Geocom Python Framework Documentation

Release 0.9.7.dev5+g3d73906

Geocom Informatik AG / VertiGIS

Feb 28, 2020

Contents:

CHAPTER 1

gpf package

1.1 Subpackages

1.1.1 gpf.common package

1.1.1.1 Submodules

1.1.1.1.1 gpf.common.const module

Module with global constants that are used throughout the *gpf* package.

`gpf.common.const.`**`ENC_DEFAULT = 'UTF-8'`**
The default encoding that the system uses (derived from locale). For most western Windows-based systems, this will be cp1252, for example.

1.1.1.1.2 gpf.common.guids module

This module contains the *Guid* class, which inherits from Pythons built-in `UUID` class. It helps validating existing GUIDs (e.g. GlobalID's) and can generate new ones. It also helps formatting the GUID for use in SQL queries.

`class` `gpf.common.guids.`**`Guid`**(*value=None*, *allow_new=False*)
Bases: `uuid.UUID`

Takes a UUID-like string or object as input and validates it. Can also be used to generate a new GUID. The *Guid* class inherits from the Python built-in UUID.

Params:

- **value** (object):

  The value to parse as a GUID. Must be set if *allow_new* is `True`.

- **allow_new** (bool):

If set to `True` and *value* is `None`, a new GUID will be generated. The default is `False`, which means that an exception will be raised if *value* is not set.

Raises

- **gpf.tools.Guid.MissingGuidError** – This exception is raised when *allow_new* is `False` and *value* is `None`.
- **gpf.tools.Guid.BadGuidError** – This exception is raised when *value* cannot be parsed to a GUID.

Examples:

```
>>> Guid(allow_new=True)
Guid('459b46ce-6370-48ae-b3cc-220026d49ec2')
>>> guid = Guid('{459b46ce-6370-48ae-b3cc-220026d49ec2}')
>>> str(guid)  # this returns the GUID for Esri SQL expressions
'{459B46CE-6370-48AE-B3CC-220026D49EC2}'
```

exception MissingGuidError

Bases: `exceptions.TypeError`

This exception is raised when `Guid` is initialized without a *value*, while *allow_new* is `False`, which is the default. Either set a *value* or set *allow_new* to `True` to prevent this error.

exception BadGuidError

Bases: `exceptions.ValueError`

This exception is raised when the GUID string cannot be successfully parsed to a valid UUID-like object.

1.1.1.1.3 gpf.common.iterutils module

1.1.1.1.4 gpf.common.textutils module

1.1.1.1.5 gpf.common.validate module

1.1.1.2 Module contents

The *common* subpackage contains several helpful multi-purpose modules, classes and functions, that are not necessarily GIS-related (for GIS tools, see the *gpf.tools* subpackage). Think evaluation, text formatting, file path handling and so on.

The functions in this subpackage are used by other *gpf* subpackages, but can also be called in user scripts.

1.1.2 gpf.tools package

1.1.2.1 Submodules

1.1.2.1.1 gpf.tools.fieldutils module

1.1.2.1.2 gpf.tools.geometry module

1.1.2.1.3 gpf.tools.maputils module

1.1.2.1.4 gpf.tools.metadata module

1.1.2.1.5 gpf.tools.queries module

1.1.2.2 Module contents

The *tools* subpackage contains a set of general classes and functions that should make it a little easier to work with ArcGIS and `arcpy`.

Some classes are wrappers for well-known `arcpy` classes, created for a more user-friendly experience and/or better performance.

1.2 Submodules

1.2.1 gpf.paths module

1.2.2 gpf.cursors module

1.2.3 gpf.lookups module

1.2.4 gpf.loggers module

1.3 Module contents

This is the documentation for the **gpf** (*Geocom Python Framework*) package for **Python 2.7**. For the Python 3 version (suitable for ArcGIS Pro), please refer to the **gpf3** package.

The *gpf* package contains several subpackages with tools and helpers for all kinds of ArcGIS-related geoprocessing and data management tasks. It is released under the Apache License 2.0 as an open-source product, allowing the community to freely use it, improve it and possibly add new features.

Several tools in this package require Esri's `arcpy` Python library, which does not make this a *free* package. However, users who have already installed and authorized ArcGIS Desktop (ArcMap, ArcCatalog etc.) should be able to work with this package without any problems.

Note: It is recommended to import `arcpy` via the `gpf` package (`from gpf import arcpy`). This will load the same (and unmodified) module as `import arcpy` would load, but it shows more useful error messages when the import fails.

CHAPTER 2

Indices and tables

- genindex
- modindex

Python Module Index

g

Index

E

G