

---

# **Ctutils Documentation**

*Release 0.2*

**Dave Hughes**

September 01, 2014



<b>1</b>	<b>Links</b>	<b>3</b>
<b>2</b>	<b>Table of Contents</b>	<b>5</b>
2.1	Installation . . . . .	5
2.2	ctagg . . . . .	6
2.3	ctinfo . . . . .	8
2.4	Change log . . . . .	8
2.5	License . . . . .	9
<b>3</b>	<b>Indices and Tables</b>	<b>11</b>



This project provides a set of small utilities for processing data from CT scanners in a variety of formats. Currently, the supported formats are:

- Xradia's TXM format
- Volume Graphics VGI+VOL format
- Grayscale TIFF stacks (8-bit, 16-bit)

At the moment, the project's goal is to produce a set of simple command line tools for batch processing of CT scans, and an API to permit easy access to scan data from Python.

---

**Note:** Please note that our understanding of several of the formats above is limited (in particular TXM and VGI+VOL), specifically in terms of the data types supported within these files.

If you encounter a file which does not work with CT Utils, please file a bug report so that we can enhance the suite. Obviously example files make debugging easier, but we understand there may be confidentiality issues with certain scans (publication embargoes, medical privacy, etc.). If this applies to your scans, please do not attach them to the bug report!

---



---

### Links

---

- The project is licensed under the [GPL v3](#) or above
- Project [source code](#) is hosted by GitHub which also hosts the [bug tracker](#)
- The [documentation](#) is hosted by ReadTheDocs





---

## Table of Contents

---

### 2.1 Installation

CT Utils is designed to be cross platform and should work happily on Microsoft Windows (under a variety of different Python frameworks), Mac OS X, and Linux. The following sections detail installation in these different environments.

#### 2.1.1 Ubuntu Linux

The simplest method of installation under Ubuntu is to use the author's PPA as a source:

```
$ sudo add-apt-repository ppa:waveform/ppa
$ sudo apt-get update
$ sudo apt-get install ctutils
```

This will ensure that CT Utils remains up to date through the usual apt upgrade system. To remove the installation:

```
$ sudo apt-get remove ctutils
```

#### 2.1.2 Other Linux

Ensure you have Python's `setuptools` and `pip` packages installed, then run the following command:

```
$ sudo pip install ctutils
```

To upgrade your installation after new releases:

```
$ sudo pip install -U ctutils
```

To remove your installation:

```
$ sudo pip uninstall ctutils
```

#### 2.1.3 Windows with Python(x,y)

Start a command window (note: not a Python console) and run the following command:

```
$ pip install ctutils
```

To upgrade your installation after new releases:

```
$ pip install -U ctutils
```

To remove your installation:

```
$ sudo pip uninstall ctutils
```

---

**Note:** Please note that Python(x,y) is only a 32-bit installation. As CT Utils relies on memory mapping for handling large files, operating on files larger than 2Gb will fail in the current version. Future versions may incorporate an emulation to work around this but operation will be considerably slower than simply using a 64-bit installation.

---

### 2.1.4 Windows with Canopy

This installation assumes you configured Canopy as your system Python installation. Start a windows command prompt (Win+R, cmd) and run the following commands:

```
$ pip uninstall PIL
$ pip install Pillow
$ pip install ctutils
```

The reason for replacing PIL (the Python Imaging Library) with Pillow is that PIL is effectively unmaintained at this point. Pillow is a fork of the PIL project which is carrying on maintenance and providing Python 3 compatibility.

To upgrade your installation after new releases:

```
$ pip install -U ctutils
```

To remove your installation:

```
$ sudo pip uninstall ctutils
```

### 2.1.5 Windows with Anaconda

This installation assumes you configured Anaconda as your system Python installation. Start a windows command prompt (Win+R, cmd) and run the following commands:

```
$ pip install -U argcomplete
$ pip install ctutils
```

Anaconda ships with a slightly out of date argcomplete package (CT Utils assumes version 0.7 or above if it finds argcomplete installed) which requires upgrading. This is a rather moot point on Windows however, as argcomplete relies on bash/zsh for its functionality and won't work under a regular Windows cmd shell.

To upgrade your installation after new releases:

```
$ pip install -U ctutils
```

To remove your installation:

```
$ sudo pip uninstall ctutils
```

## 2.2 ctagg

This utility can be used to aggregate the density readings from all scans in a CT-scanner's output into a simple CSV file, suitable for plotting in a variety of applications. File formats supported include TXM files (".txm"), VGI files (".vgi" with an equivalently named ".vol" file), or TIFF stacks (specify one of the TIFF filenames and all equivalently sized TIFFs in the directory will be loaded as part of the stack). If no output filename is specified, output will be written to a file with the same name as the input but with a ".csv" extension.

## 2.2.1 Synopsis

```
ctagg [-h] [--version] [-q] [-v] [-l FILE] [-P]
      [-s START[:STOP[:STEP]]] [-f NUM]
      input [output]
```

## 2.2.2 Description

- h, --help**  
show this help message and exit
- version**  
show program's version number and exit
- q, --quiet**  
produce less console output
- v, --verbose**  
produce more console output
- l FILE, --log-file FILE**  
log messages to the specified file
- P, --pdb**  
run under PDB (debug mode)
- s START[:STOP[:STEP]], --slice START[:STOP[:STEP]]**  
specifies the index(es) of image(s) to extract from the scan in Python slice notation, e.g. 0:50 to extract the first 50 images, 50: to extract all images from the 50th onward, or ::5 to extract every 5th image. Defaults to all images
- f NUM, --filter NUM**  
any counts below the specified filter level will be excluded from the CSV output

## 2.2.3 Examples

Extract all density measurements from `scan1.txm` into `scan1.csv`:

```
$ ctagg scan1.txm
```

Extract density measurements from every 10th slice in `scan1.txm` into `preview.csv`:

```
$ ctagg --slice ::10 scan1.txm preview.csv
```

Extract density measurements from the first slice in `scan1.vgi` into `first.csv` (this assumes a corresponding file named `scan1.vol` exists in the same directory as `scan1.vgi`):

```
$ ctagg --slice :1 scan1.vgi first.csv
```

Extract density measurements from every third slice of the TIFF stack which includes `slice001.tiff`, ignoring densities below 10 into `quick.csv`, printing verbose progress messages to the console:

```
$ ctagg -v --slice ::3 --filter 10 slice001.tiff preview.csv
```

When processing TIFF stacks, simply specify the name of one of the TIFF images in the stack. The utility will automatically locate all other TIFF images that exist in the same directory and which have the same resolution as the specified TIFF. The images will be arranged in the stack in alphabetical order by filename.

## 2.2.4 Known Issues

Our understanding of several of the formats above is limited (in particular TXM and VGI+VOL), specifically in terms of the data types supported within these files. If you encounter a file which does not work with CT Utils, please file a [bug report](#) so that we can enhance the suite.

## 2.3 ctinfo

This utility can be used to rapidly query the header of CT-scanner output in various formats. Output is written to stdout in a format conducive to script processing. File formats supported include TXM files (".txm"), VGI files (".vgi" with an equivalently named ".vol" file), or TIFF stacks (specify one of the TIFF filenames and all equivalently sized TIFFs in the directory will be loaded as part of the stack).

### 2.3.1 Synopsis

```
usage: ctinfo [-h] [--version] [-q] [-v] [-l FILE] [-P] input
```

### 2.3.2 Description

- h, --help**  
show this help message and exit
- version**  
show program's version number and exit
- q, --quiet**  
produce less console output
- v, --verbose**  
produce more console output
- l FILE, --log-file FILE**  
log messages to the specified file
- P, --pdb**  
run under PDB (debug mode)

### 2.3.3 Examples

Output information about `scan1.txm` to the console:

```
$ ctinfo scan1.txm
```

For each scan in the directory, write a text file detailing its contents:

```
$ for f in *.vgi; do ctinfo $f > ${f%.vgi}.txt; done
```

Output information on all scans in a directory hierarchy:

```
$ find -name "*.txm" -o -name "*.vgi" | xargs -n 1 ctinfo
```

## 2.4 Change log

### 2.4.1 Release 0.2 (2014-09-01)

The second release fixes some silly bugs and adds a few new features:

- Fixed a missing configparser import
- Fixed a missing default file object when an output filename was not specified under Python 2
- Added documentation including installation instructions for several scientific Python distributions under Windows
- Added Debian packaging and a PPA for Ubuntu Linux

## 2.4.2 Release 0.1 (2014-04-23)

Initial release

## 2.5 License

This file is part of ctutils.

ctutils is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

ctutils is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with ctutils. If not, see <http://www.gnu.org/licenses/>.



---

## Indices and Tables

---

- *genindex*
- *search*