SPV Reporting Tool

Excel VBA Functionalities

Golden Copy
This portion of the User Guide will go through troubleshooting techniques for certain errors as well as the procedure to modify the code in various situations. In order to understand what changes will cause errors in the code, this page will breakdown the basic functionalities of the Golden Copy. The following table sequentially lists the actions that occur to produce the Golden Copy and which module the code resides in.

### Golden Copy Report Process Flow

<table>
<thead>
<tr>
<th></th>
<th>Action</th>
<th>Description</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>User Input of date via Show Calendar button</td>
<td>The SPV data from this date will be imported to Excel. The calendar exists in the VBA component of Excel</td>
<td>Calendar, UserForm1, CONTROL_MainImport</td>
</tr>
<tr>
<td>2.</td>
<td>User Input click ‘Create Golden Copy’ button</td>
<td>This button will run the macros that generate the report.</td>
<td>CONTROL_MainImport</td>
</tr>
<tr>
<td>3a.</td>
<td>Open Main report webpage</td>
<td>An Internet Explorer window will open, direct itself to the website of the ASIA report based on the date entered.</td>
<td>CONTROL_MainImport</td>
</tr>
<tr>
<td>3b.</td>
<td>Open Sub reports on the webpage</td>
<td>A link within the Main report is triggered to display the Sub reports.</td>
<td>CONTROL_MainImport</td>
</tr>
<tr>
<td>3c.</td>
<td>Import to Excel</td>
<td>All data on the webpage is pasted into the HTMLScrape tab</td>
<td>CONTROL_MainImport</td>
</tr>
<tr>
<td>4.</td>
<td>Place Main and Sub reports to the Main and Sub tabs</td>
<td>The macro counts how many data rows of the Main/Sub report exist and pastes them into the Main/Sub report where there is an empty row.</td>
<td>HTMLScrape</td>
</tr>
</tbody>
</table>
## Golden Copy Report Process Flow cont’d

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>Repeat 3-4 for EMEA and GLOBAL reports</td>
<td>CONTROL_MainImport, HTMLScrape</td>
</tr>
<tr>
<td></td>
<td>All data on the HTMLScrape tab is deleted before reuse.</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Reformat Main tab data</td>
<td>MainCleanSheet</td>
</tr>
<tr>
<td></td>
<td>The SPV Name data in the Main report is formatted as merged cell. To allow for data mappings this merged cell is unmerged and the resulting empty columns are deleted.</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Fill the Golden Copy tab</td>
<td>GoldenCopy</td>
</tr>
<tr>
<td></td>
<td>Formulas that point to data in the Main and Sub tabs are entered for each SPV trade.</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Fill the Matured tab</td>
<td>MaturedTabCreate</td>
</tr>
<tr>
<td></td>
<td>This copies the Golden Copy tab and filters the trades to only display trades that have matured on or before the current date.</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Create new workbook with only the Golden Copy and Matured tab</td>
<td>SaveNewWorkbook</td>
</tr>
<tr>
<td></td>
<td>The Golden Copy and Matured tab are both copied and pasted into a new workbook as values. This means there will be no formulas in the cells.</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>Create a Summary tab in the new workbook</td>
<td>SummaryTab</td>
</tr>
<tr>
<td></td>
<td>The Summary tab consists of two tables. The bottom table has formulas that count the number of trades in the report against the number of trades listed in the static table above.</td>
<td></td>
</tr>
</tbody>
</table>
The following slides describe how changes in the actually Golden Copy report can be made. Changes can only be made if the data is still available in the import of data. These changes will need to be implemented in the VBA code. The following steps will describe how to open the VBA editor

- To open the VBA editor
  - Go to the tab Developer > Visual Basic
Golden Copy Output changes – Add a column in the Golden Copy

- If the Developer tab is not available:
  - Click on the Microsoft Office button > Excel Options > In the Popular field check the Show Developer tab in Ribbon box
Golden Copy Output changes – Add a column to the Golden Copy

Scenario 1: Add a column in the Golden Copy

To add a column to the Golden Copy tab, modifications to the VBA macro need to be made.

- Open VBA. In the left hand panel:
  - Expand the VBA Project for the file you have open (In this case its Create Golden Copy for 2007 Dec 5)
  - Expand the Modules folder
  - Select the GoldenCopy Module and window will open in the right pane
The code listed in the right field is that part of the macro that creates the Golden Copy report. Formulas are placed into the first row of the report and then copied and pasted for all other rows.

Underneath the text ‘// fill first data row with formulas’ begins the automatic entry of formulas.

• To add a column after column B and before column C:
  • Find the line of code that says Range("C2").Select. Above this line enter: Range("C2").Select
    ActiveCell.FormulaR1C1 = “ Enter formula here”
  • The formula that can be entered is simply an formula that can be entered in the cell of an worksheet, however, if quotes are used in the formula they need to be written as 2 quotes in the VBA code.
    Ex: cell formula =IFERROR(S2*U2, "ERROR") VBA formula =IFERROR(RC[-3]*RC[-1], ""ERROR"")
  • For each Range("C2").Select expression that follows until ‘// copy lines for the number of sub trades’, the letter needs to be changed to the next letter in the alphabet.
    Ex: previous expression: Range ("D2").Select new expression: Range("E2").Select
    Range("Z2").Select new expression: Range("AA2").Select
    Range("AB2").Select new expression: Range("AC2").Select

• Scroll down to the text ‘// Header for Golden Copy’
• Find the line of code that says Range("C1").Select. Above this line enter: Range("C1").Select
    ActiveCell.FormulaR1C1 = “ Enter column header name here”
• For each Range("D1").Select expression that follows until ‘// copy lines for the number of sub trades’, the letter needs to be changed to the next letter in the alphabet.
To add a column after column C and before column D cont’d:

- Open the MaturedTabCreate Module
- Scroll down and find the text at the bottom that says: `Sheets("Golden Copy").Select Range("AB1").Select Range(Selection, Selection.End(xlDown)).Delete`

- Edit the `Range("AB1").Select` line in the same manner that it was edited before. The letter was changed to the next letter of the alphabet. This line will now say `Range("AC1").Select` (This command deletes the last entry of the Golden Copy which was the Transaction Description Column because it was used to create the matured tab but is now no longer needed.)
- Save changes. The added column will now appear in the Golden Copy when it is run.

Depending on the location of the added column, these changes may affect other aspects of the program. For example, the Summary tab works by comparing certain columns in the Golden Copy. If these columns move in position due to adding/deleting columns, the Summary tab will need to be updated to point to the correct values. These changes will need to be made in the SummaryTab module.
Golden Copy Output changes – Delete a column to the Golden Copy

Scenario 2: Delete a column in the Golden Copy

To delete a column to the Golden Copy tab, modifications to the VBA macro need to be made.

- Open VBA. In the left hand panel:
  - Expand the VBA Project for the file you have open (In this case its Create Golden Copy for 2007 Dec 5)
  - Expand the Modules folder
  - Select the GoldenCopy Module and window will open in the right pane
The code listed in the right field is that part of the macro that creates the Golden Copy report. Formulas are placed into the first row of the report and then copied and pasted for all other rows.

Underneath the text '// fill first data row with formulas begins the automatic entry of formulas.

• To delete column C:
  • Find the consecutive lines of code that says Range("C2").Select.  
    ActiveCell.FormulaR1C1 = “Formula”
    and delete them
  • For each Range("D2").Select expression that follows until '// copy lines for the number of sub trades, the letter needs to be changed to the previous letter in the alphabet.
    Ex: previous expression: Range ("D2").Select
    Range("Z2").Select
    Range("AB2").Select
    new expression: Range("C2").Select
    new expression: Range("Y2").Select
    new expression: Range("AA2").Select
• Scroll down to the text '// Header for Golden Copy  
• Find the line of code that says Range("C1").Select.  
  ActiveCell.FormulaR1C1 = “ Header name”
  and delete them
• For each Range("D1").Select expression that follows until '// copy lines for the number of sub trades, the letter needs to be changed to the previous letter in the alphabet.
Golden Copy Output changes – Delete a column in the Golden Copy

• To delete column C cont’d:
  • Open the MaturedTabCreate Module
  • Scroll down and find the text at the bottom that says: `Sheets("Golden Copy").Select` `Range("AB1").Select` `Range(Selection, Selection.End(xlDown)).Delete`
  • Edit the `Range("AB1").Select` line in the same manner that it was edited before. The letter was changed to the `previous` letter of the alphabet. This line will now say `Range("AA1").Select` (This command deletes the last entry of the Golden Copy which was the Transaction Description Column because it was used to create the matured tab but is now no longer needed.)
  • Save changes. The deleted column no longer appear in the Golden Copy when it is run.

Depending on the location of the added column, these changes may affect other aspects of the program. For example, the Summary tab works by comparing certain columns in the Golden Copy. If these columns move in position due to adding/deleting columns, the Summary tab will need to be updated to point to the correct values. These changes will need to be made in the SummaryTab module.
If changes in the HTML Trigger report occur, this affects many aspects of the Golden Copy macros. The following modules will need to be edited:

- CONTROL_MainImport
- HTMLScrape
- MakeHeaders
- GoldenCopy
- MainCleanSheet

While many changes can occur, this user guide will only go over changes in the URL of the Trigger reports, new columns, deleted columns.
HTML changes – New URL of Trigger Reports

If the Trigger reports are moved to a new URL, changes need to be made to the VBA code.

- Open VBA. In the left hand panel:
  - Expand the VBA Project for the file you have open (In this case its Create Golden Copy for 2007 Dec 5)
  - Expand the Modules folder
  - Select the CONTROL_MainImport module and window will open in the right pane

The code listed in the right field is the part of the macro that opens the Internet Explorer window and navigates to the page where the HTML Trigger report exists for ASIA, EMEA, and GLOBAL.
Scroll down to the line of text that says `If log = “ASIA” Then
.navigate “http://use...trigger/” & getdate & “/Trigger...”`

This part of the code directs the program to the ASIA trigger report found at the listed URL. There in a & getdate & interrupting the URL because this is the user input date string that will concatenate the two URL parts and be navigated to as one string. These commands similarly exist for the other two reports in the lines that follow.

• To modify the URL:
  • Enter ASIA URL up to the section that has a date string after .navigate “
  • Enter the rest of the URL after the date string after & getdate & “
  • Make sure to close the URL fields with quotations marks after they are entered.
  • Follow the same steps for the EMEA and GLOBAL URLs

Note: These URL change instructions assume that the date string will be the unique identifier between different reports and that the date string in the URL will be in the format YYYYMMDD
If a new column is added to the **Main** report of the HTML Trigger report, changes need to be made in the following modules:

- HTMLScrape
- MakeHeaders
- GoldenCopy
- MainCleanSheet

1. First, Open VBA. In the left hand panel:
   - Expand the VBA Project for the file you have open (In this case its Create Golden Copy for 2007 Dec 5)
   - Expand the Modules folder
   - Select the HTMLScrape module and window will open in the right pane

The HTMLScrape module holds the raw copy of the HTML page. It takes the Main/Sub report fields and pastes it in the Main/Sub tab.
HTMLScrape modifications:
• Under the text '//' Pastes where the first empty cell was the line Range("A3:AN" & maincount - 1).Select needs to be edited. If one column was added, AN needs to more forward one letter to be AO.

Open the MakeHeaders module to make modifications:
• If a new column is being added after the second column and before the third column:
  • Find the line of code that says '//'Headers for Main sheet directly at the top
  • Find the line of code that says Range("C2").Select. before '//' Headers for Sub Sheet

Above this line enter:
  Range("C2").Select
  ActiveCell.FormulaR1C1 = “ Enter header name here”
• For each Range("C2").Select expression that follows until '//' copy lines for the number of sub trades, the letter needs to be changed to the next letter in the alphabet.
  Ex: previous expression: Range ("D2").Select
  Range("Z2").Select
  Range("AB2").Select
  new expression: Range("E2").Select
  new expression: Range("AA2").Select
  new expression: Range("AC2").Select
Open the GoldenCopy module to make modifications:

- Find the text 'Create unique Coper and ISN identifier in Main Column S' which is in the beginning of the code.
- This section of the code adds a unique identifier column to the Main tab which concatenates the data in column D and column E of the Main tab and reports in the last column. Because a new column has been added to the Main report, the location of this new column needs to be moved forward.
  - For the two lines that say Range(“S2”).Select, the S needs to be replaced with the next column letter, T.

Additionally, the formulas that create the Golden Copy are conditional on the location of the data in the Main and Sub reports. This means that an added or removed column in the report can cause incorrect data mappings. For Example: A field in the Golden Copy points to column G in the Main report that holds the Issue Date. If a new column is added in the Main report before column C, this means that each succeeding column is relocated the following column. Therefore the Issue date in column G is now in column H but the Golden Copy field is still pointing to column G and is retrieving incorrect data.

This means that for every field in the Golden Copy, if it is referencing data in front of the added column, the data mappings should still be correct. For every field in the Golden Copy that is referencing data after the column addition, the formulas have to be updated like so:

- previous formula: ActiveCell.FormulaR1C1 = "=Sub!RC[4]"
- new formula: ActiveCell.FormulaR1C1 = "=Sub!RC[5]"
- previous formula: ActiveCell.FormulaR1C1 = "=Sub!RC[-3]"
- new formula: ActiveCell.FormulaR1C1 = "=Sub!RC[-2]"
The reason the number after C is being changed is because the C4 means look up the value four columns ahead of the formula’s location. If a new column is added the value would have moved to the fifth column ahead of the formula’s location.

This step could potentially involve lots of changes and so it is recommended that a good level of testing accompanies these changes to verify not only that the program doesn’t create errors, but also that the correct data is placed in the right spots.

Open the MainCleanSheet module to make modifications *(Only If a new column is being added before the Status column. If not, these can be ignored)*:

- Find the expression `Columns("B:B").Select` and change both letters to the next letter if only one column is being added
- Find the expression `Columns("C:W").Select` and change both letters to the next letter if only one column is being added
If a column is removed in the **Main** report of the HTML Trigger report, changes need to be made in the following modules:

- HTMLScrape
- MakeHeaders
- GoldenCopy
- MainCleanSheet

- First, Open VBA. In the left hand panel:
  - Expand the VBA Project for the file you have open (In this case its Create Golden Copy for 2007 Dec 5)

- Expand the Modules folder

- Select the HTMLScrape module and window will open in the right pane

The HTMLScrape module holds the raw copy of the HTML page. It takes the Main/Sub report fields and pastes it in the Main/Sub tab.
HTML changes – Remove Column in Main Report

HTMLScrape modifications:
• Under the text '// Pastes where the first empty cell was the line Range("A3:AN" & maincount - 1).Select needs to be edited. If one column was removed, AN needs to more backwards one letter to be AM.

Open the MakeHeaders module to make modifications:
• If the third column is being removed:
  • Find the line of code that says '//Headers for Main sheet directly at the top
  • Find the line of code that says Range("C2").Select. before '// Headers for Sub Sheet
    ActiveCell.FormulaR1C1 = “Header Name”
    delete these two lines completely
  • For each Range("D2").Select expression that follows until '// copy lines for the number of sub trades,
    the letter needs to be changed to the next letter in the alphabet.
    Ex: previous expression: Range ("D2").Select new expression: Range("C2").Select
      Range("Z2").Select new expression: Range("Y2").Select
      Range("AB2").Select new expression: Range("AA2").Select
HTML changes – Remove Column in Main Report

Open the GoldenCopy module to make modifications:

- Find the text `//Create unique Coper and ISN identifier in Main Column S` which is in the beginning of the code.
- This section of the code adds a unique identifier column to the Main tab which concatenates the data in column D and column E of the Main tab and reports in in the last column. Because a column has been removed to the Main report, the location of this new column needs to be moved backward.
  - For the two lines that say `Range("S2").Select`, the S needs to be replaced with the previous column letter, R.

Additionally, the formulas that create the Golden Copy are conditional on the location of the data in the Main and Sub reports. This means that an added or removed column in the report can cause incorrect data mappings. For Example: A field in the Golden Copy points to column G in the Main report that holds the Issue Date. If a new column is added in the Main report before column C, this means that each succeeding column is relocated the following column. Therefore the Issue date in column G is now in column H but the Golden Copy field is still pointing to column G and is retrieving incorrect data.

This means that for every field in the Golden Copy, if it is referencing data in front of the added column, the data mappings should still be correct. For every field in the Golden Copy that is referencing data after the column deletion, the formulas have to be updated like so:

- previous formula: `ActiveCell.FormulaR1C1 = "=Sub!RC[4]"
- new formula : `ActiveCell.FormulaR1C1 = "=Sub!RC[3]"
- previous formula : `ActiveCell.FormulaR1C1 = "=Sub!RC[-3]"
- new formula : `ActiveCell.FormulaR1C1 = "=Sub!RC[-4]"
The reason the number after C is being changed is because the C4 means look up the value four columns ahead of the formula’s location. If a column is removed the value would have moved to the third column ahead of the formula’s location.

*This step could potentially involve lots of changes and so it is recommended that a good level of testing accompanies these changes to verify not only that the program doesn’t create errors, but also that the correct data is placed in the right spots.*

Open the MainCleanSheet module to make modifications *(Only if a column is being removed before the Status column. If not, these can be ignored)*:

- Find the expression Columns("B:B").Select and change both letters to the previous letter if only one column is being removed
- Find the expression Columns("C:W").Select and change both letters to the previous letter if only one column is being added
If a column is added in the **Sub** report of the HTML Trigger report, changes need to be made in the following modules:

HTMLScrape
MakeHeaders
GoldenCopy
MainCleanSheet

• First, Open VBA. In the left hand panel:
  • Expand the VBA Project for the file you have open (In this case its Create Golden Copy for 2007 Dec 5)
  • Expand the Modules folder
  • Select the HTMLScrape module and window will open in the right pane

The HTMLScrape module holds the raw copy of the HTML page. It takes the Main/Sub report fields and pastes it in the Main/Sub tab
HTML changes – Add a Column in Sub Report

HTMLScrape modifications:

• Under the text ‘// Pastes where the first empty cell was’ the line `Range("A3:AN" & maincount - 1).Select` needs to be edited. If one column was added, AN needs to move forwards one letter to be AO.
• Under the text ‘// Pastes in Sub Report’ find the line `Range("B" & maincount + 3 & ":W" & subcount).Select`. If one column was added, W needs to move forwards one letter to be X.

Open the MakeHeaders module to make modifications:

• If a column is being added after the second and before the third:
  • Find the line of code that says ‘//Headers for Main sheet directly at the top’
  • Starting after the `ActiveCell.FormulaR1C1 = “SPV Identifier”` expression, for each `Range("Y1")`.Select expression that follows until ‘// copy lines for the number of sub trades’, the letter needs to be changed to the next letter in the alphabet.
    Ex: previous expression: `Range ("D1").Select` new expression: `Range("E1").Select`
    `Range("Z1").Select` new expression: `Range("AA1").Select`
    `Range("AB1").Select` new expression: `Range("AC1").Select`

• Find the line of code that says `Range("C2").Select` before ‘// Headers for Sub Sheet’
  Above this line enter: `ActiveCell.FormulaR1C1 = “Enter header name here”`

• For each `Range("C2")`.Select expression that follows until ‘// copy lines for the number of sub trades’, the letter needs to be changed to the next letter in the alphabet.
    Ex: previous expression: `Range ("D2").Select` new expression: `Range("E2").Select`
    `Range("Z2").Select` new expression: `Range("AA2").Select`
    `Range("AB2").Select` new expression: `Range("AC2").Select`
Additionally, the formulas that create the Golden Copy are conditional on the location of the data in the Main and Sub reports. This means that an added or removed column in the report can cause incorrect data mappings. For Example: A field in the Golden Copy points to column G in the Main report that holds the Issue Date. If a new column is added in the Main report before column C, this means that each succeeding column is relocated the following column. Therefore the Issue date in column G is now in column H but the Golden Copy field is still pointing to column G and is retrieving incorrect data.

This means that for every field in the Golden Copy, if it is referencing data in front of the added column, the data mappings should still be correct. For every field in the Golden Copy that is referencing data after the column addition, the formulas have to be updated like so:

previous formula:   `ActiveCell.FormulaR1C1 = "=Sub!RC[4]"`
new formula :       `ActiveCell.FormulaR1C1 = "=Sub!RC[5]"

previous formula : `ActiveCell.FormulaR1C1 = "=Sub!RC[-3]"`
new formula :       `ActiveCell.FormulaR1C1 = "=Sub!RC[-2]"`
The reason the number after C is being changed is because the C4 means look up the value four columns ahead of the formula’s location. If a column is added the value would have moved to the fifth column ahead of the formula’s location.

*This step could potentially involve lots of changes and so it is recommended that a good level of testing accompanies these changes to verify not only that the program doesn’t create errors, but also that the correct data is placed in the right spots.*

Open the MainCleanSheet module to make modifications:

- Find the expression `Columns("C:W").Select` and change the second letter, W, to the next letter if only one column is being added.
If a column is removed in the **Sub** report of the HTML Trigger report, changes need to be made in the following modules:

- **HTMLScrape**
- **MakeHeaders**
- **GoldenCopy**
- **MainCleanSheet**

1. First, Open VBA. In the left hand panel:
   - Expand the VBA Project for the file you have open (in this case its Create Golden Copy for 2007 Dec 5)
   - Expand the Modules folder
   - Select the **HTMLScrape** module and window will open in the right pane

The **HTMLScrape** module holds the raw copy of the HTML page. It takes the Main/Sub report fields and pastes it in the Main/Sub tab.
HTML changes – Remove a Column in Sub Report

HTMLScrape modifications:
• Under the text ‘// Pastes where the first empty cell was’ the line Range("A3:AN" & maincount - 1).Select needs to be edited. If one column was removed, AN needs to move backwards one letter to be AM.
• Under the text ‘// Pastes in Sub Report’ find the line Range("B" & maincount + 3 & ":W" & subcount).Select. If one column was removed, W needs to move backwards one letter to be V.

Open the MakeHeaders module to make modifications:
• If the third column is being removed:
  • Find the line of code that says ‘// Headers for Main sheet directly at the top
  • Starting after the ActiveCell.FormulaR1C1 = “SPV Identifier” expression, for each Range(“Y1”).Select expression that follows until ‘// copy lines for the number of sub trades’, the letter needs to be changed to the previous letter in the alphabet.
    Ex: previous expression: Range (“D1”).Select    new expression: Range (“C1”).Select
     Range(“Z1”).Select    new expression: Range (“Y1”).Select
     Range(“AB1”).Select    new expression: Range (“AA1”).Select
  • Find the line of code that says Range(“C2”).Select. before ‘// Headers for Sub Sheet
    ActiveCell.FormulaR1C1 = “Header Name”
    delete these two lines.
  • For each Range(“C2”).Select expression that follows until ‘// copy lines for the number of sub trades’, the letter needs to be changed to the previous letter in the alphabet.
    Ex: previous expression: Range (“D2”).Select    new expression: Range (“C2”).Select
     Range(“Z2”).Select    new expression: Range (“Y2”).Select
     Range(“AB2”).Select    new expression: Range (“AA2”).Select
Additionally, the formulas that create the Golden Copy are conditional on the location of the data in the Main and Sub reports. This means that an added or removed column in the report can cause incorrect data mappings. For Example: A field in the Golden Copy points to column G in the Main report that holds the Issue Date. If a new column is added in the Main report before column C, this means that each succeeding column is relocated the following column. Therefore the Issue date in column G is now in column H but the Golden Copy field is still pointing to column G and is retrieving incorrect data.

This means that for every field in the Golden Copy, if it is referencing data in front of the added column, the data mappings should still be correct. For every field in the Golden Copy that is referencing data after the column **deletion**, the formulas have to be updated like so:

previous formula:  `ActiveCell.FormulaR1C1 = "=Sub!RC[4]"`
new formula:  `ActiveCell.FormulaR1C1 = "=Sub!RC[3]"

previous formula:  `ActiveCell.FormulaR1C1 = "=Sub!RC[-3]"
new formula:  `ActiveCell.FormulaR1C1 = "=Sub!RC[-4]"`
Open the MainCleanSheet module to make modifications:

• Find the expression Columns("C:W").Select and change the second letter, W, to the previous letter if only one column is being added.

The reason the number after C is being changed is because the C4 means look up the value four columns ahead of the formula’s location. If a column is removed the value would have moved back to the third column ahead of the formula’s location.

*This step could potentially involve lots of changes and so it is recommended that a good level of testing accompanies these changes to verify not only that the program doesn’t create errors, but also that the correct data is placed in the right spots.*
To apply formatting changes to any of the output reports (Golden Copy, Matured, Summary) this code can be entered in the SaveNewWorkbook module. If the user is unsure how to write the code to implement these changes, it is best to use the Record a Macro feature of Excel.

- In the View tab on the main toolbar, click on the arrow under Macros and then choose Record a macro.
• You will now be prompted to name the Macro, then select OK.
• Apply all of the formatting changes desired. The macro is recording all of the actions.
• When you are done applying changes, click on the arrow under macros again and choose Stop Recording.
• Now, open VBA. In the left hand panel:
  • Expand the VBA Project for the file you have open (In this case its Create Golden Copy for 2007 Dec 5)
  • Expand the Modules folder
  • Select the Module1 module and window will open in the right pane
• Next to Sub will be the name you have chosen for the macro. The lines of code below it are the actions you applied to the sheets. Copy the name of the macro.
• Open the SaveNewWorkbook module.
• Before the line of code that says MsgBox "Report Completed Successfully.", enter Application.Run “Name of Macro”
• Save and close and the program will now apply your formatting changes.
SPV Reporting Tool

Excel VBA Functionalities

Golden Copy
The Trigger and Error Report file works very similarly to the Golden Copy. Each file contains macros that import data from the HTML sources and places it into a Main and Sub tab. After this the Golden Copy file provides data mappings to produce the Golden Copy reports. The Trigger and Error reports will have separate data mappings to produce those reports.

For any troubleshooting issues regarding the import of data from HTML, explanations and troubleshooting guides can be seen in the Golden Copy section of this User Guide.
If another field is desired in the Error Report summary, first open VBA. In the left hand panel:

- Expand the VBA Project for the file you have open
- Expand the Modules folder
- Select the ErrorReport module and a window will open in the right pane

The ErrorReport Module contains all the code that creates data mappings and formatting to the Error tab.
• Follow the code to the first line that says For strm = 1 To m + 1
• This piece of code says:
  • If the Swap PV is a Price Error, continue to the data mappings below
  • The data mappings pick one column of the Main report and places that data into one column of the Error Report. The chosen column is the letter that appears in the statement Range("D"....).Select

• To add a column to the Error report, enter the five lines below:
  Sheets("Main").Select
  Range("E" & strm).Select
  Sheets("Errors").Select
  Range("D" & wtp).Select
  ActiveSheet.Paste

  Sheets("Main").Select
  Range("G" & strm).Select
  Sheets("Errors").Select
  Range("C" & wtp).Select
  ActiveSheet.Paste

Static Data

Data from Column D of the Main report...
• If the added column was inserted before a preexisting column, each:
  Sheets("Errors").Select
  Range("C" & wtp).Select
expressions after the addition needs to be edited so that the C is now the next letter of the alphabet.
Ex: previous expression: Range("D1").Select
  new expression: Range("E1").Select
  new expression: Range("AA1").Select
  new expression: Range("AC1").Select

• This process needs to be similarly repeated three more times.
• Find the line that says 'If Error downloading price'
• To add a column to the Error report, enter the five lines below:
  Sheets("Sub").Select
  Range("D") & strs).Copy
  Sheets("Errors").Select
  Range("C" & wtp).Select
  ActiveSheet.Paste
with the appropriate letters representing where the data is coming from and where it will be pasted.
• If the added column was inserted before a preexisting column, each:
  Sheets("Errors").Select
  Range("C" & wtp).Select
expressions after the addition needs to be edited so that the C is now the next letter of the alphabet.
• Follow these same steps after the 'If Error downloading rating' and 'If Error in Fitch Current' expressions
If a field needs to be removed in the Error Report summary, first open VBA. In the left hand panel:

- Expand the VBA Project for the file you have open
- Expand the Modules folder
- Select the ErrorReport module and a window will open in the right pane

The ErrorReport Module contains all the code that creates data mappings and formatting to the Error tab.
• Follow the code to the first line that says For strm = 1 To m + 1
• This piece of code says:
  • If the Swap PV is a Price Error, continue to the data mappings below
  • The data mappings pick one column of the Main report and places that data into one column of the Error Report. The chosen column is the letter that appears in the statement Range(“D”....).Select

• To remove a column from the Error report, enter the five lines below that correspond to that field. If you want to remove the third column look for a C in the second Range expression:
  Sheets(“Main”).Select
  Range(“D” & strm).Copy
  Sheets(“Errors”).Select
  Range(“C” & wtp).Select
  ActiveSheet.Paste

And delete all of these lines
Error Report – Delete a Column

• If the removed column preceded other columns, for each:
  Sheets(“Errors”).Select
  Range(“D” & wtp).Select
expressions after the removal needs to be edited so that the D is now the previous letter of the alphabet.
  Ex: previous expression:  Range (“D1”).Select new expression:  Range(“C1”).Select
  Range(“Z1”).Select new expression:  Range(“Y1”).Select
  Range(“AB1”).Select new expression:  Range(“AA1”).Select

• This process needs to be similarly repeated three more times.
• Find the line that says 'If Error downloading price
• To remove a column from the Error report,  find the five lines below:
  Sheets(“Sub”).Select
  Range(“D”) & strs).Copy
  Sheets(“Errors”).Select
  Range(“C” & wtp).Select
  ActiveSheet.Paste
with the appropriate letters representing where the data is coming from and where it will be pasted. And delete them
• If the removed column preceded other columns, for each:
  Sheets(“Errors”).Select
  Range(“D” & wtp).Select
expressions after the addition needs to be edited so that the D is now the previous letter of the alphabet.
• Follow these same steps after the  'If Error downloading rating and 'If Error in Fitch Current expressions
1. Expand the VBA Project for the file you have open

2. Expand the Modules folder

3. Select the `CreateTriggerReport` module and a window will open in the right pane (This module contains all the code that creates data mappings and formatting to the Trigger tab.)
1. If you want to add a column to the existing ones please follow the code to the first line that says
   \[\text{For } \text{strm} = 1 \text{ To } m + 1\]

2. To add a column to the Trigger Report from the Main/Sub output sheet, enter code similar to this one:

   First you should write the sheet you want to get the data from. Simply copy and paste the circled code and change the letter “B” with the corresponding column letter from the Main tab and continue with the letter “I” for the column which you will add to the Trigger Report. (the next column letters should be “J”, “K”, “L”, etc.)

3. Your code should look like this:

   ```vba
   Sheets("Sub").Select
   Range("B" & strs).Copy
   Sheets("Triggers").Select
   Range("I" & wtp).Select
   ActiveSheet.Paste
   ```
If the added column was inserted before a preexisting column, each:

- `Sheets(“Triggers”).Select`  
- `Range(“C” & wtp).Select`

expressions after the addition needs to be edited so that the C is now the next letter of the alphabet.

Ex: previous expression:  
- `Range(“D1”).Select`
- `Range(“Z1”).Select`
- `Range(“AB1”).Select`

new expression:  
- `Range(“E1”).Select`
- `Range(“AA1”).Select`
- `Range(“AC1”).Select`

This process needs to be similarly repeated three more times.

Find the line that says ‘Get all triggers which contain Collateral Price’.

To add a column to the Trigger report, enter the five lines below:

- `Sheets(“Sub”).Select`
- `Range(“D”) & strs).Copy`
- `Sheets(“Triggers”).Select`
- `Range(“C” & wtp).Select`
- `ActiveSheet.Paste`

with the appropriate letters representing where the data is coming from and where it will be pasted.

If the added column was inserted before a preexisting column, each:

- `Sheets(“Triggers”).Select`
- `Range(“C” & wtp).Select`

expressions after the addition needs to be edited so that the C is now the next letter of the alphabet.

Follow these same steps after the ‘Get all other triggers’ expressions.
1. Expand the VBA Project for the file you have open

2. Expand the Modules folder

3. Select the `CreateTriggerReport` module and a window will open in the right pane (This module contains all the code that creates data mappings and formatting to the Trigger tab.)
1. If you want to delete a column to the existing ones please follow the code to the first line that says

   For strm = 1 To m + 1

2. To delete a column to the Trigger Report from the Main/Sub output sheet, delete the field which contains the column name you want to remove

   2. To remove a column from the Trigger report, enter the five lines below that correspond to that field. If you want to remove the third column look for a C in the second Range expression:

   ```vba
   Sheets("Main").Select
   Range("D" & strm).Copy
   Sheets("Triggers").Select
   Range("C" & wtp).Select
   ActiveSheet.Paste
   ```

   And delete all of these lines.
• This process needs to be similarly repeated three more times.
• Find the line that says 'Get all triggers which contain Collateral Price'
• To remove a column from the Error report, find the five lines below:
  Sheets("Sub").Select
  Range("D" & wtp).Select
  expressions after the removal needs to be edited so that the D is now the previous letter of the alphabet.
  Ex: previous expression: Range ("D1").Select
      Range("Z1").Select
      Range("AB1").Select
  new expression: Range("C1").Select
      Range("Y1").Select
      Range("AA1").Select
• If the removed column preceded other columns, for each:
  Sheets("Triggers").Select
  Range("D" & wtp).Select
  expressions after the addition needs to be edited so that the D is now the previous letter of the alphabet.
  • Follow these same steps after the 'Get all other triggers rating' expressions
• If another field is desired in the Error Report summary, first open VBA. In the left hand panel:

Expand the VBA Project for the file you have open

Expand the Modules folder

Select the Maturity Warning module and a window will open in the right pane

The Maturity Warning module contains all the code that creates data mappings and formatting to the Maturity Warning tab.
Maturity Warning – Add a Column

• Follow the code to the first line that says For i = 2 To numtrades + 1
• This piece of code says:
  • If the Swap PV is a Price Error, continue to the data mappings below
  • The data mappings pick one column of the Main report and places that data into one column of the Error Report. The chosen column is the letter that appears in the statement Range("D"....).Select

• To add a column to the Maturity Warning report, enter the five lines below:
  - Sheets(“Main”).Select
  - Range(“D” & i).Copy
  - Sheets(“Maturity Warning”).Select
  - Range(“C” & vaccell).Select
  - ActiveSheet.Paste

with the appropriate letters representing where the data is coming from and where it will be pasted.
• If the added column was inserted before a preexisting column, each:
  Sheets("Maturity Warning").Select
  Range("C" & vaccell).Select
  expressions after the addition needs to be edited so that the C is now the next letter of the alphabet.
  Ex: previous expression: Range("D1").Select
      Range("Z1").Select
      Range("AB1").Select
  new expression: Range("E1").Select
      Range("AA1").Select
      Range("AC1").Select
If a field needs to be removed in the Error Report summary, first open VBA. In the left hand panel:

- Expand the VBA Project for the file you have open
- Expand the Modules folder
- Select the MaturedWarning module and a window will open in the right pane

The Matured Warning module contains all the code that creates data mappings and formatting to the Maturity Warning tab.
Maturity Warning – Delete a Column

- Follow the code to the first line that says For \( i = 2 \) To numtrades + 1
- This piece of code says:
  - If the Swap PV is a Price Error, continue to the data mappings below
  - The data mappings pick one column of the Main report and places that data into one column of the Error Report. The chosen column is the letter that appears in the statement `Range("D"...)`.Select

- To remove a column from the Maturity Warning report, enter the five lines below that correspond to that field. If you want to remove the third column look for a C in the second Range expression:

  ```vba
  Sheets("Main").Select
  Range("E" & i).Copy
  Sheets("Maturity Warning").Select
  Range("E" & vaccell).Select
  ActiveSheet.Paste
  ```

  ```vba
  Sheets("Main").Select
  Range("D" & i).Copy
  Sheets("Maturity Warning").Select
  Range("D" & vaccell).Select
  ActiveSheet.Paste
  ```

  ```vba
  Sheets("Main").Select
  Range("C" & i).Copy
  Sheets("Maturity Warning").Select
  Range("C" & vaccell).Select
  ActiveSheet.Paste
  ```

  ```vba
  ActiveSheet.Paste
  ```

And delete all of these lines.
If the removed column preceded other columns, for each:

- Sheets("Maturity Warning").Select
- Range("D" & vaccell).Select

Expressions after the removal need to be edited so that the D is now the previous letter of the alphabet.

- Previous expression: Range("D1").Select
- New expression: Range("C1").Select
- Previous expression: Range("Z1").Select
- New expression: Range("Y1").Select
- Previous expression: Range("AB1").Select
- New expression: Range("AA1").Select
Currently, the Maturity Warning tab outputs trades that are due to mature in 14 days. To change this standard, follow the steps listed below.

- Open VBA
- Expand the VBA Project for the file you have open
- Expand the Modules folder
- Select the Maturity Warning module and a window will open in the right pane
- The circled number is the upper limit of days until maturity from the current date that the code will report. Edit this number.
- **EX:** To display all trades that will mature in the next 30 days, replace 14 with 30