|  |
| --- |
| eDealer Template  Logic Design Document |
|  |

**DOCUMENT PURPOSE**

The Process Document has been designed for use on process documentation efforts. Approvals must be obtained prior to implementation of a new process or updates to this document.

Business Flow Design Document Anywhere

Template Released [Date]

Copyright © 2016 - Citi

All rights reserved. Information contained herein is for internal use and may only be used for business purposes authorized by Citi.

1/6

Document Control

1. **Document History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Status:**  **<Draft/ Final>** | **Author** | **Comment / Changes from Prior Version** |
| 1.0 | June 22, 2016 | First version | Krystian Piłat | First version |

1. **Document Reviewers/Approvers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Position (e.g. Client Manager, Project Manager etc.)** | **Reviewer (only)** | **Reviewer and Approver** |
| Krystian Piłat | Tool designer and developer |  | x |
| Mark Delossantos | Requestor and designer | x |  |
| Yuki Muraki | Requestor and designer | x |  |

Table of Contents

[1 Introduction 5](#_Toc517449060)

[2 Overview 6](#_Toc517449061)

[2.1 Problem Statement 6](#_Toc517449062)

[2.2 Metrics 6](#_Toc517449063)

[2.3 Operational Constraints 6](#_Toc517449064)

[2.4 Software Constraints 6](#_Toc517449065)

[2.5 Manual Process Description 6](#_Toc517449066)

[2.6 Target Systems 6](#_Toc517449067)

[3 Impacted Business Areas 7](#_Toc517449068)

[4 Process Diagram 7](#_Toc517449069)

[5 Process Details 7](#_Toc517449070)

[5.1 <process stage 1> 7](#_Toc517449071)

[5.2 <process stage 2> 7](#_Toc517449072)

[6 Exceptions 8](#_Toc517449073)

[6.1 Business Exceptions 8](#_Toc517449074)

[7 eDealer Template tool details 8](#_Toc517449075)

[7.1 Usage instructions 8](#_Toc517449076)

[7.2 Tool description 8](#_Toc517449077)

[8 Logic Steps 9](#_Toc517449078)

[8.1 Validation 9](#_Toc517449079)

[8.1.1 Checking mandatory fields 9](#_Toc517449080)

[8.1.2 Formatting mandatory fields 12](#_Toc517449081)

[8.2 Moving data from “Form” Sheet to “List of Added Securities” Sheet 13](#_Toc517449082)

[8.2.1 Adding SMC attributes mechanism 13](#_Toc517449083)

[8.2.2 Coping data from “Form” Sheet to “List of Added Securities” Sheet 14](#_Toc517449084)

[8.3 Cleaning content of “Form”, if chosen by user 15](#_Toc517449085)

[8.4 Deleting empty lines from “List of Added Securities” Sheet 20](#_Toc517449086)

[8.5 CSV Creation 21](#_Toc517449087)

[8.6 Additional Sheets with schedules (i.e. call) 22](#_Toc517449088)

[8.7 Password details: 24](#_Toc517449089)

[9 Notes 24](#_Toc517449090)

[10 Lesson Learned: 26](#_Toc517449091)

# Introduction

The Logic Design Document (LDD) describes logic and solution used in eDealer template tool. Solution is required as part of eDealer Integration with SMC.

# Overview

## Problem Statement

eDealer Client requires solution for adding new securities to SMC as part of eDealer Integration with SMC. Operational Support around eDealer Integration with SMC.

## Metrics

N/A

## Operational Constraints

N/A

## Software Constraints

Microsoft Office 2010 and 2016 preferred, as any older version can have compliance issues.

## Manual Process Description

N/A

## Target Systems

|  |  |
| --- | --- |
| Name | Description |
| SMC | Security Master Central |

# 

# Impacted Business Areas

ICG, ISG

# Process Diagram

N/A

# Process Details

N/A

## <process stage 1>

## <process stage 2>

# Exceptions

## Business Exceptions

N/A

# eDealer Template tool details

## Usage instructions

**Instructions:**

1. This file contains macro. Enable content to run the macro
2. Template need to be saved on PC, to support 'Create csv' functionality
3. Search for appropriate SMCI in SMC
4. Input security's data in the data fields on the left
5. Fields marked with (\*) are mandatory for SMC
6. Fields marked with (\*\*) are mandatory for eDealer
7. Fields marked with (\*\*\*) are mandatory for Regulatory
8. When IsStep is checked proper tab will be open. Please fulfil accordingly
9. Click 'Add security to the list' when done
   1. After you hit 'Add security...', the data will be copied into the 'List of added securities' table located on the second sheet
10. Click 'Clear 'Form'' and Form will be cleared and you may input information for another security
11. Click 'Create csv' to export added securities to external csv file. File will be create in the same location as macro. This file is ready for 'SMC Bulk Upload process'
12. Click 'Clear 'List of Added Securities'' to clear all created securities. **Remember to Clear form each time at the end of work!**
13. SMCP will be added automatically for new products

## Tool description

* Tool is VBA based solution, developed on Microsoft Office 2010 / 2016 Excel program

# Logic Steps

1. Validation
   1. Button driven validation (Button 20)
      1. Triggered by button. Checking if all mandatory fields are filled
      2. Triggered by button, if field is chosen as “Yes” some of optional fields will be changed to mandatory (Is Inflation Linked\*\*)
   2. Range / Cell change driven
      1. Triggered once cell / range is changed (value is added/updated/removed on a cell / range)
2. Moving data from “Form” Sheet to “List of Added Securities” Sheet
   1. Adding SMC attributes mechanism
   2. Coping data from “Form” Sheet to “List of Added Securities” Sheet
3. Cleaning content of “Form”, if chosen by user
4. Deleting empty lines from “List of Added Securities” Sheet
5. CSV creation
6. Additional Sheets with schedules (i.e. call)
7. Linked Comboboxes logic

## Validation

1. Button driven validation (Button 20)
   1. Triggered by button. Checking if all mandatory fields are filled
   2. Triggered by button, if field is chosen as “Yes” some of optional fields will be changed to mandatory (Is Inflation Linked\*\*)
2. Range / Cell change driven
   1. Triggered once cell / range is changed (value is added/updated/removed on a cell / range)

### Checking mandatory fields

#### Button driven validation (Button 20) .

* Type “a” Triggered by button. Script:

Dim rng As Range, cell As Range

Dim combo\_box As MSForms.ComboBox

Set rng = Sheets("SMC\_mandatory\_field\_list").Range("D2:D45")

i = 0

For Each cell In rng

If cell = "True" Then 'if mandatory field in empty

varia\_range = cell.Offset(, -2)

If Not IsEmpty(varia\_range) Then 'if its filled then highlight by range

Sheet1.Range(varia\_range).Interior.Color = RGB(255, 0, 0)

error\_count = i + 1

Else

varia\_box = cell.Offset(, -1) 'if its filled then highlight by combobox

Set combo\_box = Sheet1.OLEObjects(varia\_box).Object

combo\_box.BackColor = RGB(255, 0, 0)

error\_count = i + 1

End If

Else 'if mandatory field in NOT empty

varia\_range = cell.Offset(, -2)

If Not IsEmpty(varia\_range) Then 'if its filled then highlight by range

Sheet1.Range(varia\_range).Interior.Color = RGB(255, 255, 255)

Else

varia\_box = cell.Offset(, -1) 'if its filled then highlight by combobox

Set combo\_box = Sheet1.OLEObjects(varia\_box).Object

combo\_box.BackColor = RGB(255, 255, 255)

End If

End If

Next cell

If error\_count >= 1 Then

MsgBox ("Mandatory field cannot be left blank!")

Sheet1.Protect Password:="GabenPlz!!"

Exit Sub

Else

TAB SMC\_mandatory\_field\_list – source of field to make them mandatory

|  |  |  |  |
| --- | --- | --- | --- |
| field | Range | Combobox | Value |
| SMCI | D4 |  | FALSE |
| Market\_Sektor |  | ComboBox1 | FALSE |
| Sec\_lvl\_1 |  | ComboBox60 | FALSE |
| Sec\_lvl\_2 |  | ComboBox4 | FALSE |
| Description | D14 |  | FALSE |
| Country |  | ComboBox3 | FALSE |
| Issue\_Currency |  | ComboBox7 | FALSE |

Adding new field to be mandatory:

1. Go to ‘SMC\_mandatory\_field\_list’ TAB
2. Provide file name (any kind, it’s not script related)
3. Provide range / adress or Combobox name to appropriate column
4. Copy formula from Value column (=ISBLANK(Form!ID26))
5. Change range in VBA script
   1. Set rng = Sheets("SMC\_mandatory\_field\_list").Range("D2:D45") ' this is list of all mandatory fields

#### Button driven validation (Button 20).

* Type “b” Triggered by button, if field is chosen as “Yes” some of optional fields will be changed to mandatory (Is Inflation Linked\*\*):

If Is\_Inflation\_Linked = "YES" Then 'IsInflation Linked

If IsEmpty(Base\_CPI) = True Or IsEmpty(Reference\_Index) = True Then

Output2 = MsgBox("If 'Is Inflation Linked' then 'Base CPI' and 'Reference Index' cannot be left blank", vbCritical, "Error")

Sheet1.Range("D58").Interior.Color = RGB(255, 0, 0)

Sheet1.Range("D60").Interior.Color = RGB(255, 0, 0)

Exit Sub

Else

Sheet1.Range("D58").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D60").Interior.Color = RGB(255, 255, 255)

End If

Else

End If

#### Range / Cell change driven Script

* Type “a” Triggered once cell / range is changed (value is added / updated / removed on a cell / range)

Sub Worksheet\_Change(ByVal Target As Range)

If Target.Address(0, 0) = "D4" Then 'SMCI is numeric

Sheet1.Unprotect Password:="GabenPlz!!"

If Sheet1.Range("D4").Value = vbNullString Then

Sheet1.Protect Password:="GabenPlz!!"

Exit Sub

ElseIf Not IsNumeric(Sheet1.Range("D4").Value) Then

MsgBox "Please enter SMCI only as nubmer"

Sheet1.Range("D4").Interior.Color = RGB(255, 0, 0)

Sheet1.Protect Password:="GabenPlz!!"

Exit Sub

Else

Sheet1.Range("D4").Interior.Color = RGB(255, 255, 255)

Sheet1.Protect Password:="GabenPlz!!"

End If

End If

### Formatting mandatory fields

Logic same as in 8.2.1.3 Range / Cell change driven Script

## Moving data from “Form” Sheet to “List of Added Securities” Sheet

1. Moving data from “Form” Sheet to “List of Added Securities” Sheet
   1. Adding SMC attributes mechanism
   2. Coping data from “Form” Sheet to “List of Added Securities” Sheet
2. Cleaning content of “Form”, if chosen by user
3. Deleting empty lines from “List of Added Securities” Sheet

### Adding SMC attributes mechanism

* Data provided by end user are mapped to hidden lines on “Form” Sheet and then moved to “List of Added Securities” Sheet.
* It provides format changes simplification and it put ready to copy input to BULK upload format.
* It’s also easy to manage and track mapping.

### Coping data from “Form” Sheet to “List of Added Securities” Sheet

Dim rngSource As Range

Dim rngDestination As Range

Set rngSource = Sheets("Form").Range("Y4:AJ" & Cells(Rows.Count, "AA").End(xlUp).Row)

Set rngDestination = Sheets("List of Added Securities").Cells(Rows.Count, 1).End(xlUp).Offset(1, 0)

rngSource.Copy

rngDestination.PasteSpecial Paste:=xlPasteValuesAndNumberFormats

Range("AA79:AJ1000").Select

Selection.ClearContents

Range("B2").Select

End If

Call UserForm\_Initialize

Sheets("Put").Visible = False

Sheets("Step").Visible = False

Sheets("Call").Visible = False

Call KillRows

Dim answer2 As Integer

answer2 = MsgBox("Do you want to clear Form?", vbYesNo + vbQuestion, "Empty Sheet")

If answer2 = vbYes Then

Call Button195\_Click

Else

'do nothing

End If

End Sub

Private Sub UserForm\_Initialize() 'making checkboxs false as defoult

Sheet1.Range("D92").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D60").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D62").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D66").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D70").Interior.Color = RGB(255, 255, 255)

Sheet1.ComboBox13.BackColor = RGB(255, 255, 255)

Sheet1.Range("D78").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D74").Interior.Color = RGB(255, 255, 255)

Sheet1.ComboBox15.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox36.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox46.BackColor = RGB(255, 255, 255)

Sheet1.Range("I10").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D76").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D68").Interior.Color = RGB(255, 255, 255)

Sheet1.ComboBox20.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox17.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox19.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox21.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox22.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox23.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox44.BackColor = RGB(255, 255, 255)

Sheet1.Protect Password:="GabenPlz!!"

End Sub

## Cleaning content of “Form”, if chosen by user

Cleaning content of “Form”, if chosen by user (Button195\_Click)

Sub Button195\_Click()

Sheet1.Unprotect Password:="GabenPlz!!"

Range("D4:D20").Select

Selection.ClearContents

Range("D24:D26").Select

Selection.ClearContents

Range("D38:D94").Select

Selection.ClearContents

Range("I4:I74").Select

Selection.ClearContents

Range("I80:I92").Select

Selection.ClearContents

Range("AA79:AJ1000").Select

Selection.ClearContents

Sheet1.ComboBox26.Value = Null 'Is Inflation Linked\*

Sheet1.ComboBox26.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox27.Value = Null 'Is Dual Currency\*

Sheet1.ComboBox27.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox29.Value = Null 'Is Floater\*

Sheet1.ComboBox29.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox31.Value = Null 'IsStep\*

Sheet1.ComboBox31.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox15.Value = Null 'Is Put/Call Discrete\*

Sheet1.ComboBox15.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox33.Value = Null 'Business Days Indicator\*

Sheet1.ComboBox33.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox35.Value = Null 'IsSinkable\*

Sheet1.ComboBox35.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox30.Value = Null 'Fitch Rating Agency\*

Sheet1.ComboBox30.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox32.Value = Null 'Moody's Investor Services\*

Sheet1.ComboBox32.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox34.Value = Null 'Standard and Poor's Rating Services\*

Sheet1.ComboBox34.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox19.Value = Null 'RatingType Moody

Sheet1.ComboBox19.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox21.Value = Null 'RatingType Mood

S Sub Button195\_Click()

Sheet1.Unprotect Password:="GabenPlz!!"

Range("D4:D20").Select

Selection.ClearContents

Range("D24:D26").Select

Selection.ClearContents

Range("D38:D94").Select

Selection.ClearContents

Range("I4:I74").Select

Selection.ClearContents

Range("I80:I92").Select

Selection.ClearContents

Range("AA79:AJ1000").Select

Selection.ClearContents

Sheet1.ComboBox26.Value = Null 'Is Inflation Linked\*

Sheet1.ComboBox26.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox27.Value = Null 'Is Dual Currency\*

Sheet1.ComboBox27.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox29.Value = Null 'Is Floater\*

Sheet1.ComboBox29.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox31.Value = Null 'IsStep\*

Sheet1.ComboBox31.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox15.Value = Null 'Is Put/Call Discrete\*

Sheet1.ComboBox15.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox33.Value = Null 'Business Days Indicator\*

Sheet1.ComboBox33.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox35.Value = Null 'IsSinkable\*

Sheet1.ComboBox35.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox30.Value = Null 'Fitch Rating Agency\*

Sheet1.ComboBox30.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox32.Value = Null 'Moody's Investor Services\*

Sheet1.ComboBox32.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox34.Value = Null 'Standard and Poor's Rating Services\*

Sheet1.ComboBox34.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox19.Value = Null 'RatingType Moody

Sheet1.ComboBox19.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox21.Value = Null 'RatingType Mood

Sheet1.ComboBox21.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox22.Value = Null 'RatingType S&P

Sheet1.ComboBox22.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox23.Value = Null 'RatingValue S&P

Sheet1.ComboBox23.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox28.Value = Null 'Is Putable\*

Sheet1.ComboBox28.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox36.Value = Null 'IIs Put Discrete

Sheet1.ComboBox36.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox39.Value = Null

Sheet1.ComboBox39.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox41.Value = Null

Sheet1.ComboBox41.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox45.Value = Null

Sheet1.ComboBox45.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox46.Value = Null

Sheet1.ComboBox46.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox1.Value = Null

Sheet1.ComboBox1.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox3.Value = Null

Sheet1.ComboBox3.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox7.Value = Null

Sheet1.ComboBox7.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox5.Text = "NO" 'Is When Issued\*

Sheet1.ComboBox5.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox16.Value = Null

Sheet1.ComboBox16.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox18.Value = Null

Sheet1.ComboBox18.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox10.Value = Null

Sheet1.ComboBox10.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox9.Value = Null

Sheet1.ComboBox9.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox24.Value = Null

Sheet1.ComboBox24.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox60.Value = Null

Sheet1.ComboBox60.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox4.Value = Null

Sheet1.ComboBox4.BackColor = RGB(255, 255, 255)

'Sheet1.ComboBox38.Value = Null

Range("D10").Value = Null

Sheet1.Range("D10").Interior.Color = RGB(255, 255, 255)

Range("D4").Value = Null

'Sheet1.Range("D4").Interior.Color = RGB(255, 255, 255)

Range("D14").Value = Null

'Sheet1.Range("D14").Interior.Color = RGB(255, 255, 255)

Range("D20").Value = Null

Sheet1.Range("D20").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D24").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D38").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D44").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D46").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D48").Interior.Color = RGB(255, 255, 255)

Sheet1.ComboBox40.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox42.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox43.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox12.BackColor = RGB(255, 255, 255)

Sheet1.Range("I20").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("I42").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("I44").Interior.Color = RGB(255, 255, 255)

'>>>>>>>>HERE 20180604

Sheet1.Range("D4").Interior.Color = RGB(255, 255, 255)

Sheet1.Range("D14").Interior.Color = RGB(255, 255, 255)

Sheet1.ComboBox37.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox47.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox48.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox49.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox50.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox51.BackColor = RGB(255, 255, 255)

Sheet1.ComboBox52.BackColor = RGB(255, 255, 255)

Range("D12").Value = Null

Range("I82").Value = Null

Range("I84").Value = Null

Range("I86").Value = Null

Range("I88").Value = Null

Range("I90").Value = Null

Range("I92").Value = Null

Range("I54").Value = Null

Range("I60").Value = Null

Range("B2").Select

Sheet1.Protect Password:="GabenPlz!!"

End Sub

## Deleting empty lines from “List of Added Securities” Sheet

Sub KillRows()

Dim lastRow As Long

Dim i As Long

Dim ws As Worksheet

'What's the first row with data?

Const startRow As Long = 2

'Which worksheet are we dealing with?

Set ws = Sheets("List of Added Securities")

Application.ScreenUpdating = False

With ws

lastRow = .Cells(.Rows.Count, "D").End(xlUp).Row

For i = lastRow To startRow Step -1

If .Cells(i, "D").Value = "" Then

'If .Cells(i, "D").Value + .Cells(i, "").Value = 0 Then

.Cells(i, "D").EntireRow.Delete

End If

Next i

End With

Application.ScreenUpdating = True

End Sub

## CSV Creation

“Create csv” Button 189

1. Looking for “Product.IssuerSMCI” string/text, which is used as check if there is anything to be exported to csv. file
2. Defining the range and variables
3. Creating temporary file for CSV creation
4. Passing data to temp file
5. Create name with path
6. Save file and close .csv formatted file in a defined location

Sub Button189\_Click() ' create csv functionality

Dim MyFileName As String

Dim CurrentWB As Workbook, TempWB As Workbook

Dim Bulkk As String, fName As String, findMe As String

Dim match As Range

'Dim matchCheck As Boolean

findMe = "Product.IssuerSMCI"

Set match = ActiveWorkbook.Sheets("List of Added Securities").Cells.Find(findMe)

If Not match Is Nothing Then

fName = match.Offset(, 1).Value

'fName = Sheet6.Range("D5").Value

Bulkk = "BulkUpload\_"

Set CurrentWB = ActiveWorkbook

ActiveWorkbook.Sheets("List of Added Securities").UsedRange.Copy

Set TempWB = Application.Workbooks.Add(1)

TempWB.Sheets(1).Range("A1").PasteSpecial xlPasteValuesAndNumberFormats

Dim Change ' below to "- 4" to become compatible with .xls files

' MyFileName = CurrentWB.Path & "\" & Left(CurrentWB.Name, Len(CurrentWB.Name) - 5) & Format(CStr(Now), "yyy\_mm\_dd\_hh\_mm") & "\_" & ".csv"

MyFileName = CurrentWB.Path & "\" & Left(Bulkk, Len(CurrentWB.Name) - 5) & fName & "\_" & Format(CStr(Now), "yyyy\_mm\_dd\_hh\_mm") & "\_" & ".csv"

Application.DisplayAlerts = False

TempWB.SaveAs Filename:=MyFileName, FileFormat:=xlCSV, CreateBackup:=False, Local:=True

TempWB.Close SaveChanges:=False

Application.DisplayAlerts = True

MsgBox ".csv file have been created in Template location"

'ActiveWorkbook.Sheets("List of Added Securities").UsedRange.Offset(1, 0).ClearContents

Else

MsgBox ".csv file have NOT been created as there is no data available on List of Added Securities"

End If

End Sub

## Additional Sheets with schedules (i.e. call)

1. Appropriate Sheet opening (“Call”)
2. Add new line to schedule
3. Add schedule to “Form” + validation logic (CommandButton3)

* Appropriate Sheet opening (“Call”)

Sub ComboBox12\_Change() 'Is Callable\*

If ComboBox12.Value = "YES" Then

Sheets("Call").Visible = True

Sheets("Call").Activate

ElseIf ComboBox12.Value = "NO" Then

Sheets("Call").Visible = False

End If

End Sub

* Add new line to schedule

Sub CommandButton2\_Click() 'Add new line with data

i = "CallSchedule.CallStartDate"

j = "CallSchedule.CallPrice"

k = "CallSchedule.CallEndDate"

lastRow = Range("A" & Rows.Count).End(xlUp).Row

Cells(lastRow + 1, 1) = i

Cells(lastRow + 1, 3) = j

Cells(lastRow + 1, 5) = k

End Sub

* Add schedule to “Form” + validation logic (CommandButton3)

Sub CommandButton3\_Click() 'Add shedule to draft

Dim iCallStartDate As Long 'iCallStartDate check

Dim varrayCallStartDate As Variant

varrayCallStartDate = Range("A1:A" & Cells(Rows.Count, "A").End(xlUp).Row).Offset(0, 1).Value

For iCallStartDate = 1 To UBound(varrayCallStartDate, 1)

If IsEmpty(varrayCallStartDate(iCallStartDate, 1)) = True Then

MsgBox "CallStartDate cannot be empty"

Exit Sub

Else

End If

' do stuff to varrayCallStartDate(i, 1)

Next

Dim iCallPrice As Long 'CallPrice

Dim varrayCallPrice As Variant

varrayCallPrice = Range("C1:C" & Cells(Rows.Count, "C").End(xlUp).Row).Offset(0, 1).Value

For iCallPrice = 1 To UBound(varrayCallPrice, 1)

If IsEmpty(varrayCallPrice(iCallPrice, 1)) = True Then

MsgBox "CallPrice cannot be empty"

Exit Sub

Else

End If

' do stuff to varrayCallPrice(i, 1)

Next

Application.ScreenUpdating = False

Dim copySheet As Worksheet

Dim pasteSheet As Worksheet

Set copySheet = Worksheets("Call")

Set pasteSheet = Worksheets("Form")

Dim rngSource As Range

Dim rngDestination As Range

'Set rngSource = Sheets("Form").Range("Y4:AJ61") 'here is the place of data gathering. From D and I to U on 'form' sheet and then copy the range to output

Set rngSource = copySheet.Range("A2:F" & Cells(Rows.Count, "A").End(xlUp).Row)

Set rngDestination = pasteSheet.Cells(Rows.Count, 27).End(xlUp).Offset(1, 0)

rngSource.Copy

rngDestination.PasteSpecial '(xlPasteValues)

'rngDestination.PasteSpecial Paste:=xlPasteValue, Transpose:= True

rngSource.Select

Selection.ClearContents

MsgBox "Schedule have been moved to draft successfully"

End Sub

* 1. **Password details:**

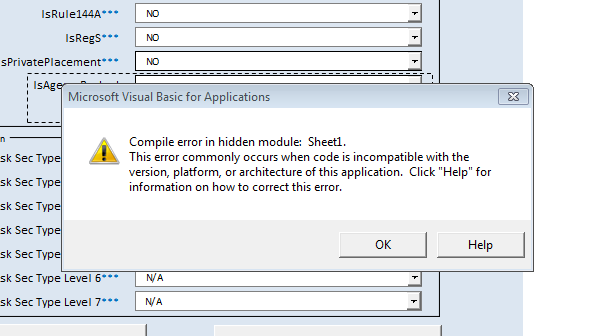
Password: GabenPlz!!

Pass used for:

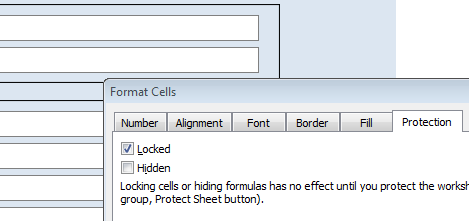
* + Script protection
  + Template cells protection:
    - On a template level
    - On a script level

# Notes

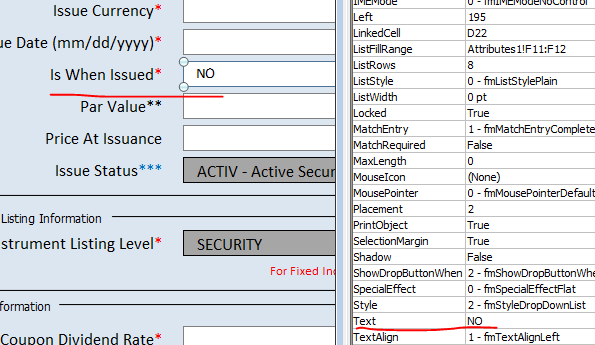
1. Error when template is left as pre-filled. Occurs because of way how combobox change functionality works. This is a reason why we need to left template cleared out at the end of each session.



1. Not used cells are formatted as locked. Thanks to that end user is able to change only those fields that are required.



1. Default value on a combobox (dropdown list)



# Lesson Learned:

* 1. Use “UseForm” for such complex templates. It have built in functionalities that developer of Sheet based template had to recreate with additional vba solutions.