

Update contents of the media folder for an ArcGIS Survey123 form item

[Try it live](#) [View source on GitHub](#)

This notebook uses the ArcGIS API for Python. For more information, see the [ArcGIS API for Python documentation and guides](#).

The media folder for a survey contains items used in various ArcGIS Survey123 questions and workflows. A survey's media folder can contain the following:

- Offline basemaps for use in the ArcGIS Survey123 field app.
- CSV files for use in `pulldata()` functions or for use as choice lists.
- Image and audio files for use in various question types.

This notebook demonstrates how you can automate the task of updating the contents of the media folder for a published survey without having to republish it in Survey123 Connect. This is useful when the items in the media folder need to be updated regularly.

IMPORTANT: Care should be taken when running this script to ensure the files you are updating have the exact name as the files currently in the survey's media folder. Also, the files must be in the same format. In the case of CSV files, the updated files must have the same column names as the original files; only the data rows should be updated or additional rows appended. If you want to rename or add columns you should update the files in the survey's media folder in ArcGIS Survey123 Connect, update the XLSForm accordingly, and then republish the survey.

It is recommended that you test the script on a copy of your survey prior to running it on your production survey. Ensure the test survey can be downloaded and updated in the Survey123 field app, checking that the external choice lists and any other media items are working as expected, before proceeding with your production survey.

The first step is to import the required modules and define our variables. The variables are as follows:

- **portalURL** - The URL for your ArcGIS organization (e.g. `www.arcgis.com`)
- **username** - Your ArcGIS organization username (e.g. `gisadmin`)
- **password** - Your ArcGIS organization password (e.g. `gisadmin1`)

- **itemID** - The item ID for the ArcGIS Survey123 form item in your ArcGIS organization (e.g. 89bc8c7844e548e09baa3aad4695e78b)
- **updated_files** - The updated file name containing the extension (e.g. myphoto.png)
- **source_loc** - Folder directory where the updated file is located (e.g. C:/Users/username/ArcGIS/My Survey Designs/...)

```
In [ ]: import arcgis
        from arcgis.gis import GIS
        import tempfile
        import zipfile
        import shutil
        import os

        portalURL = r'https://www.arcgis.com'
        username = ''           # Your Username
        # password = ''        # Your Password. Not necessary when using GIS("home")
        itemID = ''           # ItemID of your survey
        # Add one or more file names to the list.
        updated_files = ['']   # File name of the xml file for your survey
        source_loc = r''       # Local file path to survey, e.g. C:\Users\yourname\ArcGIS\My Survey Designs\MySurvey
```

Uncomment the first GIS connection if using the `portalURL`, `username`, and `password` variables from above.

```
In [ ]: # Connect to GIS
        gis = GIS(portalURL, username)
        # gis = GIS("home")
```

Download survey

Start by obtaining the properties of the Survey123 form item. These properties are used to reconstruct the ZIP file name that is uploaded. The first line in the code block below defines a Survey Manager. A survey in the Survey Manager is a single instance of a survey project that contains the item information and properties and provides access to the underlying survey dataset. For more information on Survey Manager, see the [API Reference for the ArcGIS API for Python](#).

```
In [ ]: survey_manager = arcgis.apps.survey123.SurveyManager(gis)
        surveyId = survey_manager.get(itemID)
        surveyProp = surveyId.properties
        print(surveyProp)
```

Find the form item in the `gis` and download as a ZIP file to a temporary folder directory. Additional information on downloading content can be found [here](#).

```
In [ ]: itm = arcgis.gis.Item(gis,itemID)
        print(itm)
        tmpdir = tempfile.TemporaryDirectory()
        download_folder = tmpdir.name
        savedZip = itm.download(save_path=download_folder)
```

Extract the ZIP file to an `_extracted` folder in the download location. This `_extracted` folder is where the updated media files will be copied and zipped later on.

```
In [ ]: def extractZIP(filename, folder):
        zfile = zipfile.ZipFile(filename)
        zfile.extractall(folder)

        extractZIP(savedZip, os.path.join(download_folder + "/_extracted/"))
```

Copy the updated file to the media folder, replacing the old file.

```
In [ ]: for file in updated_files:
        source_file = os.path.join(source_loc, file)
        dest_file = download_folder + "/_extracted/esriinfo/" + file
        shutil.copyfile(source_file, dest_file)
        print (file, " updated to: ", download_folder + "/_extracted/esriinfo/")
```

Delete the ZIP file you downloaded previously. This will prevent any namespace issues and ensure the process of zipping and uploading the updated survey goes smoothly.

```
In [ ]: os.remove(savedZip)
        print ("Old ZIP file deleted from: " + download_folder)
```

Upload updated survey

Zip the updated survey and place it in the download folder you defined previously. "The code below extracts the survey title from the survey properties and passes it into the file name for the ZIP file.

```
In [ ]: zipFileName = surveyProp['title']
updateZip = shutil.make_archive(zipFileName, 'zip', download_folder + '/_extracted/')
print (updateZip)
```

Upload the new ZIP file and update the form item with the new media folder content. For more information on the update property of items please visit [this link](#).

```
In [ ]: itm.update({},updateZip)
```

Clean up the intermediate data. This process deletes the updated ZIP file as well as the extracted folder containing the unzipped survey content.

```
In [ ]: tmpdir.cleanup()
print ("Temp folder deleted from: " + download_folder)
# print (zipFileName + " successfully updated with " + source_file + " and uploaded to your organization!")
print (zipFileName + " successfully updated with " + ' & '.join(map(str, updated_files)) + " and uploaded to your organ:
```

```
In [ ]:
```