Teacup Generation User Guide

# Introduction

This document is intended to function as a basic user’s guide to installing, running, and configuring the Python “teacup” script developed by RTi for use by the USBR. More detailed, low-level documentation is available within the script itself, but this user guide should provide enough information for basic use and configuration.

Contact Henry Keiter at [henry.keiter@riverside.com](mailto:henry.keiter@riverside.com) with questions about this document or software.

# Requirements and Installation

This script was built against Python 2.7.4, though any version of Python 2.7 should be fine. The core Python library can be downloaded here:

* <http://www.python.org/download/releases/2.7.4/>

Besides the standard libraries, two third-party modules are required for the teacup script. They are cx\_Oracle 5.1.2, which is used to connect to the Oracle database, and Pillow 2.0.0, which is the imaging library used to draw the teacup diagrams. These packages are freely available:

* <http://cx-oracle.sourceforge.net/>
* <https://pypi.python.org/pypi/Pillow/2.0.0#downloads>

Once Python and these two packages have been installed, the script itself can be run from anywhere on the machine.

**Note:** The Python cx\_Oracle package is necessary, but not sufficient, to make connections to an Oracle database. An Oracle client must also be installed on the machine (as detailed in the cx\_Oracle documentation). If an Oracle client is not yet available, the Oracle Instant Client can be downloaded from here:

* <http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>

# Running the Script

Once all requirements are available, the script can be run through the command line as follows:

* python gentea.py path\_to\_config\_file output\_file\_including path-

All paths should be either absolute paths, or appropriate paths relative to the current working directory. This includes the path to python.exe, if the Python environment is not on the system path.

The output file path is where the final image will be located. All other options are set up inside the specified configuration file, which may be located anywhere (see Configuration, below).

# Configuration

The teacup software is configured by altering the configuration file passed to it. This file stores data like database connection information, input base image, and the locations to draw teacups and other features. Many similar configuration files may exist anywhere on the file system, to run the teacup software for a variety of different parameters, base maps, and so on. Each configuration file is tailored specifically to a certain base map and set of features (teacups and other data such as lines, points, and text).

The configuration file should be a valid piece of Python code in its own right. This section will briefly go over its core. Concrete, working examples of each part of the configuration are available in the included file config\_files/examples/config.py.

## Input Image

The base map (on which the teacups will be drawn) is set with the input\_image property. This should be an absolute path to the location of the base image. This base image will not be modified in any way by the script.

## Database Connection

The connection to the database is configured with the database property. This item itself is composed of five elements: user, passwd, host, port, and sid. These correspond to the connection parameters required by the target database.

## Teacups and Other Attributes

The biggest portion of the configuration file is dedicated to the identification and placement of teacups, lines, points, or other text. This is configured in the property called output\_attributes. There are four types of attributes:

* teacup
* line
* point
* value

Each attribute has a slightly different set of configuration parameters, described briefly below.

### Attribute Type: Teacup

This is the core attribute: it draws a teacup of a specified size at a specified location. Attribute parameters are as follows:

* upper\_left: This is the location (in (x,y) coordinates measured from the upper-left corner of the base image) of the upper-left corner of the teacup.
* site\_id: This is the site\_id corresponding to this teacup to look up in the database. If site\_datatype\_id is provided, this parameter may be omitted.
* datatype\_id: This is the datatype\_id corresponding to this teacup to look up in the database. If site\_datatype\_id is provided, this parameter may be omitted.
* site\_datatype\_id: This is the site\_datatype\_id corresponding to this teacup to look up in the database. This parameter is optional; if it is provided, it will be used in preference to the separate site\_id and datatype\_id.
* teacup\_size: This is the relative size of the teacup on the map. It can be any real, positive number. A teacup of size 1 is 30 pixels wide at the top. This parameter may also be a function that will be used to convert capacity (floating-point value) to teacup size (floating-point value).
* capacity: The maximum capacity of the feature represented by this teacup.
* label: The name (in text) to place under the teacup as its label on the map.

### Attribute Type: Line

This attribute draws a simple line on the map.

* point\_a: This is a tuple (e.g. (0, 0) ) representing one end of the line in coordinates measured from the top-left of the image.
* point\_b: This is a tuple (e.g. (100, 100) ) representing the other end of the line in coordinates measured from the top-left of the image.
* color: This optional parameter sets the color of the line. See the drawing module for which colors are supported. The default is “green”.
* thickness: This optional parameter sets the thickness in pixels of the line that is drawn. This should be a positive integer. The default is 1.

### Attribute Type: Point

This attribute draws a simple point on the map.

* center: This is a tuple (e.g. (10, 100) ) representing the center of the point in coordinates measured from the top-left of the image.
* color: This optional parameter sets the color of the point. See the drawing module for which colors are supported. The default is “red”.
* size: This optional parameter sets the diameter in pixels of the point. Only odd, positive integers are supported (even integers are fine, but there is no difference in the point drawn at an even integer and the next integer up (e.g. 4 and 5)). The default value is 5.

### Attribute Type: Value

This attribute is used to write text values without necessarily associating them with a particular teacup. This can be used to add non-storage-level values to the output diagram. It queries the database in the same way that the teacup attributes do, and prints the results on a single line.

* upper\_left: This is the location (in (x,y) coordinates measured from the upper-left corner of the base image) of the upper-left corner of the line of text to be printed.
* site\_id: This is the site\_id to look up in the database. If site\_datatype\_id is provided, this parameter may be omitted.
* datatype\_id: This is the datatype\_id to look up in the database. If site\_datatype\_id is provided, this parameter may be omitted.
* site\_datatype\_id: This is the site\_datatype\_id to look up in the database. This parameter is optional; if it is provided, it will be used in preference to the separate site\_id and datatype\_id.
* units: The units of this value, written at the end of the text line.
* label: The name (in text) with which to label this value.

## Miscellaneous Properties

A few other optional miscellaneous properties are available. All are shown in the sample config\_files/examples/config.py file and should be fairly self-explanatory, but they are listed here for completeness:

* font\_size: The size of the font to use when writing on the image. Defaults to size 11.
* font\_face: The font face to use when writing on the image. Defaults to “Arial.ttf”.
* timestamp\_location: The location (relative to the upper-left corner of the base image) to place the diagram-creation timestamp. This defaults to (1, 1).