AUTOMATING SURVEY WORKFLOWS ACROSS THE ENTIRE ARCGIS PLATFORM

KYLE MORGAN
GIS Developer @ Wood Rodgers, Inc
Sacramento, CA
@gisdev_km
SOME BACKGROUND
SURVEY DEPT
Distributed Survey Department

(3 offices in 2 states and growing)
Team consists of Field Crews and Office personnel
A need to streamline distribution of information immediately
TYPICAL PROJECT REQUIREMENTS
Official record of Survey
Photos of marker & surrounding area
Multiple format of deliverables
HISTORICAL APPROACH
A TYPICAL DAY
Field notes are handed in and scanned by office personnel (sometimes days late)
Scanned field notes are printed and shoved in a project binder (that may wander off)
Office personnel try to keep track of attribute changes and revisions to field notes based on word of mouth.
Photos are uploaded *(usually painfully)* to a sharing service (e.g. Box) or emailed from each mobile device to the Office PM.
Office personnel download the photos to the appropriate project folder on the server by date.
Photos are rotated and resized as needed
Any reports are generated by hand using MS Word or Excel
Data is converted to various deliverable formats (CAD or GIS)
When time to deliver, additional time is incurred to double check all work (*usually by hand*)
GIS APPROACH
BEHIND THE CURTAINS

1. Hardware
2. Software
3. Automation
HARDWARE (FIELD)

1. Trimble RTK Units
2. Apple iPhone
SOFTWARE

FIELD

1. Esri Collector for ArcGIS
2. Esri Survey123

OFFICE

1. Esri ArcGIS Desktop
2. Esri ArcGIS Pro
3. Custom Web Interface
<table>
<thead>
<tr>
<th>Job #</th>
<th>Point #</th>
<th>Calc Point</th>
<th>Survey Date</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1081188</td>
<td></td>
<td>1064</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11845</td>
<td>5298</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11844</td>
<td>5299</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11843</td>
<td>5300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11842</td>
<td>5301</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11841</td>
<td>5302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11840</td>
<td>5303</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11839</td>
<td>5304</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11838</td>
<td>11838</td>
<td>5305</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11837</td>
<td>5306</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11836</td>
<td>5307</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11835</td>
<td>5308</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11834</td>
<td>5309</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11833</td>
<td>5310</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11832</td>
<td>5311</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8455.912</td>
<td>11831</td>
<td>5312</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AUTOMATION

SOFTWARE

1. ArcGIS Online
2. ArcGIS for Server
3. ArcGIS For Portal
4. Python
AUTOMATION

DELIVERABLES

1. Monument Sheet (*PDF with photos*)
2. Overview Maps
3. Compiled PNEZD files
4. Esri File Geodatabase / Shapefiles
5. CAD Drawings
FIELD WALKTHROUGH
Field crew finds/ sets monuments traditionally
Using Survey123, records the relevant information on their mobile device
Field crew syncs with ArcGIS Online when it is convenient for them
AUTOMATION WALKTHROUGH
An hourly scheduled tasks polls ArcGIS Online
Fresh data is downloaded and processed
E-mails are sent out to each survey project manager to review with links to each record.
Hi Brian,

The 7 following record(s) have been captured in the last hour for your review. Please review these at your earliest convenience.

- Point Stored 11904 / Point Calc 5028
- Point Stored 11902 / Point Calc 5345
- Point Stored 11903 / Point Calc 5346
- Point Stored 11905 / Point Calc 5355
- Point Stored 11906 / Point Calc 5356
- Point Stored 11907 / Point Calc 5357
- Point Stored 11908 / Point Calc 5395

[Survey Monument Administration Console]

Any questions, please contact [email] or [email] at [email]
# Found Monument Data Sheet

**Details**

**Project Name:** SBFC
**Survey Date:** 5/06/2017
**Job Number:** 8455 912 - SBFC

---

**Monument Characteristics**

**Monument Type:** Iron Pipe
**Material:** Aluminum
**Condition:** Good
**Stamping:** DWR 1121 L9 8067
**Height (+) above (-) below Surface:** -2

---

**Closeup Photo**

![Closeup Photo](image1)

**Location Photo**

![Location Photo](image2)
When a record is reviewed and approved, Monument Sheet PDFs are created in the appropriate folder by date, server-side.

*If a record is updated, the PDF is replaced with a new copy.*
SET MONUMENT DATA SHEET

<table>
<thead>
<tr>
<th>Project Name:</th>
<th>Job Number:</th>
<th>Date:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stored Pt. #:</td>
<td>Calculated Pt. #:</td>
<td>Party Chief: M. Gonzalez</td>
</tr>
</tbody>
</table>

### Monument Characteristics

<table>
<thead>
<tr>
<th>Monument Type:</th>
<th>Survey Tag:</th>
<th>Material:</th>
<th>Placement:</th>
<th>Condition:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iron Pipe</td>
<td>Cap</td>
<td>Plastic</td>
<td>Well</td>
<td>Good</td>
</tr>
<tr>
<td>Rail</td>
<td>Plug</td>
<td>Aluminum</td>
<td>Concrete</td>
<td>Disturbed</td>
</tr>
<tr>
<td>Deck</td>
<td>Tag</td>
<td>Brass</td>
<td>Ground</td>
<td>Bent</td>
</tr>
<tr>
<td>PK Nail</td>
<td>Washer</td>
<td>Steel</td>
<td>Asphalt</td>
<td>Other</td>
</tr>
<tr>
<td>Other:</td>
<td>Other:</td>
<td>Other:</td>
<td>Other:</td>
<td>Other:</td>
</tr>
</tbody>
</table>

### Size

- 3/8"  
- 5/8"  
- 1"  
- 1 1/2"  
- 2 1/4"  
- 3"  
- 3 3/4"

### Stamping:

- Not Stamped  
- Illegible

Height (+) above/(-) below Surface: -2.0

Notes:

---

Close Up Photo  
Location Photo
Coordinates from RTK unit are uploaded after post processing by survey office lead.
AUTOMATION MAGIC
ARCGIS FOR SERVER (PYTHON INSTANCE)

Functionality is dependent on your ArcGIS for Server license level
MATRIX LINKS

- ArcGIS Enterprise 10.5
- ArcGIS for Server 10.4

Google "ArcGIS Server [version] matrix"
ARCPY DATA ACCESS MODULE

E.g. Search, Update, Insert Cursors
What is the data access module?

The data access module, `arcpy.da`, is a Python module for working with data. It allows control of the edit session, edit operation, improved cursor support (including faster performance), functions for converting tables and feature classes to and from NumPy arrays, and support for versioning, replicas, domains, and subtypes workflows.

### Data Access functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ExtendTable</code></td>
<td>Joins the contents of a NumPy structured array to a table based on a common attribute field. The input table is updated to contain the fields from the join table.</td>
</tr>
<tr>
<td><code>FeatureClassToNumPyArray</code></td>
<td>Converts a feature class to NumPy structured array.</td>
</tr>
<tr>
<td><code>ListDomains</code></td>
<td>Lists the attribute domains belonging to a geodatabase.</td>
</tr>
<tr>
<td><code>ListFieldConflictFilters</code></td>
<td>Lists the fields in a versioned feature class or table that have field conflict filters applied.</td>
</tr>
<tr>
<td><code>ListReplicas</code></td>
<td>Lists the replicas in the workspace.</td>
</tr>
<tr>
<td><code>ListSubtypes</code></td>
<td>Returns a dictionary of the subtypes for a table or feature class.</td>
</tr>
<tr>
<td><code>ListVersions</code></td>
<td>Lists the versions in the workspace.</td>
</tr>
<tr>
<td><code>NumPyArrayToFeatureClass</code></td>
<td>Converts a NumPy structured array to a point feature class.</td>
</tr>
<tr>
<td><code>NumPyArrayToTable</code></td>
<td>Converts a NumPy structured array to a table.</td>
</tr>
<tr>
<td><code>TableToNumPyArray</code></td>
<td>Converts a table to NumPy structured array.</td>
</tr>
<tr>
<td><code>Walk</code></td>
<td>Generates directory/database structure by walking the tree top-down or bottom-up. Each directory/workspace yields a tuple of three: directory names, directory names, and file names.</td>
</tr>
</tbody>
</table>
ARCPY MAPPING MODULE

MXD Template Interface
Introduction to arcpy.mapping

What is arcpy.mapping?

ArcPy.mapping is a Python scripting module that is part of the ArcPy site package. It gets installed with ArcGIS for Desktop and is available to all licenses. It was designed primarily to manipulate the contents of existing map documents (.mxd) and layer files (.lyr). It also provides functions to automate exporting and printing. ArcPy.mapping can be used to automate map production; it extends the capabilities of Data Driven Pages and is required to build complete map books because it includes functions to export to, create, and manage PDF documents. Finally, ArcPy.mapping scripts can be published as geoprocessing services and the script functionality can be made available to web applications.

It is easiest to understand the capabilities of arcpy.mapping by describing some scenarios it facilitates. Here are only a few of the many scenarios that an arcpy.mapping script can accomplish:

- Create a report on information contained in map documents such as data frame coordinate system, layer data sources, layers with broken data sources, or layout element positioning.
- Update, repair, or replace layer data sources in a map document or layer file.
- Update layer symbology without having to physically open map documents.
- Find and replace a text string for all map documents within a folder.
- Save map documents to a prior ArcGIS version for distribution.
- Updating the map document metadata (for example, keywords, summary, and description).
- Create geographic data in batch using map export commands, such as a series of GeoTIFF images driven by a list of features in a data frame.
- Automate the creation and management of map services to be published with ArcGIS for Server.
- Build a variety of PDF map books:
  - A thematic or temporal map book with title page, multiple map pages, and any number of additional pages with supporting content such as tabular reports and contact lists.
  - A reference map book based on Data Driven Pages output.
3RD PARTY PYTHON MODULES
http://docs.python-requests.org/en/master/
Requests: HTTP for Humans

Release v2.18.1. (Installation)

Requests is the only Non-GMO HTTP library for Python, safe for human consumption.

Warning: Recreational use of the Python standard library for HTTP may result in dangerous side-effects, including security vulnerabilities, verbose code, reinventing the wheel, constantly reading documentation, depression, headaches, or even death.

Behold, the power of Requests:

```python
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u"{'type':"User"...
>>> r.json()
{u'private_gists': 419, u'total_private_repos': 77, ...}
```

See similar code, sans Requests.

Requests allows you to send organic, grass fed HTTP/1.1 requests, without the need for manual labor. There's no need to manually add query strings to your URLs, or to form-encode your POST data. Keep alive and HTTP connection pooling are 100% automatic, thanks to urllib3.

User Testimonials


Armin Ronacher--

Requests is the perfect example how beautiful an API can be with the right level of documentation.
PILLOW

https://python-pillow.org/
Welcome
This is the home of Pillow, the friendly PIL fork. PIL is the Python Imaging Library. If you have ever worried or wondered about the future of PIL, please stop. We're here to save the day.

Code
Our code is hosted on GitHub, tested on Travis CI, AppVeyor, Coveralls, Landscape and released on PyPI.

Documentation
Our documentation is hosted on readthedocs.org and includes installation instructions, handbook, API reference, release notes and more.

Discussion
Discussion occurs on GitHub, Stack Overflow and IRC. IRC conversations are logged here.

BONUS CONTENT

- ArcGIS for Portal Maps
- Daily Summary Reports