GeoSAMS GUI

Thomas Callaghan
Version 0.1

Contents

1 GeoSAMS GUI

This is the main program for the GeoSAMS GUI

The GUI has 7 tabs

1. **Math Setup** : This frame allows the user to modify the Matlab/Octave startup files.
2. **Main** : Data concerning simulation duration, configutation files in use, and recruitment period
3. **Growth** : Define parameters to compute fishing mortality
4. **Special Access** : Files used to define special areas for fishing management
5. **Fishing Mort in Special Access** : This frame in conjunction with the Special Access Frame is used to define fishing mortalities within a defined area for a specified year. If a location falls within the defined area given by the area defintions in Special Access Frame and assigned the area number.
6. **Sort By Area** : Parameters that are used to sort output data and associate with areas of interest
7. **Sort By Region** : Parameters that are used to sort output data and associate with zones defined by shapefiles

The program is started by entering the following command in the root directory of the workspace
$ python .\PythonScripts\GUI\GeoSAM**GeoSams.py** [10 8]

Where the last two number are optional and used to set limits on:

- The maximum number of areas of interest that can be defined.
- The maximum number of nodes used to specify each area of interest.

When commanded without these values the GUI defaults to 50 and 8. These values can be viewed by clicking the **SHOW Args** Button

1.1 SHOW Args

As already mention this button is used to show the setup parameters that the GUI is using for maximum number of areas, nodes, and years

1.2 START Sim

This button will start both the GeoSAMS sim and if succsessful continue with the UK interpolation. It does so by first saving the data contained in the other tabs of the GUI to configuraton files specified on this page. It will overwrite the files named if they already exist.

NOTE: The file names listed are part of the package installed when downloaded from GitHub. The user may change these names to preserve the original files. Or reinstall from GitHub to restore the original data.

1.3 SAVE ALL Configs

This is the first step in ***START Sim*** . This button will save all of the configuration files using the names given.

1.4 Year in file names

GUI specifies 2022 to 2025

- X_Y_BIOM_2022_DN Initial state as of June 1, 2022 @ 00:00, i.e. May 31, 2022 @ 24:00
- X_Y_BIOM_2023_DN Growth state as of May 31, 2023 @ 24:00, results for 1st year growth
- X_Y_BIOM_2024_DN Growth state as of May 31, 2024 @ 24:00, results for 2nd year growth

- X_Y_BIOM_2025_DN Growth state as of May 31, 2025 @ 24:00, results for 3rd year growth
- X_Y_BIOM_2026_DN Growth state as of May 31, 2026 @ 24:00, results for 4th year growth

2 Math Setup Frame

Matlab should not need any modification as these are the installed directories.

The user should not need to run any Matlab scripts as these are called from the GUI.

Octave on the other hand does require some setup. The user will need to install the desired packages from `https://gnu-octave.github.io/packages/`

- io
- geometry
- mapping
- statistics

2.1 Radio Button

The user then needs to modify .octaverc to point to where these are installed. Click the Octave radio button and edit the text box to reflect your environment. Then click Write Startup File.

2.2 Write Startup File

When ready, click Write Startup File to save the content to the appropriate startup file

- startup.m
- .octaverc

Note 1: on a Unix/MacOS platform it is assumed that Matlab is not installed. The script Unpack.sh renames startup.m to startup.xxx so it is not taken for .octaverc

Note 2: If data intialization files have not yet been created, Upack.sh is called to do so.

3 Set Fishing Mortality in Special Access Areas

Assists the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

3.1 Number Defined

The number of defined areas as determined by the user. This is limited by Max Areas of Interest. See SHOW Args button for current values.

The Number Defined is limited by default to 25. See SHOW Args for current values. The user can modify this on the command line:

.\PythonScripts\GUI\GeoSAM**GeoSams.py** Areas Nodes

Default:

.\PythonScripts\GUI\GeoSAM**GeoSams.py** 25 8

3.2 Load and Save Fishing Mortality Files

The name of the file used to hold this information. The user can load the default file 'FishingMortality.csv' or save their own configuration.

If this feature is not desired then enter NONE in the window

Use Load Fishing Mort File to load a predefined set of data

Use Save Fishing Mort File to save the currently displayed setting

3.3 Area SubFrames

Comment: Optional. Enter a comment to describe the area being specfied.

3.3.1 Year Definitions

The year for which Area N is valid

3.3.2 Corners, or Fields of Defined Mortalities by Area

3.3.2.1 Number of Corners

Corners: Specifically, the number of Fields for the year given. This is limited by Max Nodes in Area. See SHOW Args for current values. This can be changed on the command line. See above

3.3.2.2 Field Identifier by Special Area and Mortality

These are the area numbers as determined in Special Access Frame. Enter the area number and its Mortality.

4 Growth Frame

Allows the user to modify parameters that are used to define mortality computations.

4.1 Mortality

4.1.1 Fishing Mortality

This is the default fishing mortality in lieu of any other definition

4.1.2 Alpha Mortality

So for open areas, an overall fishing mortality F_{avg} would be specified and then F at each location would be computed so that:

1. The weighted average (by exploitable numbers), F, over all locations is equal to F_{avg} and
2. F at each location is proportional to $LPUE^{alpha_{mort}}$.

This would also apply to special access areas, but each one would have their own specified F, and the average would only be for those points within that access area.

$$f_{avg} = \frac{\vec{scallops}_{num} \cdot \vec{F_{mort}}_{raw}}{sum(\vec{scallops}_{num})}$$

4.1.3 Adult Mortality

Table 1 Mortality

	Adult	$Length_0$
MA	**0.25**	**65.0**
GB	**0.2**	**70.0**

4.1.4 Computing Alpha

Alpha is based on the lengths of the shell normalized by $length_0$

$$\vec{alpha} = 1 - \frac{1}{1 + exp\left(-(\vec{length}_{shell} - length_0)/10\right))}$$

4.1.5 Computing Natural Mortality

Then natural mortality is computed from juvenile natural mortality and adult natural mortality as

$$\vec{mort}_{nat} = \vec{alpha} * mort_{nat_{juv}} + \left(1 - \vec{alpha}\right) * mort_{nat_{adult}}$$

4.2 Selectivity

These parameters are used to compute the scallop selectivity as a function of its length. MA and GB have respective values for each term. GB will also distinguish between open and closed areas.

$$selectivity = 1/(1 + exp(select_a - select_b * (l_{shell} + 2.5)))$$

Table 2 Selectivity

	MA	GB Open	GB Closed
FSelect A	**20.5079**	**17.72**	**21.7345**
FSelect B	**0.19845**	**0.15795**	**0.2193**

4.3 Incidental

Table 3 Incidental

MA	**0.05**
GB	**0.1**

4.4 Discard

Discard determines how many scallops are thrown out of a catch. It is determined by scallop length and if the area is closed.

```
if ((length > cull_size) OR is_closed) then
    SetDiscard = 0.0
else
    SetDiscard = discard * selectivity
```

Table 4 Discard

	Cull Size	Discard
MA	**90.0**	**0.2**
GB	**100.0**	**0.2**

4.5 Overall Mortality, M

$$M = natural_{mortality} + Fishing_{effort} * (selectivity + incidental + discard))$$

4.6 Computing Landings Per Unit Effort, LPUE

The simulation uses the following parameters to compute LPUE

Table 5 LPUE

	Default
LPUE Slope	**0.6556**
LPUE Slope2	**2.3**
LPUE Intercept	**1094.0**
Max # of Scallops Shucked Per Day	**56000.0**
Max # of Hours Dredging Per Day	**19.0**
Dredge Width in meters	**9.144**
Towing Speed in knots	**4.8**

$$W_{expl} = \frac{EBMS}{N_{scallops}}, \text{ weight in grams}$$

$$EBMS_{tow} = EBMS * Tow_{sqm}, \text{ biomass in grams}$$

$$slope_1 = lpue_{slope} * EBMS_{tow} + lpue_{intercept}$$

$$slope_2 = LPUE_{slope_2} * EBMS_{tow}$$

$$LPUE_{limit} = max_{per_{day}} * W_{expl}/453.592$$

$$LPUE = min(slope_1, slope_2, LPUE_{limit})$$

5 Main

5.1 Growth subframe that identifies

These are the parameters used to control how long the scallop growth is simulated as well as the granularity of the growth computations

- Start Year of the simulation
- Stop Year of the simulation
- Time Steps per year
- Domain Name or region of interest, Mid-Atlantic, MA, or Georges Bannk, GB
- Sort By Statum: Used when processing Georges-Bank to break the region into quadrants due to its unique shape

5.2 Recuitment

Recruitment is only applied at a certain time of the year. These values determine this period. Combo boxes are used to format the formatting of the month and day.

- Start Day, calendar day of the year when recruitment influence begins.
- Stop Day, calendar day of the year when reruitment influence ends

5.3 Configuration Files

Files used by the sim to set up parameters. The GUI can use the default values or change the names before starting the sim. The initial names are the default names of the files when first downloaded from GitHub. The names can be changed and the GUI sets up the simulation to use the new names.

5.4 Output Selection

Checkboxes to allow the user to select the desired parameters of interest. This is used to save processing time rather than processing everything. Especially true during interpolation as it would take over and hour to do the interpolation. For example for MA with 11631 grid locations.

- Approx 2 minutes per interpolation
- Given 3 years worth of data, plus initial conditions
- 9 listed outputs

Thus 9 x 4 x 2 or 72 minutes. GB is proportionately shorter with only 6802 grid locations.

6 Sort By Area Frame

Assists the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

6.1 Number of Areas

The number of defined areas. This is limited by Max Areas of Interest. See SHOW Args button The # of Areas is limited by default to 25. See SHOW Args for current values. The user can modify this on the command line:

.\PythonScripts\GUI\GeoSAM**GeoSams.py** Areas Nodes

Default (same as started with no arguments):

.\PythonScripts\GUI\GeoSAM**GeoSams.py** 25 8

6.2 Output Parameters

This is a dropbox of the selected output parameters on the main tab. After a simulation and interpolation have been run, the user would select one of these output, click Run Sort, and the amount of that output in each of the defined areas is accumulated by year to the left of each area.

6.3 Load and Save Data Sort Files

These buttons allow the user to load a predefined set of areas or to save the current set to the named file.

6.4 Run Sort

This will start the program to check if a region grid value for a given year is within one of the specified area and if so accumulate the year sum with that value.

6.5 Area SubFrames

6.5.1 YYYY

For each year, from Start Year to Stop Year as given in the Main tab an entry box is provided to store the accumulated parameter for that year. These are not populated until after the Run Sort button has been clicked.

6.5.2 Comment

Optional. Enter a comment to describe the area being specfied.

6.5.3 # Corners

Also called nodes or sides. This is limited by Max Nodes in Area. See SHOW Args for current values. This can be changed on the command line. See above

6.5.4 Corner N

These are the coordinates of the area vertices. Enter the Longitude and Latitude of the vertices for the area. It is up to the user to ensure that a closed shape is defined.

7 Sort By Region Frame

Assists the user in viewing accumulated parameters located in zones defined by shapefiles.

7.1 Output Parameters

This is a dropbox of the selected output parameters on the main tab. After a simulation and interpolation have been run, the user would select one of these output, click Run Sort, and the amount of that output in each of the shapefile regions is accumulated by year.

7.2 Run Sort

This will start the program to check if a region grid value for a given year is within one of the specified area and if so accumulate the year sum with that value.

8 Model Buffer, SortIntoColumns

For a given point, check each of the polygons in a given shape file.

If in that polygon set BUFFER_<shape> to 1.

The region and buffer shapefiles for GB and MA are attached. **Column** "Region" for all shapefiles is the modeling region, which makes columns "REGION" and "BUFFER_REGION" in the csvs.

Buffer is larger than the modeling region, so some data outside the modeling region but close to the boundary could be within the buffer and used to construct the models. Some data will be used several times for different models if multiple modeling regions are nearby. For example, within the blue circle in the below MA plot are a few data points outside SAMS but within the buffer, which will be used by two modeling regions. For this reason, created multiple BUFFER_REGION columns for the survey data X_Y_BIOM_AL2022_XX_BUFFER.csv. The output from the R code, X_Y_BIOM_AL2022_XX_BUFFER_GAM.csv, has a longer length than the input X_Y_BIOM_AL2022_XX_BUFFER.csv because some data were used several times, and to make the query in the Python part easier I just duplicated some of the data. **Column** "BUFFER" in X_Y_BIOM_AL2022_XX_BUFFER_GAM.csv indicates which modeling region's buffer the data and GAM estimates/residuals belong to.

9 Special Access Area

This frame in conjunction with the FishingMort in Special Access frame is used to define fishing mortalities within a defined area for a specified year.

If a sim data point falls within a defined area given in this frame by the assigned area number. Then if the current year is the same as the year given in the FishingMort in Special Access frame and the area number is listed then the fishing mortality is specified by the Mortality value. Otherwise it is the default value which is defined in the Growth Frame as Fishing mortality #

9.1 Number of Areas

The number of areas the user wishes to define. This is limited by Max Areas of Interest. See SHOW Args button

The # of Areas is limited by default to 25. See SHOW Args. The user can modify this on the command line:

.\PythonScripts\GUI\GeoSAM**GeoSams.py** #Areas #Nodes Default: python
.\PythonScripts\GUI\GeoSAM**GeoSams.py** 25 8

9.2 Special Access File

The name of the file used to hold this information. The user can load the default file 'SpecialAreas.csv' or define and save their own configuration.

If this feature is not desired then enter NONE in the window

Use Load Special Area File to load a predefined set of data

Use Save Special Area File to save the currently displayed setting

9.3 Area Definitions

9.3.1 Area N

- Comment: Optional. Enter a comment to describe the area being specfied.

9.3.2 Corners:

Also called nodes or sides. This is limited by Max Nodes in Area. See SHOW Args for current values. This can be changed on the command line. See above

9.3.3 Corner N

These are the coordinates of the area vertices. Enter the Longitude and Latitude of the vertices for the area. It is up to the user to ensure that a closed shape is defined.

10 Namespace Index

10.1 Package List

Here are the packages with brief descriptions (if available):

11 Hierarchical Index

11.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

12 Class Index

12.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

13 File Index

13.1 File List

Here is a list of all files with brief descriptions:

14 Namespace Documentation

14.1 AreaManager Namespace Reference

14.1.1 Classes

class **AreaManager***This class is used to paint area grouped by.*

- class **AreaMgrSubFrame**class **Corner**
  Defines floating point data for corner defintions.

14.2 EditMathSetupFrame Namespace Reference

14.2.1 Classes

class **EditMathSetup***This class allows the user to edit the Matlab/Octave setup files to fit their environment.*

14.3 FishMortBySpecAcc Namespace Reference

14.3.1 Classes

class **FishMortBySpecAcc***This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.*

14.4 GeoSams Namespace Reference

14.4.1 Classes

class **MainApplication***This class is the parent class for the GUI.*

14.4.2 Functions

- **ComputeResiduals** (obsFile, gridFile, procID, retDict)
- **main** ()

14.4.3 Function Documentation

14.4.3.1 GeoSams.ComputeResiduals (*obsFile*, *gridFile*, *procID*, *retDict*)

14.4.3.2 GeoSams.main ()

14.5 Globals Namespace Reference

14.5.1 Functions

- **DetermineUnitsScale** (desiredParam)
- **UpdateEntry** (entry, val)
- **ShowMessage** (heading, message, type='info', timeout=2500)
  This method will display the message and then go away after the default time.

14.5.2 Variables

- str **analDir** = 'Analysis'
- str **configDir** = 'Configuration'
- str **dataDir** = 'Data'
- str **gridDir** = 'Grids'
- str **interCfgDir** = 'Interpolation'
- str **resultsDir** = 'Results'
- str **shapeFileDir** = 'Shapefiles'
- str **simCfgDir** = 'Simulation'
- str **specAccCfgDir** = 'SpecialAccess'
- str **surveyDataDir** = 'OriginalData'
- list **comboTFStr** = ['T', 'F']
- list **cornerLabelArr** = ['Corner', 'Long', 'Lat ', '0.0', '0.0']
- int **frameWidth** = 400
- int **frameHeight** = 200

- int **scrollFrameHeight** = 600
- int **helpXoffset** = 700
- int **helpYoffset** = 50
- int **meters_per_naut_mile** = 1852
- int **grid_area_sqm** = **meters_per_naut_mile****2
- str **ABUN** = 'ABUN_'
- str **BIOM** = 'BIOM_'
- str **EBMS** = 'EBMS_'
- str **FEFF** = 'FEFF_'
- str **FMOR** = 'FMOR_'
- str **LAND** = 'LAND_'
- str **LNDW** = 'LNDW_'
- str **LPUE** = 'LPUE_'
- str **RECR** = 'RECR_'
- int **scrollFrameWidth** = 900
- str **geometryStr** = '920x725+10+10'

14.5.3 Function Documentation

14.5.3.1 Globals.DetermineUnitsScale (*desiredParam*)

14.5.3.2 Globals.ShowMessage (*heading*, *message*, *type* = `'info'`, *timeout* = `2500`)

This method will display the message and then go away after the default time.

14.5.3.3 Globals.UpdateEntry (*entry*, *val*)

14.5.4 Variable Documentation

14.5.4.1 str Globals.ABUN = 'ABUN_'

14.5.4.2 str Globals.analDir = 'Analysis'

14.5.4.3 str Globals.BIOM = 'BIOM_'

14.5.4.4 list Globals.comboTFStr = ['T', 'F']

14.5.4.5 str Globals.configDir = 'Configuration'

14.5.4.6 list Globals.cornerLabelArr = ['Corner', 'Long', 'Lat ', '0.0', '0.0']

14.5.4.7 str Globals.dataDir = 'Data'

14.5.4.8 str Globals.EBMS = 'EBMS_'

14.5.4.9 str Globals.FEFF = 'FEFF_'

14.5.4.10 str Globals.FMOR = 'FMOR_'

14.5.4.11 int Globals.frameHeight = 200

14.5.4.12 int Globals.frameWidth = 400

14.5.4.13 str Globals.geometryStr = '920x725+10+10'

14.5.4.14 int Globals.grid_area_sqm = meters_per_naut_mile**2

14.5.4.15 str Globals.gridDir = 'Grids'

14.5.4.16 int Globals.helpXoffset = 700

14.5.4.17 int Globals.helpYoffset = 50

14.5.4.18 str Globals.interCfgDir = 'Interpolation'

14.5.4.19 str Globals.LAND = 'LAND_'

14.5.4.20 str Globals.LNDW = 'LNDW_'

14.5.4.21 str Globals.LPUE = 'LPUE_'

14.5.4.22 int Globals.meters_per_naut_mile = 1852

14.5.4.23 str Globals.RECR = 'RECR_'

14.5.4.24 str Globals.resultsDir = 'Results'

14.5.4.25 int Globals.scrollFrameHeight = 600

14.5.4.26 int Globals.scrollFrameWidth = 900

14.5.4.27 str Globals.shapeFileDir = 'Shapefiles'

14.5.4.28 str Globals.simCfgDir = 'Simulation'

14.5.4.29 str Globals.specAccCfgDir = 'SpecialAccess'

14.5.4.30 str Globals.surveyDataDir = 'OriginalData'

14.6 GrowthFrame Namespace Reference

14.6.1 Classes

class **Growth***This class allows the user to adjust parameters used in computing scallop growth.*

14.7 MainInputFrame Namespace Reference

14.7.1 Classes

class **MainInput***This class displays information about GeoSAMS simulation.*

14.8 PointInPolygon Namespace Reference

14.8.1 Functions

- **PointInPolygon** (polyX, polyY, x, y, nodes)

14.8.2 Function Documentation

14.8.2.1 PointInPolygon.PointInPolygon (*polyX*, *polyY*, *x*, *y*, *nodes*)

```
@param float polyX Array of horizontal, coordinates of corners
@param float polyY Array of vertical    coordinates of corners
@param float x horizontal coordinate of point we wish to determine if inside polygram
@param float y vertical   coordinate of point we wish to determine if inside polygram
@param int   nodes the number of corners, edges, that define the polygon
@returns true if point is inside polygram or if on vert or horiz edge,
         false if outside
         if point is on rise of falling edge then it may return true or false
```

14.9 shapefile Namespace Reference

14.9.1 Classes

- class **_Array**class **_Record**

- class **Reader**
- class **Shape**
- class **ShapefileException**
- class **ShapeRecord**
- class **ShapeRecords**
- class **Shapes**
- class **Writer**

14.9.2 Functions

- **b** (v, encoding='utf-8', encodingErrors='strict')
- **u** (v, encoding='utf-8', encodingErrors='strict')
- **is_string** (v)
- **pathlike_obj** (path)
- **signed_area** (coords, fast=False)
- **is_cw** (coords)
- **rewind** (coords)
- **ring_bbox** (coords)
- **bbox_overlap** (bbox1, bbox2)
- **bbox_contains** (bbox1, bbox2)
- **ring_contains_point** (coords, p)
- **ring_sample** (coords, ccw=False)
- **ring_contains_ring** (coords1, coords2)
- **organize_polygon_rings** (rings, return_errors=None)
- **test** (**kwargs)

14.9.3 Variables

- str **__version__** = "2.3.1"
- **logger** = logging.getLogger(__name__)
- bool **VERBOSE** = True
- int **NULL** = 0
- int **POINT** = 1
- int **POLYLINE** = 3
- int **POLYGON** = 5
- int **MULTIPOINT** = 8
- int **POINTZ** = 11
- int **POLYLINEZ** = 13
- int **POLYGONZ** = 15
- int **MULTIPOINTZ** = 18
- int **POINTM** = 21
- int **POLYLINEM** = 23
- int **POLYGONM** = 25
- int **MULTIPOINTM** = 28
- int **MULTIPATCH** = 31
- dict **SHAPETYPE_LOOKUP**
- int **TRIANGLE_STRIP** = 0
- int **TRIANGLE_FAN** = 1
- int **OUTER_RING** = 2
- int **INNER_RING** = 3
- int **FIRST_RING** = 4
- int **RING** = 5
- dict **PARTTYPE_LOOKUP**
- int **PYTHON3** = 3

- **xrange** = range
- **izip** = zip
- list **MISSING** = [None,'']
- int **NODATA** = -10e38
- **failure_count** = **test**()

14.9.4 Detailed Description

```
shapefile.py
Provides read and write support for ESRI Shapefiles.
authors: jlawhead<at>geospatialpython.com
maintainer: karim.bahgat.norway<at>gmail.com
Compatible with Python versions 2.7-3.x
```

14.9.5 Function Documentation

14.9.5.1 shapefile.b (*v*, *encoding* = `'utf-8'`, *encodingErrors* = `'strict'`)

14.9.5.2 shapefile.bbox_contains (*bbox1*, *bbox2*)

```
Tests whether bbox1 fully contains bbox2, returning a boolean
```

14.9.5.3 shapefile.bbox_overlap (*bbox1*, *bbox2*)

```
Tests whether two bounding boxes overlap, returning a boolean
```

14.9.5.4 shapefile.is_cw (*coords*)

```
Returns True if a polygon ring has clockwise orientation, determined
by a negatively signed area.
```

14.9.5.5 shapefile.is_string (*v*)

14.9.5.6 shapefile.organize_polygon_rings (*rings*, *return_errors* = `None`)

```
Organize a list of coordinate rings into one or more polygons with holes.
Returns a list of polygons, where each polygon is composed of a single exterior
ring, and one or more interior holes. If a return_errors dict is provided (optional),
any errors encountered will be added to it.

Rings must be closed, and cannot intersect each other (non-self-intersecting polygon).
Rings are determined as exteriors if they run in clockwise direction, or interior
holes if they run in counter-clockwise direction. This method is used to construct
GeoJSON (multi)polygons from the shapefile polygon shape type, which does not
explicitly store the structure of the polygons beyond exterior/interior ring
orientation.
```

14.9.5.7 shapefile.pathlike_obj (*path*)

14.9.5.8 shapefile.rewind (*coords*)

```
Returns the input coords in reversed order.
```

14.9.5.9 shapefile.ring_bbox (*coords*)

```
Calculates and returns the bounding box of a ring.
```

14.9.5.10 shapefile.ring_contains_point (*coords*, *p*)

```
Fast point-in-polygon crossings algorithm, MacMartin optimization.

Adapted from code by Eric Haynes
http://www.realtimerendering.com/resources/GraphicsGems//gemsiv/ptpoly_haines/ptinpoly
.c

Original description:
    Shoot a test ray along +X axis.  The strategy, from MacMartin, is to
    compare vertex Y values to the testing point's Y and quickly discard
    edges which are entirely to one side of the test ray.
```

14.9.5.11 shapefile.ring_contains_ring (*coords1*, *coords2*)

```
Returns True if all vertexes in coords2 are fully inside coords1.
```

14.9.5.12 shapefile.ring_sample (*coords*, *ccw* = `False`)

```
Return a sample point guaranteed to be within a ring, by efficiently
finding the first centroid of a coordinate triplet whose orientation
matches the orientation of the ring and passes the point-in-ring test.
The orientation of the ring is assumed to be clockwise, unless ccw
(counter-clockwise) is set to True.
```

14.9.5.13 shapefile.signed_area (*coords*, *fast* = `False`)

```
Return the signed area enclosed by a ring using the linear time
algorithm. A value >= 0 indicates a counter-clockwise oriented ring.
A faster version is possible by setting 'fast' to True, which returns
2x the area, e.g. if you're only interested in the sign of the area.
```

14.9.5.14 shapefile.test (** *kwargs*)

14.9.5.15 shapefile.u (*v*, *encoding* = 'utf-8', *encodingErrors* = 'strict')

14.9.6 Variable Documentation

14.9.6.1 str shapefile.__version__ = "2.3.1" [private]

14.9.6.2 shapefile.failure_count = test()

14.9.6.3 int shapefile.FIRST_RING = 4

14.9.6.4 int shapefile.INNER_RING = 3

14.9.6.5 shapefile.izip = zip

14.9.6.6 shapefile.logger = logging.getLogger(__name__)

14.9.6.7 list shapefile.MISSING = [None,'']

14.9.6.8 int shapefile.MULTIPATCH = 31

14.9.6.9 int shapefile.MULTIPOINT = 8

14.9.6.10 int shapefile.MULTIPOINTM = 28

14.9.6.11 int shapefile.MULTIPOINTZ = 18

14.9.6.12 int shapefile.NODATA = -10e38

14.9.6.13 int shapefile.NULL = 0

14.9.6.14 int shapefile.OUTER_RING = 2

14.9.6.15 dict shapefile.PARTTYPE_LOOKUP

```
Initial value:1 =  {
2     0: 'TRIANGLE_STRIP',
3     1: 'TRIANGLE_FAN',
4     2: 'OUTER_RING',
5     3: 'INNER_RING',
6     4: 'FIRST_RING',
7     5: 'RING'}
```

14.9.6.16 int shapefile.POINT = 1

14.9.6.17 int shapefile.POINTM = 21

14.9.6.18 int shapefile.POINTZ = 11

14.9.6.19 int shapefile.POLYGON = 5

14.9.6.20 int shapefile.POLYGONM = 25

14.9.6.21 int shapefile.POLYGONZ = 15

14.9.6.22 int shapefile.POLYLINE = 3

14.9.6.23 int shapefile.POLYLINEM = 23

14.9.6.24 int shapefile.POLYLINEZ = 13

14.9.6.25 int shapefile.PYTHON3 = 3

14.9.6.26 int shapefile.RING = 5

14.9.6.27 dict shapefile.SHAPETYPE_LOOKUP

```
Initial value:1 =  {
    0: 'NULL',
    1: 'POINT',
    3: 'POLYLINE',
    5: 'POLYGON',
    8: 'MULTIPOINT',
    11: 'POINTZ',
    13: 'POLYLINEZ',
    15: 'POLYGONZ',
     18: 'MULTIPOINTZ',
     21: 'POINTM',
     23: 'POLYLINEM',
     25: 'POLYGONM',
     28: 'MULTIPOINTM',
     31: 'MULTIPATCH'}
```

14.9.6.28 int shapefile.TRIANGLE_FAN = 1

14.9.6.29 int shapefile.TRIANGLE_STRIP = 0

14.9.6.30 bool shapefile.VERBOSE = True

14.9.6.31 shapefile.xrange = range

14.10 ShapeTest Namespace Reference

14.10.1 Classes

14.10.2 class GeoShapeVariables

- **sf** = **shapefile.Reader**("Shapefiles/MAB_Estimation_Areas_2019_UTM18_PDT.shp")
- **shapes** = sf.shapes()
- **shapeLen** = len(**sf**)
- list **shapeMA** = [**GeoShape**() for _ in range(**shapeLen**)]
- **record** = sf.record(n)
- **as_dict** = record.as_dict()
- **SAMS**
- **NewSAMS**
- **areaKm2**
- **pointLen** = len(**shapes**[n].points)
- **X**
- **Y**
- **lat**
- **lon**
- list **shapeGB** = [**GeoShape**() for _ in range(**shapeLen**)]

14.10.3 Variable Documentation

14.10.3.1 ShapeTest.areaKm2

14.10.3.2 ShapeTest.as_dict = record.as_dict()

14.10.3.3 ShapeTest.lat

14.10.3.4 ShapeTest.lon

14.10.3.5 ShapeTest.NewSAMS

14.10.3.6 ShapeTest.pointLen = len(shapes[n].points)

14.10.3.7 ShapeTest.record = sf.record(n)

14.10.3.8 ShapeTest.SAMS

14.10.3.9 ShapeTest.sf = shapefile.Reader("Shapefiles/MAB_Estimation_Areas_2019_UTM18_PDT.shp")

14.10.3.10 list ShapeTest.shapeGB = [GeoShape() for _ in range(shapeLen)]

14.10.3.11 ShapeTest.shapeLen = len(sf)

14.10.3.12 list ShapeTest.shapeMA = [GeoShape() for _ in range(shapeLen)]

14.10.3.13 ShapeTest.shapes = sf.shapes()

14.10.3.14 ShapeTest.X

14.10.3.15 ShapeTest.Y

14.11 SortByAreaFrame Namespace Reference

14.11.1 Classes

class **SortByArea***This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.*

14.12 SortByRegionFrame Namespace Reference

14.12.1 Classes

class **GeoShape***This class is used to define the shape of the regional data.*

class **SortByRegion***This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.*

14.13 SortIntoColumns Namespace Reference

14.13.1 Classes

- class **Column**class **GeoShape**

14.13.2 Variables

- **inputFile** = sys.argv[1]
- **l** = len(**inputFile**)
- **domain** = **inputFile**[**l**-2:**l**]
- **dataFile** = os.path.join('Data', **inputFile**+'.csv')
- **outfile** = os.path.join('Data', **inputFile**+'_BUFFER.csv')
- **M** = pd.read_csv(**dataFile**)
- **fileName** = os.environ['GBShapeBufferFile']
- str **subDir** = 'GB_Buffer'
- **shapeFile** = os.path.join('Shapefiles', **subDir**, **fileName**)
- **sf** = **shapefile.Reader**(**shapeFile**)
- **shapes** = sf.shapes()
- **shapeLen** = len(**sf**)
- list **shape** = [**GeoShape**() for _ in range(**shapeLen**)]
- **record** = sf.record(n)
- **as_dict** = record.as_dict()
- **Region**
- **pointLen** = len(**shapes**[n].points)
- **X**
- **Y**
- list **columns** = [**Column**() for _ in range(**shapeLen**)]
- **name**
- **X_t** = **M**['UTM_X']
- **Y_t** = **M**['UTM_Y']
- **rows** = len(**X_t**)
- **nodes** = len(**shape**[rgn].**X**)
- **sep**

- **na_rep**
- **index**

14.13.3 Variable Documentation

14.13.3.1 SortIntoColumns.as_dict = record.as_dict()

14.13.3.2 list SortIntoColumns.columns = [Column() for _ in range(shapeLen)]

14.13.3.3 SortIntoColumns.dataFile = os.path.join('Data', inputFile+'.csv')

14.13.3.4 SortIntoColumns.domain = inputFile[l-2:l]

14.13.3.5 SortIntoColumns.fileName = os.environ['GBShapeBufferFile']

14.13.3.6 SortIntoColumns.index

14.13.3.7 SortIntoColumns.inputFile = sys.argv[1]

14.13.3.8 SortIntoColumns.l = len(inputFile)

14.13.3.9 SortIntoColumns.M = pd.read_csv(dataFile)

14.13.3.10 SortIntoColumns.na_rep

14.13.3.11 SortIntoColumns.name

14.13.3.12 SortIntoColumns.nodes = len(shape[rgn].X)

14.13.3.13 SortIntoColumns.outfile = os.path.join('Data', inputFile+'_BUFFER.csv')

14.13.3.14 SortIntoColumns.pointLen = len(shapes[n].points)

14.13.3.15 SortIntoColumns.record = sf.record(n)

14.13.3.16 SortIntoColumns.Region

14.13.3.17 SortIntoColumns.rows = len(X_t)

14.13.3.18 SortIntoColumns.sep

14.13.3.19 SortIntoColumns.sf = shapefile.Reader(shapeFile)

14.13.3.20 list SortIntoColumns.shape = [GeoShape() for _ in range(shapeLen)]

14.13.3.21 SortIntoColumns.shapeFile = os.path.join('Shapefiles', subDir, fileName)

14.13.3.22 SortIntoColumns.shapeLen = len(sf)

14.13.3.23 SortIntoColumns.shapes = sf.shapes()

14.13.3.24 str SortIntoColumns.subDir = 'GB_Buffer'

14.13.3.25 SortIntoColumns.X

14.13.3.26 SortIntoColumns.X_t = M['UTM_X']

14.13.3.27 SortIntoColumns.Y

14.13.3.28 SortIntoColumns.Y_t = M['UTM_Y']

14.14 SpecialAreaFrame Namespace Reference

14.14.1 Classes

class **SpecialArea***This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.*

14.15 Widgets Namespace Reference

14.15.1 Classes

class **ScrollFrame***Scrollable Frame Class from* `https://gist.github.com/mp035/9f2027c3ef9172264532fcd6262f3b01`.

class **SubFrameElement***Generic Element.*

- class **SubFrameInterpFunction**class **SubFrameXY**
  Widget for XY label and entery.

14.15.2 Functions

- **numbersCallback** (input)
  Allows only correctly formed positive integers, ignores non-numeric characters.

- **floatCallback** (input)
  Allows only correctly formed floats, ignores non-numeric characters.

14.15.3 Function Documentation

14.15.3.1 Widgets.floatCallback (*input*)

Allows only correctly formed floats, ignores non-numeric characters.

14.15.3.2 Widgets.numbersCallback (*input*)

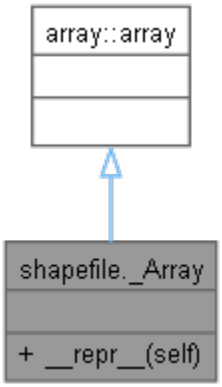
Allows only correctly formed positive integers, ignores non-numeric characters.

Only allows numeric for input

15 Class Documentation

15.1 shapefile._Array Class Reference

Inheritance diagram for shapefile._Array:



15.1.1 Public Member Functions

- **__repr__** (self)

15.1.2 Detailed Description

```
Converts python tuples to lists of the appropriate type.
```

Used to unpack different shapefile header parts.

15.1.3 Member Function Documentation

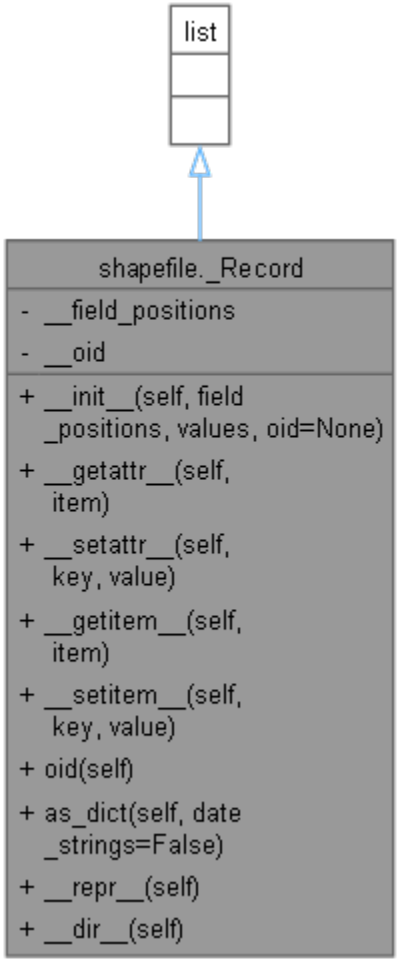
15.1.3.1 shapefile._Array.__repr__ (*self*)

15.1.3.2 The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

15.2 shapefile._Record Class Reference

Inheritance diagram for shapefile._Record:



15.2.1 Public Member Functions

- **__init__** (self, field_positions, values, **oid**=None)
- **__getattr__** (self, item)
- **__setattr__** (self, key, value)
- **__getitem__** (self, item)
- **__setitem__** (self, key, value)
- **oid** (self)
- **as_dict** (self, date_strings=False)
- **__repr__** (self)
- **__dir__** (self)

15.2.2 Private Attributes

- **__field_positions__oid**

15.2.3 Detailed Description

```
A class to hold a record. Subclasses list to ensure compatibility with
former work and to reuse all the optimizations of the builtin list.
In addition to the list interface, the values of the record
can also be retrieved using the field's name. For example if the dbf contains
```

```
a field ID at position 0, the ID can be retrieved with the position, the field name
as a key, or the field name as an attribute.

>>> # Create a Record with one field, normally the record is created by the Reader class
>>> r = _Record({'ID': 0}, [0])
>>> print(r[0])
>>> print(r['ID'])
>>> print(r.ID)
```

15.2.4 Constructor & Destructor Documentation

15.2.4.1 shapefile._Record.__init__ (*self*, *field_positions*, *values*, *oid* = None)

```
A Record should be created by the Reader class

:param field_positions: A dict mapping field names to field positions
:param values: A sequence of values
:param oid: The object id, an int (optional)
```

15.2.5 Member Function Documentation

15.2.5.1 shapefile._Record.__dir__ (*self*)

```
Helps to show the field names in an interactive environment like IPython.
See: http://ipython.readthedocs.io/en/stable/config/integrating.html

:return: List of method names and fields
```

15.2.5.2 shapefile._Record.__getattr__ (*self*, *item*)

```
__getattr__ is called if an attribute is used that does
not exist in the normal sense. For example r=Record(...), r.ID
calls r.__getattr__('ID'), but r.index(5) calls list.index(r, 5)
:param item: The field name, used as attribute
:return: Value of the field
:raises: AttributeError, if item is not a field of the shapefile
        and IndexError, if the field exists but the field's
        corresponding value in the Record does not exist
```

15.2.5.3 shapefile._Record.__getitem__ (*self*, *item*)

```
Extends the normal list item access with
access using a fieldname

For example r['ID'], r[0]
:param item: Either the position of the value or the name of a field
:return: the value of the field
```

15.2.5.4 shapefile._Record.__repr__ (*self*)

15.2.5.5 shapefile._Record.__setattr__ (*self*, *key*, *value*)

```
Sets a value of a field attribute
:param key: The field name
:param value: the value of that field
:return: None
:raises: AttributeError, if key is not a field of the shapefile
```

15.2.5.6 shapefile._Record.__setitem__ (*self*, *key*, *value*)

```
Extends the normal list item access with
access using a fieldname

For example r['ID']=2, r[0]=2
:param key: Either the position of the value or the name of a field
:param value: the new value of the field
```

15.2.5.7 shapefile._Record.as_dict (*self*, *date_strings* = `False`)

```
Returns this Record as a dictionary using the field names as keys
:return: dict
```

15.2.5.8 shapefile._Record.oid (*self*)

The index position of the record in the original shapefile

15.2.6 Member Data Documentation

15.2.6.1 shapefile._Record.__field_positions `[private]`

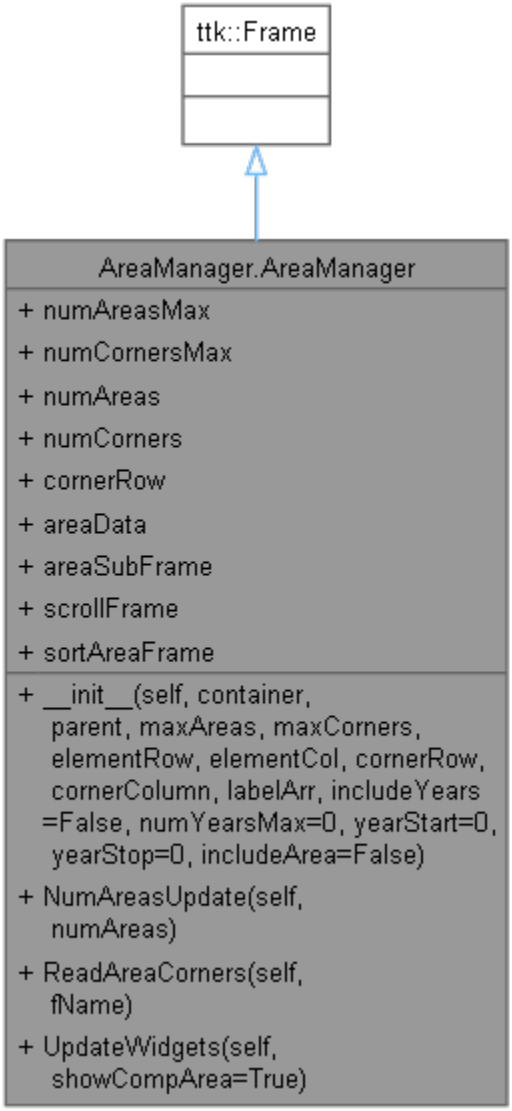
15.2.6.2 shapefile._Record.__oid `[private]`

15.2.6.3 The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

15.3 AreaManager.AreaManager Class Reference

This class is used to paint area grouped by.
Inheritance diagram for AreaManager.AreaManager:



15.3.1 Public Member Functions

- **__init__** (self, container, parent, maxAreas, maxCorners, elementRow, elementCol, **cornerRow**, cornerColumn, labelArr, includeYears=False, numYearsMax=0, yearStart=0, yearStop=0, includeArea=False)
- **NumAreasUpdate** (self, **numAreas**)
- **ReadAreaCorners** (self, fName)
- **UpdateWidgets** (self, showCompArea=True)

15.3.2 Public Attributes

- **numAreasMaxnumCornersMax**
- **numAreas**
- **numCorners**
- **cornerRow**
- **areaData**
- **areaSubFrame**

- **scrollFrame**
- **sortAreaFrame**

15.3.3 Detailed Description

This class is used to paint area grouped by.

Area N Comment Number of Nodes Update Nodes Node 1 .... Node N X data .... X data Y data .... Y data

15.3.4 Constructor & Destructor Documentation

15.3.4.1 AreaManager.AreaManager.__init__ (*self*, *container*, *parent*, *maxAreas*, *maxCorners*, *elementRow*, *elementCol*, *cornerRow*, *cornerColumn*, *labelArr*, *includeYears* = `False`, *numYearsMax* = `0`, *yearStart* = `0`, *yearStop* = `0`, *includeArea* = `False`)

15.3.5 Member Function Documentation

15.3.5.1 AreaManager.AreaManager.NumAreasUpdate (*self*, *numAreas*)

Updates the number of areas functions.

15.3.5.2 AreaManager.AreaManager.ReadAreaCorners (*self*, *fName*)

Reads an Area file and returns the number of nodes defined

15.3.5.3 AreaManager.AreaManager.UpdateWidgets (*self*, *showCompArea* = `True`)

15.3.6 Member Data Documentation

15.3.6.1 AreaManager.AreaManager.areaData

15.3.6.2 AreaManager.AreaManager.areaSubFrame

15.3.6.3 AreaManager.AreaManager.cornerRow

15.3.6.4 AreaManager.AreaManager.numAreas

15.3.6.5 AreaManager.AreaManager.numAreasMax

15.3.6.6 AreaManager.AreaManager.numCorners

15.3.6.7 AreaManager.AreaManager.numCornersMax

15.3.6.8 AreaManager.AreaManager.scrollFrame

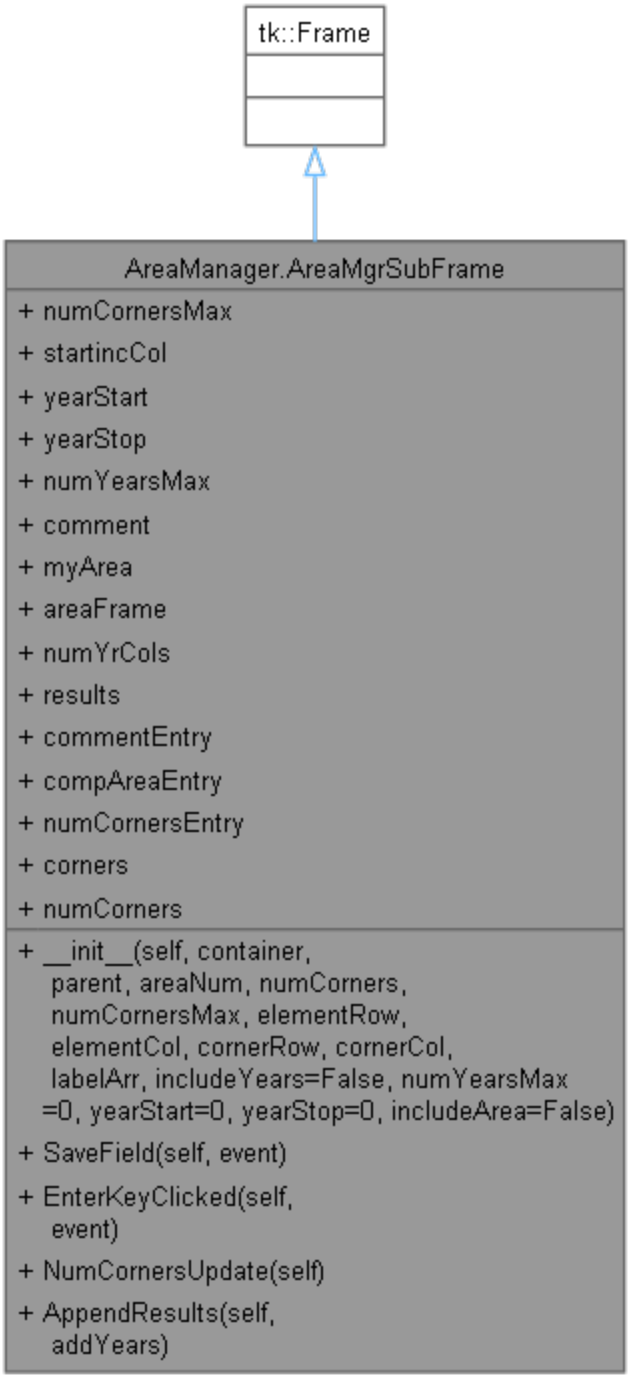
15.3.6.9 AreaManager.AreaManager.sortAreaFrame

15.3.6.10 The documentation for this class was generated from the following file:

15.3.6.11 AreaManager.py

15.4 AreaManager.AreaMgrSubFrame Class Reference

Inheritance diagram for AreaManager.AreaMgrSubFrame:



15.4.1 Public Member Functions

- **__init__** (self, container, parent, areaNum, **numCorners**, **numCornersMax**, elementRow, elementCol, cornerRow, cornerCol, labelArr, includeYears=False, **numYearsMax**=0, **yearStart**=0, **yearStop**=0, includeArea=False)
- **SaveField** (self, event)
- **EnterKeyClicked** (self, event)
- **NumCornersUpdate** (self)
- **AppendResults** (self, addYears)

This method is used to add results when the original maximum number of years is exceeded.

15.4.2 Public Attributes

- **numCornersMaxstartincCol**
- **yearStart**
- **yearStop**
- **numYearsMax**
- **comment**
- **myArea**
- **areaFrame**
- **numYrCols**
- **results**
- **commentEntry**
- **compAreaEntry**
- **numCornersEntry**
- **corners**
- **numCorners**

15.4.3 Constructor & Destructor Documentation

15.4.3.1 AreaManager.AreaMgrSubFrame.__init__ (*self*, *container*, *parent*, *areaNum*, *numCorners*, *numCornersMax*, *elementRow*, *elementCol*, *cornerRow*, *cornerCol*, *labelArr*, *includeYears* = `False`, *numYearsMax* = `0`, *yearStart* = `0`, *yearStop* = `0`, *includeArea* = `False`)

15.4.4 Member Function Documentation

15.4.4.1 AreaManager.AreaMgrSubFrame.AppendResults (*self*, *addYears*)

This method is used to add results when the original maximum number of years is exceeded.

15.4.4.2 AreaManager.AreaMgrSubFrame.EnterKeyClicked (*self*, *event*)

15.4.4.3 AreaManager.AreaMgrSubFrame.NumCornersUpdate (*self*)

15.4.4.4 AreaManager.AreaMgrSubFrame.SaveField (*self*, *event*)

15.4.5 Member Data Documentation

15.4.5.1 AreaManager.AreaMgrSubFrame.areaFrame

15.4.5.2 AreaManager.AreaMgrSubFrame.comment

15.4.5.3 AreaManager.AreaMgrSubFrame.commentEntry

15.4.5.4 AreaManager.AreaMgrSubFrame.compAreaEntry

15.4.5.5 AreaManager.AreaMgrSubFrame.corners

15.4.5.6 AreaManager.AreaMgrSubFrame.myArea

15.4.5.7 AreaManager.AreaMgrSubFrame.numCorners

15.4.5.8 AreaManager.AreaMgrSubFrame.numCornersEntry

15.4.5.9 AreaManager.AreaMgrSubFrame.numCornersMax

15.4.5.10 AreaManager.AreaMgrSubFrame.numYearsMax

15.4.5.11 AreaManager.AreaMgrSubFrame.numYrCols

15.4.5.12 AreaManager.AreaMgrSubFrame.results

15.4.5.13 AreaManager.AreaMgrSubFrame.startincCol

15.4.5.14 AreaManager.AreaMgrSubFrame.yearStart

15.4.5.15 AreaManager.AreaMgrSubFrame.yearStop

15.4.5.16 The documentation for this class was generated from the following file:

15.4.5.17 AreaManager.py

15.5 SortIntoColumns.Column Class Reference

15.5.1 Public Member Functions

- **__init__** (self)

15.5.2 Public Attributes

- **nameinBox**

15.5.3 Constructor & Destructor Documentation

15.5.3.1 SortIntoColumns.Column.__init__ (*self*)

15.5.4 Member Data Documentation

15.5.4.1 SortIntoColumns.Column.inBox

15.5.4.2 SortIntoColumns.Column.name

15.5.4.3 The documentation for this class was generated from the following file:

15.5.4.4 SortIntoColumns.py

15.6 AreaManager.Corner Class Reference

Defines floating point data for corner defintions.

15.6.1 Public Member Functions

- **__init__** (self, maxCorners)

15.6.2 Public Attributes

- **longlat**
- **numCorners**

15.6.3 Detailed Description

Defines floating point data for corner defintions.

long, lat have become interchangeable with x, y

15.6.4 Constructor & Destructor Documentation

15.6.4.1 AreaManager.Corner.__init__ (*self*, *maxCorners*)

15.6.5 Member Data Documentation

15.6.5.1 AreaManager.Corner.lat

15.6.5.2 AreaManager.Corner.long

15.6.5.3 AreaManager.Corner.numCorners

15.6.5.4 The documentation for this class was generated from the following file:

15.6.5.5 AreaManager.py

15.7 EditMathSetupFrame.EditMathSetup Class Reference

This class allows the user to edit the Matlab/Octave setup files to fit their environment.

Inheritance diagram for EditMathSetupFrame.EditMathSetup:

15.7.1 Public Member Functions

- **__init__** (self, container)
  Constructor for Growth Class.

- **on_visibility** (self, event)
  Opens either startup.m or .octaverc depending if user selected Matlab or Octave resp.

- **LoadStartupFile** (self)
- **ReadFile** (self, showMsg)
- **WriteStartupFile** (self)
- **pop_up** (self)

15.7.2 Public Attributes

- **matlabFNameoctaveFName**
- **editMathFrame**
- **editText**
- **editTextButton**
- **isMatlab**
- **usingMatlab**
- **useMatlabRB**
- **useOctaveRB**
- **on_visibility**

15.7.3 Detailed Description

This class allows the user to edit the Matlab/Octave setup files to fit their environment.

15.7.4 Constructor & Destructor Documentation

15.7.4.1 EditMathSetupFrame.EditMathSetup.__init__ (*self*, *container*)

Constructor for Growth Class.

15.7.5 Member Function Documentation

15.7.5.1 EditMathSetupFrame.EditMathSetup.LoadStartupFile (*self*)

15.7.5.2 EditMathSetupFrame.EditMathSetup.on_visibility (*self*, *event*)

Opens either startup.m or .octaverc depending if user selected Matlab or Octave resp.

15.7.5.3 EditMathSetupFrame.EditMathSetup.pop_up (*self*)

15.7.5.4 EditMathSetupFrame.EditMathSetup.ReadFile (*self*, *showMsg*)

15.7.5.5 EditMathSetupFrame.EditMathSetup.WriteStartupFile (*self*)

15.7.6 Member Data Documentation

15.7.6.1 EditMathSetupFrame.EditMathSetup.editMathFrame

15.7.6.2 EditMathSetupFrame.EditMathSetup.editText

15.7.6.3 EditMathSetupFrame.EditMathSetup.editTextButton

15.7.6.4 EditMathSetupFrame.EditMathSetup.isMatlab

15.7.6.5 EditMathSetupFrame.EditMathSetup.matlabFName

15.7.6.6 EditMathSetupFrame.EditMathSetup.octaveFName

15.7.6.7 EditMathSetupFrame.EditMathSetup.on_visibility

15.7.6.8 EditMathSetupFrame.EditMathSetup.useMatlabRB

15.7.6.9 EditMathSetupFrame.EditMathSetup.useOctaveRB

15.7.6.10 EditMathSetupFrame.EditMathSetup.usingMatlab

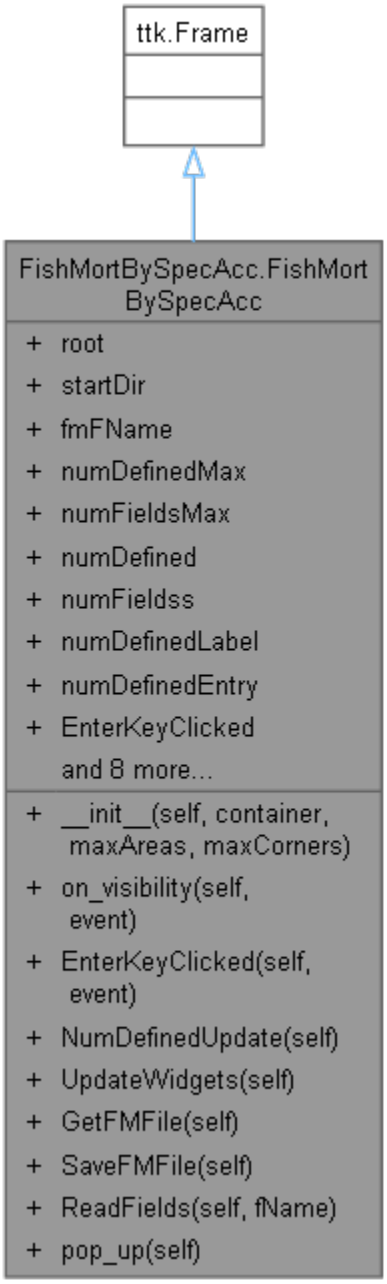
15.7.6.11 The documentation for this class was generated from the following file:

15.7.6.12 EditMathSetupFrame.py

15.8 FishMortBySpecAcc.FishMortBySpecAcc Class Reference

This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

Inheritance diagram for FishMortBySpecAcc.FishMortBySpecAcc:



15.8.1 Public Member Functions

- **__init__** (self, container, maxAreas, maxCorners)

- **on_visibility** (self, event)
- **EnterKeyClicked** (self, event)
- **NumDefinedUpdate** (self)
- **UpdateWidgets** (self)
- **GetFMFile** (self)
- **SaveFMFile** (self)
- **ReadFields** (self, fName)
- **pop_up** (self)
  Help Window for Fishing Mortatlity in Special Access Area.

15.8.2 Public Attributes

- **rootstartDir**
- **fmFName**
- **numDefinedMax**
- **numFieldsMax**
- **numDefined**
- **numFieldss**
- **numDefinedLabel**
- **numDefinedEntry**
- **EnterKeyClicked**
- **fishMortFile**
- **fishMortFileLabel**
- **openFMFileButton**
- **saveFMFileButton**
- **yearEntry**
- **areaMgr**
- **on_visibility**
- **numAreas**

15.8.3 Detailed Description

This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

15.8.4 Constructor & Destructor Documentation

15.8.4.1 FishMortBySpecAcc.FishMortBySpecAcc.__init__ (*self*, *container*, *maxAreas*, *maxCorners*)

15.8.5 Member Function Documentation

15.8.5.1 FishMortBySpecAcc.FishMortBySpecAcc.EnterKeyClicked (*self*, *event*)

15.8.5.2 FishMortBySpecAcc.FishMortBySpecAcc.GetFMFile (*self*)

15.8.5.3 FishMortBySpecAcc.FishMortBySpecAcc.NumDefinedUpdate (*self*)

15.8.5.4 FishMortBySpecAcc.FishMortBySpecAcc.on_visibility (*self*, *event*)

15.8.5.5 FishMortBySpecAcc.FishMortBySpecAcc.pop_up (*self*)

Help Window for Fishing Mortatlity in Special Access Area.

15.8.5.6 FishMortBySpecAcc.FishMortBySpecAcc.ReadFields (*self*, *fName*)

```
Reads an Area file and returns the number of fields.
```

Fields have a Special Area number for the x value with a Mortality setting for the y value.

15.8.5.7 FishMortBySpecAcc.FishMortBySpecAcc.SaveFMFile (*self*)

15.8.5.8 FishMortBySpecAcc.FishMortBySpecAcc.UpdateWidgets (*self*)

15.8.6 Member Data Documentation

15.8.6.1 FishMortBySpecAcc.FishMortBySpecAcc.areaMgr

15.8.6.2 FishMortBySpecAcc.FishMortBySpecAcc.EnterKeyClicked

15.8.6.3 FishMortBySpecAcc.FishMortBySpecAcc.fishMortFile

15.8.6.4 FishMortBySpecAcc.FishMortBySpecAcc.fishMortFileLabel

15.8.6.5 FishMortBySpecAcc.FishMortBySpecAcc.fmFName

15.8.6.6 FishMortBySpecAcc.FishMortBySpecAcc.numAreas

15.8.6.7 FishMortBySpecAcc.FishMortBySpecAcc.numDefined

15.8.6.8 FishMortBySpecAcc.FishMortBySpecAcc.numDefinedEntry

15.8.6.9 FishMortBySpecAcc.FishMortBySpecAcc.numDefinedLabel

15.8.6.10 FishMortBySpecAcc.FishMortBySpecAcc.numDefinedMax

15.8.6.11 FishMortBySpecAcc.FishMortBySpecAcc.numFieldsMax

15.8.6.12 FishMortBySpecAcc.FishMortBySpecAcc.numFieldss

15.8.6.13 FishMortBySpecAcc.FishMortBySpecAcc.on_visibility

15.8.6.14 FishMortBySpecAcc.FishMortBySpecAcc.openFMFileButton

15.8.6.15 FishMortBySpecAcc.FishMortBySpecAcc.root

15.8.6.16 FishMortBySpecAcc.FishMortBySpecAcc.saveFMFileButton

15.8.6.17 FishMortBySpecAcc.FishMortBySpecAcc.startDir

15.8.6.18 FishMortBySpecAcc.FishMortBySpecAcc.yearEntry

15.8.6.19 The documentation for this class was generated from the following file:

15.8.6.20 FishMortBySpecAcc.py

15.9 ShapeTest.GeoShape Class Reference

15.9.1 Public Member Functions

- **__init__** (self)

15.9.2 Public Attributes

- **XY**
- **lat**
- **lon**
- **SAMS**
- **NewSAMS**
- **areaKm2**

15.9.3 Constructor & Destructor Documentation

15.9.3.1 ShapeTest.GeoShape.__init__ (*self*)

15.9.4 Member Data Documentation

15.9.4.1 ShapeTest.GeoShape.areaKm2

15.9.4.2 ShapeTest.GeoShape.lat

15.9.4.3 ShapeTest.GeoShape.lon

15.9.4.4 ShapeTest.GeoShape.NewSAMS

15.9.4.5 ShapeTest.GeoShape.SAMS

15.9.4.6 ShapeTest.GeoShape.X

15.9.4.7 ShapeTest.GeoShape.Y

15.9.4.8 The documentation for this class was generated from the following file:

15.9.4.9 ShapeTest.py

15.10 SortByRegionFrame.GeoShape Class Reference

This class is used to define the shape of the regional data.

15.10.1 Public Member Functions

- **__init__** (self)

15.10.2 Public Attributes

- **XY**
- **lat**
- **lon**
- **Zone**
- **area**

15.10.3 Detailed Description

This class is used to define the shape of the regional data.

This class uses PyShp libraries as defined in `https://github.com/GeospatialPython/pyshp` ==> Code ==> Download ZIP

15.10.4 Constructor & Destructor Documentation

15.10.4.1 SortByRegionFrame.GeoShape.__init__ (*self*)

15.10.5 Member Data Documentation

15.10.5.1 SortByRegionFrame.GeoShape.area

15.10.5.2 SortByRegionFrame.GeoShape.lat

15.10.5.3 SortByRegionFrame.GeoShape.lon

15.10.5.4 SortByRegionFrame.GeoShape.X

15.10.5.5 SortByRegionFrame.GeoShape.Y

15.10.5.6 SortByRegionFrame.GeoShape.Zone

15.10.5.7 The documentation for this class was generated from the following file:

15.10.5.8 SortByRegionFrame.py

15.11 SortIntoColumns.GeoShape Class Reference

15.11.1 Public Member Functions

- **__init__** (self)

15.11.2 Public Attributes

- **XY**
- **lat**
- **lon**
- **Region**

15.11.3 Constructor & Destructor Documentation

15.11.3.1 SortIntoColumns.GeoShape.__init__ (*self*)

15.11.4 Member Data Documentation

15.11.4.1 SortIntoColumns.GeoShape.lat

15.11.4.2 SortIntoColumns.GeoShape.lon

15.11.4.3 SortIntoColumns.GeoShape.Region

15.11.4.4 SortIntoColumns.GeoShape.X

15.11.4.5 SortIntoColumns.GeoShape.Y

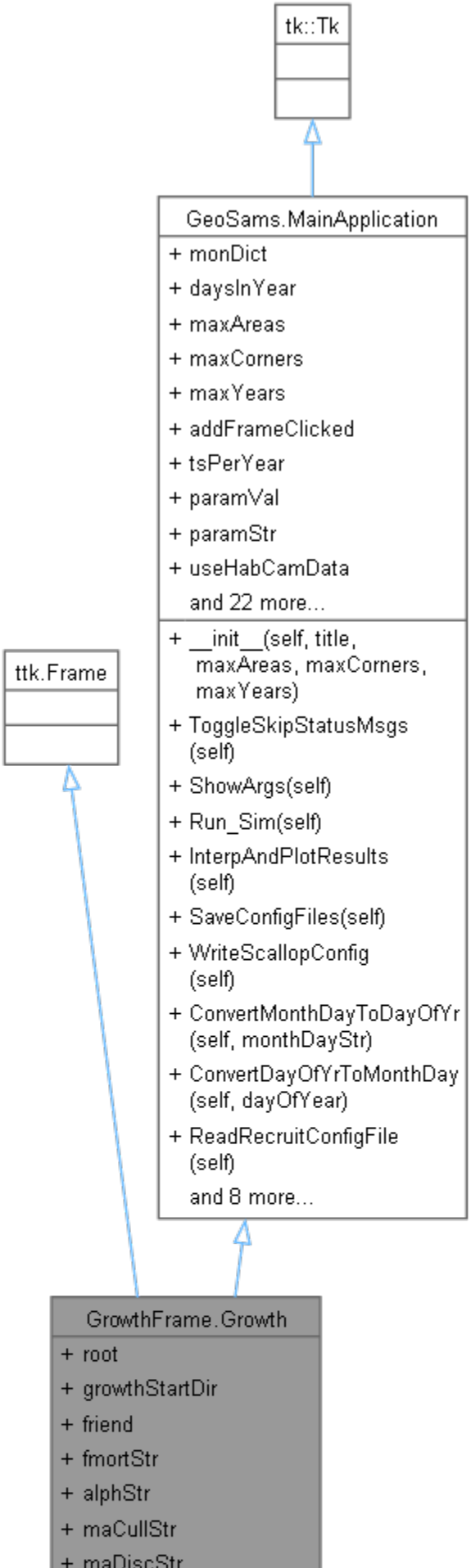
15.11.4.6 The documentation for this class was generated from the following file:

15.11.4.7 SortIntoColumns.py

15.12 GrowthFrame.Growth Class Reference

This class allows the user to adjust parameters used in computing scallop growth.

Inheritance diagram for GrowthFrame.Growth:


tk::Tk
GeoSams.MainApplication
+ monDict
+ daysInYear
+ maxAreas
+ maxCorners
+ maxYears
+ addFrameClicked
+ tsPerYear
+ paramVal
+ paramStr
+ useHabCamData
and 22 more...
+ __init__(self, title, maxAreas, maxCorners, maxYears)
+ ToggleSkipStatusMsgs (self)
+ ShowArgs(self)
+ Run_Sim(self)
+ InterpAndPlotResults (self)
+ SaveConfigFiles(self)
+ WriteScallopConfig (self)
+ ConvertMonthDayToDayOfYr (self, monthDayStr)
+ ConvertDayOfYrToMonthDay (self, dayOfYear)
+ ReadRecruitConfigFile (self)
and 8 more...
ttk.Frame
GrowthFrame.Growth
+ root
+ growthStartDir
+ friend
+ fmortStr
+ alphStr
+ maCullStr

15.12.1 Public Member Functions

- **__init__** (self, container, **friend**)
  Constructor for Growth Class.

- **on_visibility** (self, event)
- **GetGrowthConfigFName** (self)
  Calls the filedialog method asksaveasfilename to name a file to be used for the Mortality Configuration file.

- **LoadGrowthData** (self)
- **UpdateWidgets** (self)
- **UpdateValues** (self)
  Method to read Mortality Configuration file and set values accordingly.

- **pop_up** (self)

15.12.1.1 Public Member Functions inherited from GeoSams.MainApplication

- **ToggleSkipStatusMsgs** (self)
- **ShowArgs** (self)
  Display setup limits here Messagebox blocks entry widgets if attempted to open before the main window completes.

- **Run_Sim** (self)
  *Starts the GeoSAMS simulatation **ScallopPopDensity** .*

- **InterpAndPlotResults** (self)
- **SaveConfigFiles** (self)
  Save all of the defined configuration files.

- **WriteScallopConfig** (self)
  Saves simulation configuration file.

- **ConvertMonthDayToDayOfYr** (self, monthDayStr)
  This method is used to converty the recruitment start and stop dates from a string month numerical day into days in a year.

- **ConvertDayOfYrToMonthDay** (self, dayOfYear)
- **ReadRecruitConfigFile** (self)
  Read in the (tag, value) parameters from the recruitment to update parameters.

- **WriteRecruitmentConfig** (self)
  Saves recruitment parameters to a configuration file.

- **WriteGrowthConfig** (self)

Saves mortality parameters to a configuration file.

- **WriteGridMgrConfig** (self)
  Saves grid manager parameters to a configuration file.

- **WriteSpatialFncsConfig** (self, cfgFile, functions, numFncsEntry)
  Saves spatial function parameters to a configuration file.

- **ReadConfigFile** (self, fName)
  Reads a typical configuration file to recover the tags and values.

- **ReadSimConfigFile** (self)
  Read in the (tag, value) parameters from the simulation configuration file.

- **ReadGridMgrConfigFile** (self)
  Read in the (tag, value) parameters from the grid manager configuration file.

15.12.2 Public Attributes

- **rootgrowthStartDir**
- **friend**
- **fmortStr**
- **alphStr**
- **maCullStr**
- **maDiscStr**
- **gbCullStr**
- **gbDiscStr**
- **maFSelAStr**
- **maFSelBStr**
- **gbClFSelAStr**
- **gbClFSelBStr**
- **gbOpFSelAStr**
- **gbOpFSelBStr**
- **maAdultMortStr**
- **gbAdultMortStr**
- **maIncidStr**
- **gbIncidStr**
- **maLen0Str**
- **gbLen0Str**
- **fmortFileName**
- **lpueSlStr**
- **lpueSl2Str**
- **lpueIntcStr**
- **maxPerDayStr**
- **maxTimeStr**
- **dredgeWdStr**
- **towSpdStr**
- **fishMort**
- **alphaMort**

- **maAdultMort**
- **gbAdultMort**
- **maLength0**
- **gbLength0**
- **maFSelectA**
- **maFSelectB**
- **gbClosedFSelectA**
- **gbClosedFSelectB**
- **gbOpenFSelectA**
- **gbOpenFSelectB**
- **lpueSlope**
- **lpueSlope2**
- **lpueIntcept**
- **maxPerDay**
- **maxTime**
- **dredgeWth**
- **towSpeed**
- **maIncident**
- **gbIncident**
- **maCullSize**
- **maDiscard**
- **gbCullSize**
- **gbDiscard**
- **growthCfgFile**
- **saveGrowthConfigButton**
- **loadGrowthConfigButton**
- **on_visibility**

15.12.2.1 Public Attributes inherited from GeoSams.MainApplication

- **monDict**
- **daysInYear**
- **maxAreas**
- **maxCorners**
- **maxYears**
- **addFrameClicked**
- **tsPerYear**
- **paramVal**
- **paramStr**
- **useHabCamData**
- **domainName**
- **yearStart**
- **yearStop**
- **simConfigFile**
- **style**
- **root**
- **notebook**
- **frame1**
- **recrConfigFile**
- **frame2**
- **frame3**
- **frame5**
- **frame6**

- **frame7**
- **frame8**
- **gmConfigFile**
- **specAccFileStr**
- **isSkip**
- **skipStatusMsgs**
- **skipStatusMsgsRB**
- **recruitYrStrt**
- **recruitYrStop**

15.12.3 Detailed Description

This class allows the user to adjust parameters used in computing scallop growth.

15.12.4 Constructor & Destructor Documentation

15.12.4.1 GrowthFrame.Growth.__init__ (*self*, *container*, *friend*)

Constructor for Growth Class.

Reimplemented from **GeoSams.MainApplication** (*p.69*).

15.12.5 Member Function Documentation

15.12.5.1 GrowthFrame.Growth.GetGrowthConfigFName (*self*)

Calls the filedialog method asksaveasfilename to name a file to be used for the Mortality Configuration file.

It then writes out the defined parameters to this file using the 'tag = value' format.

15.12.5.2 GrowthFrame.Growth.LoadGrowthData (*self*)

15.12.5.3 GrowthFrame.Growth.on_visibility (*self*, *event*)

15.12.5.4 GrowthFrame.Growth.pop_up (*self*)

Reimplemented from **GeoSams.MainApplication** (*p.70*).

15.12.5.5 GrowthFrame.Growth.UpdateValues (*self*)

Method to read Mortality Configuration file and set values accordingly.

15.12.5.6 GrowthFrame.Growth.UpdateWidgets (*self*)

15.12.6 Member Data Documentation

15.12.6.1 GrowthFrame.Growth.alphaMort

15.12.6.2 GrowthFrame.Growth.alphStr

15.12.6.3 GrowthFrame.Growth.dredgeWdStr

15.12.6.4 GrowthFrame.Growth.dredgeWth

15.12.6.5 GrowthFrame.Growth.fishMort

15.12.6.6 GrowthFrame.Growth.fmortFileName

15.12.6.7 GrowthFrame.Growth.fmortStr

15.12.6.8 GrowthFrame.Growth.friend

15.12.6.9 GrowthFrame.Growth.gbAdultMort

15.12.6.10 GrowthFrame.Growth.gbAdultMortStr

15.12.6.11 GrowthFrame.Growth.gbClFSelAStr

15.12.6.12 GrowthFrame.Growth.gbClFSelBStr

15.12.6.13 GrowthFrame.Growth.gbClosedFSelectA

15.12.6.14 GrowthFrame.Growth.gbClosedFSelectB

15.12.6.15 GrowthFrame.Growth.gbCullSize

15.12.6.16 GrowthFrame.Growth.gbCullStr

15.12.6.17 GrowthFrame.Growth.gbDiscard

15.12.6.18 GrowthFrame.Growth.gbDiscStr

15.12.6.19 GrowthFrame.Growth.gbIncident

15.12.6.20 GrowthFrame.Growth.gbIncidStr

15.12.6.21 GrowthFrame.Growth.gbLen0Str

15.12.6.22 GrowthFrame.Growth.gbLength0

15.12.6.23 GrowthFrame.Growth.gbOpenFSelectA

15.12.6.24 GrowthFrame.Growth.gbOpenFSelectB

15.12.6.25 GrowthFrame.Growth.gbOpFSelAStr

15.12.6.26 GrowthFrame.Growth.gbOpFSelBStr

15.12.6.27 GrowthFrame.Growth.growthCfgFile

15.12.6.28 GrowthFrame.Growth.growthStartDir

15.12.6.29 GrowthFrame.Growth.loadGrowthConfigButton

15.12.6.30 GrowthFrame.Growth.lpueIntcept

15.12.6.31 GrowthFrame.Growth.lpueIntcStr

15.12.6.32 GrowthFrame.Growth.lpueSl2Str

15.12.6.33 GrowthFrame.Growth.lpueSlope

15.12.6.34 GrowthFrame.Growth.lpueSlope2

15.12.6.35 GrowthFrame.Growth.lpueSlStr

15.12.6.36 GrowthFrame.Growth.maAdultMort

15.12.6.37 GrowthFrame.Growth.maAdultMortStr

15.12.6.38 GrowthFrame.Growth.maCullSize

15.12.6.39 GrowthFrame.Growth.maCullStr

15.12.6.40 GrowthFrame.Growth.maDiscard

15.12.6.41 GrowthFrame.Growth.maDiscStr

15.12.6.42 GrowthFrame.Growth.maFSelAStr

15.12.6.43 GrowthFrame.Growth.maFSelBStr

15.12.6.44 GrowthFrame.Growth.maFSelectA

15.12.6.45 GrowthFrame.Growth.maFSelectB

15.12.6.46 GrowthFrame.Growth.maIncident

15.12.6.47 GrowthFrame.Growth.maIncidStr

15.12.6.48 GrowthFrame.Growth.maLen0Str

15.12.6.49 GrowthFrame.Growth.maLength0

15.12.6.50 GrowthFrame.Growth.maxPerDay

15.12.6.51 GrowthFrame.Growth.maxPerDayStr

15.12.6.52 GrowthFrame.Growth.maxTime

15.12.6.53 GrowthFrame.Growth.maxTimeStr

15.12.6.54 GrowthFrame.Growth.on_visibility

15.12.6.55 GrowthFrame.Growth.root

15.12.6.56 GrowthFrame.Growth.saveGrowthConfigButton

15.12.6.57 GrowthFrame.Growth.towSpdStr

15.12.6.58 GrowthFrame.Growth.towSpeed

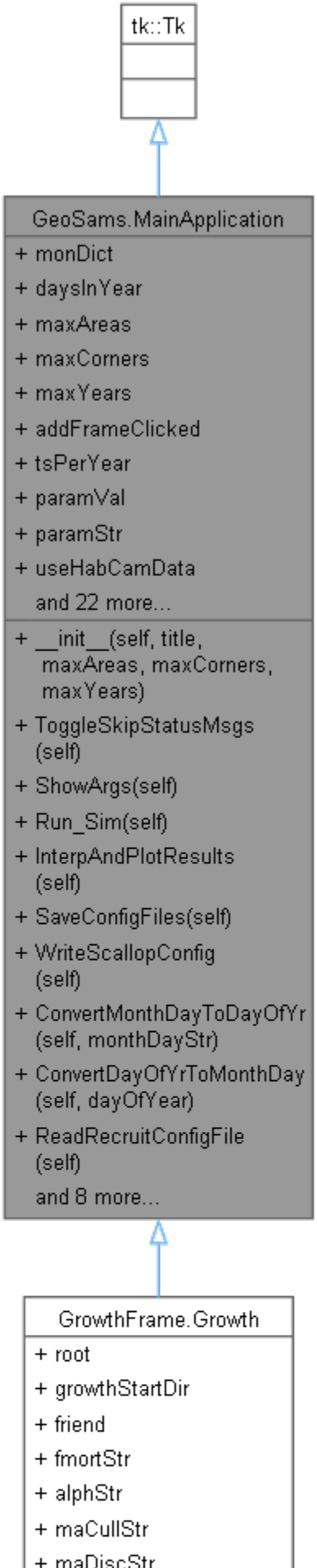
15.12.6.59 The documentation for this class was generated from the following file:

15.12.6.60 GrowthFrame.py

15.13 GeoSams.MainApplication Class Reference

This class is the parent class for the GUI.

Inheritance diagram for GeoSams.MainApplication:


tk::Tk
GeoSams.MainApplication
+ monDict
+ daysInYear
+ maxAreas
+ maxCorners
+ maxYears
+ addFrameClicked
+ tsPerYear
+ paramVal
+ paramStr
+ useHabCamData
and 22 more...
+ __init__(self, title, maxAreas, maxCorners, maxYears)
+ ToggleSkipStatusMsgs (self)
+ ShowArgs(self)
+ Run_Sim(self)
+ InterpAndPlotResults (self)
+ SaveConfigFiles(self)
+ WriteScallopConfig (self)
+ ConvertMonthDayToDayOfYr (self, monthDayStr)
+ ConvertDayOfYrToMonthDay (self, dayOfYear)
+ ReadRecruitConfigFile (self)
and 8 more...
GrowthFrame.Growth
+ root
+ growthStartDir
+ friend
+ fmortStr
+ alphStr
+ maCullStr

15.13.1 Public Member Functions

- **__init__** (self, title, **maxAreas**, **maxCorners**, **maxYears**)
- **ToggleSkipStatusMsgs** (self)
- **ShowArgs** (self)
  Display setup limits here Messagebox blocks entry widgets if attempted to open before the main window completes.

- **Run_Sim** (self)
  Starts the GeoSAMS simulatation ***ScallopPopDensity*** .

- **InterpAndPlotResults** (self)
- **SaveConfigFiles** (self)
  Save all of the defined configuration files.

- **WriteScallopConfig** (self)
  Saves simulation configuration file.

- **ConvertMonthDayToDayOfYr** (self, monthDayStr)
  This method is used to converty the recruitment start and stop dates from a string month numerical day into days in a year.

- **ConvertDayOfYrToMonthDay** (self, dayOfYear)
- **ReadRecruitConfigFile** (self)
  Read in the (tag, value) parameters from the recruitment to update parameters.

- **WriteRecruitmentConfig** (self)
  Saves recruitment parameters to a configuration file.

- **WriteGrowthConfig** (self)
  Saves mortality parameters to a configuration file.

- **WriteGridMgrConfig** (self)
  Saves grid manager parameters to a configuration file.

- **WriteSpatialFncsConfig** (self, cfgFile, functions, numFncsEntry)
  Saves spatial function parameters to a configuration file.

- **ReadConfigFile** (self, fName)
  Reads a typical configuration file to recover the tags and values.

- **ReadSimConfigFile** (self)
  Read in the (tag, value) parameters from the simulation configuration file.

- **ReadGridMgrConfigFile** (self)
  Read in the (tag, value) parameters from the grid manager configuration file.

- **pop_up** (self)

15.13.2 Public Attributes

- **monDictdaysInYear**
- **maxAreas**
- **maxCorners**
- **maxYears**
- **addFrameClicked**
- **tsPerYear**
- **paramVal**
- **paramStr**
- **useHabCamData**
- **domainName**
- **yearStart**
- **yearStop**
- **simConfigFile**
- **style**
- **root**
- **notebook**
- **frame1**
- **recrConfigFile**
- **frame2**
- **frame3**
- **frame5**
- **frame6**
- **frame7**
- **frame8**
- **gmConfigFile**
- **specAccFileStr**
- **isSkip**
- **skipStatusMsgs**
- **skipStatusMsgsRB**
- **recruitYrStrt**
- **recruitYrStop**

15.13.3 Detailed Description

This class is the parent class for the GUI.

15.13.4 Constructor & Destructor Documentation

15.13.4.1 GeoSams.MainApplication.__init__ (*self*, *title*, *maxAreas*, *maxCorners*, *maxYears*)

Reimplemented in **GrowthFrame.Growth** (*p.62*).

15.13.5 Member Function Documentation

15.13.5.1 GeoSams.MainApplication.ConvertDayOfYrToMonthDay (*self*, *dayOfYear*)

15.13.5.2 GeoSams.MainApplication.ConvertMonthDayToDayOfYr (*self*, *monthDayStr*)

This method is used to converty the recruitment start and stop dates from a string month numerical day into days in a year.

Changed entry to combo box to guarantee format

15.13.5.2.1 Parameters

monthDayStr	string that holds month and day in alpha format. That is 'JAN 01'

The Growth year starts on June 1st, actually May 31 at 2400 Jun 1st @ 0600 is day 0.25 which is = 0.25 /365.2425 = 0.00068 years June 1st @ 1200 is day 0.50 which is = 0.50 /365.2425 = 0.00137

June 1st @ 1800 is day 0.75 which is = 0.75 /365.2425 = 0.00205

June 1st @ 2359 is day 0.99 which is = 0.99931/365.2425 = 0.002736

Jun2 2nd @ 0000 is day 1 which is = 1.00000/365.2425 = 0.00274

Jun2 2nd @ 2400 is day 2 which is = 2.00000/365.2425 = 0.00548

Dec 31st @ 2400 is day 214 which is = 214. /365.2425 = 0.58591

Jan 1st @ 2400 is day 215 which is = 215. /365.2425 = 0.58865

= 1 + DayOfYear(12,31) - DayOfYear(5,31) Apr 10 @ 2400 is day 314 which is = 314. /365.2425 = 0.85970

if leap year 315 which is = 315. /365.2425 = 0.86244 However, leap year is handled in the main loop in which it is considered only for the current year

15.13.5.3 GeoSams.MainApplication.InterpAndPlotResults (*self*)

15.13.5.4 GeoSams.MainApplication.pop_up (*self*)

Reimplemented in **GrowthFrame.Growth** (*p.62*).

15.13.5.5 GeoSams.MainApplication.ReadConfigFile (*self*, *fName*)

Reads a typical configuration file to recover the tags and values.

The parameters in these files all have the following format:

- # indicates that the line is a comment. Otherwise
- 'tag' = 'value

15.13.5.5.1 Parameters

fName	The name of the file to read.

15.13.5.5.2 Returns

An array of tuples showing (tag, value) found in the file

15.13.5.6 GeoSams.MainApplication.ReadGridMgrConfigFile (*self*)

Read in the (tag, value) parameters from the grid manager configuration file.

15.13.5.7 GeoSams.MainApplication.ReadRecruitConfigFile (*self*)

Read in the (tag, value) parameters from the recruitment to update parameters.

15.13.5.8 GeoSams.MainApplication.ReadSimConfigFile (*self*)

Read in the (tag, value) parameters from the simulation configuration file.

15.13.5.9 GeoSams.MainApplication.Run_Sim (*self*)

Starts the GeoSAMS simulatation **ScallopPopDensity** .

1a) TrawlData5mmbin(Year, 'DN') this will Delete bin5mm2022AL.csv INPUT: OriginalData\dredgetowbysize7917.csv OUTPUT: Data\bin5mm<yyyy><dn>.csv

15.13.5.10 GeoSams.MainApplication.SaveConfigFiles (*self*)

Save all of the defined configuration files.

15.13.5.11 GeoSams.MainApplication.ShowArgs (*self*)

Display setup limits here Messagebox blocks entry widgets if attempted to open before the main window completes.

15.13.5.12 GeoSams.MainApplication.ToggleSkipStatusMsgs (*self*)

15.13.5.13 GeoSams.MainApplication.WriteGridMgrConfig (*self*)

Saves grid manager parameters to a configuration file.

15.13.5.14 GeoSams.MainApplication.WriteGrowthConfig (*self*)

Saves mortality parameters to a configuration file.

15.13.5.15 GeoSams.MainApplication.WriteRecruitmentConfig (*self*)

Saves recruitment parameters to a configuration file.

15.13.5.16 GeoSams.MainApplication.WriteScallopConfig (*self*)

Saves simulation configuration file.

It does so by writeing the parameters for the to the name file as well as keeping helpfule comments.

15.13.5.17 GeoSams.MainApplication.WriteSpatialFncsConfig (*self*, *cfgFile*, *functions*, *numFncsEntry*)

Saves spatial function parameters to a configuration file.

15.13.6 Member Data Documentation

15.13.6.1 GeoSams.MainApplication.addFrameClicked

15.13.6.2 GeoSams.MainApplication.daysInYear

15.13.6.3 GeoSams.MainApplication.domainName

15.13.6.4 GeoSams.MainApplication.frame1

15.13.6.5 GeoSams.MainApplication.frame2

15.13.6.6 GeoSams.MainApplication.frame3

15.13.6.7 GeoSams.MainApplication.frame5

15.13.6.8 GeoSams.MainApplication.frame6

15.13.6.9 GeoSams.MainApplication.frame7

15.13.6.10 GeoSams.MainApplication.frame8

15.13.6.11 GeoSams.MainApplication.gmConfigFile

15.13.6.12 GeoSams.MainApplication.isSkip

15.13.6.13 GeoSams.MainApplication.maxAreas

15.13.6.14 GeoSams.MainApplication.maxCorners

15.13.6.15 GeoSams.MainApplication.maxYears

15.13.6.16 GeoSams.MainApplication.monDict

15.13.6.17 GeoSams.MainApplication.notebook

15.13.6.18 GeoSams.MainApplication.paramStr

15.13.6.19 GeoSams.MainApplication.paramVal

15.13.6.20 GeoSams.MainApplication.recrConfigFile

15.13.6.21 GeoSams.MainApplication.recruitYrStop

15.13.6.22 GeoSams.MainApplication.recruitYrStrt

15.13.6.23 GeoSams.MainApplication.root

15.13.6.24 GeoSams.MainApplication.simConfigFile

15.13.6.25 GeoSams.MainApplication.skipStatusMsgs

15.13.6.26 GeoSams.MainApplication.skipStatusMsgsRB

15.13.6.27 GeoSams.MainApplication.specAccFileStr

15.13.6.28 GeoSams.MainApplication.style

15.13.6.29 GeoSams.MainApplication.tsPerYear

15.13.6.30 GeoSams.MainApplication.useHabCamData

15.13.6.31 GeoSams.MainApplication.yearStart

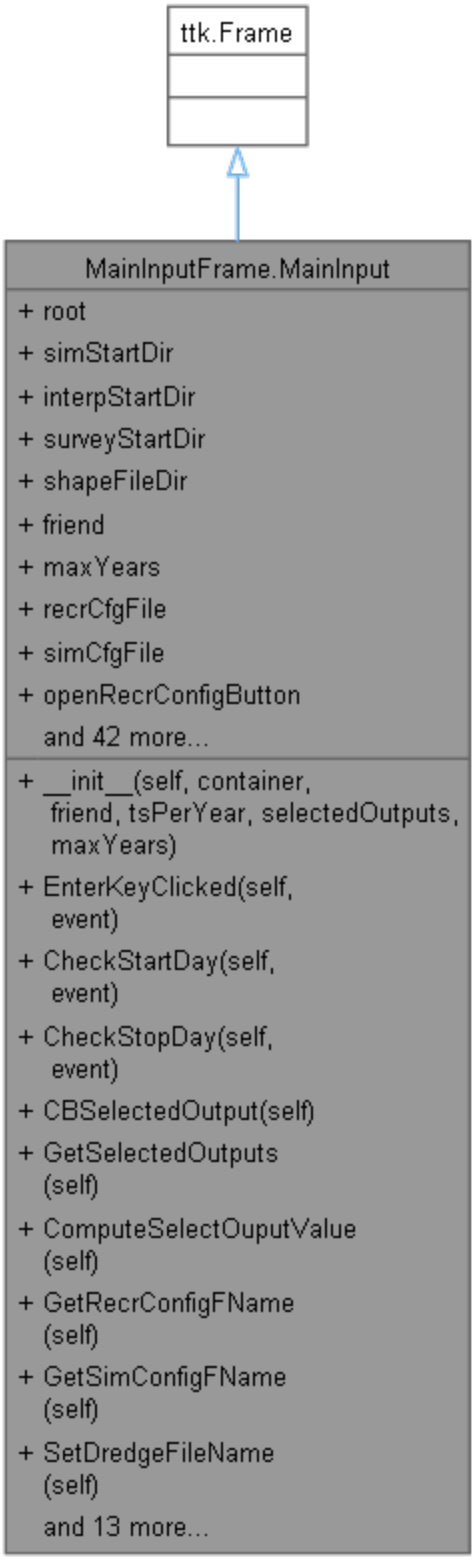
15.13.6.32 GeoSams.MainApplication.yearStop

15.13.6.33 The documentation for this class was generated from the following file:

15.13.6.34 GeoSams.py

15.14 MainInputFrame.MainInput Class Reference

This class displays information about GeoSAMS simulation.
Inheritance diagram for MainInputFrame.MainInput:



15.14.1 Public Member Functions

- **__init__** (self, container, **friend**, **tsPerYear**, selectedOutputs, **maxYears**)

- **EnterKeyClicked** (self, event)
  This method is called on both Enter Key clicked and goes out of focus.

- **CheckStartDay** (self, event)
  Checks start day to validate date is appropriate for month.

- **CheckStopDay** (self, event)
  Checks stop day to validate date is appropriate for month.

- **CBSelectedOutput** (self)
  Determines the value for which outputs are selected as they are checked.

- **GetSelectedOutputs** (self)
  Updates the final value from which outputs are selected.

- **ComputeSelectOuputValue** (self)
  Bit shifts (multiplies) checkbuttons and computes bit position value.

- **GetRecrConfigFName** (self)
  Calls the filedialog method asksaveasfilename to name a file to be used for the Recruitment Configuration file.

- **GetSimConfigFName** (self)
  Calls the filedialog method asksaveasfilename to name a file to be used for the Simulation Configuration file.

- **SetDredgeFileName** (self)
- **SetDredgeFileEnvVar** (self)
- **SetHabCamFileName** (self)
- **SetHabCamFileEnvVar** (self)
- **OpenPDF** (self)
- **SetMaShapeFile** (self)
- **SetMaShapeFileEnvVar** (self)
- **SetGbShapeFile** (self)
- **SetGBShapeFileEnvVar** (self)
- **SetMaShapeBufferFile** (self)
- **SetMaShapeBufferFileEnvVar** (self)
- **SetGbShapeBufferFile** (self)
- **SetGBShapeBufferFileEnvVar** (self)
- **pop_up** (self)

15.14.2 Public Attributes

- **rootsimStartDir**
- **interpStartDir**
- **surveyStartDir**
- **shapeFileDir**
- **friend**

- **maxYears**
- **recrCfgFile**
- **simCfgFile**
- **openRecrConfigButton**
- **openSimConfigButton**
- **openPDFButton**
- **dredgeDataFile**
- **habCamDataFile**
- **setDredgeDataButton**
- **setHabCamDataButton**
- **maShapeFileEntry**
- **gbShapeFileEntry**
- **maShapeBufferFileEntry**
- **gbShapeBufferFileEntry**
- **setMaShapeFileButton**
- **setGbShapeFileButton**
- **setMaShapeBufferFileButton**
- **setGbShapeBufferFileButton**
- **monthsArr**
- **startDayLabel**
- **startDayComboMonth**
- **CheckStartDay**
- **startDayComboDay**
- **stopDayLabel**
- **stopDayComboMonth**
- **CheckStopDay**
- **stopDayComboDay**
- **recrYrStrt**
- **recrYrStop**
- **numYrsAvg**
- **startYr**
- **stopYr**
- **tsPerYear**
- **domainNameLabel**
- **domainNameCombo**
- **lpueVar**
- **ebmsVar**
- **bmsVar**
- **abunVar**
- **lndwVar**
- **landVar**
- **feffVar**
- **fmortVar**
- **recrVar**
- **desiredOutput**
- **areaFName**

15.14.3 Detailed Description

This class displays information about GeoSAMS simulation.

This same information is used on the command line when starting SRC\ScallopPopDensity

15.14.4 Constructor & Destructor Documentation

15.14.4.1 MainInputFrame.MainInput.__init__ (*self*, *container*, *friend*, *tsPerYear*, *selectedOutputs*, *maxYears*)

15.14.5 Member Function Documentation

15.14.5.1 MainInputFrame.MainInput.CBSelectedOutput (*self*)

Determines the value for which outputs are selected as they are checked.

15.14.5.2 MainInputFrame.MainInput.CheckStartDay (*self*, *event*)

Checks start day to validate date is appropriate for month.

Does not consider if leap year

15.14.5.3 MainInputFrame.MainInput.CheckStopDay (*self*, *event*)

Checks stop day to validate date is appropriate for month.

Does not consider if leap year

15.14.5.4 MainInputFrame.MainInput.ComputeSelectOuputValue (*self*)

Bit shifts (multiplies) checkbuttons and computes bit position value.

15.14.5.5 MainInputFrame.MainInput.EnterKeyClicked (*self*, *event*)

This method is called on both Enter Key clicked and goes out of focus.

15.14.5.6 MainInputFrame.MainInput.GetRecrConfigFName (*self*)

Calls the filedialog method asksaveasfilename to name a file to be used for the Recruitment Configuration file.

It then writes out the defined parameters to this file using the 'tag = value' format.

15.14.5.7 MainInputFrame.MainInput.GetSelectedOutputs (*self*)

Updates the final value from which outputs are selected.

15.14.5.8 MainInputFrame.MainInput.GetSimConfigFName (*self*)

Calls the filedialog method asksaveasfilename to name a file to be used for the Simulation Configuration file.

It then writes out the defined parameters to this file using the 'tag = value' format.

15.14.5.9 MainInputFrame.MainInput.OpenPDF (*self*)

15.14.5.10 MainInputFrame.MainInput.pop_up (*self*)

15.14.5.11 MainInputFrame.MainInput.SetDredgeFileEnvVar (*self*)

15.14.5.12 MainInputFrame.MainInput.SetDredgeFileName (*self*)

15.14.5.13 MainInputFrame.MainInput.SetGbShapeBufferFile (*self*)

15.14.5.14 MainInputFrame.MainInput.SetGBShapeBufferFileEnvVar (*self*)

15.14.5.15 MainInputFrame.MainInput.SetGbShapeFile (*self*)

15.14.5.16 MainInputFrame.MainInput.SetGBShapeFileEnvVar (*self*)

15.14.5.17 MainInputFrame.MainInput.SetHabCamFileEnvVar (*self*)

15.14.5.18 MainInputFrame.MainInput.SetHabCamFileName (*self*)

15.14.5.19 MainInputFrame.MainInput.SetMaShapeBufferFile (*self*)

15.14.5.20 MainInputFrame.MainInput.SetMaShapeBufferFileEnvVar (*self*)

15.14.5.21 MainInputFrame.MainInput.SetMaShapeFile (*self*)

15.14.5.22 MainInputFrame.MainInput.SetMaShapeFileEnvVar (*self*)

15.14.6 Member Data Documentation

15.14.6.1 MainInputFrame.MainInput.abunVar

15.14.6.2 MainInputFrame.MainInput.areaFName

15.14.6.3 MainInputFrame.MainInput.bmsVar

15.14.6.4 MainInputFrame.MainInput.CheckStartDay

15.14.6.5 MainInputFrame.MainInput.CheckStopDay

15.14.6.6 MainInputFrame.MainInput.desiredOutput

15.14.6.7 MainInputFrame.MainInput.domainNameCombo

15.14.6.8 MainInputFrame.MainInput.domainNameLabel

15.14.6.9 MainInputFrame.MainInput.dredgeDataFile

15.14.6.10 MainInputFrame.MainInput.ebmsVar

15.14.6.11 MainInputFrame.MainInput.feffVar

15.14.6.12 MainInputFrame.MainInput.fmortVar

15.14.6.13 MainInputFrame.MainInput.friend

15.14.6.14 MainInputFrame.MainInput.gbShapeBufferFileEntry

15.14.6.15 MainInputFrame.MainInput.gbShapeFileEntry

15.14.6.16 MainInputFrame.MainInput.habCamDataFile

15.14.6.17 MainInputFrame.MainInput.interpStartDir

15.14.6.18 MainInputFrame.MainInput.landVar

15.14.6.19 MainInputFrame.MainInput.lndwVar

15.14.6.20 MainInputFrame.MainInput.lpueVar

15.14.6.21 MainInputFrame.MainInput.maShapeBufferFileEntry

15.14.6.22 MainInputFrame.MainInput.maShapeFileEntry

15.14.6.23 MainInputFrame.MainInput.maxYears

15.14.6.24 MainInputFrame.MainInput.monthsArr

15.14.6.25 MainInputFrame.MainInput.numYrsAvg

15.14.6.26 MainInputFrame.MainInput.openPDFButton

15.14.6.27 MainInputFrame.MainInput.openRecrConfigButton

15.14.6.28 MainInputFrame.MainInput.openSimConfigButton

15.14.6.29 MainInputFrame.MainInput.recrCfgFile

15.14.6.30 MainInputFrame.MainInput.recrVar

15.14.6.31 MainInputFrame.MainInput.recrYrStop

15.14.6.32 MainInputFrame.MainInput.recrYrStrt

15.14.6.33 MainInputFrame.MainInput.root

15.14.6.34 MainInputFrame.MainInput.setDredgeDataButton

15.14.6.35 MainInputFrame.MainInput.setGbShapeBufferFileButton

15.14.6.36 MainInputFrame.MainInput.setGbShapeFileButton

15.14.6.37 MainInputFrame.MainInput.setHabCamDataButton

15.14.6.38 MainInputFrame.MainInput.setMaShapeBufferFileButton

15.14.6.39 MainInputFrame.MainInput.setMaShapeFileButton

15.14.6.40 MainInputFrame.MainInput.shapeFileDir

15.14.6.41 MainInputFrame.MainInput.simCfgFile

15.14.6.42 MainInputFrame.MainInput.simStartDir

15.14.6.43 MainInputFrame.MainInput.startDayComboDay

15.14.6.44 MainInputFrame.MainInput.startDayComboMonth

15.14.6.45 MainInputFrame.MainInput.startDayLabel

15.14.6.46 MainInputFrame.MainInput.startYr

15.14.6.47 MainInputFrame.MainInput.stopDayComboDay

15.14.6.48 MainInputFrame.MainInput.stopDayComboMonth

15.14.6.49 MainInputFrame.MainInput.stopDayLabel

15.14.6.50 MainInputFrame.MainInput.stopYr

15.14.6.51 MainInputFrame.MainInput.surveyStartDir

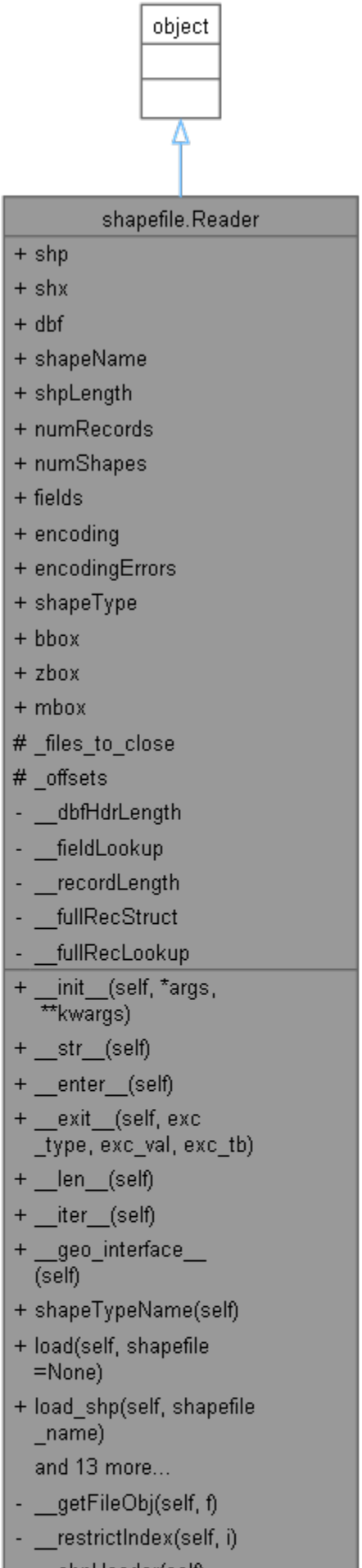
15.14.6.52 MainInputFrame.MainInput.tsPerYear

15.14.6.53 The documentation for this class was generated from the following file:

15.14.6.54 MainInputFrame.py

15.15 shapefile.Reader Class Reference

Inheritance diagram for shapefile.Reader:


object
shapefile.Reader
+ shp
+ shx
+ dbf
+ shapeName
+ shpLength
+ numRecords
+ numShapes
+ fields
+ encoding
+ encodingErrors
+ shapeType
+ bbox
+ zbox
+ mbox
# _files_to_close
# _offsets
- __dbfHdrLength
- __fieldLookup
- __recordLength
- __fullRecStruct
- __fullRecLookup
+ __init__(self, *args, **kwargs)
+ __str__(self)
+ __enter__(self)
+ __exit__(self, exc_type, exc_val, exc_tb)
+ __len__(self)
+ __iter__(self)
+ __geo_interface__(self)
+ shapeTypeName(self)
+ load(self, shapefile=None)
+ load_shp(self, shapefile_name)
and 13 more...
- __getFileObj(self, f)
- __restrictIndex(self, i)

15.15.1 Public Member Functions

- **__init__** (self, *args, **kwargs)
- **__str__** (self)
- **__enter__** (self)
- **__exit__** (self, exc_type, exc_val, exc_tb)
- **__len__** (self)
- **__iter__** (self)
- **__geo_interface__** (self)
- **shapeTypeName** (self)
- **load** (self, shapefile=None)
- **load_shp** (self, shapefile_name)
- **load_shx** (self, shapefile_name)
- **load_dbf** (self, shapefile_name)
- **__del__** (self)
- **close** (self)
- **shape** (self, i=0, **bbox**=None)
- **shapes** (self, **bbox**=None)
- **iterShapes** (self, **bbox**=None)
- **record** (self, i=0, **fields**=None)
- **records** (self, **fields**=None)
- **iterRecords** (self, **fields**=None)
- **shapeRecord** (self, i=0, **fields**=None, **bbox**=None)
- **shapeRecords** (self, **fields**=None, **bbox**=None)
- **iterShapeRecords** (self, **fields**=None, **bbox**=None)

15.15.2 Public Attributes

- **shpshx**
- **dbf**
- **shapeName**
- **shpLength**
- **numRecords**
- **numShapes**
- **fields**
- **encoding**
- **encodingErrors**
- **shapeType**
- **bbox**
- **zbox**
- **mbox**

15.15.3 Protected Attributes

- **_files_to_close_offsets**

15.15.4 Private Member Functions

- **__getFileObj** (self, f)
- **__restrictIndex** (self, i)
- **__shpHeader** (self)
- **__shape** (self, oid=None, **bbox**=None)
- **__shxHeader** (self)
- **__shxOffsets** (self)
- **__shapeIndex** (self, i=None)

- **__dbfHeader** (self)
- **__recordFmt** (self, **fields**=None)
- **__recordFields** (self, **fields**=None)
- **__record** (self, fieldTuples, recLookup, recStruct, oid=None)

15.15.5 Private Attributes

- **__dbfHdrLength__fieldLookup**
- **__recordLength**
- **__fullRecStruct**
- **__fullRecLookup**

15.15.6 Detailed Description

```
Reads the three files of a shapefile as a unit or
separately.  If one of the three files (.shp, .shx,
.dbf) is missing no exception is thrown until you try
to call a method that depends on that particular file.
The .shx index file is used if available for efficiency
but is not required to read the geometry from the .shp
file. The "shapefile" argument in the constructor is the
name of the file you want to open, and can be the path
to a shapefile on a local filesystem, inside a zipfile,
or a url.

You can instantiate a Reader without specifying a shapefile
and then specify one later with the load() method.

Only the shapefile headers are read upon loading. Content
within each file is only accessed when required and as
efficiently as possible. Shapefiles are usually not large
but they can be.
```

15.15.7 Constructor & Destructor Documentation

15.15.7.1 shapefile.Reader.__init__ (*self*, * *args*, ** *kwargs*)

15.15.7.2 shapefile.Reader.__del__ (*self*)

15.15.8 Member Function Documentation

15.15.8.1 shapefile.Reader.__dbfHeader (*self*)`[private]`

Reads a dbf header. Xbase-related code borrows heavily from ActiveState Python Cookbook Recipe 362715 by Raymond Hettinger

15.15.8.2 shapefile.Reader.__enter__ (*self*)

```
Enter phase of context manager.
```

15.15.8.3 shapefile.Reader.__exit__ (*self*, *exc_type*, *exc_val*, *exc_tb*)

```
Exit phase of context manager, close opened files.
```

15.15.8.4 shapefile.Reader.__geo_interface__ (*self*)

15.15.8.5 shapefile.Reader.__getFileObj (*self*, *f*)[private]

```
Checks to see if the requested shapefile file object is
```
available. If not a ShapefileException is raised.

15.15.8.6 shapefile.Reader.__iter__ (*self*)

Iterates through the shapes/records in the shapefile.

15.15.8.7 shapefile.Reader.__len__ (*self*)

Returns the number of shapes/records in the shapefile.

15.15.8.8 shapefile.Reader.__record (*self*, *fieldTuples*, *recLookup*, *recStruct*, *oid* = None)[private]

```
Reads and returns a dbf record row as a list of values. Requires specifying
a list of field info tuples 'fieldTuples', a record name-index dict 'recLookup',
and a Struct instance 'recStruct' for unpacking these fields.
```

15.15.8.9 shapefile.Reader.__recordFields (*self*, *fields* = None)[private]

```
Returns the necessary info required to unpack a record's fields,
restricted to a subset of fieldnames 'fields' if specified.
Returns a list of field info tuples, a name-index lookup dict,
and a Struct instance for unpacking these fields. Note that DeletionFlag
is not a valid field.
```

15.15.8.10 shapefile.Reader.__recordFmt (*self*, *fields* = None)[private]

```
Calculates the format and size of a .dbf record. Optional 'fields' arg
specifies which fieldnames to unpack and which to ignore. Note that this
always includes the DeletionFlag at index 0, regardless of the 'fields' arg.
```

15.15.8.11 shapefile.Reader.__restrictIndex (*self*, *i*)[private]

```
Provides list-like handling of a record index with a clearer
```
error message if the index is out of bounds.

15.15.8.12 shapefile.Reader.__shape (*self*, *oid* = None, *bbox* = None) [private]

Returns the header info and geometry for a single shape.

15.15.8.13 shapefile.Reader.__shapeIndex (*self*, *i* = None) [private]

```
Returns the offset in a .shp file for a shape based on information
```
in the .shx index file.

15.15.8.14 shapefile.Reader.__shpHeader (*self*) [private]

Reads the header information from a .shp file.

15.15.8.15 shapefile.Reader.__shxHeader (*self*) [private]

Reads the header information from a .shx file.

15.15.8.16 shapefile.Reader.__shxOffsets (*self*) [private]

Reads the shape offset positions from a .shx file

15.15.8.17 shapefile.Reader.__str__ (*self*)

```
Use some general info on the shapefile as __str__
```

15.15.8.18 shapefile.Reader.close (*self*)

15.15.8.19 shapefile.Reader.iterRecords (*self*, *fields* = None)

```
Returns a generator of records in a dbf file.
Useful for large shapefiles or dbf files.
To only read some of the fields, specify the 'fields' arg as a
list of one or more fieldnames.
```

15.15.8.20 shapefile.Reader.iterShapeRecords (*self*, *fields* = None, *bbox* = None)

```
Returns a generator of combination geometry/attribute records for
all records in a shapefile.
To only read some of the fields, specify the 'fields' arg as a
list of one or more fieldnames.
To only read entries within a given spatial region, specify the 'bbox'
arg as a list or tuple of xmin,ymin,xmax,ymax.
```

15.15.8.21 shapefile.Reader.iterShapes (*self*, *bbox* = None)

```
Returns a generator of shapes in a shapefile. Useful
for handling large shapefiles.
To only read shapes within a given spatial region, specify the 'bbox'
arg as a list or tuple of xmin,ymin,xmax,ymax.
```

15.15.8.22 shapefile.Reader.load (*self*, *shapefile* = None)

```
Opens a shapefile from a filename or file-like
object. Normally this method would be called by the
```

constructor with the file name as an argument.

15.15.8.23 shapefile.Reader.load_dbf (*self*, *shapefile_name*)

```
Attempts to load file with .dbf extension as both lower and upper case
```

15.15.8.24 shapefile.Reader.load_shp (*self*, *shapefile_name*)

```
Attempts to load file with .shp extension as both lower and upper case
```

15.15.8.25 shapefile.Reader.load_shx (*self*, *shapefile_name*)

```
Attempts to load file with .shx extension as both lower and upper case
```

15.15.8.26 shapefile.Reader.record (*self*, *i* = 0, *fields* = None)

```
Returns a specific dbf record based on the supplied index.
To only read some of the fields, specify the 'fields' arg as a
list of one or more fieldnames.
```

15.15.8.27 shapefile.Reader.records (*self*, *fields* = None)

```
Returns all records in a dbf file.
To only read some of the fields, specify the 'fields' arg as a
list of one or more fieldnames.
```

15.15.8.28 shapefile.Reader.shape (*self*, *i* = 0, *bbox* = None)

```
Returns a shape object for a shape in the geometry
record file.
If the 'bbox' arg is given (list or tuple of xmin,ymin,xmax,ymax),
returns None if the shape is not within that region.
```

15.15.8.29 shapefile.Reader.shapeRecord (*self*, *i* = 0, *fields* = None, *bbox* = None)

```
Returns a combination geometry and attribute record for the
supplied record index.
To only read some of the fields, specify the 'fields' arg as a
list of one or more fieldnames.
If the 'bbox' arg is given (list or tuple of xmin,ymin,xmax,ymax),
returns None if the shape is not within that region.
```

15.15.8.30 shapefile.Reader.shapeRecords (*self*, *fields* = None, *bbox* = None)

```
Returns a list of combination geometry/attribute records for
all records in a shapefile.
To only read some of the fields, specify the 'fields' arg as a
list of one or more fieldnames.
To only read entries within a given spatial region, specify the 'bbox'
arg as a list or tuple of xmin,ymin,xmax,ymax.
```

15.15.8.31 shapefile.Reader.shapes (*self*, *bbox* = None)

```
Returns all shapes in a shapefile.
To only read shapes within a given spatial region, specify the 'bbox'
arg as a list or tuple of xmin,ymin,xmax,ymax.
```

15.15.8.32 shapefile.Reader.shapeTypeName (*self*)

15.15.9 Member Data Documentation

15.15.9.1 shapefile.Reader.__dbfHdrLength `[private]`

15.15.9.2 shapefile.Reader.__fieldLookup `[private]`

15.15.9.3 shapefile.Reader.__fullRecLookup `[private]`

15.15.9.4 shapefile.Reader.__fullRecStruct `[private]`

15.15.9.5 shapefile.Reader.__recordLength `[private]`

15.15.9.6 shapefile.Reader._files_to_close `[protected]`

15.15.9.7 shapefile.Reader._offsets `[protected]`

15.15.9.8 shapefile.Reader.bbox

15.15.9.9 shapefile.Reader.dbf

15.15.9.10 shapefile.Reader.encoding

15.15.9.11 shapefile.Reader.encodingErrors

15.15.9.12 shapefile.Reader.fields

15.15.9.13 shapefile.Reader.mbox

15.15.9.14 shapefile.Reader.numRecords

15.15.9.15 shapefile.Reader.numShapes

15.15.9.16 shapefile.Reader.shapeName

15.15.9.17 shapefile.Reader.shapeType

15.15.9.18 shapefile.Reader.shp

15.15.9.19 shapefile.Reader.shpLength

15.15.9.20 shapefile.Reader.shx

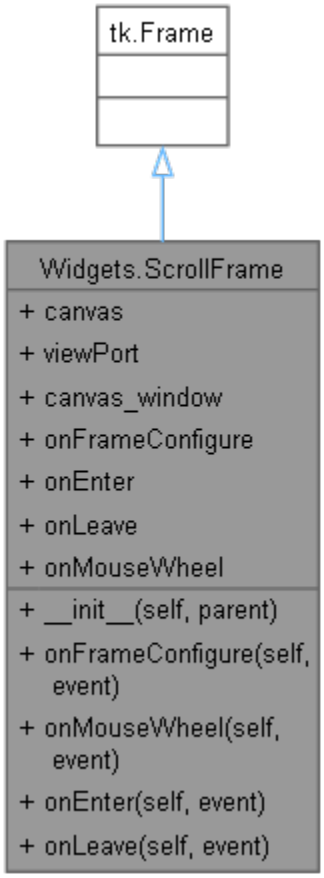
15.15.9.21 shapefile.Reader.zbox

15.15.9.22 The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

15.16 Widgets.ScrollFrame Class Reference

Scrollable Frame Class from
`https://gist.github.com/mp035/9f2027c3ef9172264532fcd6262f3b01`.
Inheritance diagram for Widgets.ScrollFrame:



15.16.1 Public Member Functions

- **__init__** (self, parent)
- **onFrameConfigure** (self, event)
  whenever the size of the frame changes, alter the scroll region respectively.

- **onMouseWheel** (self, event)
  cross platform scroll wheel event

- **onEnter** (self, event)
  bind wheel events when the cursor enters the control

- **onLeave** (self, event)
  unbind wheel events when the cursorl leaves the control

15.16.2 Public Attributes

- **canvasviewPort**
- **canvas_window**
- **onFrameConfigure**
- **onEnter**
- **onLeave**
- **onMouseWheel**

15.16.3 Detailed Description

Scrollable Frame Class from `https://gist.github.com/mp035/9f2027c3ef9172264532fcd6262f3b01`.

15.16.4 Constructor & Destructor Documentation

15.16.4.1 Widgets.ScrollFrame.__init__ (*self*, *parent*)

15.16.5 Member Function Documentation

15.16.5.1 Widgets.ScrollFrame.onEnter (*self*, *event*)

bind wheel events when the cursor enters the control

15.16.5.2 Widgets.ScrollFrame.onFrameConfigure (*self*, *event*)

whenever the size of the frame changes, alter the scroll region respectively.

Reset the scroll region to encompass the inner frame

15.16.5.3 Widgets.ScrollFrame.onLeave (*self*, *event*)

unbind wheel events when the cursorl leaves the control

15.16.5.4 Widgets.ScrollFrame.onMouseWheel (*self*, *event*)

cross platform scroll wheel event

15.16.6 Member Data Documentation

15.16.6.1 Widgets.ScrollFrame.canvas

15.16.6.2 Widgets.ScrollFrame.canvas_window

15.16.6.3 Widgets.ScrollFrame.onEnter

15.16.6.4 Widgets.ScrollFrame.onFrameConfigure

15.16.6.5 Widgets.ScrollFrame.onLeave

15.16.6.6 Widgets.ScrollFrame.onMouseWheel

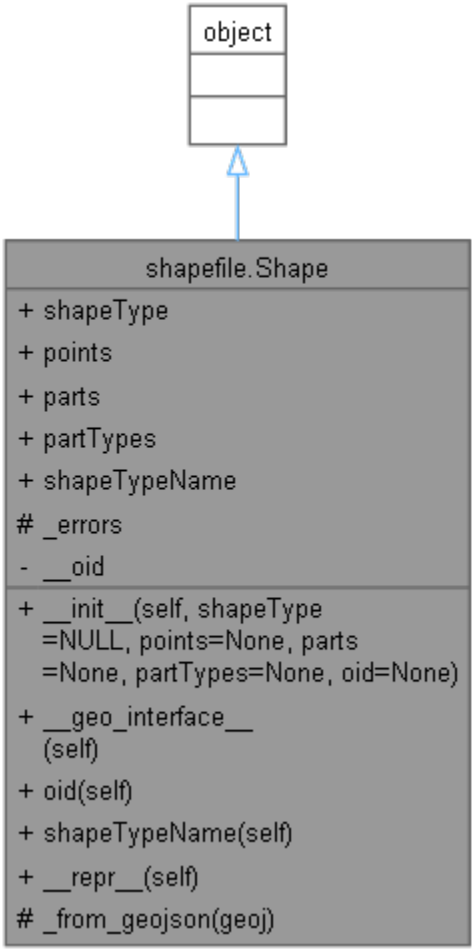
15.16.6.7 Widgets.ScrollFrame.viewPort

15.16.6.8 The documentation for this class was generated from the following file:

15.16.6.9 Widgets.py

15.17 shapefile.Shape Class Reference

Inheritance diagram for shapefile.Shape:



15.17.1 Public Member Functions

- **__init__** (self, **shapeType**=**NULL**, **points**=None, **parts**=None, **partTypes**=None, **oid**=None)
- **__geo_interface__** (self)
- **oid** (self)
- **shapeTypeName** (self)
- **__repr__** (self)

15.17.2 Public Attributes

- **shapeTypepoints**
- **parts**
- **partTypes**
- **shapeTypeName**

15.17.3 Static Protected Member Functions

- **_from_geojson** (geoj)

15.17.4 Protected Attributes

15.17.5 _errorsPrivate Attributes

__oid

15.17.6 Constructor & Destructor Documentation

15.17.6.1 shapefile.Shape.__init__ (*self*, *shapeType* = `NULL`, *points* = `None`, *parts* = `None`, *partTypes* = `None`, *oid* = `None`)

```
Stores the geometry of the different shape types
specified in the Shapefile spec. Shape types are
usually point, polyline, or polygons. Every shape type
except the "Null" type contains points at some level for
example vertices in a polygon. If a shape type has
multiple shapes containing points within a single
geometry record then those shapes are called parts. Parts
are designated by their starting index in geometry record's
list of shapes. For MultiPatch geometry, partTypes designates
the patch type of each of the parts.
```

15.17.7 Member Function Documentation

15.17.7.1 shapefile.Shape.__geo_interface__ (*self*)

15.17.7.2 shapefile.Shape.__repr__ (*self*)

15.17.7.3 shapefile.Shape._from_geojson (*geoj*)`[static], [protected]`

15.17.7.4 shapefile.Shape.oid (*self*)

The index position of the shape in the original shapefile

15.17.7.5 shapefile.Shape.shapeTypeName (*self*)

15.17.8 Member Data Documentation

15.17.8.1 shapefile.Shape.__oid [private]

15.17.8.2 shapefile.Shape._errors [protected]

15.17.8.3 shapefile.Shape.parts

15.17.8.4 shapefile.Shape.partTypes

15.17.8.5 shapefile.Shape.points

15.17.8.6 shapefile.Shape.shapeType

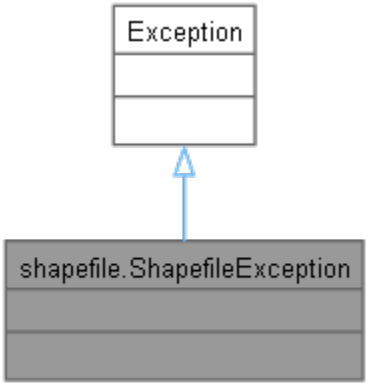
15.17.8.7 shapefile.Shape.shapeTypeName

15.17.8.8 The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

15.18 shapefile.ShapefileException Class Reference

Inheritance diagram for shapefile.ShapefileException:



15.18.1 Detailed Description

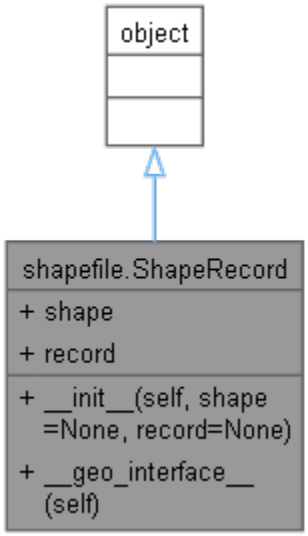
An exception to handle shapefile specific problems.

The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

15.19 shapefile.ShapeRecord Class Reference

Inheritance diagram for shapefile.ShapeRecord:



15.19.1 Public Member Functions

- **__init__** (self, **shape**=None, **record**=None)
- **__geo_interface__** (self)

15.19.2 Public Attributes

- **shaperecord**

15.19.3 Detailed Description

```
A ShapeRecord object containing a shape along with its attributes.
```

Provides the GeoJSON __geo_interface__ to return a Feature dictionary.

15.19.4 Constructor & Destructor Documentation

15.19.4.1 shapefile.ShapeRecord.__init__ (*self*, *shape* = `None`, *record* = `None`)

15.19.5 Member Function Documentation

15.19.5.1 shapefile.ShapeRecord.__geo_interface__ (*self*)

15.19.6 Member Data Documentation

15.19.6.1 shapefile.ShapeRecord.record

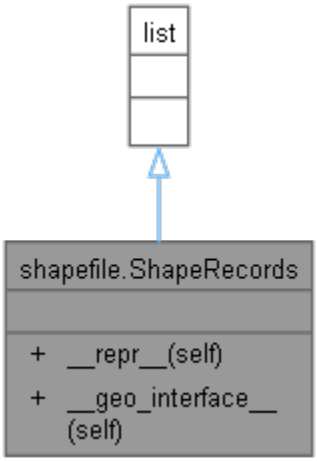
15.19.6.2 shapefile.ShapeRecord.shape

15.19.6.3 The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

15.20 shapefile.ShapeRecords Class Reference

Inheritance diagram for shapefile.ShapeRecords:



15.20.1 Public Member Functions

- **__repr__** (self)
- **__geo_interface__** (self)

15.20.2 Detailed Description

```
A class to hold a list of ShapeRecord objects. Subclasses list to ensure compatibility
with
former work and to reuse all the optimizations of the builtin list.
In addition to the list interface, this also provides the GeoJSON __geo_interface__
```

to return a FeatureCollection dictionary.

15.20.3 Member Function Documentation

15.20.3.1 shapefile.ShapeRecords.__geo_interface__ (*self*)

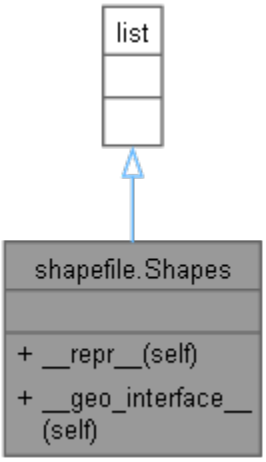
15.20.3.2 shapefile.ShapeRecords.__repr__ (*self*)

15.20.3.3 The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

15.21 shapefile.Shapes Class Reference

Inheritance diagram for shapefile.Shapes:



15.21.1 Public Member Functions

- **__repr__** (self)
- **__geo_interface__** (self)

15.21.2 Detailed Description

```
A class to hold a list of Shape objects. Subclasses list to ensure compatibility with
former work and to reuse all the optimizations of the builtin list.
In addition to the list interface, this also provides the GeoJSON __geo_interface__
```

to return a GeometryCollection dictionary.

15.21.3 Member Function Documentation

15.21.3.1 shapefile.Shapes.__geo_interface__ (*self*)

15.21.3.2 shapefile.Shapes.__repr__ (*self*)

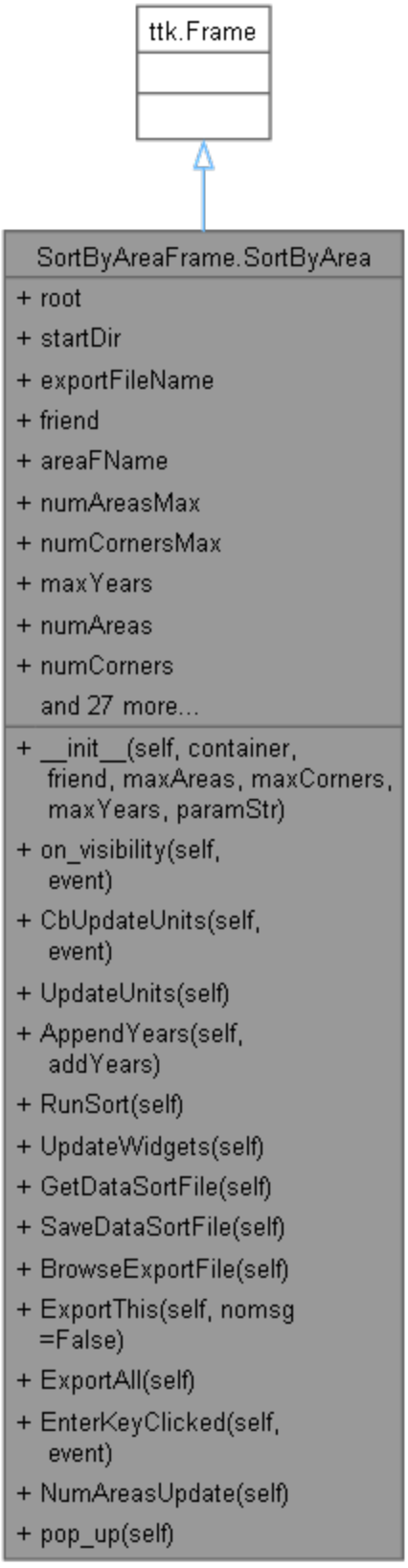
15.21.3.3 The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

15.22 SortByAreaFrame.SortByArea Class Reference

This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

Inheritance diagram for SortByAreaFrame.SortByArea:


ttk.Frame
SortByAreaFrame.SortByArea
+ root
+ startDir
+ exportFileName
+ friend
+ areaFName
+ numAreasMax
+ numCornersMax
+ maxYears
+ numAreas
+ numCorners
and 27 more...
+ __init__(self, container, friend, maxAreas, maxCorners, maxYears, paramStr)
+ on_visibility(self, event)
+ CbUpdateUnits(self, event)
+ UpdateUnits(self)
+ AppendYears(self, addYears)
+ RunSort(self)
+ UpdateWidgets(self)
+ GetDataSortFile(self)
+ SaveDataSortFile(self)
+ BrowseExportFile(self)
+ ExportThis(self, nomsg =False)
+ ExportAll(self)
+ EnterKeyClicked(self, event)
+ NumAreasUpdate(self)
+ pop_up(self)

15.22.1 Public Member Functions

- **__init__** (self, container, **friend**, maxAreas, maxCorners, **maxYears**, **paramStr**)
- **on_visibility** (self, event)
- **CbUpdateUnits** (self, event)
- **UpdateUnits** (self)
- **AppendYears** (self, addYears)
- **RunSort** (self)
- **UpdateWidgets** (self)
- **GetDataSortFile** (self)
- **SaveDataSortFile** (self)
- **BrowseExportFile** (self)
- **ExportThis** (self, nomsg=False)
  This method exports the current page of data, just a single output parameter.

- **ExportAll** (self)
  Export all select parameters.

- **EnterKeyClicked** (self, event)
- **NumAreasUpdate** (self)
- **pop_up** (self)
  Help Window for Sort By Area.

15.22.2 Public Attributes

- **rootstartDir**
- **exportFileName**
- **friend**
- **areaFName**
- **numAreasMax**
- **numCornersMax**
- **maxYears**
- **numAreas**
  f.write('AREA,YEAR,' + outStr + ' ('+ units + ')
  ') Write Header

- **numCornersparamStr**
- **yearStart**
  f.write('AREA,YEAR,' + outStr + ' ('+ units + ')
  ') Write Header

- **yearStopnumYears**
  f.write('AREA,YEAR,' + outStr + ' ('+ units + ')
  ') Write Header

- **areaDatascrollFrame**
- **numAreasLabel**
- **numAreasEntry**

- **EnterKeyClicked**
- **outputParmLabel**
- **comboParameter**
- **CbUpdateUnits**
- **dataSortFileLabel**
- **dataSortFileEntry**
- **dataSortUnitsLabel**
- **dataSortUnitsEntry**
- **openDataSortButton**
- **saveDataSortButton**
- **runSortButton**
- **exportThisSortButton**
- **exportAllSortButton**
- **exportFileLabel**
- **exportFileEntry**
- **browseExportButton**
- **areas**
- **on_visibility**
- **domainName**

15.22.3 Detailed Description

This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

15.22.3.1.1 Parameters

friend	is used to access Start and Stop year from the Main Frame
maxAreas	defined at start up for the maximum allowed areas
maxCorners	defined at start up for the maximum allowed corners or nodes
maxYears	defined at start up for the maximum allowed year range
paramStr	defined at start up for the desired outputs

15.22.4 Constructor & Destructor Documentation

15.22.4.1 SortByAreaFrame.SortByArea.__init__ (*self*, *container*, *friend*, *maxAreas*, *maxCorners*, *maxYears*, *paramStr*)

15.22.5 Member Function Documentation

15.22.5.1 SortByAreaFrame.SortByArea.AppendYears (*self*, *addYears*)

15.22.5.2 SortByAreaFrame.SortByArea.BrowseExportFile (*self*)

15.22.5.3 SortByAreaFrame.SortByArea.CbUpdateUnits (*self*, *event*)

15.22.5.4 SortByAreaFrame.SortByArea.EnterKeyClicked (*self*, *event*)

15.22.5.5 SortByAreaFrame.SortByArea.ExportAll (*self*)

Export all select parameters.

For each parameter

- Verify data file exists, Lat_Lon_Grid_ + ABUN_ + AL + _ 2015_2017

15.22.5.6 SortByAreaFrame.SortByArea.ExportThis (*self*, *nomsg* = `False`)

This method exports the current page of data, just a single output parameter.

First row : AREA YEAR PARAMETER (UNITS) 1 StartYear

1 ... 1 StopYear ... ... N StartYear

N ... N StopYear

15.22.5.7 SortByAreaFrame.SortByArea.GetDataSortFile (*self*)

15.22.5.8 SortByAreaFrame.SortByArea.NumAreasUpdate (*self*)

Updates the number of areas functions.

15.22.5.9 SortByAreaFrame.SortByArea.on_visibility (*self*, *event*)

15.22.5.10 SortByAreaFrame.SortByArea.pop_up (*self*)

Help Window for Sort By Area.

15.22.5.11 SortByAreaFrame.SortByArea.RunSort (*self*)

15.22.5.12 SortByAreaFrame.SortByArea.SaveDataSortFile (*self*)

15.22.5.13 SortByAreaFrame.SortByArea.UpdateUnits (*self*)

15.22.5.14 SortByAreaFrame.SortByArea.UpdateWidgets (*self*)

15.22.6 Member Data Documentation

15.22.6.1 SortByAreaFrame.SortByArea.areaData

15.22.6.2 SortByAreaFrame.SortByArea.areaFName

15.22.6.3 SortByAreaFrame.SortByArea.areas

15.22.6.4 SortByAreaFrame.SortByArea.browseExportButton

15.22.6.5 SortByAreaFrame.SortByArea.CbUpdateUnits

15.22.6.6 SortByAreaFrame.SortByArea.comboParameter

15.22.6.7 SortByAreaFrame.SortByArea.dataSortFileEntry

15.22.6.8 SortByAreaFrame.SortByArea.dataSortFileLabel

15.22.6.9 SortByAreaFrame.SortByArea.dataSortUnitsEntry

15.22.6.10 SortByAreaFrame.SortByArea.dataSortUnitsLabel

15.22.6.11 SortByAreaFrame.SortByArea.domainName

15.22.6.12 SortByAreaFrame.SortByArea.EnterKeyClicked

15.22.6.13 SortByAreaFrame.SortByArea.exportAllSortButton

15.22.6.14 SortByAreaFrame.SortByArea.exportFileEntry

15.22.6.15 SortByAreaFrame.SortByArea.exportFileLabel

15.22.6.16 SortByAreaFrame.SortByArea.exportFileName

15.22.6.17 SortByAreaFrame.SortByArea.exportThisSortButton

15.22.6.18 SortByAreaFrame.SortByArea.friend

15.22.6.19 SortByAreaFrame.SortByArea.maxYears

15.22.6.20 SortByAreaFrame.SortByArea.numAreas

```
f.write('AREA,YEAR,' + outStr + ' ('+ units + ')
') Write Header
```

15.22.6.21 SortByAreaFrame.SortByArea.numAreasEntry

15.22.6.22 SortByAreaFrame.SortByArea.numAreasLabel

15.22.6.23 SortByAreaFrame.SortByArea.numAreasMax

15.22.6.24 SortByAreaFrame.SortByArea.numCorners

15.22.6.25 SortByAreaFrame.SortByArea.numCornersMax

15.22.6.26 SortByAreaFrame.SortByArea.numYears

```
f.write('AREA,YEAR,' + outStr + ' ('+ units + ')
') Write Header
```

15.22.6.27 SortByAreaFrame.SortByArea.on_visibility

15.22.6.28 SortByAreaFrame.SortByArea.openDataSortButton

15.22.6.29 SortByAreaFrame.SortByArea.outputParmLabel

15.22.6.30 SortByAreaFrame.SortByArea.paramStr

15.22.6.31 SortByAreaFrame.SortByArea.root

15.22.6.32 SortByAreaFrame.SortByArea.runSortButton

15.22.6.33 SortByAreaFrame.SortByArea.saveDataSortButton

15.22.6.34 SortByAreaFrame.SortByArea.scrollFrame

15.22.6.35 SortByAreaFrame.SortByArea.startDir

15.22.6.36 SortByAreaFrame.SortByArea.yearStart

```
f.write('AREA,YEAR,' + outStr + ' ('+ units + ')
') Write Header
```

15.22.6.37 SortByAreaFrame.SortByArea.yearStop

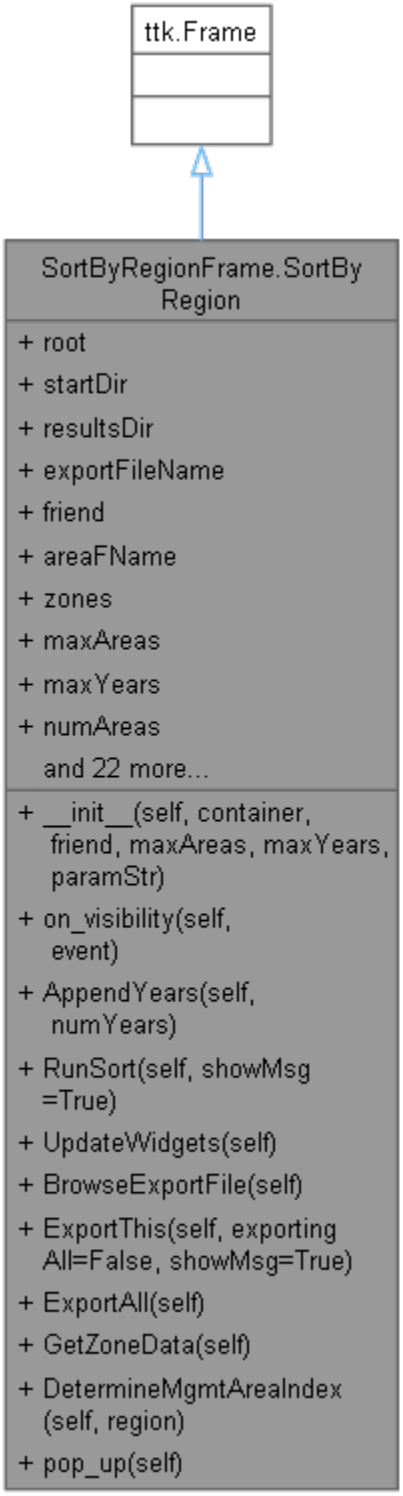
15.22.6.38 The documentation for this class was generated from the following file:

15.22.6.39 SortByAreaFrame.py

15.23 SortByRegionFrame.SortByRegion Class Reference

This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

Inheritance diagram for SortByRegionFrame.SortByRegion:



15.23.1 Public Member Functions

- **__init__** (self, container, **friend**, **maxAreas**, **maxYears**, **paramStr**)
- **on_visibility** (self, event)
- **AppendYears** (self, **numYears**)
  Max Number of years has increased, need to add additional columns.

- **RunSort** (self, showMsg=True)
- **UpdateWidgets** (self)
- **BrowseExportFile** (self)
- **ExportThis** (self, exportingAll=False, showMsg=True)
  This method exports ouput parameter table to its own file name.

- **ExportAll** (self)
  Export all select parameters.

- **GetZoneData** (self)
  Gets shape data and places it into a array of GeoShape.

- **DetermineMgmtAreaIndex** (self, region)
  determine Management Area Index

- **pop_up** (self)
  Help Window for Sort By Area.

15.23.2 Public Attributes

- **rootstartDir**
- **resultsDir**
- **exportFileName**
- **friend**
- **areaFName**
- **zones**
- **maxAreas**
- **maxYears**
- **numAreas**
- **numCorners**
- **paramStr**
- **areaKm2**
- **yearStart**
- **yearStop**
- **numYears**
- **scrollFrame**
- **sortAreaFrame**
- **outputParmLabel**
- **comboParameter**
- **runSortButton**
- **exportThisSortButton**
- **exportAllSortButton**

- **exportFileLabel**
- **exportFileEntry**
- **browseExportButton**
- **tableRows**
- **tableCols**
- **firstYrCol**
- **table**
- **on_visibility**
- **domainName**

15.23.3 Detailed Description

This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

15.23.3.1.1 Parameters

friend	is used to access Start and Stop year from the Main Frame
maxAreas	defined at start up for the maximum allowed areas
maxCorners	defined at start up for the maximum allowed corners or nodes
maxYears	defined at start up for the maximum allowed year range
paramStr	defined at start up for the desired outputs

15.23.4 Constructor & Destructor Documentation

15.23.4.1 SortByRegionFrame.SortByRegion.__init__ (*self*, *container*, *friend*, *maxAreas*, *maxYears*, *paramStr*)

15.23.5 Member Function Documentation

15.23.5.1 SortByRegionFrame.SortByRegion.AppendYears (*self*, *numYears*)

Max Number of years has increased, need to add additional columns.

15.23.5.2 SortByRegionFrame.SortByRegion.BrowseExportFile (*self*)

15.23.5.3 SortByRegionFrame.SortByRegion.DetermineMgmtAreaIndex (*self*, *region*)

determine Management Area Index

15.23.5.4 SortByRegionFrame.SortByRegion.ExportAll (*self*)

Export all select parameters.

For each parameter

- Verify data file exists, Lat_Lon_Grid_ + ABUN_ + AL + _ 2015_2017

15.23.5.5 SortByRegionFrame.SortByRegion.ExportThis (*self*, *exportingAll* = `False`, *showMsg* = `True`)

This method exports ouput parameter table to its own file name.

15.23.5.6 SortByRegionFrame.SortByRegion.GetZoneData (*self*)

Gets shape data and places it into a array of GeoShape.

MA regions, if selected, are first placed into array followed by GB regions, selected. Logic will always have either MA, GB, or both

15.23.5.7 SortByRegionFrame.SortByRegion.on_visibility (*self*, *event*)

15.23.5.8 SortByRegionFrame.SortByRegion.pop_up (*self*)

Help Window for Sort By Area.

15.23.5.9 SortByRegionFrame.SortByRegion.RunSort (*self*, *showMsg* = `True`)

15.23.5.10 SortByRegionFrame.SortByRegion.UpdateWidgets (*self*)

15.23.6 Member Data Documentation

15.23.6.1 SortByRegionFrame.SortByRegion.areaFName

15.23.6.2 SortByRegionFrame.SortByRegion.areaKm2

15.23.6.3 SortByRegionFrame.SortByRegion.browseExportButton

15.23.6.4 SortByRegionFrame.SortByRegion.comboParameter

15.23.6.5 SortByRegionFrame.SortByRegion.domainName

15.23.6.6 SortByRegionFrame.SortByRegion.exportAllSortButton

15.23.6.7 SortByRegionFrame.SortByRegion.exportFileEntry

15.23.6.8 SortByRegionFrame.SortByRegion.exportFileLabel

15.23.6.9 SortByRegionFrame.SortByRegion.exportFileName

15.23.6.10 SortByRegionFrame.SortByRegion.exportThisSortButton

15.23.6.11 SortByRegionFrame.SortByRegion.firstYrCol

15.23.6.12 SortByRegionFrame.SortByRegion.friend

15.23.6.13 SortByRegionFrame.SortByRegion.maxAreas

15.23.6.14 SortByRegionFrame.SortByRegion.maxYears

15.23.6.15 SortByRegionFrame.SortByRegion.numAreas

15.23.6.16 SortByRegionFrame.SortByRegion.numCorners

15.23.6.17 SortByRegionFrame.SortByRegion.numYears

15.23.6.18 SortByRegionFrame.SortByRegion.on_visibility

15.23.6.19 SortByRegionFrame.SortByRegion.outputParmLabel

15.23.6.20 SortByRegionFrame.SortByRegion.paramStr

15.23.6.21 SortByRegionFrame.SortByRegion.resultsDir

15.23.6.22 SortByRegionFrame.SortByRegion.root

15.23.6.23 SortByRegionFrame.SortByRegion.runSortButton

15.23.6.24 SortByRegionFrame.SortByRegion.scrollFrame

15.23.6.25 SortByRegionFrame.SortByRegion.sortAreaFrame

15.23.6.26 SortByRegionFrame.SortByRegion.startDir

15.23.6.27 SortByRegionFrame.SortByRegion.table

15.23.6.28 SortByRegionFrame.SortByRegion.tableCols

15.23.6.29 SortByRegionFrame.SortByRegion.tableRows

15.23.6.30 SortByRegionFrame.SortByRegion.yearStart

15.23.6.31 SortByRegionFrame.SortByRegion.yearStop

15.23.6.32 SortByRegionFrame.SortByRegion.zones

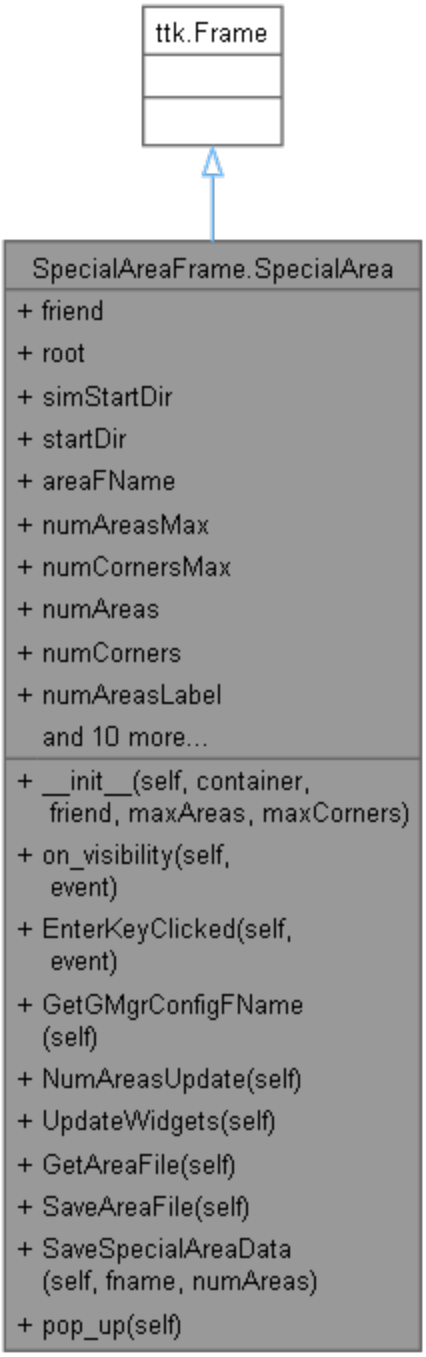
15.23.6.33 The documentation for this class was generated from the following file:

15.23.6.34 SortByRegionFrame.py

15.24 SpecialAreaFrame.SpecialArea Class Reference

This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

Inheritance diagram for SpecialAreaFrame.SpecialArea:



15.24.1 Public Member Functions

- **__init__** (self, container, **friend**, maxAreas, maxCorners)
- **on_visibility** (self, event)
- **EnterKeyClicked** (self, event)
- **GetGMgrConfigFName** (self)
  Calls the filedialog method asksaveasfilename to name a file to be used for the Grid Manager Configuration file.

- **NumAreasUpdate** (self)
- **UpdateWidgets** (self)
- **GetAreaFile** (self)
- **SaveAreaFile** (self)
- **SaveSpecialAreaData** (self, fname, **numAreas**)
- **pop_up** (self)
  Help Window for Special Access Area.

15.24.2 Public Attributes

- **friendroot**
- **simStartDir**
- **startDir**
- **areaFName**
- **numAreasMax**
- **numCornersMax**
- **numAreas**
- **numCorners**
- **numAreasLabel**
- **numAreasEntry**
- **EnterKeyClicked**
- **gmCfgFile**
- **openGmgrConfigButton**
- **openAreaFileButton**
- **specAccFile**
- **specAccFileLabel**
- **saveAreaFileButton**
- **areaMgr**
- **on_visibility**

15.24.3 Detailed Description

This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.

15.24.4 Constructor & Destructor Documentation

15.24.4.1 SpecialAreaFrame.SpecialArea.__init__ (*self*, *container*, *friend*, *maxAreas*, *maxCorners*)

15.24.5 Member Function Documentation

15.24.5.1 SpecialAreaFrame.SpecialArea.EnterKeyClicked (*self*, *event*)

15.24.5.2 SpecialAreaFrame.SpecialArea.GetAreaFile (*self*)

15.24.5.3 SpecialAreaFrame.SpecialArea.GetGMgrConfigFName (*self*)

Calls the filedialog method asksaveasfilename to name a file to be used for the Grid Manager Configuration file.

It then writes out the defined parameters to this file using the 'tag = value' format.

15.24.5.4 SpecialAreaFrame.SpecialArea.NumAreasUpdate (*self*)

15.24.5.5 SpecialAreaFrame.SpecialArea.on_visibility (*self*, *event*)

15.24.5.6 SpecialAreaFrame.SpecialArea.pop_up (*self*)

Help Window for Special Access Area.

15.24.5.7 SpecialAreaFrame.SpecialArea.SaveAreaFile (*self*)

15.24.5.8 SpecialAreaFrame.SpecialArea.SaveSpecialAreaData (*self*, *fname*, *numAreas*)

15.24.5.9 SpecialAreaFrame.SpecialArea.UpdateWidgets (*self*)

15.24.6 Member Data Documentation

15.24.6.1 SpecialAreaFrame.SpecialArea.areaFName

15.24.6.2 SpecialAreaFrame.SpecialArea.areaMgr

15.24.6.3 SpecialAreaFrame.SpecialArea.EnterKeyClicked

15.24.6.4 SpecialAreaFrame.SpecialArea.friend

15.24.6.5 SpecialAreaFrame.SpecialArea.gmCfgFile

15.24.6.6 SpecialAreaFrame.SpecialArea.numAreas

15.24.6.7 SpecialAreaFrame.SpecialArea.numAreasEntry

15.24.6.8 SpecialAreaFrame.SpecialArea.numAreasLabel

15.24.6.9 SpecialAreaFrame.SpecialArea.numAreasMax

15.24.6.10 SpecialAreaFrame.SpecialArea.numCorners

15.24.6.11 SpecialAreaFrame.SpecialArea.numCornersMax

15.24.6.12 SpecialAreaFrame.SpecialArea.on_visibility

15.24.6.13 SpecialAreaFrame.SpecialArea.openAreaFileButton

15.24.6.14 SpecialAreaFrame.SpecialArea.openGmgrConfigButton

15.24.6.15 SpecialAreaFrame.SpecialArea.root

15.24.6.16 SpecialAreaFrame.SpecialArea.saveAreaFileButton

15.24.6.17 SpecialAreaFrame.SpecialArea.simStartDir

15.24.6.18 SpecialAreaFrame.SpecialArea.specAccFile

15.24.6.19 SpecialAreaFrame.SpecialArea.specAccFileLabel

15.24.6.20 SpecialAreaFrame.SpecialArea.startDir

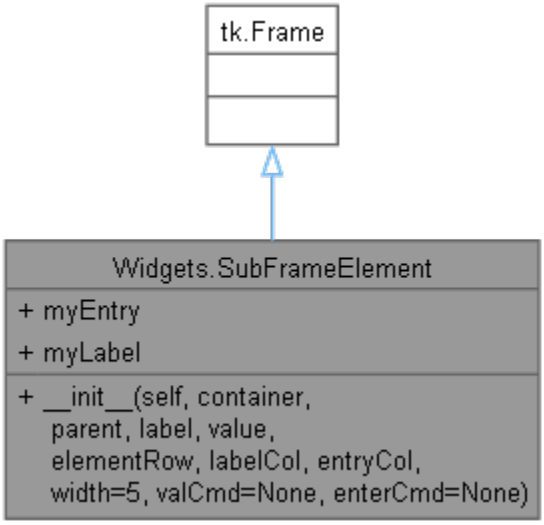
15.24.6.21 The documentation for this class was generated from the following file:

15.24.6.22 SpecialAreaFrame.py

15.25 Widgets.SubFrameElement Class Reference

Generic Element.

Inheritance diagram for Widgets.SubFrameElement:



15.25.1 Public Member Functions

- **__init__** (self, container, parent, label, value, elementRow, labelCol, entryCol, width=5, valCmd=None, enterCmd=None)

15.25.2 Public Attributes

- **myEntrymyLabel**

15.25.3 Detailed Description

Generic Element.

Provides a label and an entery field. Optionally allows programmer to specify a method to validate entry and another method to respond to Enter Key.

15.25.4 Constructor & Destructor Documentation

15.25.4.1 Widgets.SubFrameElement.__init__ (*self*, *container*, *parent*, *label*, *value*, *elementRow*, *labelCol*, *entryCol*, *width* = `5`, *valCmd* = `None`, *enterCmd* = `None`)

15.25.5 Member Data Documentation

15.25.5.1 Widgets.SubFrameElement.myEntry

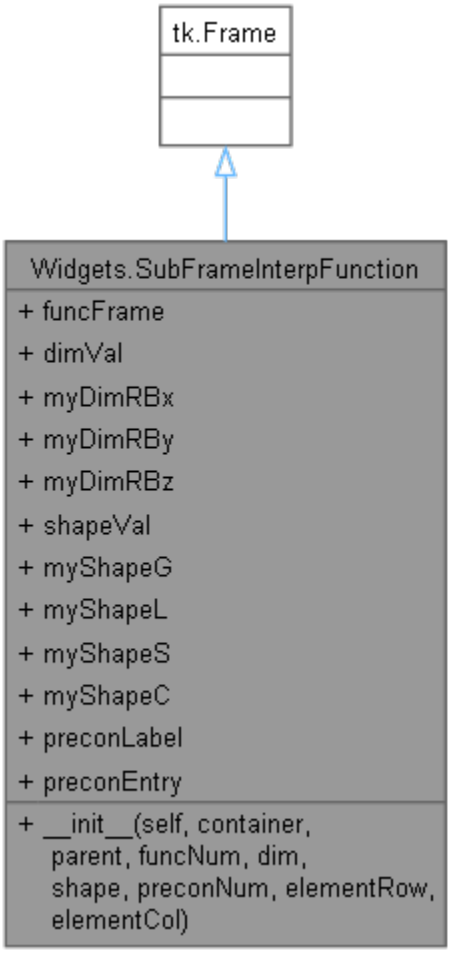
15.25.5.2 Widgets.SubFrameElement.myLabel

15.25.5.3 The documentation for this class was generated from the following file:

15.25.5.4 Widgets.py

15.26 Widgets.SubFrameInterpFunction Class Reference

Inheritance diagram for Widgets.SubFrameInterpFunction:



15.26.1 Public Member Functions

- **__init__** (self, container, parent, funcNum, dim, shape, preconNum, elementRow, elementCol)

15.26.2 Public Attributes

- **funcFramedimVal**
- **myDimRBx**

- **myDimRBy**
- **myDimRBz**
- **shapeVal**
- **myShapeG**
- **myShapeL**
- **myShapeS**
- **myShapeC**
- **preconLabel**
- **preconEntry**

15.26.3 Constructor & Destructor Documentation

15.26.3.1 Widgets.SubFrameInterpFunction.__init__ (*self*, *container*, *parent*, *funcNum*, *dim*, *shape*, *preconNum*, *elementRow*, *elementCol*)

15.26.4 Member Data Documentation

15.26.4.1 Widgets.SubFrameInterpFunction.dimVal

15.26.4.2 Widgets.SubFrameInterpFunction.funcFrame

15.26.4.3 Widgets.SubFrameInterpFunction.myDimRBx

15.26.4.4 Widgets.SubFrameInterpFunction.myDimRBy

15.26.4.5 Widgets.SubFrameInterpFunction.myDimRBz

15.26.4.6 Widgets.SubFrameInterpFunction.myShapeC

15.26.4.7 Widgets.SubFrameInterpFunction.myShapeG

15.26.4.8 Widgets.SubFrameInterpFunction.myShapeL

15.26.4.9 Widgets.SubFrameInterpFunction.myShapeS

15.26.4.10 Widgets.SubFrameInterpFunction.preconEntry

15.26.4.11 Widgets.SubFrameInterpFunction.preconLabel

15.26.4.12 Widgets.SubFrameInterpFunction.shapeVal

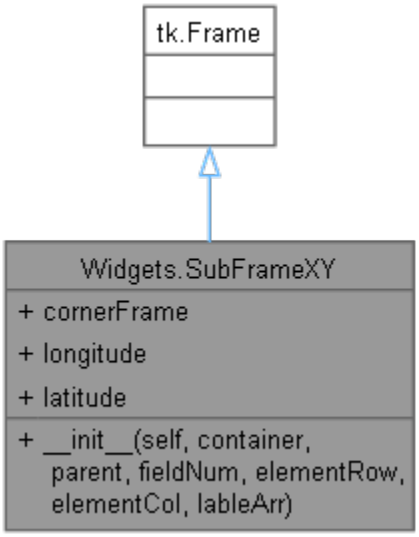
15.26.4.13 The documentation for this class was generated from the following file:

15.26.4.14 Widgets.py

15.27 Widgets.SubFrameXY Class Reference

Widget for XY label and entery.

Inheritance diagram for Widgets.SubFrameXY:



15.27.1 Public Member Functions

- **__init__** (self, container, parent, fieldNum, elementRow, elementCol, lableArr)

15.27.2 Public Attributes

- **cornerFramelongitude**
- **latitude**

15.27.3 Detailed Description

Widget for XY label and entery.

Longitude, Latitude have become interchangeable with X, Y

15.27.4 Constructor & Destructor Documentation

15.27.4.1 Widgets.SubFrameXY.__init__ (*self*, *container*, *parent*, *fieldNum*, *elementRow*, *elementCol*, *lableArr*)

15.27.5 Member Data Documentation

15.27.5.1 Widgets.SubFrameXY.cornerFrame

15.27.5.2 Widgets.SubFrameXY.latitude

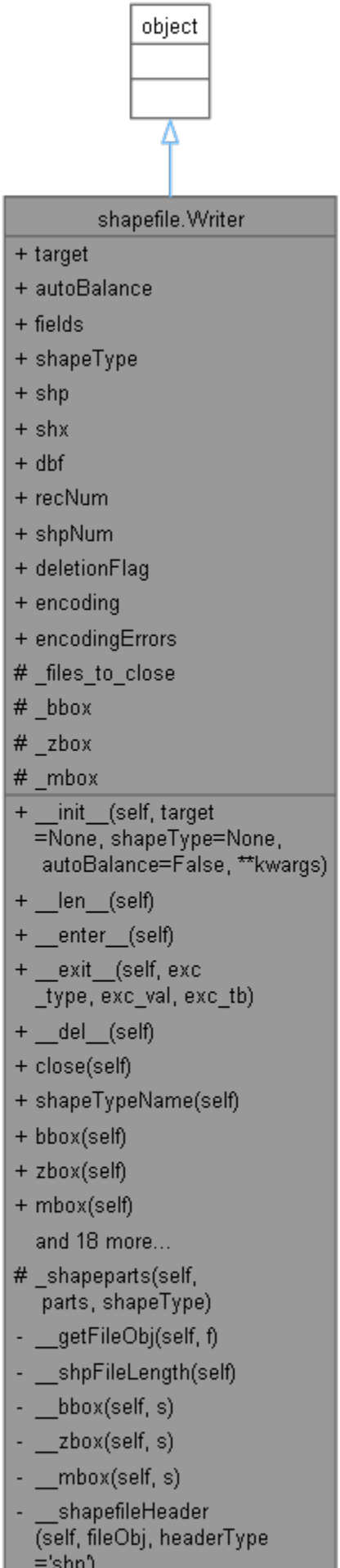
15.27.5.3 Widgets.SubFrameXY.longitude

15.27.5.4 The documentation for this class was generated from the following file:

15.27.5.5 Widgets.py

15.28 shapefile.Writer Class Reference

Inheritance diagram for shapefile.Writer:


object
shapefile.Writer
+ target
+ autoBalance
+ fields
+ shapeType
+ shp
+ shx
+ dbf
+ recNum
+ shpNum
+ deletionFlag
+ encoding
+ encodingErrors
# _files_to_close
# _bbox
# _zbox
# _mbox
+ __init__(self, target =None, shapeType=None, autoBalance=False, **kwargs)
+ __len__(self)
+ __enter__(self)
+ __exit__(self, exc _type, exc_val, exc_tb)
+ __del__(self)
+ close(self)
+ shapeTypeName(self)
+ bbox(self)
+ zbox(self)
+ mbox(self)
and 18 more...
_shapeparts(self, parts, shapeType)
- __getFileObj(self, f)
- __shpFileLength(self)
- __bbox(self, s)
- __zbox(self, s)
- __mbox(self, s)
- __shapefileHeader (self, fileObj, headerType ='shp')

15.28.1 Public Member Functions

- **__init__** (self, **target**=None, **shapeType**=None, **autoBalance**=False, **kwargs)
- **__len__** (self)
- **__enter__** (self)
- **__exit__** (self, exc_type, exc_val, exc_tb)
- **__del__** (self)
- **close** (self)
- **shapeTypeName** (self)
- **bbox** (self)
- **zbox** (self)
- **mbox** (self)
- **shape** (self, s)
- **record** (self, *recordList, **recordDict)
- **balance** (self)
- **null** (self)
- **point** (self, x, y)
- **pointm** (self, x, y, m=None)
- **pointz** (self, x, y, z=0, m=None)
- **multipoint** (self, points)
- **multipointm** (self, points)
- **multipointz** (self, points)
- **line** (self, lines)
- **linem** (self, lines)
- **linez** (self, lines)
- **poly** (self, polys)
- **polym** (self, polys)
- **polyz** (self, polys)
- **multipatch** (self, parts, partTypes)
- **field** (self, name, fieldType="C", size="50", decimal=0)

15.28.2 Public Attributes

- **targetautoBalance**
- **fields**
- **shapeType**
- **shp**
- **shx**
- **dbf**
- **recNum**
- **shpNum**
- **deletionFlag**
- **encoding**
- **encodingErrors**

15.28.3 Protected Member Functions

- **_shapeparts** (self, parts, **shapeType**)

15.28.4 Protected Attributes

- **_files_to_close_bbox**
- **_zbox**

- **_mbox**

15.28.5 Private Member Functions

- **__getFileObj** (self, f)
- **__shpFileLength** (self)
- **__bbox** (self, s)
- **__zbox** (self, s)
- **__mbox** (self, s)
- **__shapefileHeader** (self, fileObj, headerType='**shp**')
- **__dbfHeader** (self)
- **__shpRecord** (self, s)
- **__shxRecord** (self, offset, length)
- **__dbfRecord** (self, **record**)

15.28.6 Detailed Description

Provides write support for ESRI Shapefiles.

15.28.7 Constructor & Destructor Documentation

15.28.7.1 shapefile.Writer.__init__ (*self*, *target* = `None`, *shapeType* = `None`, *autoBalance* = `False`, ** *kwargs*)

15.28.7.2 shapefile.Writer.__del__ (*self*)

15.28.8 Member Function Documentation

15.28.8.1 shapefile.Writer.__bbox (*self*, *s*)`[private]`

15.28.8.2 shapefile.Writer.__dbfHeader (*self*)`[private]`

Writes the dbf header and field descriptors.

15.28.8.3 shapefile.Writer.__dbfRecord (*self*, *record*)`[private]`

Writes the dbf records.

15.28.8.4 shapefile.Writer.__enter__ (*self*)

```
Enter phase of context manager.
```

15.28.8.5 shapefile.Writer.__exit__ (*self*, *exc_type*, *exc_val*, *exc_tb*)

```
Exit phase of context manager, finish writing and close the files.
```

15.28.8.6 shapefile.Writer.__getFileObj (*self*, *f*)[private]

Safety handler to verify file-like objects

15.28.8.7 shapefile.Writer.__len__ (*self*)

```
Returns the current number of features written to the shapefile.
If shapes and records are unbalanced, the length is considered the highest
```
of the two.

15.28.8.8 shapefile.Writer.__mbox (*self*, *s*)[private]

15.28.8.9 shapefile.Writer.__shapefileHeader (*self*, *fileObj*, *headerType* = 'shp')[private]

```
Writes the specified header type to the specified file-like object.
Several of the shapefile formats are so similar that a single generic
```
method to read or write them is warranted.

15.28.8.10 shapefile.Writer.__shpFileLength (*self*)[private]

Calculates the file length of the shp file.

15.28.8.11 shapefile.Writer.__shpRecord (*self*, *s*)[private]

15.28.8.12 shapefile.Writer.__shxRecord (*self*, *offset*, *length*)[private]

Writes the shx records.

15.28.8.13 shapefile.Writer.__zbox (*self*, *s*)[private]

15.28.8.14 shapefile.Writer._shapeparts (*self*, *parts*, *shapeType*)[protected]

```
Internal method for adding a shape that has multiple collections of points (parts):
lines, polygons, and multipoint shapes.
```

15.28.8.15 shapefile.Writer.balance (*self*)

```
Adds corresponding empty attributes or null geometry records depending
on which type of record was created to make sure all three files
```
are in synch.

15.28.8.16 shapefile.Writer.bbox (*self*)

```
Returns the current bounding box for the shapefile which is
the lower-left and upper-right corners. It does not contain the
```
elevation or measure extremes.

15.28.8.17 shapefile.Writer.close (*self*)

```
Write final shp, shx, and dbf headers, close opened files.
```

15.28.8.18 shapefile.Writer.field (*self*, *name*, *fieldType* = "C", *size* = "50", *decimal* = 0)

Adds a dbf field descriptor to the shapefile.

15.28.8.19 shapefile.Writer.line (*self*, *lines*)

```
Creates a POLYLINE shape.
```
Lines is a collection of lines, each made up of a list of xy values.

15.28.8.20 shapefile.Writer.linem (*self*, *lines*)

```
Creates a POLYLINEM shape.
Lines is a collection of lines, each made up of a list of xym values.
```
If the m (measure) value is not included, it defaults to None (NoData).

15.28.8.21 shapefile.Writer.linez (*self*, *lines*)

```
Creates a POLYLINEZ shape.
Lines is a collection of lines, each made up of a list of xyzm values.
If the z (elevation) value is not included, it defaults to 0.
```
If the m (measure) value is not included, it defaults to None (NoData).

15.28.8.22 shapefile.Writer.mbox (*self*)

Returns the current m extremes for the shapefile.

15.28.8.23 shapefile.Writer.multipatch (*self*, *parts*, *partTypes*)

```
Creates a MULTIPATCH shape.
Parts is a collection of 3D surface patches, each made up of a list of xyzm values.
PartTypes is a list of types that define each of the surface patches.
The types can be any of the following module constants: TRIANGLE_STRIP,
TRIANGLE_FAN, OUTER_RING, INNER_RING, FIRST_RING, or RING.
If the z (elevation) value is not included, it defaults to 0.
```
If the m (measure) value is not included, it defaults to None (NoData).

15.28.8.24 shapefile.Writer.multipoint (*self*, *points*)

```
Creates a MULTIPOINT shape.
```

Points is a list of xy values.

15.28.8.25 shapefile.Writer.multipointm (*self*, *points*)

```
Creates a MULTIPOINTM shape.
Points is a list of xym values.
```

If the m (measure) value is not included, it defaults to None (NoData).

15.28.8.26 shapefile.Writer.multipointz (*self*, *points*)

```
Creates a MULTIPOINTZ shape.
Points is a list of xyzm values.
If the z (elevation) value is not included, it defaults to 0.
```

If the m (measure) value is not included, it defaults to None (NoData).

15.28.8.27 shapefile.Writer.null (*self*)

Creates a null shape.

15.28.8.28 shapefile.Writer.point (*self*, *x*, *y*)

Creates a POINT shape.

15.28.8.29 shapefile.Writer.pointm (*self*, *x*, *y*, *m* = `None`)

```
Creates a POINTM shape.
```

If the m (measure) value is not set, it defaults to NoData.

15.28.8.30 shapefile.Writer.pointz (*self*, *x*, *y*, *z* = `0`, *m* = `None`)

```
Creates a POINTZ shape.
If the z (elevation) value is not set, it defaults to 0.
```

If the m (measure) value is not set, it defaults to NoData.

15.28.8.31 shapefile.Writer.poly (*self*, *polys*)

```
Creates a POLYGON shape.
Polys is a collection of polygons, each made up of a list of xy values.
Note that for ordinary polygons the coordinates must run in a clockwise direction.
```

If some of the polygons are holes, these must run in a counterclockwise direction.

15.28.8.32 shapefile.Writer.polym (*self*, *polys*)

```
Creates a POLYGONM shape.
Polys is a collection of polygons, each made up of a list of xym values.
Note that for ordinary polygons the coordinates must run in a clockwise direction.
If some of the polygons are holes, these must run in a counterclockwise direction.
```

If the m (measure) value is not included, it defaults to None (NoData).

15.28.8.33 shapefile.Writer.polyz (*self*, *polys*)

```
Creates a POLYGONZ shape.
Polys is a collection of polygons, each made up of a list of xyzm values.
Note that for ordinary polygons the coordinates must run in a clockwise direction.
If some of the polygons are holes, these must run in a counterclockwise direction.
If the z (elevation) value is not included, it defaults to 0.
```

If the m (measure) value is not included, it defaults to None (NoData).

15.28.8.34 shapefile.Writer.record (*self*, * *recordList*, ** *recordDict*)

```
Creates a dbf attribute record. You can submit either a sequence of
field values or keyword arguments of field names and values. Before
adding records you must add fields for the record values using the
field() method. If the record values exceed the number of fields the
extra ones won't be added. In the case of using keyword arguments to specify
field/value pairs only fields matching the already registered fields
```

will be added.

15.28.8.35 shapefile.Writer.shape (*self*, *s*)

15.28.8.36 shapefile.Writer.shapeTypeName (*self*)

15.28.8.37 shapefile.Writer.zbox (*self*)

Returns the current z extremes for the shapefile.

15.28.9 Member Data Documentation

15.28.9.1 shapefile.Writer._bbox [protected]

15.28.9.2 shapefile.Writer._files_to_close [protected]

15.28.9.3 shapefile.Writer._mbox [protected]

15.28.9.4 shapefile.Writer._zbox [protected]

15.28.9.5 shapefile.Writer.autoBalance

15.28.9.6 shapefile.Writer.dbf

15.28.9.7 shapefile.Writer.deletionFlag

15.28.9.8 shapefile.Writer.encoding

15.28.9.9 shapefile.Writer.encodingErrors

15.28.9.10 shapefile.Writer.fields

15.28.9.11 shapefile.Writer.recNum

15.28.9.12 shapefile.Writer.shapeType

15.28.9.13 shapefile.Writer.shp

15.28.9.14 shapefile.Writer.shpNum

15.28.9.15 shapefile.Writer.shx

15.28.9.16 shapefile.Writer.target

15.28.9.17 The documentation for this class was generated from the following file:

- PyshpMaster/**shapefile.py**

16 File Documentation

16.1 AreaManager.py File Reference

16.1.1 Classes

class **AreaManager.AreaManager***This class is used to paint area grouped by.*

class **AreaManager.Corner***Defines floating point data for corner defintions.*

16.1.2 class AreaManager.AreaMgrSubFrameNamespaces

- namespace **AreaManager**

16.2 EditMathSetupFrame.py File Reference

16.2.1 Classes

class **EditMathSetupFrame.EditMathSetup***This class allows the user to edit the Matlab/Octave setup files to fit their environment.*

16.2.2 Namespaces

- namespace **EditMathSetupFrame**

16.3 FishMortBySpecAcc.py File Reference

16.3.1 Classes

class **FishMortBySpecAcc.FishMortBySpecAcc***This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.*

16.3.2 Namespaces

- namespace **FishMortBySpecAcc**

16.4 GeoSams.py File Reference

16.4.1 Classes

class **GeoSams.MainApplication***This class is the parent class for the GUI.*

16.4.2 Namespaces

- namespace **GeoSams**

16.4.3 Functions

- **GeoSams.ComputeResiduals** (obsFile, gridFile, procID, retDict)
- **GeoSams.main** ()

16.5 Globals.py File Reference

16.5.1 Namespaces

- namespace **Globals**

16.5.2 Functions

- **Globals.DetermineUnitsScale** (desiredParam)
- **Globals.UpdateEntry** (entry, val)
- **Globals.ShowMessage** (heading, message, type='info', timeout=2500)
  This method will display the message and then go away after the default time.

16.5.3 Variables

- str **Globals.analDir** = 'Analysis'
- str **Globals.configDir** = 'Configuration'
- str **Globals.dataDir** = 'Data'
- str **Globals.gridDir** = 'Grids'
- str **Globals.interCfgDir** = 'Interpolation'
- str **Globals.resultsDir** = 'Results'
- str **Globals.shapeFileDir** = 'Shapefiles'
- str **Globals.simCfgDir** = 'Simulation'
- str **Globals.specAccCfgDir** = 'SpecialAccess'
- str **Globals.surveyDataDir** = 'OriginalData'
- list **Globals.comboTFStr** = ['T', 'F']
- list **Globals.cornerLabelArr** = ['Corner', 'Long', 'Lat ', '0.0', '0.0']
- int **Globals.frameWidth** = 400
- int **Globals.frameHeight** = 200
- int **Globals.scrollFrameHeight** = 600
- int **Globals.helpXoffset** = 700
- int **Globals.helpYoffset** = 50
- int **Globals.meters_per_naut_mile** = 1852
- int **Globals.grid_area_sqm** = **meters_per_naut_mile****2
- str **Globals.ABUN** = 'ABUN_'
- str **Globals.BIOM** = 'BIOM_'
- str **Globals.EBMS** = 'EBMS_'
- str **Globals.FEFF** = 'FEFF_'
- str **Globals.FMOR** = 'FMOR_'
- str **Globals.LAND** = 'LAND_'
- str **Globals.LNDW** = 'LNDW_'
- str **Globals.LPUE** = 'LPUE_'
- str **Globals.RECR** = 'RECR_'
- int **Globals.scrollFrameWidth** = 900
- str **Globals.geometryStr** = '920x725+10+10'

16.6 GrowthFrame.py File Reference

16.6.1 Classes

class **GrowthFrame.Growth***This class allows the user to adjust parameters used in computing scallop growth.*

16.6.2 Namespaces

- namespace **GrowthFrame**

16.7 MainInputFrame.py File Reference

16.7.1 Classes

class **MainInputFrame.MainInput***This class displays information about GeoSAMS simulation.*

16.7.2 Namespaces

- namespace **MainInputFrame**

16.8 PointInPolygon.py File Reference

16.8.1 Namespaces

- namespace **PointInPolygon**

16.8.2 Functions

- **PointInPolygon.PointInPolygon** (polyX, polyY, x, y, nodes)

16.9 PyshpMaster/shapefile.py File Reference

16.9.1 Classes

- class **shapefile._Array**class **shapefile.Shape**
- class **shapefile._Record**
- class **shapefile.ShapeRecord**
- class **shapefile.Shapes**
- class **shapefile.ShapeRecords**
- class **shapefile.ShapefileException**
- class **shapefile.Reader**
- class **shapefile.Writer**

16.9.2 Namespaces

- namespace **shapefile**

16.9.3 Functions

- **shapefile.b** (v, encoding='utf-8', encodingErrors='strict')
- **shapefile.u** (v, encoding='utf-8', encodingErrors='strict')
- **shapefile.is_string** (v)
- **shapefile.pathlike_obj** (path)
- **shapefile.signed_area** (coords, fast=False)
- **shapefile.is_cw** (coords)
- **shapefile.rewind** (coords)
- **shapefile.ring_bbox** (coords)
- **shapefile.bbox_overlap** (bbox1, bbox2)
- **shapefile.bbox_contains** (bbox1, bbox2)
- **shapefile.ring_contains_point** (coords, p)
- **shapefile.ring_sample** (coords, ccw=False)
- **shapefile.ring_contains_ring** (coords1, coords2)

- **shapefile.organize_polygon_rings** (rings, return_errors=None)
- **shapefile.test** (**kwargs)

16.9.4 Variables

- str **shapefile.__version__** = "2.3.1"
- **shapefile.logger** = logging.getLogger(__name__)
- bool **shapefile.VERBOSE** = True
- int **shapefile.NULL** = 0
- int **shapefile.POINT** = 1
- int **shapefile.POLYLINE** = 3
- int **shapefile.POLYGON** = 5
- int **shapefile.MULTIPOINT** = 8
- int **shapefile.POINTZ** = 11
- int **shapefile.POLYLINEZ** = 13
- int **shapefile.POLYGONZ** = 15
- int **shapefile.MULTIPOINTZ** = 18
- int **shapefile.POINTM** = 21
- int **shapefile.POLYLINEM** = 23
- int **shapefile.POLYGONM** = 25
- int **shapefile.MULTIPOINTM** = 28
- int **shapefile.MULTIPATCH** = 31
- dict **shapefile.SHAPETYPE_LOOKUP**
- int **shapefile.TRIANGLE_STRIP** = 0
- int **shapefile.TRIANGLE_FAN** = 1
- int **shapefile.OUTER_RING** = 2
- int **shapefile.INNER_RING** = 3
- int **shapefile.FIRST_RING** = 4
- int **shapefile.RING** = 5
- dict **shapefile.PARTTYPE_LOOKUP**
- int **shapefile.PYTHON3** = 3
- **shapefile.xrange** = range
- **shapefile.izip** = zip
- list **shapefile.MISSING** = [None,'']
- int **shapefile.NODATA** = -10e38
- **shapefile.failure_count** = **test**()

16.10 ShapeTest.py File Reference

16.10.1 Classes

16.10.2 class ShapeTest.GeoShapeNamespaces

- namespace **ShapeTest**

16.10.3 Variables

- **ShapeTest.sf** = **shapefile.Reader**("Shapefiles/MAB_Estimation_Areas_2019_UTM18_PDT.shp")
- **ShapeTest.shapes** = sf.shapes()
- **ShapeTest.shapeLen** = len(**sf**)
- list **ShapeTest.shapeMA** = [**GeoShape**() for _ in range(**shapeLen**)]
- **ShapeTest.record** = sf.record(n)
- **ShapeTest.as_dict** = record.as_dict()
- **ShapeTest.SAMS**

- **ShapeTest.NewSAMS**
- **ShapeTest.areaKm2**
- **ShapeTest.pointLen** = len(**shapes**[n].points)
- **ShapeTest.X**
- **ShapeTest.Y**
- **ShapeTest.lat**
- **ShapeTest.lon**
- list **ShapeTest.shapeGB** = [**GeoShape**() for _ in range(**shapeLen**)]

16.11 SortByAreaFrame.py File Reference

16.11.1 Classes

class **SortByAreaFrame.SortByArea***This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.*

16.11.2 Namespaces

- namespace **SortByAreaFrame**

16.12 SortByRegionFrame.py File Reference

16.12.1 Classes

class **SortByRegionFrame.GeoShape***This class is used to define the shape of the regional data.*

class **SortByRegionFrame.SortByRegion***This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.*

16.12.2 Namespaces

- namespace **SortByRegionFrame**

16.13 SortIntoColumns.py File Reference

16.13.1 Classes

- class **SortIntoColumns.GeoShape**class **SortIntoColumns.Column**

16.13.2 Namespaces

- namespace **SortIntoColumns**

16.13.3 Variables

- **SortIntoColumns.inputFile** = sys.argv[1]
- **SortIntoColumns.l** = len(**inputFile**)
- **SortIntoColumns.domain** = **inputFile**[**l**-2:**l**]
- **SortIntoColumns.dataFile** = os.path.join('Data', **inputFile**+'.csv')
- **SortIntoColumns.outfile** = os.path.join('Data', **inputFile**+'_BUFFER.csv')
- **SortIntoColumns.M** = pd.read_csv(**dataFile**)
- **SortIntoColumns.fileName** = os.environ['GBShapeBufferFile']
- str **SortIntoColumns.subDir** = 'GB_Buffer'
- **SortIntoColumns.shapeFile** = os.path.join('Shapefiles', **subDir**, **fileName**)
- **SortIntoColumns.sf** = **shapefile.Reader**(**shapeFile**)
- **SortIntoColumns.shapes** = sf.shapes()

- **SortIntoColumns.shapeLen** = len(**sf**)
- list **SortIntoColumns.shape** = [**GeoShape**() for _ in range(**shapeLen**)]
- **SortIntoColumns.record** = sf.record(n)
- **SortIntoColumns.as_dict** = record.as_dict()
- **SortIntoColumns.Region**
- **SortIntoColumns.pointLen** = len(**shapes**[n].points)
- **SortIntoColumns.X**
- **SortIntoColumns.Y**
- list **SortIntoColumns.columns** = [**Column**() for _ in range(**shapeLen**)]
- **SortIntoColumns.name**
- **SortIntoColumns.X_t** = **M**['UTM_X']
- **SortIntoColumns.Y_t** = **M**['UTM_Y']
- **SortIntoColumns.rows** = len(**X_t**)
- **SortIntoColumns.nodes** = len(**shape**[rgn].**X**)
- **SortIntoColumns.sep**
- **SortIntoColumns.na_rep**
- **SortIntoColumns.index**

16.14 SpecialAreaFrame.py File Reference

16.14.1 Classes

class **SpecialAreaFrame.SpecialArea***This class is used to assist the user in defining areas of interest to assess accumulated parameters located in these areas of interest.*

16.14.2 Namespaces

- namespace **SpecialAreaFrame**

16.15 Widgets.py File Reference

16.15.1 Classes

class **Widgets.SubFrameElement***Generic Element.*

- class **Widgets.SubFrameInterpFunction**class **Widgets.SubFrameXY**
  Widget for XY label and entery.

  class **Widgets.ScrollFrame***Scrollable Frame Class from* `https://gist.github.com/mp035/9f2027c3ef9172264532fcd6262f3b01`.

16.15.2 Namespaces

- namespace **Widgets**

16.15.3 Functions

- **Widgets.numbersCallback** (input)
  Allows only correctly formed positive integers, ignores non-numeric characters.

- **Widgets.floatCallback** (input)
  Allows only correctly formed floats, ignores non-numeric characters.

17 Index