# EO4VisTrails Documentation

***Release 1.0.0***

**Meraka.CSIR**

March 08, 2013

# CONTENTS

Contents:

CHAPTER

ONE

# OVERVIEW

EO4VisTrails adds spatial and temporal data access, data pre-processing and data analysis capabilities to VisTrails, the Python scientific workflow tool. This includes cloud, analytics and standards-based spatial data access.

Current data modules allow access to OGC Web Services, such as WFS, WMS, SOS, as well as other spatial data sources, including PostGIS databases and netCDF datacubes. Additional modules provides wrappers around standard R, Octave and SQL scripts for data analysis capabilities.

Spatial data can be visualised using a mapping module (via the QGIS API), as well as more conventional plots, for example, for time-series and windrose data.

CHAPTER

TWO

# PACKAGES

## 2.1 DataAnalytics

This package enables spatial and statistical analysis capabilities for EO4VisTrails.

### 2.1.1 RasterLang

### 2.1.2 PySAL

This module ???

**class** dataanalytics.PySAL.**PySALModule**

**class** dataanalytics.PySAL.**PySALModule**

dataanalytics.PySAL.**W**
alias of <Mock object at 0x28c05d0>

### 2.1.3 PyDAP

### 2.1.4 rpy2Stats

### 2.1.5 netcdf4

**Note:** add brief description of what this netCDF client does

dataanalytics.netcdf4.**netcdf4Reader**
alias of <Mock object at 0x2836590>

dataanalytics.netcdf4.**netcdf4ConfigurationWidget**
alias of <Mock object at 0x2836b10>

### 2.1.6 octave

## 2.2 GeoInf

This package provides GIS capabilities for EO4VisTrails.

In particular, it provides GRASS, PostGIS, OGC web clients and some visualisation capabilities through the QGIS API.

### 2.2.1 DataModels

This package provides consistent internal data models for feature, raster and time-series data. Proposes use of gdal, ogr (or GeoJson) and hdf

#### DataRequest

This module forms part of the eo4vis1trails capabilities - it is used as a base module for building requests.

geoinf.datamodels.DataRequest.**DataRequest**
alias of <Mock object at 0x318e3d0>

#### Feature

This module provides an generic OGC Simple Features data model via OGR. All eo4vistrails modules dealing with feature data must extend one of the provided classes.

geoinf.datamodels.Feature.**initialize**(**args*, ***keywords*)
Add module to the Vistrails registry; specify input & output ports.

geoinf.datamodels.Feature.**FeatureModel**
alias of <Mock object at 0x35093d0>

geoinf.datamodels.Feature.**MemFeatureModel**
alias of <Mock object at 0x3509810>

geoinf.datamodels.Feature.**FileFeatureModel**
alias of <Mock object at 0x3509a50>

geoinf.datamodels.Feature.**FeatureModelGeometryComparitor**
alias of <Mock object at 0x35097d0>

#### QgsLayer

#### QgsLayerWriter

This module provides a means for saving raster and vector data to file, in the format defined by QGIS.

geoinf.datamodels.QgsLayerWriter.**QgsLayerWriter**
alias of <Mock object at 0x35362d0>

#### Raster

This module provides a generic coverage/raster data model via GDAL. All eo4vistrails modules dealing with raster data must extend this class.

geoinf.datamodels.Raster.**initialize**(**args*, ***keywords*)
Add module to the Vistrails registry; specify input & output ports.

**class** geoinf.datamodels.Raster.**_GdalMemModel**

Used as both a storage mechanism for MemRasterModel instances and as a general GDAL constructor, i.e. whether you want an in-memory GDAL data provider or not, data will always be passed into GDAL control via instances of _GdalMemModel

**dumpToFile** (*filesource*, *datasetType='GTiff'*)

**loadContentFromFile** (*sourceDS*, *getStatement=''*)

Loads content off filesystem, e.g. from a geotiff

**Parameters**

- **uri** (*str*) – a path to a file
- **getStatement** (*str*) – the XML of the request parameters

**Return type** None

Expects datasets with one layer, so some arcane formats are out...

**loadContentFromURI** (*uri*, *getStatement=''*)

Loads content off web service, feed etc.; e.g., a WCS

**Parameters**

- **uri** (*str*) – the URI of the service endpoint
- **getStatement** (*str*) – the XML of the request parameters

**Return type** None

These arguments allow creation of GET/POST requests, and also allow us to make GDAL sensibly deal with the inputs.

For WCS, we expect a WCS GetCoverage request to be incoming. GDAL, our raster "swiss army knife", expects to access a WCS from a config file on the filesystem, that looks like:

```
<WCS_GDAL>
    <ServiceURL>
        http://ict4eo.meraka.csir.co.za/geoserver/wcs?
    </ServiceURL>
    <CoverageName>
        nurc:Img_Sample
    </CoverageName>
</WCS_GDAL>
```

and is called something like *ict4eowcs.wcs*.

GDAL needs read/write access to this file, for it then writes the capabilities to it for later reference.

geoinf.datamodels.Raster.**RasterModel**

alias of <Mock object at 0x35c7e10>

### RasterImport

### TemporalVectorLayer

### WebRequest

This module forms part of the eo4vistrails capabilities - it is used to handle web request (e.g. for WFS, WCS or SOS) processing inside vistrails.

geoinf.datamodels.WebRequest.**WebRequest**
alias of <Mock object at 0x37da2d0>

### 2.2.2 GeoStrings

This package provides core and general purpose code for eo4vistrails geoinf module. In particular, it provides geo-constanst for use in other modules.

#### GeoJSON

This module holds a GeoJSON type that can be passed around between modules.

geoinf.geostrings.GeoJSON.**GeoJSONModule**
alias of <Mock object at 0x387c450>

geoinf.geostrings.GeoJSON.**GJFeature**
alias of <Mock object at 0x387ca10>

#### GeoStrings

This abstract module provides string specialisations that allow modules to understand what kind of data is being passed into a method as a string.

geoinf.geostrings.GeoStrings.**GeoStringConstantWidget**
alias of <Mock object at 0x3917990>

### 2.2.3 Helpers

This package provides core and general purpose helper code for eo4vistrails geoinf module. In particular, provides a tool for defining Areas, Lines and Points of interest for use elsewhere in GIS environments and statistical environments.

#### AOI_Utils

### 2.2.4 PostGIS

This package provides GIS capabilities for eo4vistrails. In particular, it provides PostGIS clients via psycopg2.

#### PostGIS

#### pgLoadersDumpers

#### DataTransformations

### 2.2.5 Web

This package provides OGC capabilities for eo4vistrails. In particular, provides WFS, WCS and SOS clients through the owslib (http://gispython.org/owslib) library, and hence has a dependency thereupon. owslib is dependent on libxml or elementree

### Common

This module provides a small API for fetching the service and dealing with OGC (Open Geospatial Consortium) Web Service metadata, which is common to all the various services (via owslib).

**Requirements:** owslib 0.3.4

NOTE:

As at 2010-11-09, you will need to patch version 0.3.4b for owslib sos.py:

```
self.filters=filter.Filter_Capabilities(val)
```

should be:

```
self.filters=filter.FilterCapabilities(val)
```

**class** geoinf.web.Common.**OgcService**(*service_url*, *service_type*, *service_version*, *timeout=180*, *capabilities=None*)

Base class for OGC service operations.

Input ports:

**service_url: string** URL for OGC service

**service_type: string** 'wfs', 'sos', 'wcs'

**service_version: string** currently only version 1.0 for OGC services (but see Notes)

**timeout: integer** in seconds - some lines/servers experiences slow speeds and/or the XML being returned is in a large document, taking time to construct (see inline comments about usage in different services)

**capabilities:** a full capabilities document - if provided, no call will be made by owslib to the service_url, and this will be used instead

Notes:

Namespace issues pervade WFS in owslib e.g. * cannot handle when geoserver shortcuts a namespace with an abbreviation * wfs will never load a 1.1.0 instance, but it should be loaded as wfs200

**setOperations**(*service_list*)
service identification metadata is structured differently for the different services - parse appropriately

**setProviderIdentification**(*provider_dict*)
provider metadata is structured differently for the different services - parse appropriately

**setServiceIdentification**(*service_dict*)
service identification metadata is structured differently for the different services - parse appropriately

### FTP

This module provides FTP functionality.

geoinf.web.FTP.**FTPReader**
alias of <Mock object at 0x3d09bd0>

### OgcConfigurationWidget

### OgcService

### PortConfigurationWidget

This module provides general OGC (Open Geospatial Consortium) Web Service selection widgets for configuring geoinf.web modules.

This refers primarily to GetCapabilities requests

**class** geoinf.web.PortConfigurationWidget.**Port**(*id=None*, *name=None*, *sigstring='edu.utah.sci.vistrails.basic:String'*, *type='input'*, *sort_key=-1*)

TODO - write docstring

**value**()

Create a tuple-form of a port, for use by Vistrails module

geoinf.web.PortConfigurationWidget.**PortConfigurationWidget**

alias of <Mock object at 0x3f45f50>

### SOS

### SOSFeeder

### WCS

### WFS

### WPS

CHAPTER

THREE

# GLOSSARY

**source directory** The directory which, including its subdirectories, contains all source files for one Sphinx project.

CHAPTER

FOUR

# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

# PYTHON MODULE INDEX

## d



## g