

2012-07-13

# Evaluating collection disposal: life cycle perspectives for integrated library collection management

Ambler, Chelsea

---

Ambler, C. (2012). Evaluating collection disposal: life cycle perspectives for integrated library collection management (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from <https://prism.ucalgary.ca>. doi:10.11575/PRISM/25968

<http://hdl.handle.net/11023/118>

*Downloaded from PRISM Repository, University of Calgary*

## **APPENDIX D: DESIGN**

### **D.1. Tool instructions**

The following is a set of brief instructions to accompany the supportive tool. It is positioned to assist collection management professionals in accomplishing two tasks in order to inform collection disposal decision-making through the evaluative framework: the collection of data and the visual presentation of this information. It is apparent that many libraries may not collect information regarding collection disposal. This could be as there are no established methods or tools to assist in this work. This tool allows for the collection disposal related data. In addition, it can be customized to reflect the unique properties of collection materials and the unique composition of individual library collections. In other sectors, material flow analysis is often accomplished through the use of Sankey diagrams which then inform material flow management. As collection disposal from library collections represents a flow of material, data analysis within the tool compiles collection disposal data and displays a similar flow diagram.

### **D.2. Opening and using the tool**

The tool is an Excel based program. In order to open and use the tool, save the file, ucalgary\_2012\_Ambler\_Chelsea\_Tool.xlsx, to your computer. After opening the file ensure that you enable Macros. This can be done following the prompt that appears. The Tool workbook contains several named worksheets and two User Forms to assist data entry and analysis as follows:

- Collection Disposal Data (worksheet)
- Source (worksheet)

- Designation (worksheet)
- Material (worksheet)
- Disposal (worksheet)
- Compiled Data (worksheet)
- Flow Diagram (worksheet)
- Collection Disposal Data (user form)
- Flow Diagram Creator (user form)

### ***D.2.1. Adding data***

Data can be entered several ways. The Add Data button will launch the Collection Disposal Data user form allowing data to be entered quickly and consistently into the Collection Disposal Data worksheet.

### ***D.2.2. Using default lists***

The Begin Adding Data button on the first page of the Collection Disposal Data user form allows users to quickly begin collecting disposal data. Default lists are comprised of the recommended general material designations and specific material designations found within IFLA's *International Standard Bibliographic Description* (2007). Additional categories and list parameters were derived from the research associated with this master's thesis. Simply choose from the dropdown lists to begin entering data. Not all fields are required. However, more complete data entry will enable analysis of collection disposal flows from multiple perspectives.

### ***D.2.3. Direct entry***

Alternatively, data can be directly entered into the Collection Disposal Data worksheet. With this method care must be taken to enter data with consistent use of designations, and categories so that the underlying programming can accurately display flow diagrams. Greiner (2010) details collection use studies with Excel 2007 and covers how collection data may be exported from integrated library systems and manipulated in spreadsheet programs. This may be one method to glean collection disposal information and data regarding material designations.

### ***D.2.4. Customizing lists***

Lists can be created by using the Collection Disposal Data user form and selecting the button Create Custom Lists. Enter list information according to the instructions and use the associated button for each field to create the custom list so it may be used. Lists are stored on their respective sheets as *named ranges*. Those comfortable with Excel may choose to create custom lists manually to the sheets by creating their own named ranges. Take care that any lists created this way are named ranges and follow the organization of the default lists. To use customized lists select the Add Data button and choose Use Custom List from the Collection Disposal Data userform. Follow the instructions provided to begin adding data using custom lists.

### ***D.2.5. Creating a flow diagram***

When data has been collected choose the Create Flow button. This can be done at any stage so long as there is data in the Collection Disposal Data worksheet and can be repeated any time without impact to the collected data. The Flow Diagram Creator user form prompts the selection of three parameters to organize the diagram. Select Create Flow Diagram when finished. The

diagram is now displayed on the Flow Diagram worksheet. The compiled data used to create the diagram is available on the Compiled Data worksheet.

### **D.3. VBA code**

While user scenarios and errors were anticipated to the best of the author's ability it is possible that errors may still result upon customization or in using the tool differently than as described here. Errors encountered in these situations are not likely to be severe or detrimental as the Macros developed relate only to facilitating data entry, creating named ranges, interpreting data, and creating a visual image within the confines of the specified file. The VBA code underlying the support tool is provide below for those wishing to make modifications and improvements so that their needs are better met.

#### ***D.3.1. Worksheet objects***

```
Private Sub CommandButton1_Click()
UserForm1.Show
End Sub
Private Sub CommandButton2_Click()

lastrowr = Worksheets("Collection Disposal Data").Cells.Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row

With Worksheets("Collection Disposal Data").Range("N3:N" & lastrowr)
If Application.WorksheetFunction.CountA(.Cells) = 0 Then
UserForm2.OptionButton7.Enabled = False
Else:
UserForm2.OptionButton7.Enabled = True
End If
End With
Load UserForm2
UserForm2.Show
End Sub
```

**D.3.2. Userform 1**

```

Private Sub NumbersOnly()
'validate entry to accept only numbers (decimals allowed)
Dim a As String
Dim ac As Control

Set ac = MultiPage1.SelectedItem.ActiveControl
If ac.Name = "ComboBox2" Or ac.Name = "ComboBox3" Or ac.Name = "CommandButton4"
Or ac.Name = "ComboBox12" Or ac.Name = "ComboBox13" Or ac.Name =
"CommandButton15" Or ac.Name = "CommandButton10" Or ac.Name = "CommandButton11"
Then
Exit Sub
Else:
a = Replace(Replace(Replace(ac.Value, ".", "1"), ".", "1"), Chr(13) & Chr(10), "1")

If Not IsNumeric(a) And ac.Value <> "" Then
MsgBox "Enter numbers only."
ac.Value = Left(ac.Value, Len(ac.Value) - 1)
End If
End If
End Sub
Private Sub IntegersOnly()
'validate entry to accept only numbers
Dim ac As Control
Set ac = MultiPage1.SelectedItem.ActiveControl

If Not IsNumeric(ac.Value) And ac.Value <> "" Then
MsgBox "Enter numbers only."
ac.Value = Left(ac.Value, Len(ac.Value) - 1)
End If
End Sub
Private Sub UniqueListnm()
Dim ac As Control
Dim illegal, a As String
Dim i As Integer
Set ac = MultiPage1.SelectedItem.ActiveControl
illegal = "!@#%&*()+-=\\[{}|;':./?><`~""""

If ac.Name = "CommandButton9" Or ac.Name = "CommandButton12" Or ac.Name =
"CommandButton13" Or ac.Name = "CommandButton14" Then
Exit Sub
Else:
For i = 1 To Len(illegal)
If StrComp(Mid(illegal, i, 1), Right(ac.Text, 1)) = 0 Then

```

```

    MsgBox "This is not a valid character."
    ac.Text = Left(ac.Text, Len(ac.Text) - 1)
    i = i + 1
    End If
Next

a = Replace(ac.Text, " ", "_")

For Each nm In ThisWorkbook.Names
If nm.Name = a Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken." & Chr(10) & "Choose
another for this list."
Cancel = True
ac.SelStart = 0
ac.SelLength = Len(ac.Text)
End If

Next nm
End If
End Sub
Private Sub CommandButton1_Click()
'go to add data tab
Me.MultiPage1.Value = 1
End Sub
Private Sub CommandButton3_Click()
'go to add custom data tab
Me.MultiPage1.Value = 2
End Sub
Private Sub CommandButton16_Click()
'go to create custom lists tab
Me.MultiPage1.Value = 3
End Sub
Private Sub CommandButton6_Click()
Unload Me
End Sub
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
If CloseMode <> 1 Then Cancel = 1
End Sub
Private Sub ComboBox2_Exit(ByVal Cancel As MSForms.ReturnBoolean)
'exit general designation list and use selection to determine default material and unit weight if
available
Dim list, a As String
list = "general_designation_default"
a = ComboBox2.Text

```

```

With Range(list)
    Set c = .Find(a, LookIn:=xlValues)
    If Not c Is Nothing Then
        ComboBox4.Value = c.Offset(0, 1).Value
        TextBox23.Value = c.Offset(0, 2).Value
    End If
End With

End Sub
Private Sub ComboBox3_Enter()
'enter specific designation box and use selection from general designation to inform RowSource
if available
Dim list, a As String
list = "general_designation_default"
a = ComboBox2.Value

With Range(list)
    Set c = .Find(a, LookIn:=xlValues)
    If Not c Is Nothing Then
        ComboBox3.RowSource = Replace(ComboBox2.Value, " ", "_")
    End If
End With

End Sub
Private Sub ComboBox3_Exit(ByVal Cancel As MSForms.ReturnBoolean)
'exit specific designation list and use selection to determine default material and unit weight if
available
Dim list, a As String
list = Replace(ComboBox2.Text, " ", "_")
a = ComboBox3.Text

If a = "" Then
Exit Sub
Else:
With Range("general_designation_default")
    Set c = .Find(ComboBox2.Text, LookIn:=xlValues)
    If Not c Is Nothing Then
        With Range(list)
            Set c = .Find(a, LookIn:=xlValues)
            If Not c Is Nothing Then
                ComboBox4.Value = c.Offset(0, 1).Value
                TextBox23.Value = c.Offset(0, 2).Value
            Else:
                ComboBox4.Value = ""
                TextBox23.Value = ""
            End If
        End With
    End If
End With

```



```

        End If
    End With
End If
End With
End If
End Sub
Private Sub TextBox2_Change()
IntegersOnly
End Sub
Private Sub TextBox23_Enter()
If TextBox2.Value = "" Then
MsgBox "Quantity is a required field."
End If
End Sub
Private Sub TextBox23_Change()
NumbersOnly
End Sub
Private Sub TextBox24_Change()
NumbersOnly
End Sub
Private Sub CommandButton4_Click()
'add data button on add data tab - this adds a new line of data to the collection disposal data
worksheet
Dim b As Long
Dim a, lastrow As Integer

If TextBox2.Value = "" Then
MsgBox "Quantity is a required field."
Cancel = True
TextBox2.SetFocus

Else:
Set wscdd = Worksheets("Collection Disposal Data")
lastrow = wscdd.Range("L:L").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row + 1

wscdd.Range("B" & lastrow).Value = ComboBox1.Text
wscdd.Range("D" & lastrow).Value = ComboBox2.Text
wscdd.Range("E" & lastrow).Value = ComboBox3.Text
wscdd.Range("G" & lastrow).Value = ComboBox4.Text
wscdd.Range("I" & lastrow).Value = ComboBox5.Text
wscdd.Range("J" & lastrow).Value = ComboBox6.Text
wscdd.Range("L" & lastrow).Value = TextBox2.Text

If TextBox24.Value = "" Then

```

```
wscdd.Range("M" & lastrow).Value = TextBox23.Text
Else: wscdd.Range("M" & lastrow).Value = ""
End If
```

```
If TextBox24.Value = "" Then
  If TextBox23.Value = "" Then
    wscdd.Range("N" & lastrow).Value = ""
  Else: a = TextBox2.Value
    b = Val(TextBox23.Value)
    wscdd.Range("N" & lastrow).Value = Val(a * b * 0.001)
  End If
Else: wscdd.Range("N" & lastrow).Value = Val(TextBox24.Value)
End If
wscdd.Range("O" & lastrow).Value = TextBox3.Text
```

```
ComboBox1.Text = ""
ComboBox2.Text = ""
ComboBox3.Text = ""
ComboBox4.Text = ""
ComboBox5.Text = ""
ComboBox6.Text = ""
TextBox2.Text = ""
TextBox23.Text = ""
TextBox24.Text = ""
TextBox3.Text = ""
```

```
ComboBox1.SetFocus
```

```
End If
End Sub
Private Sub ComboBox9_Change()
'select from named source lists
ComboBox10.RowSource = ComboBox9.Text
End Sub
Private Sub ComboBox11_Change()
'select from named general material designation lists
ComboBox12.RowSource = ComboBox11.Text
End Sub
Private Sub ComboBox12_Change()
'select general designation list and use selection to determine default material and weight if
available
Dim list, a As String
list = ComboBox11.Text
a = ComboBox12.Text
```

```

If list <> "" Then
With Range(list)
  Set c = .Find(a, LookIn:=xlValues)
  If Not c Is Nothing Then
    ComboBox15.Value = c.Offset(0, 1).Value
    TextBox25.Value = c.Offset(0, 2).Value
  Else:
    ComboBox15.Value = ""
    TextBox25.Value = ""
  End If
End With
End If
End Sub
Private Sub ComboBox13_Enter()
'enter specific designation box and use general designation selection to determine RowSource if
available
Dim a As String
a = ComboBox12.Text

ComboBox13.RowSource = ""

For Each nm In ActiveWorkbook.Names
If nm.Name = Replace(a, " ", "_") Then
ComboBox13.RowSource = Replace(a, " ", "_")
Exit For
End If
Next nm

End Sub
Private Sub ComboBox13_Change()
' select custom specific designation and use selection to determine default material and unit
weight if available
Dim list, a As String
list = Replace(ComboBox12.Text, " ", "_")
a = ComboBox13.Text

If a <> "" Then

For Each nm In ThisWorkbook.Names
If nm.Name = list Then
  With Range(list)
    Set c = .Find(a, LookIn:=xlValues)
    If Not c Is Nothing Then
      ComboBox15.Value = c.Offset(0, 1).Value
      TextBox25.Value = c.Offset(0, 2).Value
    
```

```

Else:
  ComboBox15.Value = ""
  TextBox25.Value = ""
End If
End With

Exit For
End If

Next nm

End If
End Sub
Private Sub ComboBox14_Change()
'select from named material lists
ComboBox15.RowSource = ComboBox14.Text
End Sub
Private Sub ComboBox16_Change()
'select from named disposal method lists
ComboBox17.RowSource = ComboBox16.Text
End Sub
Private Sub ComboBox18_Change()
'select from named disposal reason lists
ComboBox19.RowSource = ComboBox18.Text
End Sub
Private Sub TextBox22_Change()
IntegersOnly
End Sub
Private Sub TextBox25_Enter()
If TextBox22.Value = "" Then
MsgBox "Quantity is a required field."
End If
End Sub
Private Sub TextBox25_Change()
NumbersOnly
End Sub
Private Sub TextBox26_Change()
NumbersOnly
End Sub
Private Sub CommandButton15_Click()
Dim a, laswtrw As Integer
Dim b As Long
If TextBox22.Value = "" Then
MsgBox "Quantity is a required field. Only unit weight and total weight fields are optional."
Cancel = True

```

TextBox22.SetFocus

Else:

Set wscdd = Worksheets("Collection Disposal Data")  
 lastrow = wscdd.Range("L:L").Find("\*", SearchOrder:=xlByRows,  
 SearchDirection:=xlPrevious).Row + 1

wscdd.Range("B" & lastrow).Value = ComboBox10.Text  
 wscdd.Range("D" & lastrow).Value = ComboBox12.Text  
 wscdd.Range("E" & lastrow).Value = ComboBox13.Text  
 wscdd.Range("G" & lastrow).Value = ComboBox15.Text  
 wscdd.Range("I" & lastrow).Value = ComboBox17.Text  
 wscdd.Range("J" & lastrow).Value = ComboBox19.Text  
 wscdd.Range("L" & lastrow).Value = TextBox22.Text

If TextBox26.Value = "" Then  
 wscdd.Range("M" & lastrow).Value = Val(TextBox25.Text)  
 Else: wscdd.Range("M" & lastrow).Value = ""  
 End If

If TextBox26.Value = "" Then  
 If TextBox25.Value = "" Then  
 wscdd.Range("N" & lastrow).Value = ""  
 Else: a = TextBox22.Value  
 b = Val(TextBox25.Value)  
 wscdd.Range("N" & lastrow).Value = Val(a \* b \* 0.001)  
 End If  
 Else: wscdd.Range("N" & lastrow).Value = Val(TextBox26.Value)  
 End If  
 wscdd.Range("O" & lastrow).Value = TextBox21.Text

ComboBox10.Text = ""  
 ComboBox12.Text = ""  
 ComboBox13.Text = ""  
 ComboBox15.Text = ""  
 ComboBox17.Text = ""  
 ComboBox19.Text = ""  
 TextBox22.Text = ""  
 TextBox21.Text = ""  
 TextBox25.Text = ""  
 TextBox26.Text = ""

ComboBox10.SetFocus

End If

```

End Sub
Private Sub MultiPage1_Change()
'load newly created lists into 'use custom lists' form
If Me.MultiPage1.Value = 2 Then
Dim contains As Variant
Dim nm As Name
Dim a, lname As String
Dim i, X As Integer

ComboBox9.Clear
ComboBox11.Clear
ComboBox14.Clear
ComboBox16.Clear
ComboBox18.Clear

For Each nm In ThisWorkbook.Names
    lname = nm.RefersTo
    If lname Like "=Source!" & "*" Then
        ComboBox9.AddItem nm.Name
    ElseIf lname Like "=Designation!$A$" & "*" Then
        ComboBox11.AddItem nm.Name
    ElseIf lname Like "=Material!" & "*" Then
        ComboBox14.AddItem nm.Name
    ElseIf lname Like "=Disposal!$B$" & "*" Then
        ComboBox16.AddItem nm.Name
    ElseIf lname Like "=Disposal!$D$" & "*" Then
        ComboBox18.AddItem nm.Name
    End If
Next nm
End If
End Sub
Private Sub TextBox4_Change()
UniqueListnm
End Sub
Private Sub CommandButton9_Click()
'create a custom source list
Dim listnm, listcnt, newnm, newrng, dlistnm, dlistct, X() As String
Dim i, lastrow, r, sr, e As Integer

listnm = TextBox4.Text
listcnt = Replace(TextBox5.Text, Chr(10), ",")
newnm = Replace(listnm, " ", "_")
dlistnm = ":" & listnm & ":"
dlistct = ":" & Replace(TextBox5.Text, Chr(13) & Chr(10), ":") & ":"

```

```

If TextBox5.Text = "" Then
MsgBox "There need to be list contents for this list to be added."
TextBox5.SetFocus
Cancel = True

Else:

For Each nm In ThisWorkbook.Names
    tnewnm = ":" & Replace(Replace(TextBox5.Text, Chr(13) & Chr(10), ":"), " ", "_") & ":"
    tnm = ":" & nm.Name & ":"

    If nm.Name = newnm Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken."
TextBox4.SetFocus
Exit Sub

    ElseIf InStr(tnewnm, tnm) <> 0 Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken."
TextBox5.SetFocus
Exit Sub
    End If
Next nm

    Set wss = Worksheets("Source")
    lastrow = wss.Range("B:B").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row + 2
    wss.Range("A" & lastrow).Value = listnm
    r = lastrow
    'add listcnt to first cell of range
    wss.Range("B" & lastrow).Value = listcnt

    'pass elements of list contents to range
    X = Split(wss.Range("B" & r), ",")
    For i = 0 To UBound(X)
    wss.Range("B" & r).Value = X(i)
    r = r + 1
    Next i

    'add new defined name
    newrng = "'Source!' & Range("B" & lastrow & ":B" & r - 1).Address
    ActiveWorkbook.Names.Add Name:=newnm, RefersTo:=newrng
    TextBox4.Value = ""
    TextBox5.Value = ""
    TextBox8.SetFocus

```

```

End If
End Sub
Private Sub TextBox8_Change()
UniqueListnm
End Sub
Private Sub TextBox7_Change()
Dim illegal, a, b As String
illegal = "!@#$$%^&*()+-=[][{}];':./?><~""""

For L = 1 To Len(illegal)
    If StrComp(Mid(illegal, L, 1), Right(TextBox7.Text, 1), vbBinaryCompare) = 0 Then
        MsgBox "This is not a valid character."
        TextBox7.Text = Left(TextBox7.Text, Len(TextBox7.Text) - 1)
    End If
Next

a = ":" & Replace(Replace(TextBox7.Text, Chr(13) & Chr(10), ":"), " ", "_")

For Each nm In ThisWorkbook.Names
b = ":" & nm.Name & ":"
If InStr(a, b) <> 0 Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken." & Chr(10) & "Choose
another name for this entry."
Cancel = True
If Len(a) = Len(nm.Name) Then
TextBox7.SelStart = InStr(a, nm.Name) - 1
Else:
TextBox7.SelStart = InStr(a, nm.Name) - 2
End If
TextBox7.SelLength = Len(nm.Name)
End If

Next nm
End Sub
Private Sub TextBox28_Change()
NumbersOnly
End Sub
Private Sub CommandButton10_Click()
'create a custom general material designation list
Dim listnm, listcnt, listcntdm, listcntdw, newnm, newrng, dlistnm, dlistct, tnm, tnewnm, X() As
String
Dim i, r, sr, lastrow, b, e As Integer

listnm = TextBox8.Text
listcnt = Replace(TextBox7.Text, Chr(13) & Chr(10), ",")

```



```
listcntdm = Replace(TextBox27.Text, Chr(13) & Chr(10), ",")
listcntdw = Replace(TextBox28.Text, Chr(13) & Chr(10), ",")
newnm = Replace(listnm, " ", "_")
dlistnm = ":" & listnm & ":"
dlistct = ":" & Replace(TextBox7.Text, Chr(13) & Chr(10), ":") & ":"
```

```
If TextBox7.Text = "" Then
MsgBox "There need to be list contents for this list to be added."
TextBox7.SetFocus
Cancel = True
```

```
ElseIf InStr(dlistct, dlistnm) <> 0 Then
MsgBox " The list name " & TextBox8.Text & " is duplicated in the list contents. Each entry
must be unique."
TextBox7.SetFocus
Cancel = True
```

Else:

```
For Each nm In ThisWorkbook.Names
tnewnm = ":" & Replace(Replace(TextBox7.Text, Chr(13) & Chr(10), ":"), " ", "_") & ":"
tnm = ":" & nm.Name & ":"
```

```
If nm.Name = newnm Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken."
TextBox8.SetFocus
Exit Sub
```

```
ElseIf InStr(tnewnm, tnm) <> 0 Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken."
TextBox7.SetFocus
Exit Sub
End If
```

Next nm

```
Set wsd = Worksheets("Designation")
lastrow = wsd.Cells.Find("*", SearchOrder:=xlByRows, SearchDirection:=xlPrevious).Row +
2
wsd.Range("A" & lastrow).Value = listnm
r = lastrow + 1
```

```
'add listcnt to first cell of range
wsd.Range("A" & r).Value = listcnt
wsd.Range("B" & r).Value = listcntdm
wsd.Range("C" & r).Value = listcntdw
```

```
'pass elements of list contents to range
X = Split(wsd.Range("A" & r), ",")
For i = 0 To UBound(X)
wsd.Range("A" & r).Value = X(i)
r = r + 1
Next i
```

```
newrng = "'Designation!' & Range("A" & lastrow + 1 & ":A" & r - 1).Address
ActiveWorkbook.Names.Add Name:=newnm, RefersTo:=newrng
```

```
r = lastrow + 1
X = Split(wsd.Range("B" & r), ",")
For i = 0 To UBound(X)
wsd.Range("B" & r).Value = X(i)
r = r + 1
Next i
```

```
r = lastrow + 1
X = Split(wsd.Range("C" & r), ",")
For i = 0 To UBound(X)
wsd.Range("C" & r).Value = Val(X(i))
r = r + 1
Next i
```

```
'add new defined name
```

```
ComboBox8.RowSource = newrng
ComboBox8.Value = ComboBox8.list(0)
TextBox9.SetFocus
```

```
End If
End Sub
Private Sub TextBox29_Change()
NumbersOnly
End Sub
Private Sub CommandButton11_Click()
'create a custom specific designation list based on new general material designation list
Dim glistnm, listnm, listcnt, listcntdm, listcntdw, newnm, newrng, X() As String
Dim glistnmRow, i, r, c, c1, c2, li, rc As Integer
glistnm = TextBox8.Text
listnm = ComboBox8.Text
listcnt = Replace(TextBox9.Text, Chr(13) & Chr(10), ",")
listcntdm = Replace(TextBox30.Text, Chr(13) & Chr(10), ",")
listcntdw = Replace(TextBox29.Text, Chr(13) & Chr(10), ",")
```

```

newnm = Replace(listnm, " ", "_")
tnewnm = ":" & Replace(Replace(TextBox9.Text, Chr(13) & Chr(10), ":"), " ", "_") & ":"
tnm = ":" & Replace(glistnm, " ", "_") & ":"

```

```

If InStr(tnewnm, tnm) <> 0 Then
MsgBox "List contents cannot duplicate the general material designation " & glistnm & " ."
TextBox9.SetFocus
Exit Sub
End If

```

```

'add listnm(s) to same row as glistnm and offset each by 3 columns
Set wsd = Worksheets("Designation")
glistnmRow = wsd.Cells.Find(glistnm, SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row

```

```

c = 4
Do Until Cells(glistnmRow, c).Value = ""
c = c + 3
Loop

```

```

wsd.Cells(glistnmRow, c).Value = listnm
c1 = c + 1
c2 = c + 2
r = glistnmRow + 1

```

```

'add listcnt to first cell of range
wsd.Cells(r, c).Value = listcnt
wsd.Cells(r, c1).Value = listcntdm
wsd.Cells(r, c2).Value = listcntdw

```

```

'pass elements of list contents to range
X = Split(wsd.Cells(r, c), ",")
For i = 0 To UBound(X)
wsd.Cells(r, c).Value = X(i)
r = r + 1
Next i

```

```

newrng = "'Designation'!" & Range(Cells(glistnmRow + 1, c), Cells(r - 1, c)).Address
ActiveWorkbook.Names.Add Name:=newnm, RefersTo:=newrng

```

```

r = glistnmRow + 1
X = Split(wsd.Cells(r, c1), ",")
For i = 0 To UBound(X)
wsd.Cells(r, c1).Value = X(i)
r = r + 1

```

```

Next i

r = glistnmRow + 1
X = Split(wsd.Cells(r, c2), ",")
For i = 0 To UBound(X)
wsd.Cells(r, c2).Value = Val(X(i))
r = r + 1
Next i

'set focus back to sd comboBox and clear textbox
TextBox9.Text = ""
TextBox30.Text = ""
TextBox29.Text = ""
li = ComboBox8.ListIndex + 1
rc = ComboBox8.ListCount
If li > rc - 1 Then
TextBox13.SetFocus
MsgBox "You have finished creating the specific designations for the list " & TextBox8.Text
& " ."
ComboBox8.Value = ""
TextBox9.Text = ""
TextBox30.Text = ""
TextBox29.Text = ""
TextBox8.Text = ""
TextBox7.Text = ""
TextBox27.Text = ""
TextBox28.Text = ""
Else:
ComboBox8.Value = ComboBox8.list(li)
TextBox9.SetFocus
End If

End Sub
Private Sub TextBox13_Change()
UniqueListnm
End Sub
Private Sub CommandButton12_Click()
Dim listnm, listcnt, newnm, newrng, X() As String
Dim i, r, lastrow As Integer
listnm = TextBox13.Text
listcnt = Replace(TextBox12.Text, Chr(10), ",")
newnm = Replace(listnm, " ", "_")

If TextBox12.Text = "" Then
MsgBox "There need to be list contents for this list to be added."

```

```

TextBox12.SetFocus
Cancel = True

ElseIf TextBox13.Text <> "" Then

For Each nm In ThisWorkbook.Names
If nm.Name = Replace(listnm, " ", "_") Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken." & Chr(10) & "Choose
another for this list."
ac.SelStart = 0
ac.SelLength = Len(ac.Text)
Exit Sub
End If

Next nm

Else:
'add listnm to end of column A
Set wsm = Worksheets("Material")
lastrow = wsm.Cells.Find("*", SearchOrder:=xlByRows, SearchDirection:=xlPrevious).Row +
2
wsm.Range("A" & lastrow).Value = listnm
r = lastrow

'add listcnt to first cell of range
wsm.Range("B" & lastrow).Value = listcnt

'pass elements of list contents to range
X = Split(wsm.Range("B" & r), ",")
For i = 0 To UBound(X)
wsm.Range("B" & r).Value = X(i)
r = r + 1
Next i

'add new defined name
newrng = "=" & Range("B" & lastrow & ":B" & r - 1).Address
ActiveWorkbook.Names.Add Name:=newnm, RefersTo:=newrng
TextBox16.SetFocus
End If
End Sub
Private Sub TextBox16_Change()
UniqueListnm
End Sub
Private Sub CommandButton13_Click()
Dim listnm, listcnt, newnm, newrng, X() As String

```

```

Dim i, lastrow, r As Integer
listnm = TextBox16.Text
listcnt = Replace(TextBox15.Text, Chr(10), ",")
newnm = Replace(listnm, " ", "_")

If TextBox15.Text = "" Then
MsgBox "There need to be list contents for this list to be added."
TextBox12.SetFocus
Cancel = True

ElseIf TextBox16.Text <> "" Then

For Each nm In ThisWorkbook.Names
If nm.Name = Replace(listnm, " ", "_") Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken." & Chr(10) & "Choose
another for this list."
ac.SelStart = 0
ac.SelLength = Len(ac.Text)
Exit Sub
End If

Next nm

Else:
'add listnm to end of column A
Set wsdp = Worksheets("Disposal")
lastrow = wsdp.Cells.Find("*", SearchOrder:=xlByRows, SearchDirection:=xlPrevious).Row
+ 2
wsdp.Range("A" & lastrow).Value = listnm
r = lastrow
'add listcnt to first cell of range
wsdp.Range("B" & r).Value = listcnt

'pass elements of list contents to range
X = Split(wsdp.Range("B" & r), ",")
For i = 0 To UBound(X)
wsdp.Range("B" & r).Value = X(i)
r = r + 1
Next i

'add new defined name
newrng = "=" & Range("B" & lastrow & ":B" & r - 1).Address
ActiveWorkbook.Names.Add Name:=newnm, RefersTo:=newrng
TextBox20.SetFocus
End If

```

```

End Sub
Private Sub TextBox20_Change()
UniqueListnm
End Sub
Private Sub CommandButton14_Click()
Dim listnm, listcnt, newnm, newrng, X() As String
Dim i, r, glistnmRow As Integer
glistnm = TextBox16.Text
listnm = TextBox20.Text
listcnt = Replace(TextBox17.Text, Chr(10), ",")
newnm = Replace(listnm, " ", "_")

If TextBox20.Text = "" Then
MsgBox "There need to be list contents for this list to be added."
TextBox12.SetFocus
Cancel = True

ElseIf TextBox17.Text <> "" Then

For Each nm In ThisWorkbook.Names
If nm.Name = newnm Then
MsgBox "The name " & Replace(nm.Name, "_", " ") & " is taken." & Chr(10) & "Choose
another for this list."
TextBox17.SelStart = 0
TextBox17.SelLength = Len(ac.Text)
Exit Sub
End If

Next nm

Else:
'add listnm to column C same row as new disposal method list name
Set wsdp = Worksheets("Disposal")
glistnmRow = wsdp.Cells.Find(glistnm, SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row
wsdp.Range("C" & glistnmRow).Value = listnm
r = glistnmRow

'add listcnt to first cell of range
wsdp.Range("D" & r).Value = listcnt

'pass elements of list contents to range
X = Split(wsdp.Range("D" & r), ",")
For i = 0 To UBound(X)
wsdp.Range("D" & r).Value = X(i)

```

```

    r = r + 1
Next i

'add new defined name
newrng = "'Disposal!' & Range("D" & glistnmRow & ":D" & r - 1).Address
ActiveWorkbook.Names.Add Name:=newnm, RefersTo:=newrng
End If
End Sub
Private Sub CommandButton7_Click()
'move forward a tab
Dim i As String
i = MultiPage1.Value
i = i + 1
Me.MultiPage1.Value = i
End Sub
Private Sub CommandButton8_Click()
'move backward a tab
Dim i As String
i = MultiPage1.Value
i = i - 1
Me.MultiPage1.Value = i
End Sub

```

### ***D.3.3. Userform 2***

```

Private Sub CommandButton1_Click()
Application.ScreenUpdating = False

Dim flow, disposal, analyse As String

'determine which columns of data to use from userform selection
If OptionButton1.Value = True Then
flow = "D"
ElseIf OptionButton2.Value = True Then
flow = "E"
ElseIf OptionButton3.Value = True Then
flow = "G"
End If

If OptionButton4.Value = True Then
disposal = "I"
ElseIf OptionButton5.Value = True Then
disposal = "J"
End If

```



```

If OptionButton6.Value = True Then
analyse = "L"
ElseIf OptionButton7.Value = True Then
analyse = "N"
End If

Dim wscdd, wscd As Worksheet
Dim i, lastrowr, lastrowd, cstringr, nr As Integer
Dim r, cstring As Range
Dim a, b, c, d, ustring As String

Set wscdd = Worksheets("Collection Disposal Data")
Set wscd = Worksheets("compiled data")
Set fd = Worksheets("Flow Diagram")

lastrowr = wscdd.Cells.Find("*", SearchOrder:=xlByRows, SearchDirection:=xlPrevious).Row

wscd.Range("A2:L2", wscd.Range("A2:L2").End(xlDown)).Clear

For Each Shape In fd.Shapes
    Shape.Delete
Next

'determine range to be compiled from worksheet("Collection Disposal Data")
If analyse = "N" Then
With wscdd.Range("N3:N" & lastrowr)
If Application.WorksheetFunction.CountA(.Cells) = 0 Then
MsgBox "A diagram cannot be created. Choose to analyze by quantity or ensure that weight
information is entered."
Exit Sub
ElseIf Application.WorksheetFunction.CountIf(.Cells, "") > 0 Then
MsgBox "Entries without weight will be excluded from the diagram."
End If
End With
End If

For Each r In wscdd.Range("B3:N" & lastrowr).Rows
    i = r.Row
    If wscdd.Range(analyse & i).Value = "" Then

        GoTo nextdata:
    End If
    If wscdd.Range("B" & i).Value = "" Then
        a = "unidentified source"
    Else: a = wscdd.Range("B" & i).Value

```

```

End If
If wscdd.Range(flow & i).Value = "" Then
b = "unidentified material"
Else: b = wscdd.Range(flow & i).Value
End If
If wscdd.Range(disposal & i).Value = "" Then
c = "unidentified disposal"
Else: c = wscdd.Range(disposal & i).Value
End If
d = wscdd.Range(analyse & i).Value

ustring = a & b & c

'if there is data in "compiled data" find range and search for a match
With wscd.Range("A:A")
Set cstring = .Find(ustring, LookIn:=xlValues, LookAt:=xlWhole)
If Not cstring Is Nothing Then
firstAddress = cstring.Address
cstringr = cstring.Row

'if there is a match increase quantity (d value)
If d = "" Then
wscd.Range("E" & cstringr) = wscd.Range("E" & cstringr) + 0
Else: wscd.Range("E" & cstringr) = wscd.Range("E" & cstringr) + d
End If
'if there is no match add a new entry
Else:
lastrowd = wscd.Range("A:A").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row
If lastrowd = "" Then
nr = 2
Else: nr = lastrowd + 1
End If
wscd.Range("A" & nr).Value = ustring
.Range("B" & nr).Value = a
.Range("C" & nr).Value = b
.Range("D" & nr).Value = c
.Range("E" & nr).Value = d
End If
End With
nextdata: Next

Dim sstring, dstring As Range
Dim sr, dr, lastrowus, lastrowud, sstringr, dstringr As Integer
Dim e, f, g As String

```

```
lastrowd = wscd.Range("A:A").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row
```

```
'sort compiled data
wscd.Sort.SortFields.Clear
wscd.Sort.SortFields.Add Key:=Range("B2:B" & lastrowd), SortOn:=xlSortOnValue,
Order:=xlAscending, DataOption:=xlSortNormal
wscd.Sort.SortFields.Add Key:=Range("C2:C" & lastrowd), SortOn:=xlSortOnValue,
Order:=xlAscending, DataOption:=xlSortNormal
wscd.Sort.SortFields.Add Key:=Range("D2:D" & lastrowd), SortOn:=xlSortOnValue,
Order:=xlAscending, DataOption:=xlSortNormal
```

```
With wscd.Sort
    .SetRange Range("A1:E" & lastrowd)
    .Header = xlYes
    .MatchCase = xlFalse
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With
```

```
'create a unique counts table on same page for source
For Each r In wscd.Range("A2:E" & lastrowd).Rows
    i = r.Row
    e = wscd.Range("B" & i).Value
    f = wscd.Range("D" & i).Value
    g = wscd.Range("E" & i).Value
```

```
With wscd.Range("G:G")
Set sstring = .Find(e, LookIn:=xlValues, LookAt:=xlWhole)
    If Not sstring Is Nothing Then
        firstAddress = sstring.Address
        sstringr = sstring.Row
```

```
'if there is a match increase quantity
wscd.Range("H" & sstringr) = wscd.Range("H" & sstringr) + g
```

```
'if no match add a new entry
Else:
    lastrowus = wscd.Range("G:G").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row
    If lastrowus = "" Then
        sr = 2
    Else: sr = lastrowus + 1
    End If
```

```
wscd.Range("G" & sr).Value = e
wscd.Range("H" & sr).Value = g
End If
```

```
With wscd.Range("I:I")
Set dstring = .Find(f, LookIn:=xlValues, LookAt:=xlWhole)
If Not dstring Is Nothing Then
firstAddress = dstring.Address
dstringr = dstring.Row
```

```
'if there is a match increase quantity
wscd.Range("J" & dstringr) = wscd.Range("J" & dstringr) + g
```

```
'if no match add a new entry
Else:
```

```
lastrowud = wscd.Range("I:I").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row
```

```
If lastrowud = "" Then
```

```
dr = 2
```

```
Else:
```

```
dr = lastrowud + 1
```

```
End If
```

```
wscd.Range("I" & dr).Value = f
```

```
wscd.Range("J" & dr).Value = g
```

```
End If
```

```
End With
```

```
End With
```

```
Next
```

```
'create chart with shapes!
```

```
wscd.Range("L2") = Application.WorksheetFunction.Sum(wscd.Range("E2:E" & lastrowd))
```

```
lastrowus = wscd.Range("G:G").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row
```

```
lastrowud = wscd.Range("I:I").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row
```

```
lastrowm = wscd.Range("C:C").Find("*", SearchOrder:=xlByRows,
SearchDirection:=xlPrevious).Row
```

```
ys = 100
```

```
yd = 100
```

```
sps = 50 / (lastrowus - 1)
```

```
spd = 50 / (lastrowud - 1)
```

```
'add chart
```

```
Dim MyChart As Chart
```

```
fd.Shapes.AddChart(, 10, 10, 800, 645).Name = "Collection Flow Diagram"
fd.ChartObjects("Collection Flow Diagram").Activate
```

```
Set MyChart = ActiveChart
```

```
MyChart.Shapes.AddLabel(msoTextOrientationHorizontal, 125, 50, 100, 50).Name = "source"
  With MyChart.Shapes("source")
    .TextFrame.AutoSize = True
    .TextFrame.Characters.Text = "Sources of Collection Disposal"
    .TextFrame.Characters.Font.Size = 8
    .IncrementLeft -MyChart.Shapes("source").Width / 2
  End With
```

```
MyChart.Shapes.AddLabel(msoTextOrientationHorizontal, 650, 50, 100, 50).Name =
"destination"
  With MyChart.Shapes("destination")
    .TextFrame.AutoSize = True
    .TextFrame.Characters.Text = "Destinations of Collection Disposal"
    .TextFrame.Characters.Font.Size = 8
    .IncrementLeft -MyChart.Shapes("destination").Width / 2 + 50
  End With
```

```
MyChart.Shapes.AddLabel(msoTextOrientationHorizontal, 400, 10, 100, 50).Name = "title"
  With MyChart.Shapes("title")
    .TextFrame.AutoSize = True
    .TextFrame.Characters.Text = "Collection Flow Diagram"
    .TextFrame.Characters.Font.Size = 18
    .IncrementLeft -MyChart.Shapes("title").Width / 2
  End With
```

```
For Each r In wscd.Range("G2:G" & lastrowus).Rows
```

```
  i = r.Row
```

```
  t = wscd.Range("L2")
```

```
  snm = wscd.Range("G" & i)
```

```
  sl = wscd.Range("H" & i)
```

```
  spsizes = sl / t * 400
```

```
  Dim ps As Integer
```

```
  ps = sl / t * 100
```

```
  'add shapes to represent source categories
```

```
  MyChart.Shapes.AddShape(msoShapeRectangle, 100, ys, 50, spsizes).Name = snm
```

```
  With MyChart.Shapes(snm)
```

```
    .Fill.ForeColor.SchemeColor = 44
```

```
  '.Fill.Transparency = 0.5
```

```

.Line.Visible = msoFalse
MyChart.Shapes.AddLine(150, ys, 220, ys).Name = "up" & snm
With MyChart.Shapes("up" & snm)
.Line.ForeColor.SchemeColor = 44
.Line.Transparency = 0.75
End With
MyChart.Shapes.AddLine(150, ys + MyChart.Shapes(snm).Height - 0.5, 220, ys +
MyChart.Shapes(snm).Height - 0.5).Name = "down" & snm
With MyChart.Shapes("down" & snm)
.Line.ForeColor.SchemeColor = 44
.Line.Transparency = 0.75
End With
'add label to describe shape
MyChart.Shapes.AddLabel(msoTextOrientationHorizontal, 100, ys +
MyChart.Shapes(snm).Height, 100, 50).Name = "t" & snm
With MyChart.Shapes("t" & snm)
.Fill.Visible = msoFalse
.Line.Visible = msoFalse
.TextFrame.Characters.Text = snm & " (" & ps & ")"
.TextFrame.AutoSize = True
.TextFrame.Characters.Font.ColorIndex = 1
.TextFrame.MarginRight = 1
.TextFrame.MarginLeft = 1
.TextFrame.MarginTop = 1
.TextFrame.MarginBottom = 1
.TextFrame.Characters.Font.Size = 8
.IncrementLeft -MyChart.Shapes("t" & snm).Width - 5
.IncrementTop -MyChart.Shapes("t" & snm).Height + 2
End With
End With
ys = ys + spsizes + sps
Next

For Each r In wscd.Range("I2:I" & lastrowud).Rows
i = r.Row
dnm = wscd.Range("I" & i)
dl = wscd.Range("J" & i)
spsized = dl / t * 400
Dim pd As Integer
pd = dl / t * 100

'add shapes to represent destination categories
MyChart.Shapes.AddShape(msoShapeRectangle, 650, yd, 50, spsized).Name = dnm
With MyChart.Shapes(dnm)
.Fill.ForeColor.SchemeColor = 50

```

```

.Fill.Transparency = 0.5
.Line.Visible = msoFalse
  MyChart.Shapes.AddLine(580, yd, 650, yd).Name = "up" & dnm
  With MyChart.Shapes("up" & dnm)
  .Line.ForeColor.SchemeColor = 50
  .Line.Transparency = 0.75
  End With
  MyChart.Shapes.AddLine(580, yd + MyChart.Shapes(dnm).Height - 0.5, 650, yd +
MyChart.Shapes(dnm).Height - 0.5).Name = "down" & dnm
  With MyChart.Shapes("down" & dnm)
  .Line.ForeColor.SchemeColor = 50
  .Line.Transparency = 0.75
  End With
'add label to describe shape
  MyChart.Shapes.AddLabel(msoTextOrientationHorizontal, 650, yd +
MyChart.Shapes(dnm).Height, 100, 50).Name = "t" & dnm
  With MyChart.Shapes("t" & dnm)
  .Fill.Visible = msoFalse
  .Line.Visible = msoFalse
  .TextFrame.Characters.Text = dnm & " (" & pd & ")"
  .TextFrame.AutoSize = True
  .TextFrame.Characters.Font.ColorIndex = 1
  .TextFrame.MarginRight = 1
  .TextFrame.MarginLeft = 1
  .TextFrame.MarginTop = 1
  .TextFrame.MarginBottom = 1
  .TextFrame.Characters.Font.Size = 8
  .IncrementLeft 55
  .IncrementTop -MyChart.Shapes("t" & snm).Height + 2
  End With
End With
yd = yd + spsized + spd
Next r

```

```

'add shapes to represent flow
For Each r In wscd.Range("C2:C" & lastrowm).Rows
  Dim st, dt As Long
  Dim s, m, dst, q As String
  Dim astr, adtr As Integer
  i = r.Row
  s = wscd.Range("B" & i)
  m = wscd.Range("C" & i)
  dst = wscd.Range("D" & i)
  q = wscd.Range("E" & i) / t * 400
  Dim pf As Integer

```

```

pf = wscd.Range("E" & i) / t * 100

'determine where to put shape in relation to source
With wscd.Range("B:B")
  Set ast = .Find(s, After:=Range("B" & i), SearchDirection:=xlPrevious, LookAt:=xlWhole)
  If Not ast Is Nothing Then
    firstAddress = ast.Address
    astr = ast.Row
    Dim ar As Integer
    ar = wscd.Range("E" & astr) / t * 100
    If astr < i Then
      sa = wscd.Range("B" & astr) & ">" & wscd.Range("C" & astr) & ">" &
wscd.Range("D" & astr) & "(" & ar & ")"
      st = MyChart.Shapes(sa).Vertices(4, 2)
      Else:
      st = MyChart.Shapes(s).Top
      End If
    Else: st = MyChart.Shapes(s).Top
    End If
  End With
'determine where to put shape in relation to destination
With wscd.Range("D:D")
  Set adt = .Find(dst, After:=Range("D" & i), SearchDirection:=xlPrevious,
LookAt:=xlWhole)
  If Not adt Is Nothing Then
    firstAddress = adt.Address
    adtr = adt.Row
    Dim ad As Integer
    ad = wscd.Range("E" & adtr) / t * 100
    If adtr < i Then
      da = wscd.Range("B" & adtr) & ">" & wscd.Range("C" & adtr) & ">" &
wscd.Range("D" & adtr) & "(" & ad & ")"
      dt = MyChart.Shapes(da).Vertices(3, 2)
      Else: dt = MyChart.Shapes(dst).Top
      End If
    Else: dt = MyChart.Shapes(dst).Top
    End If
  End With

'determine points of new shape
Dim mshape(1 To 5, 1 To 2) As Single
mshape(1, 1) = 220
mshape(1, 2) = st
mshape(2, 1) = 580
mshape(2, 2) = dt

```



```

mshape(3, 1) = 580
mshape(3, 2) = dt + q
mshape(4, 1) = 220
mshape(4, 2) = st + q
mshape(5, 1) = 220
mshape(5, 2) = st

```

```

MyChart.Shapes.AddPolyline(mshape).Name = s & ">" & m & ">" & dst & "(" & pf & ")"
With MyChart.Shapes(s & ">" & m & ">" & dst & "(" & pf & ")")
    .Fill.ForeColor.SchemeColor = 51
    .Fill.Transparency = 0.5
    .Line.Visible = msoFalse

```

'add labels to either end

```

MyChart.Shapes.AddLabel(msoTextOrientationHorizontal, 220, MyChart.Shapes(s & ">" &
m & ">" & dst & "(" & pf & ")").Vertices(4, 2), 100, 50).Name = "f" & m & i
With MyChart.Shapes("f" & m & i)
    .Fill.Visible = msoFalse
    .Line.Visible = msoFalse
    .TextFrame.Characters.Text = m & "(" & pf & ")"
    .TextFrame.AutoSize = True
    .TextFrame.Characters.Font.ColorIndex = 1
    .TextFrame.MarginRight = 1
    .TextFrame.MarginLeft = 1
    .TextFrame.MarginTop = 1
    .TextFrame.MarginBottom = 1
    .TextFrame.Characters.Font.Size = 6
    .IncrementLeft -MyChart.Shapes("f" & m & i).Width
    .IncrementTop -MyChart.Shapes("f" & m & i).Height + 2
End With

```

```

MyChart.Shapes.AddLabel(msoTextOrientationHorizontal, 580, MyChart.Shapes(s & ">" &
m & ">" & dst & "(" & pf & ")").Vertices(3, 2), 100, 50).Name = "b" & m & i
With MyChart.Shapes("b" & m & i)
    .Fill.Visible = msoFalse
    .Line.Visible = msoFalse
    .TextFrame.Characters.Text = m & "(" & pf & ")"
    .TextFrame.AutoSize = True
    .TextFrame.Characters.Font.ColorIndex = 1
    .TextFrame.MarginRight = 1
    .TextFrame.MarginLeft = 1
    .TextFrame.MarginTop = 1
    .TextFrame.MarginBottom = 1
    .TextFrame.Characters.Font.Size = 6
    .IncrementLeft

```

```

        .IncrementTop -MyChart.Shapes("b" & m & i).Height + 2
    End With

End With

Next r

With MyChart.Shapes
    numShapes = .Count
    numAutoShapes = 1
    ReDim autoShpArray(1 To numShapes)
    For i = 1 To numShapes
        autoShpArray(numAutoShapes) = .Item(i).Name
        numAutoShapes = numAutoShapes + 1
    Next
    ReDim Preserve autoShpArray(1 To numAutoShapes)
    Set asRange = .Range(autoShpArray)

asRange.Group
asRange.Name = "group"
asRange.Top = (fd.ChartObjects("Collection Flow Diagram").Height - asRange.Height) / 2
asRange.Left = (fd.ChartObjects("Collection Flow Diagram").Width - asRange.Width) / 2
End With

fd.ChartObjects("Collection Flow Diagram").Select
ActiveWindow.Zoom = True
fd.Range("B3").Select
Application.ScreenUpdating = True
Unload Me
fd.Activate

End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub

```