Extending Python

James Tedrick, GIS Specialist, M-NCPPC
Two Examples
Mailing List Generator
Mailing List Generator
Mailing List Generator
<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Address 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONGRESSWOMAN DONNA EDWARDS</td>
<td>U.S. HOUSE OF REPRESENTATIVES</td>
<td>434 CANNON HOUSE OFFICE BUILDING</td>
</tr>
<tr>
<td>EDGAR NEAL</td>
<td>EAST GERMANTOWN CITIZENS ASSN.</td>
<td>11809 C…</td>
</tr>
<tr>
<td>GARY VALEN</td>
<td>SUGARLOAF CITIZENS ASSN. INC.</td>
<td>18325 D…</td>
</tr>
<tr>
<td>MERRITT W. EDNIE</td>
<td>BOYDS CIVIC ASSN.</td>
<td>P.O. BOX</td>
</tr>
<tr>
<td>None CONTACT</td>
<td>CLARKSBURG CIVIC ASSN.</td>
<td>P.O. BOX</td>
</tr>
<tr>
<td>None None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>PAT OLSON</td>
<td>GERMANTOWN CITIZENS ASSN.</td>
<td>18413 K…</td>
</tr>
<tr>
<td>PATRICK DARBY</td>
<td>CLARKSBURG CHAMBER OF COMMERCE</td>
<td>6125 T…</td>
</tr>
<tr>
<td>TOM AND MELAN HOFFMAN</td>
<td>CLARKSBURG INITIATIVES ASSN.</td>
<td>23801 P…</td>
</tr>
</tbody>
</table>
Mailing List Generator (PDF)

CONGRESSWOMAN DONNA EDWARDS
U.S. HOUSE OF REPRESENTATIVES
434 CANEY HILL OFFICE BUILDING
WASHINGTON, D.C. 20515-2004

EDGAR WEALE
EAST GERMANTOWN CITIZENS ASSN.
11904 COLLINS DRIVE
GERMANTOWN, MD 20876

GARY VALE
SUGARLOAF CITIZENS ASSN. INC.
18325 DAMESTOWN ROAD
POOKESVILLE, MD 20837

MERRITT W. EDWIN
BOYDS CIVIC ASSN.
P.O. BOX 185
BOYDS, MD 20841
E-mail a Map
E-mail a Map
E-mail a Map
What happened?

- Both programs use Python scripts
- Python is a popular open source programming language
- Main geoprocessing script programming language
- Popular language -> many libraries
- Open Source -> many free libraries
Analysis in ArcGIS

- ArcToolBox
- ModelBuilder
- Python Scripts
  - By themselves
  - In a Script Tool
  - In a Model
Mailing List (Excel/PDF)

- **pyExcelerator -> xlwt**
  - Libraries are sometimes iffy, come & go
  - Minimal instructions (supplied by 3rd party)

- **ReportLab (PDF)**
  - Supported by a company
  - Extensive manual
  - Requires “Run in separate process” option
def SaveToPDFLabels(values, outfile):
    from reportlab.pdfgen import canvas
    out_pdf = canvas.Canvas(outfile, pagesize = letter)
    out_pdf.setFont("Courier", 10)
    for A_value in values:
        #Page Change; skipped code for column changes (every 10)
        if count > 0 and count % 30 == 0:
            out_pdf.showPage()
            out_pdf.setFont("Courier", 10)
            x_pos = 0.15 * inch
            y_pos = 10.5 * inch
            #We skipped formatting the label text
            label = out_pdf.beginText()
            label.setTextOrigin(x_pos, y_pos)
            label.textLine(label_1stline)
            label.textLine(lastLine)
            out_pdf.drawText(label)
            y_pos = y_pos - 1.02 * inch
            count = count + 1
        out_pdf.showPage()
    out_pdf.save()
def SaveToExcel(headers, values, outfile):
    import xlwt as xl
    # Open new workbook
    outdoc = xl.Workbook()
    # Add a worksheet
    outsheet = outdoc.add_sheet("Addresses")
    # Create a Bold font for headers
    header_font = xl.Font(); header_font.bold = True
    # Convert font to style
    header_style = xl.XFStyle(); header_style.font = header_font
    # Write headers
    for col, value in enumerate(headers):
        outsheet.write(0, col, value, header_style)
    # Write values
    for row_num, row_values in enumerate(values):
        row_num += 1  # Start at row 1
        for col, value in enumerate(row_values):
            outsheet.write(row_num, col, value)
    outdoc.save(outfile)
E-mail a map

- 3 libraries
  - Send the e-mail: smtplib
  - Get the maps (from REST): urllib
  - Work with the images: Perl Imaging Library (PIL)

- REST: REpresentational State Transfer
  - Everything is accessed via URL
  - Released with ArcGIS Server 9.3
  - Makes access to maps, data, tasks trivial
from urllib import urlretrieve
def getImage(Xcoord, Ycoord, ACCT, outimagepath):
    #Cut out: setting up size, scale, bounding box, image format
    #HTTP path to base map
    base_map = "http://localhost/ArcGIS/rest/services/BaseMap/MapServer/export?"
    #HTTP path to overlay map
    overlay_map = "http://localhost/ArcGIS/rest/services/querymap/MapServer/export?"
    #additional setup for overlay- set transparent background, find parcel by account
    overlay = "transparent=true&layers=show:0&layerDefs=0:ACCT='' + ACCT + ''"
    #Full URL: base path + bounding box + size + image format
    base_map_url = basemapservice + boundingbox + size + '&' + imageformat
    #Get the image & any messages (ex. 404 Not Found)
    (baseimagefilepath, message) = urlretrieve(basemap_url)
    #Do the same for Overlay with additional setup
    overlay_map_url = overlay_map + boundingbox + size + '&' + overlay + '&' + imageformat
    (overlayimagefilepath, overlaymessage) = urlretrieve(propmap_url)
from PIL import Image

#
base_image = Image.open(baseimagefilepath)
overlay_image = Image.open(overlayimagefilepath)
base_image.paste(overlay_image, (0,0), overlay_image)
base_image.save(outimagepath)
def sendMail(to, subject, text, files=[], server="smtp.gmail.com"):  
    from = "M-NCPPC GIS <MNCPPC.GIS@gmail.com>"
    msg = MIMEMultipart()
    #Message setup: From, To, Date, Subject
    msg['From'] = from; msg['To'] = COMMASPACE.join(to);
    msg['Date'] = formatdate(localtime=True); msg['Subject'] = subject
    #Attach the 1st part: the text
    msg.attach( MIMEText(text,'html') )
    #Assume that there is 1 more attachment and it’s an image
    fp = open(files[0],"rb")
    part = MIMEImage(fp.read())
    fp.close()
    msg.attach(part)

smtp = smtplib.SMTP(server, 587)
#Log-in to gmail
smtp.login("MNCPPC.GIS@gmail.com",PASSWORD)
smtp.sendmail(fro, to, msg.as_string() )
smtp.close()
You don’t need to use GIS!

- E-mailing a map script uses no GIS procedures!
- ArcGIS Server is a platform to run the program (equivalent to ASP), easily hooks into web pages, other programs
- Utilizes ArcGIS Server’s job handling for temporary file management
Use cases

- Analysis follow-up/reports
- Automatic map/slide generation
- Automated Notification (twitter, e-mail for download)
- Application submittal: feature creation/verification
- ArcGIS Desktop HTML pop-ups: Always-ready analysis/map making