

Utility Network, arcpy.mp, and JSAPI Enable Map Product Delivery via Portal

Mark Cederholm
UniSource Energy Services
Flagstaff, Arizona

DEVELOPER SUMMIT
2021



GeoNet Blog Posts

- Migrating a Plot from ArcObjects and ArcMap to arcpy.mp and Portal
<https://tinyurl.com/4grg9n5k>
- Follow-up to this presentation



Versions Used in Examples

- ArcGIS Pro: 2.5.2, 2.7.1
- ArcGIS Enterprise:
 - 10.8: Web Tool, WAB
 - 10.8.1: Utility Network, [EB?]



Demo

The screenshot displays the ArcGIS Pro interface for a project named "System_Cert_UN [Read Only] - System Cert Maps". The ribbon includes tabs for Project, Map, Insert, Analysis, View, Edit, Imagery, and Share. The main map area shows a geographic map with several red star markers and blue star markers, each labeled with a unique ID. A blue shaded area is visible on the map. The left sidebar contains a "Contents" panel with a search bar and a "Drawing Order" list. The right sidebar contains an "Element" panel with the text "Select an element." The status bar at the bottom shows a scale of 1:90,775, coordinates of 436,159.65E 3,900,794.82N, and "Selected Features: 0".

System	Component	ID
POC Valves Feeding System 1	Valve	WME001 - WILLIAMS BORDER STAT
Stations Feeding System 1	Station	04-1232 - DOG HAUS REGULATOR S
Stations Feeding System 1	Station	04-2402 - FLAGSTAFF CITY GATE #2
Stations Feeding System 1	Station	04-1251 - HILLCREST REGULATOR S
Stations Feeding System 1	Station	04-1222 - IZABEL REGULATOR STAT
Stations Feeding System 1	Station	04-1273 - JAY LANE REGULATOR ST
Stations Feeding System 1	Station	04-1259 - LITTLE AMERICA REGULA
Stations Feeding System 1	Station	04-1241
Stations Feeding System 1	Station	04-1251
Stations Feeding System 1	Station	04-1222
Stations Feeding System 1	Station	04-1232
Stations Feeding System 1	Station	04-1273
Stations Feeding System 1	Station	04-1259
Stations Feeding System 1	Station	04-1281
Stations Feeding System 1	Station	04-1245
Stations Feeding System 1	Station	04-1254
Stations Feeding System 1	Station	04-2402



Utility Network and Mapping

- Subnet lines and attributes: MAOP, number of valves, number of meters
- Useful network feature attributes: subnetwork Names



Utility Network and Mapping

- Map service layers may be used in ArcGIS Pro projects and web maps
- Layers may be queried using REST API, JSAPI, or Python API



arcpy.mp

- Load and modify symbols
- Change and manipulate renderers
- Make layout elements visible or invisible
- Make modifications at the CIM level*

*Works in Pro 2.5.2, crashes 2.7.1



The Python Toolbox

```
def getParameterInfo(self):  
    param0 = arcpy.Parameter(  
        displayName = "Project Folder",  
        name = "project_folder",  
        datatype = "GPString",  
        parameterType = "Optional",  
        direction = "Input")
```

```
sDir = parameters[0].valueAsText  
if sDir == " " or sDir == "" or sDir == "#" or sDir == None:  
    # Use this as a placeholder for publishing the GP service  
    parameters[self._iResult].value = sResult  
return
```




```
param6 = arcpy.Parameter(  
    displayName = "Output PDF",  
    name = "output_pdf",  
    datatype = "DEFile",  
    parameterType = "Derived",  
    direction = "Output")
```

```
sOutput = os.path.join(arcpy.env.scratchFolder, sOutName)  
layout.exportToPDF(sOutput)  
self.__sOutput = sOutput  
self.__bOutput = True
```

```
if self.__bOutput:  
    self.__Message("Output = " + self.__sOutput)  
    parameters[self._iOutput].value = self.__sOutput
```



10.8.1 and Layout.exportToPDF

- Depending on Portal configuration, exportToPDF may cause "OSError"
- Print Service patch does not fix it
- Workaround: use 10.8



Getting the current credentials

```
# Get portal (verify_cert is needed for web tool)
gis = arcgis.gis.GIS("pro", verify_cert=False)
sPortalUrl = gis.url
sPortalUrl = sPortalUrl.replace("/portal", "")
if sPortalUrl[len(sPortalUrl) - 1:] != "/":
    sPortalUrl += "/"
#gis = None
#sPortalUrl = "https://gisq.uns.com/"
sMapUrl = sPortalUrl + self._sMapService
```



Querying subnet lines

```
# Query subnet lines
SystemPressures = {}
sSubnetUrl = sMapUrl + "/4"
sWhere = "ISDIRTY = 0 And TIERNAME = 3"
Fields = ["SUBNETWORKNAME", "maoprecord"]
lyrSubnet = arcgis.features.FeatureLayer(sSubnetUrl, gis)
Features = lyrSubnet.query(sWhere, Fields, return_geometry=False)
for feat in Features.features:
    attr = feat.attributes
    sName = attr["SUBNETWORKNAME"]
    dmaop = attr["maoprecord"]
    SystemPressures[sName] = dmaop
```



Querying stations

```
sAssemblyUrl = sMapUrl + "/0"
sWhere = "ASSETGROUP IN (3,5) And lifecyclestatus = 8"
sWhere += " And SUPPORTEDSUBNETWORKNAME Like '%" + sSystemName + "%'"
Fields = ["name", "assetid", "SUPPORTEDSUBNETWORKNAME"]
lyrAssembly = arcgis.features.FeatureLayer(sAssemblyUrl, gis)
Features = lyrAssembly.query(sWhere, Fields, return_geometry=False)
for feat in Features.features:
    attr = feat.attributes
    sStationName = attr["name"]
    sAssetID = attr["assetid"]
    sSubnets = attr["SUPPORTEDSUBNETWORKNAME"]
    Subnets = set(sSubnets.split("::"))
    if sSystemName not in Subnets:
        continue
    AssetIDs[sStationName] = sAssetID
    StationSystems = []
    for sName in Subnets:
        if sName not in SystemPressures:
            continue
        StationSystems.append(sName)
```



Determine if station is feeding or fed

```
if len(StationSystems) == 1:
    StationsFeeding.add(sStationName)
    continue
sSystem0 = StationSystems[0]
dMAOP0 = SystemPressures[sSystem0]
sSystem1 = StationSystems[1]
dMAOP1 = SystemPressures[sSystem1]
if sSystem0 == sSystemName and dMAOP0 > dMAOP1:
    StationsFed.add(sStationName)
    SystemsFed.add(sSystem1)
    continue
elif sSystem1 == sSystemName and dMAOP1 > dMAOP0:
    StationsFed.add(sStationName)
    SystemsFed.add(sSystem0)
    continue
StationsFeeding.add(sStationName)
```



Query extent of systems

```
# Query extent of affected systems
SystemsAffected = [sSystemName]
for sName in SystemsFed:
    SystemsAffected.append(sName)
sWhere = self.__BuildInClause("SUBNETWORKNAME", SystemsAffected)
sWhere = "TIERNAME = 3 And " + sWhere
result = lyrSubnet.query(sWhere, return_extent_only=True)
env = arcgis.geometry.Geometry(result["extent"])
```



Convert to arcpy and expand

```
# Create expanded extent
# [BUG: Envelope.as_arcpy wipes out the spatial reference]
ext = env.as_arcpy
sr = env.spatial_reference.as_arcpy
dRatio = 1.05
dXMid = (ext.XMin + ext.XMax) / 2
dYMid = (ext.YMin + ext.YMax) / 2
dW = ext.width * dRatio / 2
dH = ext.height * dRatio / 2
extMap = arcpy.Extent(dXMid - dW, dYMid - dH, dXMid + dW, dYMid + dH,
                      spatial_reference=sr)
```



Open APRX and get map and layout

```
# Open project, get layout, and update it
aprx = arcpy.mp.ArcGISProject(self.__sProject)
layout = aprx.listLayouts(self._sLayout)[0]
```

```
frame = layout.listElements("MAPFRAME_ELEMENT")[0]
map = frame.map

# Set up map layers
for lyr in map.listLayers():
    sName = lyr.name
```

```
# Set map extent
frame.camera.setExtent(self.__extMap)
```



Modify Layers

```
elif sName[:25] == "Stations Fed From System ":
    if len(self.__StationsFed) == 0:
        lyr.visible = False
    else:
        lyr.name = sName[:25] + self.__sSystemNumber
        sWhere = self.__BuildInClause("name", self.__StationsFed)
        lyr.definitionQuery = "ASSETGROUP IN (3, 5) And " + sWhere
        self.__UpdateStationRenderer(lyr, False, self.__StationsFed)
        lyr.visible = True
elif sName[:30] == "Distribution Systems Fed From ":
    if len(self.__SystemsFed) == 0:
        lyr.visible = False
    else:
        lyr.name = sName[:30] + sTownName + " " + self.__sSystemNumber
        sWhere = self.__BuildInClause("SUBNETWORKNAME", self.__SystemsFed)
        lyr.definitionQuery = "TIERNAME = 3 And " + sWhere
        self.__UpdateSystemRenderer(lyr, False, self.__SystemsFed)
        lyr.visible = True
```



Update text elements*

*Works in 10.8, but not 10.8.1

```
for txt in layout.listElements("TEXT_ELEMENT"):
    sName = txt.name
    dW0 = txt.elementWidth
    dX0 = txt.elementPositionX
    bReposition = True
    if sName == "main_title":
        txt.text = sTownName + " System # " + self.__sSystemNumber + " Certification"
    elif sName == "subtitle":
        txt.text = self.__sSubtitle
    elif sName == "date_now":
        txt.text = datetime.strftime(datetime.now(), "%m/%d/%Y")
        bReposition = False
    if bReposition:
        dW1 = txt.elementWidth
        dX1 = txt.elementPositionX
        dX2 = dX1 - ((dW1 - dW0) / 2)
        txt.elementPositionX = dX2
```

Center anchor points do not work properly
at 2.5.2, 10.8 – fixed at 2.7.1



Update text elements via CIM*

```
cim = layout.getDefinition("V2")
for e in cim.elements:
    bUpdate = False
    sName = e.name
    if sName == "main_title":
        sTxt = sTownName + " System # " + self.__sSystemNumber + " Certification"
        bUpdate = True
    elif sName == "subtitle":
        sTxt = self.__sSubtitle
        bUpdate = True
    elif sName == "date_now":
        sTxt = datetime.strftime(datetime.now(), "%m/%d/%Y")
        bUpdate = True
    if not bUpdate:
        continue
    g = e.graphic
    g.text = sTxt
layout.setDefinition(cim)
```

*Works in Pro 2.5.2, crashes 2.7.1



Loading symbols from .stylx*

```
bUseSavedSymbol = False
Symbols = symbolSaved.listSymbolsFromGallery(sSymbolName)
if len(Symbols) == 0:
    self.__Message("Symbol not found: " + sSymbolName)
    bUseSavedSymbol = True
```

```
if bUseSavedSymbol:
    itm.symbol = symbolSaved
else:
    itm.symbol.applySymbolFromGallery(sSymbolName)
```

*Works in 10.8.1/10.8, but not in Pro 2.5.2 or 2.7.1



listMissingValues*

```
if bUseList:  
    missing = sym.renderer.listMissingValues()  
    grp = missing[0]  
    sym.renderer.addValue({sHeading:grp.items})  
else:  
    sym.renderer.addValue({sHeading:Values})
```

*Works in 10.8, but not 10.8.1



Creating an empty result

The screenshot shows a dialog box titled "System Cert Map ()" with a close button (X) in the top right corner. The dialog has a green header bar with a checkmark icon and the text "Completed.". Below this, it displays the following information:

- Started:** Today at 11:51:47 AM
- Completed:** Today at 11:51:48 AM
- Elapsed Time:** 1 Second

There are three expandable sections:

- Errors and warnings:** Indicated by a right-pointing chevron (>).
- Parameters:** Indicated by a downward-pointing chevron (v). This section contains a list of parameters:
 - Project Folder
 - Town Name
 - System Number
 - Subtitle
 - Show other mains: false
 - Result: No results.
 - Output PDF
- Environments:** Indicated by a right-pointing chevron (>).
- Messages:** Indicated by a right-pointing chevron (>).

A small grid icon is visible in the bottom right corner of the dialog box.



Publishing the Web Tool

Overwrite Web Tool ? ▾ ⌵ ✕

Overwrite SystemCertMapTool Web Tool

General Configuration Content Messages

Item Details

Name
SystemCertMapTool

Summary

Tags

Data

Finish Sharing



Setting the Message Level

Overwrite Web Tool ? ▾ 🔍 ✕

Overwrite SystemCertMapTool Web Tool

General Configuration Content Messages

Capabilities

Upload

Parameters

Execution Mode

Asynchronous (recommended for long-running tc
 View output in map image layer
 Synchronous

Properties

Message Level Info ▾

Maximum number of records returned
by server

Finish Sharing



Running the Web Tool

ArcGIS REST Services Directory

[Home](#) > [services](#) > [DevSummit](#) > [SystemCertMapTool \(GPServer\)](#) > [System Cert Map](#) > [j5a](#)

[JSON](#)


Result Parameter: output_pdf (System Cert Map)


```
{
  "paramName": "output_pdf",
  "dataType": "GPDataFile",
  "value": {"url": "https://30029327ung.unisource.corp/server/rest/directories/arcgisjobs/devsummit/s"}
}
```




A Custom Web AppBuilder Widget


System Cert Maps with Web AppBuilder for ArcGIS


Find address or place 

 Nellis Air Force Range

System Cert Map Widget 

OK.

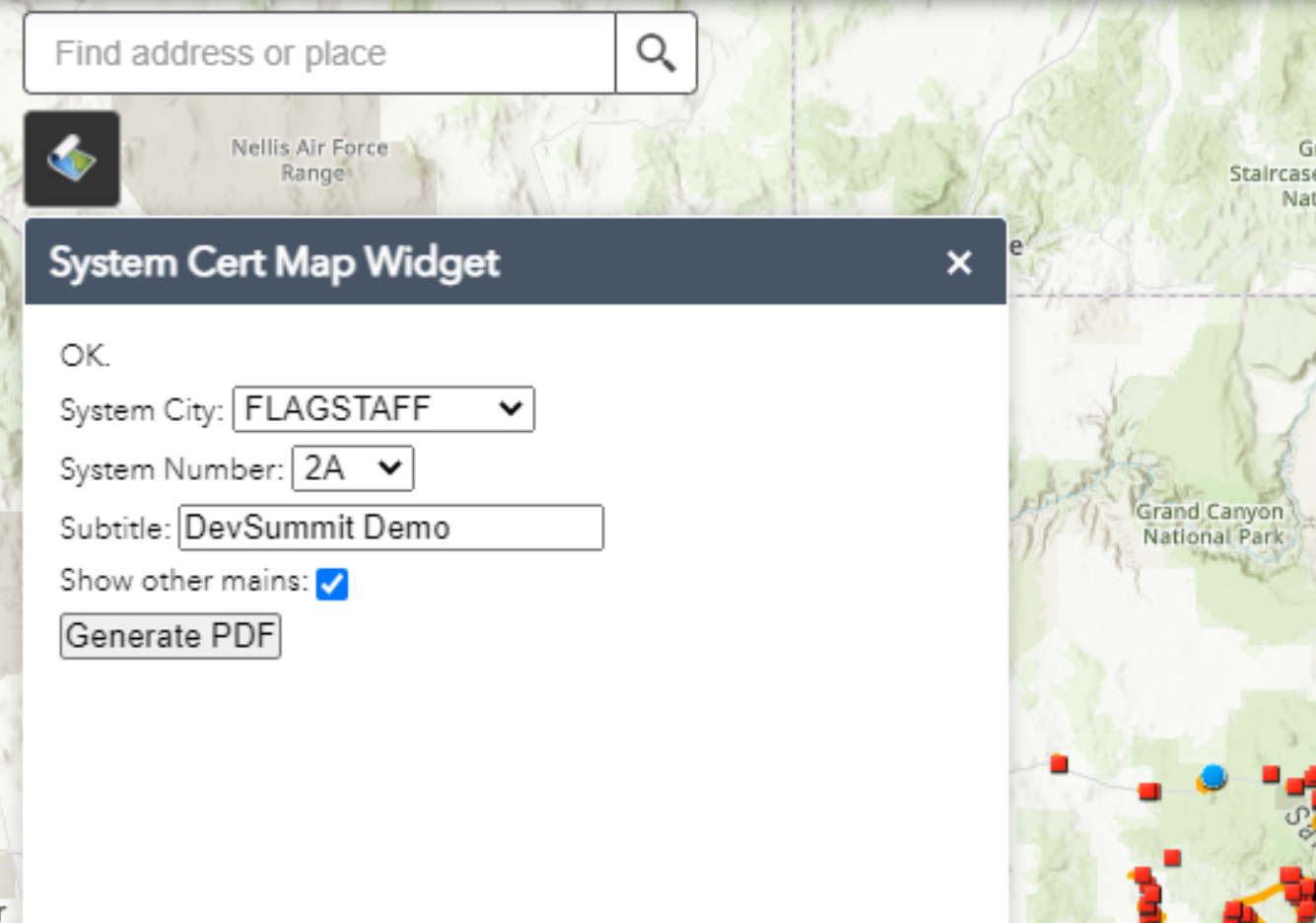
System City: 

System Number: 

Subtitle:

Show other mains:

Grand Canyon National Park



Querying Subnet Lines

```
this.status("Querying subnets...");  
var task: QueryTask = new QueryTask(this._urlSubnet);  
var query: Query = new Query();  
query.where = "ISDIRTY = 0 And TIERNAME = 3";  
query.returnGeometry = false;  
query.outFields = ["SUBNETWORKNAME"];  
task.on("complete", (result: any) => { this.onQueryTaskComplete(result); } );  
task.execute(query);
```



Parsing Subnet Names

```
let Features: FeatureSet = evt.featureSet;
let Graphics: Graphic[] = Features.features;
let Systems: Object = Object();
let SystemTowns: string[] = [];
var SystemNumbers: string[];
for (var i: number = 0; i < Graphics.length; i++)
{
    let g: Graphic = Graphics[i];
    let sSystem: string = g.attributes["SUBNETWORKNAME"];
    let tok: string[] = sSystem.split(" ");
    let sNum: string = tok[tok.length - 1];
    let sTown = sSystem.replace(" " + sNum, "");
    if (Systems.hasOwnProperty(sTown))
        SystemNumbers = Systems[sTown];
    else
    {
        SystemTowns.push(sTown);
        SystemNumbers = [];
        Systems[sTown] = SystemNumbers;
    }
    SystemNumbers.push(sNum);
}
```



Sorting and Populating Results

```
SystemTowns.sort();
for (var i: number = 0; i < SystemTowns.length; i++)
{
    let sTown: string = SystemTowns[i];
    SystemNumbers = Systems[sTown];
    SystemNumbers.sort((s1: string, s2: string) =>
        { return this.compareSystemNumber(s1, s2); })
}
this._Systems = Systems;
this.populateSelect(this._selSystemTown, SystemTowns);
// This does not trigger _onSystemTownChange
this._selSystemTown.selectedIndex = 0;
this.getSystemNumbers();
this._btnGenerate.disabled = false;
this.status("OK.")
```



System Town Change Event

```
props["_onSystemTownChange"] = () => { this.getSystemNumbers(); }
```

```
private getSystemNumbers(): void  
{  
    let sTown: string = this._selSystemTown.value;  
    let SystemNumbers: string[] = this._Systems[sTown];  
    this.populateSelect(this._selSystemNumber, SystemNumbers);  
    this._selSystemNumber.selectedIndex = 0;  
}
```



Create and Call Geoprocessor

```
this._gp = new Geoprocessor(this._urlGP);
this._gp.on("status-update", (info: JobInfo) =>
  { this.onStatusUpdate(info); });
this._gp.on("job-complete", (info: JobInfo) =>
  { this.onJobComplete(info); });
this._gp.on("get-result-data-complete", (result: any) =>
  { this.onGetResultDataComplete(result); });
```

```
this.status("Generating output...");
this._btnGenerate.disabled = true;
let sOtherMains: string = "false"
if (this._chkOtherMains.checked)
  sOtherMains = "true"
let params: any =
{
  "project_folder": this._dir,
  "town_name": this._selSystemTown.value,
  "system_number": this._selSystemNumber.value,
  "subtitle": this._txtSubtitle.value,
  "show_other_mains": sOtherMains
};
this._sJobId = "";
this._gp.submitJob(params);
```



Display Status and Get Results

```
private onStatusUpdate(evt: any): void
{
    let info: JobInfo = evt.jobInfo;
    let sMsg: string = info.jobStatus + "...";
    let Messages: GPMessage[] = info.messages;
    let iLen: number = Messages.length;
    if (iLen > 0)
        sMsg = Messages[iLen - 1].description;
    this.status(sMsg);
}

private onJobComplete(evt: any): void
{
    let info: JobInfo = evt.jobInfo;
    this._sJobId = info.jobId;
    this._gp.getResultData(info.jobId, "result");
}
```



Get PDF and Open Window

```
private onGetResultDataComplete(evt: any): void
{
    let val: ParameterValue = evt.result;
    let sName: string = val.paramName;
    if (sName === "output_pdf")
    {
        this.status("Done.");
        window.open(val.value.url);
        this._btnGenerate.disabled = false;
        return;
    }
    let sVal: string = val.value;
    if (sVal !== "Success.")
        return;
    if (this._sJobId === "")
        return; // This should never happen
    this._gp.getResultData(this._sJobId, "output_pdf");
    this.status(sVal);
}
```



TO DO:

- Resolve 10.8.1 issues
- Email notification for long jobs
- Experience Builder



Questions?

- Mark Cederholm
mcederholm@uesaz.com
- These slides, and sample code, may be downloaded at:
<http://www.pierssen.com/arccgis/python.htm>

