# Overview

Setting up the Project Performance Assessment (PPA) tool, version 2.0, entails the following components:

1. Code repository, input data layers, and related files
2. Python dependency packages, beyond the python standard library, that must be installed in order to run the code
3. An enterprise geodatabase (GDB) that is registered with ArcServer or ArcGIS Portal
4. ArcGIS Pro, to publish the tool
5. ArcServer, to host the tool

## Using This Guide for Making Your Own PPA Tool

Many of the processes described in this developer guide are oriented toward a SACOG-specific deployment of the PPA tool. However, if you are working independently or for another agency, you should be able to use this guide to understand what steps and data layers are needed and accordingly adapt the inputs to be specific to your organization and available data layers.

## PPA Setup on Federated ArcGIS Server



Further reading on SACOG’s Federated ArcGIS Server - <https://github.com/djconly85/PPA2_0_code/blob/master/SACOG_EnterpriseGDBSummary.pdf>

## Source Code Repository

All source python and SQL code are available on <https://github.com/djconly85/PPA2_0_code>

## Software Requirements

* ArcGIS Pro 2.x
* ArcServer 10.7.x
* Relational database management system (RDBMS) such as Microsoft SQL Server, MySQL, or PostgreSQL
* Python 3.x, preferably provided through the Anaconda environment that ships with ArcGIS Pro

## Assumed Knowledge and Further Reading

The processes described in this developer guide assume the reader is familiar with and/or has access to colleagues who are familiar with the following programs and services:

* Programs provided by ESRI, Inc.
  + ArcGIS Pro
  + ArcServer
  + ArcGIS Portal
  + ArcGIS Enterprise geodatabase setup and registration
  + ArcGIS Online and Web Apps
* Python and [all dependency packages](#_Python_Dependencies) used in the PPA tool

# Input Data

The PPA uses many data files to calculate its performance metrics. This section describes how these layers were created and how PPA-specific fields within these layers were calculated. Comprehensive tables of the data files and fields within each file are available in the Appendix.

## Land Use and Travel Behavior Data

SACOG’s Integrated Transportation and Land Use (ILUT) data are prepared from a combination of SACOG’s land use forecast and outputs of its SACSIM travel demand model. It is provided at the parcel level as points and polygons.

### Parcel Points

* Columns added:
  + SUBRAD, from TAZ07 file
    - Stands for Sub-Regional Analysis District
    - Used for making interface maps
  + TYPE\_CODE values from SACOG’s Envision Tomorrow land use data
    - 1 – High density
    - 2 - Med-High density
    - 3 - Med density
    - 4 – Low density
    - 5 – Very low/Rural residential
    - 6 – Mix use
    - TYPE\_CODE used to make simpler bar charts showing housing diversity/mix within an area
  + Type code – LUTYPE lookup table corresponds type code ID numbers to plain-english descriptions. It is included with GIS materials as lutype\_typecode\_xref.csv
  + TYPE\_CODE\_DESC
    - Plain-English version of TYPE CODE, based on coding described under the TYPE\_CODE description

### Parcel Polygons

* Has subset of ILUT attributes; built by simple join of parcel polygon feature class with ILUT point data.
* Used for metrics that depend on measuring acres of something. E.g., acres of ag land use within a buffer.

### Using Different Land Use Data

The PPA is designed to use point-level parcel data as its primary source of land use data. Using other land use geography types (e.g. census tracts, traffic analysis zones (TAZs)) will require the user to assign that more aggregate data to a parcel-level point file in order to have a small enough geographic scale.

## Community Types

Community types can be thought of as “types of built environments”. In the SACOG region, for example, some of the community types are “urban core”, “small town main street”, and “rural residential”. Community types are polygon areas, and the PPA identifies which community type a project falls within. Tagging projects to community types allows apples-to-apples comparison between projects (e.g., compare rural projects to other rural projects, urban core projects to other urban core projects, etc.).

Community types in PPA2 for the SACOG region are based on community types in SACOG’s 2020 MTP-SCS. However, for PPA2 they were further subdivided based on jurisdiction to better capture variation and, for example, differences between the “urban core” downtown of Sacramento and “rural mainstreet” downtown of Winters. There is a lookup table (comm\_type\_x\_juris\_create.xlsx) corresponding the new, subdivided community types to the more generic community types used in the 2020 MTP.

## NPMRDS TMC Data

The National Performance Measurement Research Data Set (NPMRDS) is provided by FHWA to state DOTs, MPOs, and contractors working for these agencies. It provides national travel time data for hundreds of thousands of road segments called traffic message channels, or TMCs. The speed data, at the time of this writing, are provided to FHWA by Inrix.

Important to note is that NPMRDS speeds are provided for free to state DOTs and MPOs for roads on the National Highway System (NHS). Data are available for roads off the NHS, but require an additional purchase of the data from Inrix.

### Speed Data

SQL queries generate speed metrics for each TMC, which are then joined to a GIS feature class of TMCs. A list of SQL query scripts is available in the table of scripts in the appendix

Directions are based on the NORTHBOUND / SOUTHBOUND / EASTBOUND / WESTBOUND direction names from the TMC\_identification.csv file that comes with each NPMRDS download and NOT based on the N/S/E/W direction tags that come in the NPMRDS shapefile. This is because (1) NPMRDS shapefile only has data for TMCs on the National Highway System (NHS) and (2) because the directions given in the SHP file are true cardinal directions, rather than signed directions. This means, for example, I-5 NB between SMF airport and Woodland has a SHP direction of “W” for west, but it’s signed direction is ‘NORTHBOUND’ because I-5 is, overall, an officially north-south highway.

### Truck count data

* From Caltrans 2016 truck counts point SHP file
* For state highways only
* Used in PPA to calculate share of traffic on road that is truck traffic, for the freeway report only (not the arterial or state-of-good-repair reports).
* Process to tag truck count data to TMCs:
  + As needed, manually separate out count points to be on correct sides of interchanges so their data spatially join to correct TMC.
  + Spatially join count point data to TMCs in GIS.
  + Ensure to divide count values by 2 as needed to get correct directional volumes. Assume that percent values are the same for each direction.
  + Manually clean up any spatial join errors

Truck counts were only taken at select locations and so were not tagged to all highway TMCs for PPA2. However, options to fill in the remaining TMCs include:

* Use Inrix-sourced truck volumes from NPMRDS data set (used for PM3 metrics), but those data are proprietary and there’s no information on how they were gathered. They also show significant discrepancy with Caltrans-based truck data. Use at your own risk.
* It is possible to manually propagate Caltrans truck volumes to “downstream” TMCs, but this would be a somewhat laborious, manual process and would require making assumptions about truck volumes at some locations.

## Hex Layers

Hex layers are used in the tool interface to visualize key ILUT data such as population, VMT per capita, jobs, dwelling units, etc.

NOTE – Hex layers and their calculation does not in any way affect the outcomes calculated by the PPA tool. They are purely a means by which to visualize data in the user map interface.

### Mix Index Hex Layer

Procedure to calculate mix index[[1]](#footnote-2) value:

1. For each parcel polygon, calculate its mix index based on 1-mile buffer
2. Do GIS intersect between parcel polygons and hex polygons, then calculate area-weighted average mix index value for each hex.

Refer to the script guide in the appendix to find which script to use.

Example execution of procedure to calculate mix index at the hex level:

1. Hex contains 10 net acres[[2]](#footnote-3) and intersected by parcel A and parcel B
2. Parcel A’s mix index is 0.5, parcel B’s mix index is 0.3
3. Within Hex’s 10 acres, 8 are intersected by parcel A, the remaining 2 are intersected by parcel B
4. Therefore the Hex’s mix index would by ((8\*0.5) + (2\*0.3)) / 10, or 0.46

Notes on creating the hex mix index value using SACOG’s land use parcel polygon data:

* The mix index is a “two-way” score: it returns a lower score the farther the mix travels from the ‘optimal’ mix, regardless of which direction it travels. So, for example, a location with many jobs, retail establishments, etc. but few households will score low because there are not many people who live nearby enough to walk to bike easily to those jobs/destinations. Conversely, a hex that is mostly housing but has few jobs or retail will also return a low mix index score.
* In some instances, the sum of the area on parcels within a hex is greater than the area of the whole hex. For most cases, the difference is very small (< 1 square foot). In a few cases, the difference is larger (E.g., for 41 hexes the sum of the areas on the parcel pieces within the hex is more than 10% larger than the area of the entire hex). These exceptions appear to be due to duplicate and/or overlapping parcels and are concentrated along county boundaries.

## Sugar Access Data

### About Sugar Access

Sugar Access software, created by Citilabs (now a division of Bentley as of 2020), performs accessibility analyses. Example questions it can answer are:

* From each census tract, how many jobs can I reach within a 30-minute drive?
* From each TAZ, how many grocery stores can I walk to within 30 minutes?
* If I build a new bridge, how much will the new connection affect accessibility? Where will it affect accessibility?

The PPA uses outputs from Sugar Access as inputs to allow the PPA to calculate its accessibility metrics for each project.

### Using Sugar in PPA

* Analysis time period was weekdays, 7am-9am for 2018. This means that traffic congestion and transit service availability is based on conditions during this time period.
* Street network: all-streets network provided by Sugar, with HERE-based traffic speeds aggregated by peak vs. Off-peak time periods. As mentioned above, all accessibility analyses are based on AM peak congestion data.
* All trips are assumed to start at polygon (in this case, block group or block) centroid, connect to nearest all-streets network segment, then use provided all-streets network and/or transit network to arrive at destination polygon centroid.
* Sugar’s decay curve feature was enabled. Decay curves consider mode, trip purpose, and travel time and assume that as a destination gets farther away, fewer people are willing to travel to access it and therefore it counts as fewer destinations[[3]](#footnote-4).

#### Destination Types

* Total jobs
* Low-income jobs (rationale: transit access to jobs that are more likely to have low-income and thus transit-dependent employees)
* Total “POIs”, equal to sum of:
  + Parks
  + “education facilities”
  + libraries
  + “medical facilities”
  + grocery stores
  + pharmacies
  + clothing stores
  + banks
* “Education facilities” = Higher\_Ed + Schools
* “Medical facilities” = Hospital + Medical\_Service

#### Modes

* WALK = 30-minute walk
* BIKE = 30-min bike ride
* TRAN = 45-minute transit trip, inclusive of walk time for stop access/egress
* AUTO:
  + 30min drive for Job, Medical, and education destinations
  + 15min drive for POIs

## Travel Model Highway Network

The travel model highway network is a DBF created by exporting the Cube NET highway file resulting from a SACSIM model run. Among the data relevant to the PPA it contains are:

* Modeled vehicle volumes for each model link, broken out by vehicle occupancy.
* After a joining data from the [transit links file](#_Transit_Links_File), the number of modeled transit trips taken on the highway segment. This is only used for the PPA’s freeway report.

Base and future year model data come from the model runs performed by SACOG as part of the Environmental Impact Report (EIR) submitted with its 2020 Metropolitan Transportation Plan (MTP).

### Transit Links File

The transit links file (trans.link.all.dbf) is a DBF created after a SACSIM model run. Through aggregation at the A\_B link level, you can calculate the total number of transit trips that occurred on each highway link.

## Transit Stop Event Point Data

* Based on 2016 GTFS points
* Steps to create:
  + Create point file of all transit stop events for all transit providers in region based on the stops.txt and stop\_times.txt GTFS files
  + Perform GIS dissolve to points based on stop X/Y coordinates. Resulting file will be unique point locations with count of vehicle events (one bus stopping at one bus stop is one vehicle event) that happen at each point.
  + Resulting file is NOT a good indicator of where every bus stop is in the system; PPA uses it to show where transit vehicle activity occurs.

## Road Intersection Point Data

To create point file from roadway network in ArcGIS:

1. Make sure your roadway centerline file is projected to a reference system with very fine X/Y coordinate values (e.g. feet or meters, not WGS84 degrees).
2. Run Vertices to Points tool, specifying that you want points created for both the start and end points of each line segment in the centerline file.
3. Calculate the X/Y values for each point
4. Add columns, as long integer data type, for X\_int and Y\_int.
5. Set these int columns equal to the X/Y values you calculated in step 3. This will truncate the X/Y values.
6. Run GIS Summarize tool, summarizing by the X\_int and Y\_int values and getting a count of points that grouped.
7. With the resulting table from the Summarize, change it into a feature class based on the X\_int and Y\_int values. This is your resulting intersection file used in the PPA tool The COUNT column contains the number of legs extending from each intersection. For PPA purposes, only 3- and 4-way intersections are counted as intersections. E.g., cul-de-sacs and two adjacent links on the same road are not considered intersections.

## Collision Point Data

### Process to create

1. Download from TIMS (collision data, not party/victim tables)
2. Geocode using TIMS processor script
   1. NOTE – generally 10-15% of collisions get geocoded to the wrong location
3. Use TIMS freeway tagger script to tag whether a collision happened on a freeway or not.
4. Manual post-process may be needed to tag collisions on non-freeway state highways as being “non-freeway”, because script may tag them as being on freeways just because they’re on state highways. This must be done as a semi-manual process in GIS.

# Registering Inputs with ArcServer

For the tool to run online as a hosted geoprocessing service, it needs access to its input data layers (land use data, congestion data, etc.). To be accessible by the tool, the data layers must be in a location that is registered with ArcServer.

## What needs to be registered

* All input data layers
* APRX map file used
* Any configuration or parameter files used (e.g. CSVs, XLSXs, etc.)

## Registering a Folder

1. Go to Portal - <https://portal.sacog.org/portal/homev>
2. Go to Content > Add Item > A Data Store
3. Specify title of Data Store object, type, and tags
4. Enter the file path using full UNC machine name (e.g. [\\arcserver-svr\](file:///\\arcserver-svr\)...)
5. Validate

## Registering an Enterprise Geodatabase

The tool can run from a file geodatabase contained within a folder registered as a Data Store, but for stability when multiple users run the tool at once, the tool data must be in an enterprise geodatabase, not a file geodatabase. To set up an enterprise geodatabase:

1. Go to [Server Manager](#_Accessing_Server_Manager) > Site > Data Stores
2. Select to Register a database

### Accessing Server Manager Site Through Portal

1. Go to Portal home page > Organization > Settings > Servers
2. Resulting page lists URLs for the server manager sites for each server linked to the Portal. As of April 2020, PPA2 is using the “hosting” server.

### Setting up the SDE and using it in script

1. In ArcGIS Pro, pull up the Create Database Connection tool
2. Fill out parameters, ensuring that “database authentication” and “save username/password” boxes are checked.
3. Save the SDE file in the same folder that you keep the data layers in.
4. To use the SDE in script, use the SDE file’s path just like you’d use a file geodatabase path.

# Setting Up Python Environments

## Overview

* Dependencies = python packages that must be installed on the computer and in the environment that runs the tool.
  + List of dependencies is in [Appendix](#_Python_Dependencies)
* All dependencies were managed (installed/removed/updated) using the [conda](https://docs.conda.io/en/latest/) package manager that is included with ArcGIS Pro.
* Dependencies must be installed on all computers on which the PPA2 code will be running. For the 2020 PPA tool development, this required installing dependencies on the following machines:
  + WIN10-MODEL-2, which was used for testing and running local versions of the tool in ArcGIS Pro and through Spyder IDE
  + ARCSERVERGIS-SVR, which was the machine on which the published tool code was executed when the tool was run via Portal (either in web browser or ArcGIS Pro Portal interface).

### Conda environments

[Cheat sheet for working with conda environments](https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf)

ArcGIS Pro has a “default” conda environment called “arcgispro-py3”. If you are a non-admin user you must make a clone of this environment if you want to install or remove packages from it. However, since cloned environments are local (i.e., tied to a specific user account), they cannot be used by the arcgis script service (shown in Windows Task Manager as ArcSOC.exe, or as domain user “arcgis”). Therefore, for the machine that the online tool will run the tool code from, you must make sure that all dependencies are installed in the default environment and not a clone environment.

## Managing Dependencies

### Using Conda

As explained above in the [Conda Environments section](#_Conda_environments), you must use the default environment “arcgispro-py3”. However, you must run as an administrator in order to make any changes to the default environment. So when opening the command prompt to enter conda commands, make sure you open it as an administrator.

### Installing non-conda packages into your conda environment

*NOTE - When possible, use conda for all package management!*

1. Open command prompt as administrator
2. Change directory to C:\Program Files\ArcGIS\Server\framework\runtime\ArcGIS\bin\Python\envs\arcgispro-py3, or wherever Python is kept for Arc products on that server machine.
   1. Or adapt to be the correct environment file path
3. Enter python -m pip install <package name>
   1. Note – all packages must be installed in the default environment because that is the environment used by the arcgis script service (usually named ArcSOC.EXE). Cloned environments are
   2. To see which user is running a script, have script do AddMessage(sys.executable)

## Gotcha issues with xlwings module

*Update 4/3/2020 – At the time of this writing, the PPA tool currently does not have a PDF publishing function available and is thus not using the Xlwings module. However, this section of the documentation is included as reference.*

* Note – xlwings module’s exclusive purpose is to publish Excel version of report as PDF.
* Excel must be installed on any computers executing code that uses the xlwings module.
* Must set xlwings.App.visible = True
  + <https://stackoverflow.com/questions/14037412/cannot-access-excel-file>
* User “arcgis”, which is the impersonation of arcgis processes, must have launch and activation permissions for Excel application, which can be set by:
  1. Opening Component Services > Console Root > Component Services > Computers > My Computer > DCOM Config
  2. Right-click Microsoft Excel Application > Properties > Security
  3. Set Launch and Activation Permissions to ‘customize’
  4. Click on Edit > Add > Advanced > Find Now
  5. Select ‘arcgis arcgis’ user from list of users that appears
  6. Click OK
  7. Give that user full permissions.
* To be safe, also do this:
  1. Find folder that EXCEL.EXE is in
  2. Give full permissions to the ‘arcgis arcgis’ user to folder two levels above the folder in which EXCEL.EXE is stored.
* Another, obscure fix to make is to add an empty folder called “Desktop” to each of these locations:
  + C:\Windows\system32\config\systemprofile
  + C:\Windows\SysWow64\config\systemprofile
  + Reference on [Stack Overflow](https://stackoverflow.com/questions/17177612/excel-access-denied-with-win32-python-pywin32) article (4/3/2020)

**Changing publishing parameters without republishing**

Example task to change: want to set notifications from “error” to “info” to display more messages when running.

1. Go to <https://services.sacog.org/hosting/manager>
2. Select the GP service from Manage Services > Folders > Site
3. In tool page, go to Parameters and edit as needed
4. After finishing, click ‘Save and Restart’

## Potential Issues When Updating ArcGIS Pro

Updating Pro seems to disable or corrupt Conda environments that were previously installed. If you get any errors related to modules not being found:

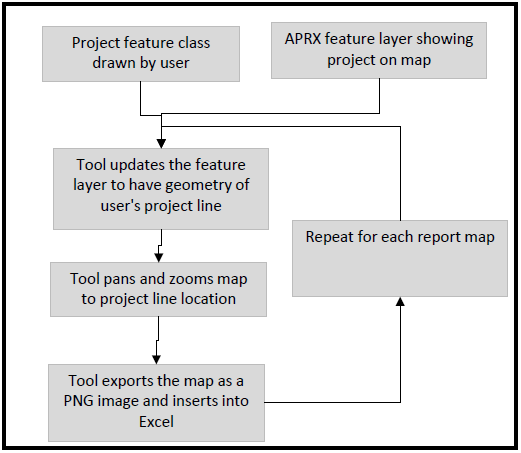
* Make sure you’re using the [default environment](#_Conda_environments) “arcgispro-py3”
* Confirm that all needed dependencies are installed by entering command conda list while in the default environment

# Code Gotchas to be Aware Of

* When you publish code, ArcServer may change certain hard-coded file paths based on what it believes are necessary to ensure all of your data are accessible to the server. However, sometimes this ends up making changes that break your script, and you must go to the published version of your script to manually revert the change.
* Use os.path.join for any and all file path joins. Using any other method can cause a “can’t find feature class” error.
* Avoid storing temporary feature classes in the GIS “memory” space. Instead use arcpy.env.scratchGDB, which is more stable.

# Making and Inserting Maps into Reports

The figure below summarizes how maps get inserted into the output Excel report



* On the server machine that your input data layers are stored on, in the folder registered with Portal, you must have the following:
  + PPA2 data layers
  + APRX project containing all maps and layouts you want to include in report
  + CSV specifying parameters for each map produced
  + CSV specifying which maps go into each report and where within each report

# Publishing the Tool

Tool publishing was done through ArcGIS Pro’s “share as web tool” feature, as detailed in [ESRI’s documentation](https://pro.arcgis.com/en/pro-app/help/analysis/geoprocessing/share-analysis/publishing-web-tools-in-arcgis-pro.htm).

## Things to keep in mind about publishing

When publishing, remember to:

* Share the tool with “everyone”
* Make sure the message level is set to “info”, so all status messages will be displayed when running the tool
* Analyze before publishing to resolve any warnings or errors.

When you publish through Pro, it creates a copy of the python scripts associated with the tool. However, as mentioned in the [Code Gotchas](#_Code_Gotchas_to) section, if you have certain key words or file path strings in your script, the publishing process will make changes to these. Most of the time these changes will not affect how the script runs, but sometimes they can cause errors. If such errors arise, you must go in to the published copy of the script and manually change the value back to what it was. There are examples of this in the [Troubleshooting section](#_Troubleshooting).

## Turning a Web Tool Into A Widget

Widget interfaces have more parameters and functionality than a simple web tool interface and are a better platform to test out published online tools.

1. Make sure you have published the tool you wish to turn into a widget.
2. Open or create a new Web App in WebAppBuilder
3. Open Widget tab > Select Geoprocessing Widget
4. Get Widget URL from Portal (navigate via tabs at top of window)
5. Select the Web Tool you want to use
6. Set Parameters

# Troubleshooting

## Errors During Run

* PermissionError: [Errno 13] Permission denied: ~\\mkl\_fft\\\_\_init\_\_.py
  + *Solution 1:* Make sure that, on the hosting server machine, that the default conda environment, “argispro-py3” is the active environment.
  + *Solution 2:*
    - Ensure that user “arcgis arcgis” has access to the parent folder of the Conda environment being used on the computer that’s hosting the published scripts.
* FileNotFoundError: [Errno 2] No such file or directory: '<scratch Folder>\\<imagefile>.jpg'
  + *Solution 1:* In the published ppa\_input\_params.py script, manually edit to make sure the server\_folder variable points to the correct folder on the data store folder (as of 3/5/2020, it is [\\arcserver-svr\D\PPA\_v2\_SVR](file:///\\arcserver-svr\D\PPA_v2_SVR)). This is an example of the ArcServer making changes to file paths during the publishing process, as described in the [Code Gotchas section](#_Code_Gotchas_to).
    - Change line: *fgdb = os.path.join(server\_folder, g\_ESRI\_variable\_2)* to instead be: fgdb *= os.path.join(server\_folder, 'PPA\_V2.gdb')*
    - May also need to [restart GP service](#_Things_to_turn) for tool in Server Manager website, refresh tool page, restart Pro after making the edit
  + *Solution 2:* Confirm that the Project\_Line\_Template feature class is in the geodatabase that your tool is using—this is needed in order to make the APRX maps that get exported to images. If this fails then you’ll get the “file not found” error when trying to find the map image.
* ModuleNotFoundError: No module named ‘Pillow’
  + *Testing method*: Use the GetVersionSVR tool in the TestingPPA toolbox. This is a quick script to see if a given module is installed.
    - Make sure that Pillow is install in the environment that the tool is being run in.
    - Also try making a quick python script that imports openpyxl module and makes an Image object out of any JPG file. This Image() step is what normally triggers the error. Make sure you’re running the python on the same machine that the tool runs on.
  + *Solution 1:* Happened after installing xlwings. Ensure that you can import it via command line when in default server environment (arcgispro-py3 on E drive)
  + *Solution 3:* Try updating openpyxl
  + *Solution 4:* Try python -m pip install Pillow after navigating to default conda env folder, or can do through normal conda install Pillow if you’re running the cmd as an administrator
  + *Solution 5*: Restart computer/server.
* ERROR 000464: Cannot get exclusive schema lock. Either being edited or in use by another application or service. Failed to execute (Delete).
  + *Solution 1:* Make sure no one has any of the affected layers open on any computers.
  + *Solution 2:* In tool’s Server Manager setting, make sure up to 50 instances of tool can run at once; and that refresh is once per hour
  + *Solution 3*:
    - For all affected feature layers, give the feature layer name a time.clock() suffix, and also create the layer in arcpy.env.scratchGDB to make sure it is unique every time it is made or deleted. E.g., instead of fl = “feature\_layer”, make it fl = os.path.join(arcpy.env.scratchGDB,”feature\_layer{}”.format(int(time.clock())+1)
    - *Solution 4*: This issue primarily arises when the tool is being run by 2 or more people at the same time, or if 2+ run sessions overlap each other at any point. Instead of hosting input data layers in a file geodatabase, put them in an Enterprise geodatabase, which is meant to handle multiple users accessing the same database object at the same time.

### XLWings Errors

*NOTE – As of 4/3/2020, XLWings is not being used in the tool, but these troubleshooting steps are being kept as reference for possible future debugging and implementation efforts. However, per* [*Microsoft’s own developer network*](https://support.microsoft.com/en-us/help/257757/considerations-for-server-side-automation-of-office)*, it is not recommended to open and run Excel through an automated process such as the xlwings/pywin32 module, and Microsoft offers no support nor does it make any guarantees about reliability or performance.*

* No module ‘xlwings’
  + *Solution 1:* Need to confirm which environment the tool is running in and ensure xlwings is installed in that environment. Conda install must be to default env and use conda to install (run as admin if needed).
  + *Solution 2:* Use conda install first. If that doesn’t fix issue, use python -m pip install method
* 'pywintypes.com\_error: (-2147352567, \'Exception occurred.\', (0, \'Microsoft Excel\', "Microsoft Excel cannot access the file <output XLSX file (not template)>…There are several possible reasons:\\n\\n• The file name or path does not exist.\\n• The file is being used by another program.\\n• The workbook you are trying to save has the same name as a currently open workbook.", \'xlmain11.chm\', 0, -2146827284), None)')
  + *Solution 1*: Confirm that ppa utils.py script, xlwings.App.Visible = True
  + *Solution 2*: Confirm in the params.py file that in the dict corresponding performance outcomes to Excel sheet tabs, that the sheet tab referred to in the dictionary has a corresponding tab in the Excel template workbook.
  + *Solution 3:* Double check to make sure none of the template Excel files are currently open on any computers.
  + *Notes*
    - This error recurred and neither of the normal solutions worked. But after editing the script to just add in some messages saying where the XLSX file was, it started working (though the messages showing where the XLSX was did not appear. NOTE this appears to be a more general stability issue.
* pywintypes.com\_error: (-2147024891, 'Access is denied.', None, None)
  + *Solution 1:* Make sure the directory C:\Windows\SysWOW64\config\systemprofile\Desktop exists. You may need to create the ‘Desktop’ folder. Then make sure user ‘arcgis arcgis’ has access to the Desktop folder
    - Also make sure to go through all troubleshooting steps in the “xlwings gotcha” section above. E.g., making Desktop folders, enabling arcgis user access to Excel through Component Services, etc.
  + *Solution 2:* In addition to giving permissions to “arcgis user”, if using Dev Portal, make sure to give permissions to “dconly”. In security settings, navigate to Edit > Add, and in Locations, select the machine name (i.e., not the default “nt-domain.sacog.org”. The machine name should have a user named “WIN10-MODEL-2\dconly”
* pywintypes.com\_error: (-2147352567, 'Exception occurred.', (0, 'Microsoft Excel', 'Open method of Workbooks class failed', 'xlmain11.chm', 0, -2146827284), None
  + *Only seems to affect dev portal installed on same machine used to publish on ArcPro*
  + *Solution 1*: Same checks as for permission-denied error, but also check:
    - Make sure dconly (Local user, not domain userDConly ) has access to all scripts and Excel Template files.
    - Ensure that local user dconly is in same Administrator group as dconly (Search > Edit local users and groups > Local Users and Groups > Users > Select dconly (or whichever local ID online tool users > Properties > Member of > Add, then do normal steps to make sure user is added to ‘Administrators’ group
    - Restart computer
* "pywintypes.com\_error: (-2147352567, 'Exception occurred.', (0, 'Microsoft Excel', 'Microsoft Excel cannot open or save any more documents because there is not enough available memory or disk space. \\n\\n• To make more memory available, close workbooks or programs you no longer need. \\n\\n• To free disk space, delete files you no longer need from the disk you are saving to.',
  + *Solution 1:* None found as of 4/7/2020
* "pywintypes.com\_error: (-2146777998, 'OLE error 0x800ac472', None, None)"
  + *Solution 1:* Make sure that the input template XLSX file is closed on all computers

## Errors After Run Completes

* When trying to download PDF, link takes to “Error code 400 invalid URL” page.
  + *Solution 1:* None found as of 4/7/2020
* Map images are fine in XLSX output, but in PDF outputs they appear as blank rectangles with very tiny message saying “The picture can’t be displayed”
  + Affects every part of PDFs, immediately after being exported from Excel via xlwings
  + Does not produce any error messages.
  + Does not affect desktop
  + Did not always affect online. Started affecting it sometime during week of 3/2-3/6/2020.
  + *Solution 1:* None found as of 4/7/2020. See [disclaimer](#_XLWings_Errors) about issues converting Excel to PDF
* Tool online calculates incorrect distances
  + *Solution* – this is a projection issue. In desktop, all layers were set to EPSG 2226, but basemaps for Portal are by default in EPSG 4326 (WGS84). Jeanie had a way of making Portal basemap use EPSG 2226. Other potential option is to use EPSG 4326 for everything, but if you do, make sure that the line distances are correct using GIS measuring tool or [Daft Logic’s online tool](https://www.daftlogic.com/projects-google-maps-distance-calculator.htm).

### Map Issues

* Heat map layers and vector polygon layers will render when tool is run from Desktop, but not when published to Portal (even when Portal version is run through Pro interface).
  + *Solution 1*: As a workaround:
    - The heat map layers are in the APRX maps as map images, not feature class layers. This prevents them from disappearing from output images (JPEGs/PNGs)
    - Legends are manually built using images for heat color range, rather than having the legend dynamically generated from a feature class.
    - This is not an optimal solution but is stable as of 3/26/2020.
  + ESRI tech support was consulted about this, and during testing of test tool, neither ESRI nor SACOG could replicate the issue. But it persisted on main tool APRX/script.

## Tools to help with debugging

* Remember to run these both on desktop and in Portal to compare environment differences between each machine, if the hosted Portal scripts are in a different machine from the scripts run via Pro Desktop.
* Script tools within the TestingPPA Toolbox
  + GetVersionSVR – gets version and environment info during run time, and user info too. Pretty simple info. Useful information if you’re getting error messages related to not being able to find a specific module. Helps identify whether you installed into correct env and onto correct machine.
  + ListSystemEnvironmentsGP – much more verbose info on environment variables during runtime. Good for confirming who’s running, what conda env, which modules loaded, etc.
  + TestGetLineLen – simple tool that returns the length of a user-drawn line. Used to make sure that the tool is calculating the correct line length.

## Things to turn off and on to effect script changes

If you make an edit to a script used by the tool, the changes may require taking one or more of the steps below in order to take effect:

* Confirm that changes saved to python script
* Close and restart ArcGIS Pro
* Restart tool’s GP Service via Server Manager website
  1. Go to [Server Manager](#_Accessing_Server_Manager) > Services
  2. Open the tool you want to restart
  3. Click ‘save and restart’
* If above fails, check with GIS team before trying:
  + Restarting server service
  + Restarting server machine

# Setting up a separate Development Portal

*NOTE – As of April 2020, this is for reference only. SACOG did not end up using its development Portal much for testing, but the setup process is listed here in case SACOG or any other user of the PPA data decide to make such a single-machine setup in the future.*

## Purpose

* Many issues arise in production because processes are spread on 3 different computers, as described in the [Setup](#_PPA_Setup_on) section.
* Being distributed like this makes testing and debugging challenging because, for any computer on which code is run:
  + Specific permissions to files are needed for the “arcgis arcgis” domain user
  + Specific python packages are needed
  + Conda environment must be set correctly
  + These need to be correct for all computers
* The Dev Portal setup combines these processes onto a single PC.

## Setup Process

1. ESRI Article on [Planning a Base Deployment](https://enterprise.arcgis.com/en/enterprise/latest/install/windows/plan-a-base-deployment.htm)
2. Pete put Enterprise Builder file onto WIN10-MODEL-2 E: drive
3. Open builder file and follow prompts
   1. PRVC Authorization file is available via MyEsri > License ESRI Products
4. Master admin login info:
5. Need secure web certificate for HTTPS to work. Was kind of complicated to set this up and you needed Pete to help with this. Involved IIS (internet information services, searchable through Start Menu)

## Portal user-related info

* Import getpass module in python (is part of standard python library)
* getpass.getuser() value = “dconly” when run from Portal; “DConly” from desktop
* sys.executable() = E:\...ArcSOC.exe when run from Portal; C:\...\ArcGISPro.exe when run from desktop.

## Post-Setup Conda Issues

* Installation of Portal/Enterprise created a new Conda env named arcgispro-py3, with a site-packages path on the E: drive, where Enterprise was installed. However, this name duplicated the existing arcgispro-py3 env that referred to the one from ArcGIS Pro on the C drive.
  + Tools, once published, run in this E drive version of arcgispro-py3 env
* When I changed directory to E drive and did activate arcgispro-py3, it activated the E drive version and the C drive version no longer appeared in conda env list.
* To check if the C drive version of arcgispro-py3 permanently disappeared, I changed directory back to C and did conda env list.
  + Need to figure out how to reclaim it. Cannot remember how this was done.
* Effect of editing system PATH variable:
  + If you set PATH to call the Enterprise conda EXE installation folder (E drive, see below), then it will list the arcgispro-py3 version on E drive.
  + If you have PATH call the C drive conda EXE installation folder…?
  + If you have both folder paths in the PATH variable, does it create a conflict?

### Conda EXE installations

As of April 2020

* Enterprise - E:\ArcGIS\Server\framework\runtime\ArcGIS\bin\Python\Scripts\conda.exe
* ArcGIS Pro - C:\Program Files\ArcGIS\Pro\bin\Python\Scripts\conda.exe

### Conda default envs

As of April 2020

* E:\ArcGIS\Server\framework\runtime\ArcGIS\bin\Python\envs\arcgispro-py3 (default for Enterprise)
* C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3 (default for ArcGIS Pro)

## Debugging issues

* On published scripts, if you add a variable to a published script (e.g. add a variable to the params.py file), that update may not be recognized until you restart the server.

# Appendices

## Python Dependencies

The table below lists which python packages (dependencies) are needed that do not come with the Python Standard Library but are needed to run the PPA tool.

|  |  |
| --- | --- |
| **Package Name** | **Description** |
| Pandas | Data analysis package for various types of database and spreadsheet analyses |
| XLWings | Allows python processes to manipulate Excel workbook files using pywin32 python-windows interface |
| Openpyxl | Allows reading and writing content to/from Excel files |
| Arcpy | Does all spatial and GIS-related python tasks. ESRI proprietary library. Requires license form ESRI. |
|  |  |
|  |  |

## GitHub Source Code Repository

URL - <https://github.com/djconly85/PPA2_0_code>

1. Refer to Project Performance Tool Documentation for further information on how the mix index, a.k.a. the Land Use Diversity Index, is calculated for each parcel. [↑](#footnote-ref-2)
2. Net acres = developable acres within hex that are on parcels. Excludes areas within the parcel that is over water, on public rights-of-way, etc. [↑](#footnote-ref-3)
3. Sundquist et al. 2017, *Accessibility in Practice: A Guide for Transportation and Land Use Decision Making*. <https://www.ssti.us/wp/wp-content/uploads/2018/01/Accessibility_Guide_Final.pdf> [↑](#footnote-ref-4)