Building Live Data Feeds using Python
Paul Dodd & Derrick Burke
What are they?
Use Cases – what’s happening now?

http://www.esri.com/services/disaster-response
Use Cases – what’s happening now?
Use Cases – who’s affected? what if?
BREAKING NEWS
SUPERMARKETS, HOMES TOPPLE
Interactive maps show most powerful videos
Latest Examples – NDFD Precipitation…

http://tmservices1/arcgis/rest/services/LiveFeeds/NDFD_Precipitation/MapServer
Data Automation options…

• Aggregated Live Feeds Methodology
  - Bulk processing
  - Periodic updates
  - Heavy Geoprocessing
  - Raster manipulation

• GeoEvent Processor
  - Rapid turnaround
  - Live Streaming in/out
  - Alerting via GeoFencing
  - Sensor/Device ID tracking
Feed requirements

- Accessible

- Timely

- Robust architecture (desktop → cloud, scalable)

- Extensible (cross platform, rapid deployment)

- Automated
Feed construction

• Pre-requisites:
  - Python
  - Arcpy
  - Others (.py, 7zip, degrib, ...)

• Components
  - ALFlib.py
    - Function / Class Library
  - ALFprocessor.py
    - Feed Processor +
    - ‘<feed file>.cfg’ (data processing logic)
  - ALFdeployer.py
    - Archive Distribution
Architecture of a feed

Internet

Data

Published Map Service

ArcGIS Server

GIS User

Log Files

Feed Routine

Work FileGDB

Deployment Logic

Live FileGDB

Data

Published Map Service

GIS User

Internet

Log Files

Feed Routine

Work FileGDB

Deployment Logic

Live FileGDB

Data

Published Map Service

GIS User

Internet

Log Files

Feed Routine

Work FileGDB

Deployment Logic

Live FileGDB

Data

Published Map Service

GIS User

Internet

Log Files

Feed Routine

Work FileGDB

Deployment Logic

Live FileGDB

Data

Published Map Service

GIS User

Internet

Log Files

Feed Routine

Work FileGDB

Deployment Logic

Live FileGDB

Data

Published Map Service

GIS User
Let’s dive in

```python
for h3 in page.find_all("h3"):
    value = (h3.contents[0])
    if value != "Afdeling":
        print >> txt, value
import codecs
f = codecs.open("alle.txt", "r", encoding="utf-8")
text = f.read()
f.close()
# open the file again for writing
f = codecs.open("alle.txt", "w", encoding="utf-8")
f.write(value+"\n")
# write the original contents
f.write(text)
f.close()
```
For additional Help detail:  
Use: 'ALFlib.py -h [name]' 

To execute a Function call:  
Use: 'ALFlib.py <func_name> [params][ ...]'  
Ex: ALFlib.py getDownload "ftp://my.server.domain/test.dat'  
   userName="me" password="myself"  
   (don't forget to double-escape strings with quotes! Like "'MyString'"

To return a Property value: 
Use: 'ALFlib.py <prop_name>'

Available Classes:
AppLock(<lockFile>)
Constants()
CountryCodeLoader([indexOn])
DataError(<exception_text>)
LogLevel(<nickName>, <hostFolder>[, <retention>[, <singleton>]])
OrderedDict()
Progress([interval])
S3cmd([configFile][, <verbose>])
TimeoutError(<exception_text>)
WeatherStationLoader([indexOn])

Available Functions:
c2f(<tempCelsius>)
callCommandLine([callParams][, <description>[, <timeout>[, <timerHook>[/[<verbose>]]]]])
check?zip([retry][, <verboseHandle>])
copyFileQDB([source], [destination])[* Deprecated]*
copyFile([source][, [destination][, <exclusions>][, <overwrite>]], <recursive>[, <showProgress>][, <verbose>])
dms2dd([degrees][, <minutes>][, <seconds>][, <bearing>])
f2c(<tempFahrenheit>)
getDownload([url][, workPath][, <timeStamping>[, <timeout>][, <readSizeInKB>] [', <authentication>[, <userName>[, <password>[, <verbose>]]]]])
getVerboseHandles([<verbose>])
heatIndex([tempCelsius][, percentHumidity])
minorVersion([major][, <minor>][, <bug>])
recursePath([path][, <mask>][, <exclusions>][, <matches>][, <recursive>[, <verboseHandler>]], [tempCelsius][, <verboseHandler>])
relativeHumidity([tempCelsius], [dewPointCelsius])
setFileTime([filename], [<timestamp>])
unzipFile([<zipFile>[, <destination>][, <verbose>][, <password>]])
version()
windChill([tempCelsius][, <windSpeedKmph>])
xmNormailzer([node])
Building Live Data Feeds using Python

```
G:\> ALFprocessor.py -h
ALFprocessor.py, v0.8.0, Paul Dodd - esri

Aggregated Live Feed processor tool.
Usage: ALFprocessor.py -h [-c] <configFile>

- h = Show this usage.
- c = Create template <configFile> if it doesn’t already exist.
<configFile> = File path and/or name of ALF configuration
file to process. Will only read file with
’cfg’ extension, specification of which is
not required here.

G:\> ALFdeployer.py -h
ALFdeployer.py, v1.0.1, Paul Dodd - esri

File Geodatabase archive download and deployment tool.
Usage: ALFdeployer.py [-h] <configFile>

- h = Show this usage.
<configFile> = Configuration file path and/or name + ’cfg’ extension.
New file is generated if not found or specified.

G:\>
```
# reflect the minimum version level you require.
# Returns True if the version is at least the minimum specified, False otherwise.

def main( self):
    # Start of Main logic
    if len( sys.argv) < 3:
        print( " * No Input Shapefile specified *
"
else:
    inputFile = os.path.splitext( sys.argv[2])[0] + ".shp"
    print( "\nProcessing: '0').format( inputFile)"

outputFileGDB = "HurricaneSandy.gdb"
outputFeatureclass = "CurrentPrecip"

# Initialize Work FileGDB
workspace = self.prepWorkspace( workPath, outputFileGDB, recreate=True)

if workspace:
    outputFeatureclass = workspace + os.sep + outputFeatureclass

    print( "\nImporting data...")
    self.logResults( arcpy.CopyFeatures_management( inputFile, outputFeatureclass))

    print( "\nCreating Spatial Index...")
    self.logResults( arcpy.AddSpatialIndex_management( outputFeatureclass, 5, 0, 0))

    print( "\nDeploying...")
    ALFlib.copyFiles( workspace, livePath, exclusions="*.lock")

    ALFlog.archive( " Successfully imported: "0').format( inputFile)"

    print( "\nDone!")
Building Live Data Feeds using Python

Copying From:
'C:\EPC_Sandy\Work\HurricaneSandy.gdb'
to:'C:\EPC_Sandy\Live'...

Files Copied: 0
Files Overwritten: 44
Files Skipped: 0

Done!

Sandy - ArcMap

SandyPrecip Routine Started: 11:46:54 Tue 03/19/2013

Importing ArcPy...
Processing: 'Sandy_Oct30_01.shp'

Creating Spatial Index...
Importing data...
Creating Spatial Index...
Deploying...
Gathering file list...

Copying from:
'C:\EPC_Sandy\Work\HurricaneSandy.gdb'
to:
'C:\EPC_Sandy\Live'...

Copy completed!
Files Copied: 0
Files Overwritten: 44
Files Skipped: 0
Done!

C:\EPC_Sandy>ALFprocessor.py SandyPrecip Sandy_Oct30_02
SandyPrecip Routine Started: 11:48:02 Tue 03/19/2013

Importing ArcPy...
Processing: 'Sandy_Oct30_02.shp'
Preparing Workspace...
Removing File Geodatabase...
Creating File Geodatabase...
Importing data...
Creating Spatial Index...
Deploying...
Gathering file list...

Copying from:
'C:\EPC_Sandy\Work\HurricaneSandy.gdb'
to:
'C:\EPC_Sandy\Live'...

Copy completed!
Files Copied: 0
Files Overwritten: 44
Files Skipped: 0
Done!

C:\EPC_Sandy>
SandyPrecip Routine Started: 12:07:08 Tue 03/19/2013

Importing ArcPy...
  arcpy: v10.1.1
  configuration: SandyPrecip.cfg, v1.0.0
  processor: ALFprocessor.py, v0.8.0
  library: ALFlib.py, v1.6.0

Processing: 'sandy_oct30_00.shp'

Preparing workspace...
  <directory>: 'C:\EPC_Sandy\Work'
  gdbname: 'Hurricanesandy.gdb'
  <create>: True

Removing file geodatabase...
  Created File Geodatabase...
  Returned: 'C:\EPC_Sandy\Work\Hurricanesandy.gdb'

Importing data...

Executing: CopyFeatures Sandy_oct30_00.shp C:\EPC_Sandy\Work\Hurricanesandy.gdb\CurrentPrecip # 0 0 0
  Start Time: Tue Mar 19 12:07:16 2013
  Succeeded at Tue Mar 19 12:07:17 2013 (Elapsed Time: 0.00 seconds)

Creating Spatial Index...

Executing: AddSpatialIndex C:\EPC_Sandy\Work\Hurricanesandy.gdb\CurrentPrecip # 0 0 0
  Start Time: Tue Mar 19 12:07:17 2013
  Succeeded at Tue Mar 19 12:07:17 2013 (Elapsed Time: 0.00 seconds)

Deploying...

Gathering file list...

Copying from:
  'C:\EPC_Sandy\Work\Hurricanesandy.gdb'

to:
  'C:\EPC_Sandy\Live'...

Updating: 'Hurricanesandy.gdb\a00000001.gdbindexes'
  updating: 'Hurricanesandy.gdb\a00000001.gdbtable'
Cloud / Distributed

Deployment (step 1 of 2)

Feed Aggregator
ALFprocessor.py

Internet

Data

Log Files

Feed Routine

Work FileGDB

Deployment Logic

Live FileGDB

Zip File

FileGDB Archive

Common Storage
(External or Internal)

Amazon S3
Cloud / Distributed (cont)

Deployment (step 2 of 2)

Common Storage
(External or Internal)

Amazon S3

Internet

One or More Archives

Download and Deployment Logic

Work area

Live FileGDB

Data Deployment
ALFdeployer.py

Published Map Service

ArcGIS Server

One or More Archives

ArcGIS Server

Published Map Service
Cloud / Distributed
(Deployment Diagram)

Source Data
(NOAA / USGS / …)

Feed Routine

Work FileGDB

Aggregation

One or more Machines dedicated to Aggregation

Zip File

FileGDB Archive

Distribution

Multiple ArcGIS Server Machines

Deployment

Live FileGDB

Consumption

Published Map Service

ArcGIS Server

Internet

Common Storage
(External or Internal)

ALF-Lite – File-Storage / ArcGIS Server Deployment

Amazon S3

Building Live Data Feeds using Python

ALF-Lite – File-Storage / ArcGIS Server Deployment

Source Data
(NOAA / USGS / …)

Feed Routine

Work FileGDB

Aggregation

One or more Machines dedicated to Aggregation

Zip File

FileGDB Archive

Distribution

Multiple ArcGIS Server Machines

Deployment

Live FileGDB

Consumption

Published Map Service

ArcGIS Server

Internet

Common Storage
(External or Internal)

Building Live Data Feeds using Python

One or more Machines dedicated to Aggregation

Multiple ArcGIS Server Machines

Consumption

Published Map Service

ArcGIS Server

Internet

Common Storage
(External or Internal)
Hosted Feature Services

• ALF inclusion
  - Underway

• Feedback – need your ideas / comments
  - dburke@esri.com
  - pdodd@esri.com
Current Development

• Dynamic GeoEnrichment – Analytics
  - GeoEnrich Authoritative data (near real time)

• Predictive Feeds – NDFD…others to come!

• Feeds available on ArcGIS Online (Living Atlas)
  - Live Feeds Group -
    http://www.arcgis.com/home/group.html?owner=esri&title=Live%20Feeds

• Your feeds! -> Aggregated Live Feed Community
  http://esriurl.com/LiveFeed
Go forth and hack

- AGOL Community Page (samples, scripts) http://esriurl.com/LiveFeed
- Live Data Feeds (Prototypes) http://tmservices1.esri.com/arcgis/rest/services/LiveFeeds
- ArcGIS Online: Search for ‘Live Feeds’ group!
Thank you…

• Please fill out the session survey:

First Offering ID: 2129

Online – www.esri.com/ucsessionsurveys

Presenter contact:

pdodd@esri.com, dburke@esri.com or @derrickburke