user_dma, adhoc_file_name

Subfunctions:

None

Used In:

Single_DMA_Loop

POWERPOINT_OBJECT_TEXT_SIZE:

Description:

This function takes the table shape object on the slide and iterates through each row and column in that table to pass through every possible cell. Each cell has its text size set to the Parameters value for the chart. Similarly, rows and columns are resized to the values from the Parameters worksheet.

Required Input Parameters:

shape_name: Shape Object, typically ppt_shape

slide_obj: Slide Object, typically ppt_slide

Variables:

row_index (index), column_index (index), minimum_width, pasted_table, text_size, cell_height, cell_width

Subfunctions:

None

Used In:

Current_Chart_Loop

POWERPOINT_SLIDE_MOVE:

Description:

Accepts a slide and shape then repositions that shape on the slide based on position values from the Parameters worksheet. If the slide is centered, the horizontal and vertical positioning values are ignored.

Required Input Parameters:

shape_obj: Shape Object; the chart from the slide, typically ppt_shape

slide_obj: Slide Object; the current slide, typically new_presentation

slide_index: Integer; the index number of the slide_obj

Variables:

slide_obj, shape_obj, slide_index

Subfunctions:

None

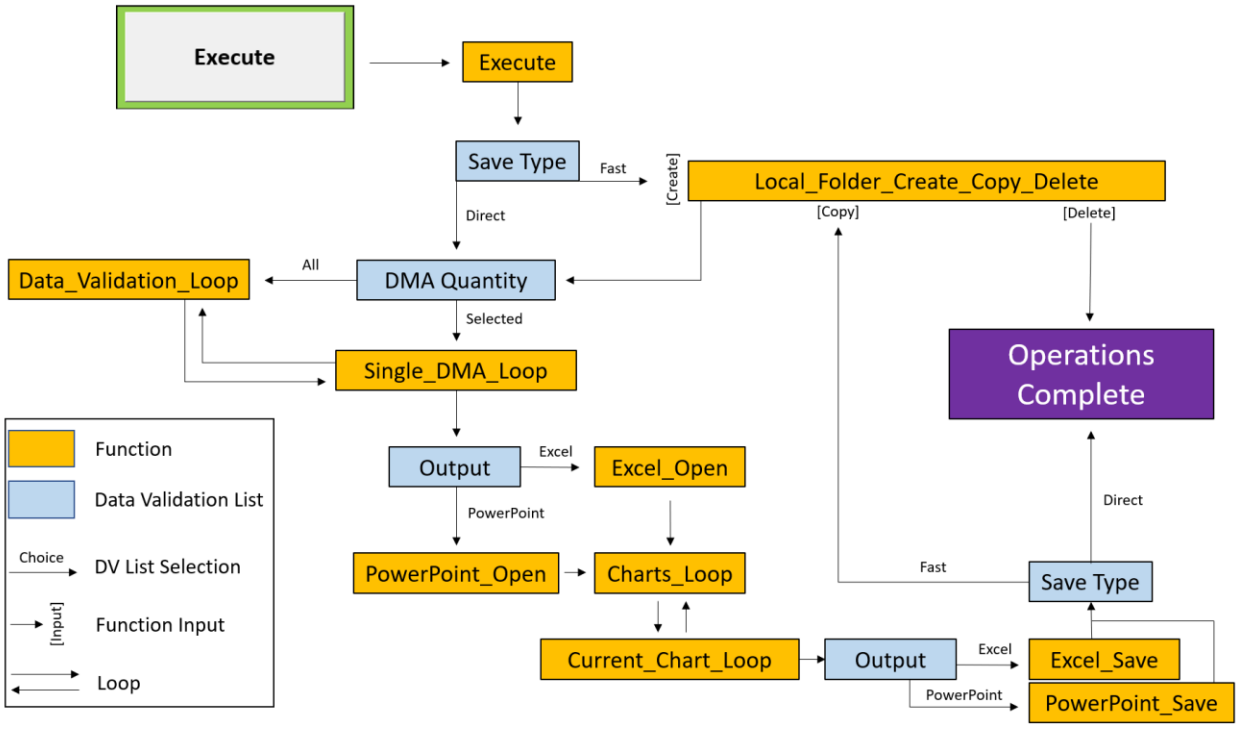
Used In:
Current_Chart_Loop

COMMON VBA METHODS:

This section covers some of the more commonly used VBA methods in the code. For the most detailed information possible, google the method and read Microsoft official documentation.

- Application.CutCopyMode: clear clipboard
- Application.ScreenUpdating: prevents application from updating the screen as it calculates
- Application.DisplayAlerts: suppresses alerts from applications
- StrComp: compares two strings, equality = 0

FUNCTION FLOW:



ERRORS AND SOLUTIONS:

Despite the best efforts of developers, VBA is a complicated programming language that imperfectly interfaces with a closed application interface. VBA errors are mercurial and can arise from many different sources. The best way to prevent these errors is to ensure the main

analysis tool workbook is unaltered except where this document specifies. Most errors arise from variable type mismatches and overloaded Excel/PowerPoint application requests.

To ensure the fewest errors:

- Close all other Excel workbooks
- Close all PowerPoint presentations
- **Close and reopen Excel and try again to reset the VBA environment**
    - You must close **all** Excel processes, best achieved via Task Manager

Errors will still happen, but there are best practices to resolve them. **As always, Google and Microsoft Documentation are your friends.** This section covers common errors that crop up in VBA. It is structured as follows:

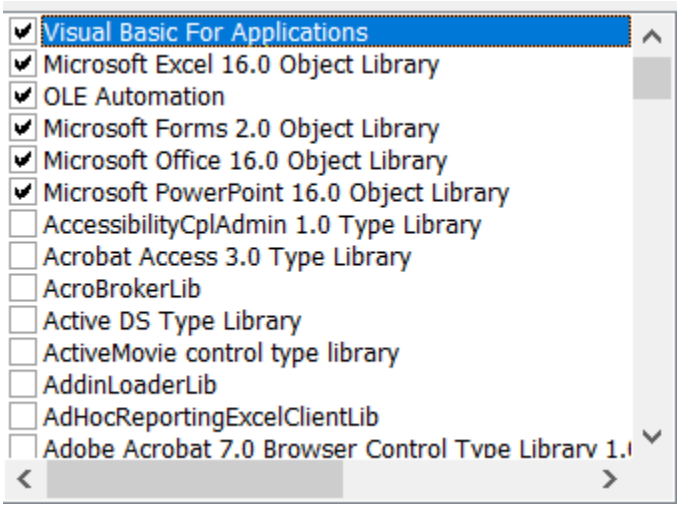
Error

- Description
- Resolution

COMMON ERRORS

Error 429:

- This error occurs when the user does not have the correct VBA Microsoft libraries enabled.
- Open the VBA environment and navigate to the Tools tab then References and enable the following libraries (specifically Microsoft PowerPoint 16.0 Object Library):



Error 1004 – PasteSpecial method of class Range failed:

- The clipboard is probably empty, and the code is trying to paste nothing.
- Make sure you are passing a worksheet to Copy_Paste_Wait if it fails on the Excel export, or a slide if it fails on the PowerPoint export. Open and Close Excel fully and attempt again.

Error 70 – Permission Denied:

- This is an error that can crop up when VBA attempts to save over an open PowerPoint or Excel file.
- Reset the VBA environment by closing Excel and PowerPoint fully and opening it again. Be sure the local temporary folder or any subfiles are not open.