Crime Mapping – Technical Summary

The two main aspects of and type of incident mapping are creating the point and manipulating the legend. In this case, the point is created from a stored x,y coordinate using the XY Event Theme object. An event theme object is a theme that is temporary. It is deleted when ArcView exits. If you want to save the xytheme as an ftheme, you must export it. Here is the code. The comments in the code explain what is happening.

```
'If there is a selection map the crimes
if (vtCrimeData.GetSelection.Count > 0) then
 'get the bitmap from the crime virtual table
  btmCrimeData = vtCrimeData.GetSelection
  'set the the definition bitmap. This insures that only those records selected will be exported
  vtCrimeData.SetDefBitmap(btmCrimeData)
  vtCrimeData.UpdateDefBitMap
 'create the XY source name
  srcXYName = XYName.Make( vtCrimeData, vtCrimeData.FindField("x_coord"),
                              vtCrimeData.FindField("y_coord") )
  'create the x,y event theme
  thmXY  = Theme.Make( srcXYName )
 'create the file name for the Ftheme, the user selects the name
  fnCrOut = FileDialog.Put("c:\temp\crime1.shp".AsFileName, "*.shp", "Create Output File")
  if the user hits cancel the end the script
  if (fnCrOut = nil) then return nil end
  'export the xytheme to a ftheme virtual table
  ftbXYNew = thmXY.GetFTab.Export ( fnCrOut, Shape, TRUE )
  'create the new ftheme from the exported virtual table
  thmCrimes  = FTheme.Make(ftbXYNew)
  'add the theme to the view
  vwCrime.AddTheme(thmCrimes)
  'ask the user to name the theme
  strTitle = MsgBox.Input("Enter New Theme Name", "New Theme", "Crime1")
 'set the name of the theme
  thmCrimes.SetName(strTitle)
end
```

After a new theme is added to the View I use code to render the legend. Now this is a bit different that normal. I have a saved legend file that has all of the crime types in it. I just want to use that saved file to pick out the crime types that have been mapped. So I first set the legend using the saved file, then I go back and manipulate the Symbol List and Classification List objects of the Legend object. The code is on the following pages. The comments in the code explain what is happening.

```
‘get the legend of the theme and set it from a saved legend file.
 legCrimes = thmCrimes.GetLegend
 legCrimes.SetLegendType(#LEGEND_TYPE_UNIQUE)
 legCrimes.Unique(thmCrimes, "filetype")
 fnleg = FileName.Make("\\opgis\fiji\police\incident\crimes.avl")
 legCrimes.Load(fnleg, #LEGEND_LOADTYPE_CLASSESANDSYMBOLS)

 thmCrimes.UpdateLegend

 vwCrime.InvalidateTOC(nil)

‘start altering the legend
 ‘get the legend object, the current classification list and the current symbols list
 legActive = thmCrimes.GetLegend
 lstClassifications = legActive.GetClassifications
 lstSymbols = legActive.GetSymbols

create an empty list and populate it with only the crime types in the current theme.
 lstTypes = {}
 for each rec in thmCrimes.GetFtab

lstTypes.Add(thmCrimes.GetFTab.ReturnValueString(thmCrimes.GetFTab.FindField("FileType"),
rec))
 end
‘remove duplicate crime types and sort the list
 lstTypes.RemoveDuplicates
 lstTypes.Sort(true)

‘make copies (clone) of the current classification and symbols list
 lstCloneClasses = lstClassifications.Clone
 lstCloneSymbols = lstSymbols.Clone

‘each entry in the legend has an index number. Iterate through the list of crime types and get the
legend index number and add it to the legend index list. The result is a list of index numbers
 lstLegIdx = {}

 for each type in lstTypes

  lstLegIdx.Add(legActive.GetIndex({type}))

 end

 lstLegIdx.RemoveDuplicates

‘create the empty lists for the new, good classification list and symbols list.
 lstGoodSymbols = {}
 lstGoodClasses = {}
```

CONTINUED ON NEXT PAGE

```
‘iterate through all of the classifications
 for each cls in lstClassifications
  ‘return the index number of the classification
  numCls = lstClassifications.Find(cls)
  ‘if the classification index number is in the list of legend index numbers then add the
classification and the symbol to the new lists
  if (lstLegIdx.FindByValue(numCls) <> -1) then

   lstGoodClasses.Add(cls)
   lstGoodSymbols.Add(lstSymbols.Get(lstClassifications.Find(cls)))

  end
 end

‘create a dummy class and symbol to initialize the list.
 clsJunk = Classification.Make("J", "Q")
 symJunk = Symbol.Make(#SYMBOL_MARKER)

 lstGoodClasses.Add(clsJunk)
 lstGoodSymbols.Add(symJunk)

‘empty the current classification and symbol lists and merge the new “good” lists.
 lstClassifications.Empty
 lstClassifications = lstClassifications.Merge(lstGoodClasses)

 lstSymbols.Empty
 lstSymbols = lstSymbols.Merge(lstGoodSymbols)

‘update the legend and redraw the themes
 thmCrimes.UpdateLegend

 thmCrimes.SetVisible(true)
 thmCrimes.SetActive(true)
```

If you have any questions regarding this code please feel free to contact me at:

Robert Meier, GIS Analyst
City of Overland Park, Kanas
8500 Santa Fe Dr.
Overland Park, KS 66212
913/895-6203
rmeier@opkansas.org