v0.9.3

GoldQC User Guide: Connecting GoldSim with PHREEQC

KEIRAN HINES

10-10-2017

ACKNOWLEDGEMENTS

I would like to thank Dr. David Faulkner of RMIT for this project idea. David offered continued help and support in understanding how the chemistry should interact within the model, this included building many test models and interpreting the results as well as providing tutelage in the use of both GoldSim and PHREEQC. Without this guidance this project would not have been successful.

Table of Contents

List of Figures

List of Tables

1 Introduction

GoldQC provides a link between the GoldSim (GoldSim Technology Group, 2017) simulation package and PHREEQC (USGS, n.d.). The purpose of this package is to allow for a simple and easy to use method of informing GoldSim about the correct chemical balance present in a solution of water. The module makes use of the external interface element in GoldSim to connect with the Python programming language and transfer chemical mass data from GoldSim to PHREEQC using the PHREEQC COM interface connection. In the current version GoldQC expects that GoldSim passes in a vector of element concentration in mg/l and expects that the adjusted values of the elements be passed back to GoldSim in a vector in the same order and units.

1.1 GoldSim External Element

To connect the GoldQC module to GoldSim two DLL files have been provided for use with 32bit or 64bit versions of Python, they are named GoldQC.dll and GoldQC_x64.dll respectively. These DLL files were built using the Linking GoldSim to Python documentation (GoldSim Technology Group, 2017). The DLLs have been built and tested on 32bit and 63bit versions of Python using GoldSim 12.

1.2 GoldQC Python Module

The GoldQC Python Module (GoldQC.py) was developed with inspiration from the Linking GoldSim to Python documentation (GoldSim Technology Group, 2017). The aim of GoldQC.py is to encompass all the necessary features outlined by the documentation for linking GoldSim and Python whilst also enabling the link to PHREEQC's COM interface (IPhreeqc). GoldQC relies on comtypes, a third-party Python package for communication with the COM interface (Heller, n.d.). GoldQC is responsible for setting up the connection to IPhreeqc, receiving data from GoldSim parsing it to a PHREEQC expected format, receiving data back from PHREEQC, parsing the data back to GoldSim expected format and finally returning the results to GoldSim. GoldQC also uses PrettyTable for printing inputs and outputs to the log file in a human readable format (Maurits, n.d.). Configuration details of GoldQC can be set in the GoldQC.config file. The config file is split into three sections, phreeqc, GoldSim and GoldQC. A brief description of these sections can be found in Table 1. See Appendix A for a template of the GoldQC.config file.

Table 1: Configuration file section descriptions

Config Section	Description
phreeqc	The phreeqc configuration section has one option that needs to be configured, database. That is the file path to the databased you would like PHREEQC to use for its calculations. The phreeqc section can also optionally specify values for pH, pe, redox, temp and charge as per PHREEQC specifications as well as a list of equilibrium phases. Equilibrium phases should be a list equilibrium phases with the element, saturation index and amount in moles specified. See Appendix A for a config template.
GoldSim	The GoldSim section has an elements section that needs to be set. The elements section should contain a list of element symbols for the GoldSim vector passed as input to GoldQC in the correct order they appear in GoldSim.
GoldQC	The GoldQC section has four options, all of which are optional. The first is the log_file, this is the name or file path to where you would like the log file to be generated, if no file is specified GoldQC.log will be used. This will allow you access to any error or warning messages provided from PHREEQC. The second option you can specify is the debug_level, by default this is set to 0 meaning only errors and warnings will be logged to the logfile. If debug_level is set to 1 the inputs from GoldSim and outputs from PHREEQC will be logged to help see issues in simulation. Finally, if you set debug_level to 2 the log file will include the inputs from GoldSim and outputs from PHREEQC as well as the fully compiled input string that is passed from GoldQC to PHREEQC. Notice that having debug set will significantly increase the runtime of your model. The third option is suppress_warnings, this option can be set to either 'True' or 'False', if suppress_warnings is set to True and debug is set to 0 the log file will not contain any warning message, if warnings are supressed yet debug is on warnings will still be logged. Finally you can specify use_Config_pH to either True or False to allow GoldSim to be informed about the pH balance from either a set config value or a dynamically changing pH value from GoldSim.

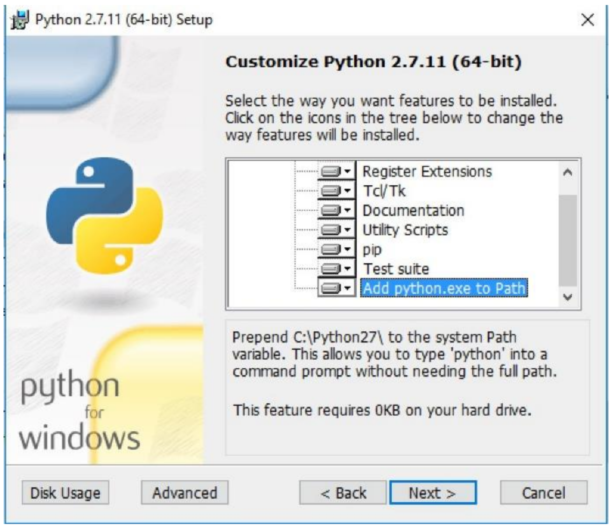
2 Setting up GoldQC

GoldQC is designed to be used with GoldSim version 12, the below installation instructions assume you have installed GoldSim and are familiar with the basic workings of GoldSim. GoldQC relies on IPhreeqc and Python 2.7.x to function correctly, section 2.1 and 2.2 will guide you through installation and setup of the IPhreeqc and Python components.

2.1 Installing Python

If you already have Python 2.7.x installed on your computer please skip to step 4: installing comtypes.

1. Download Python 2.7.x, available from www.python.org. Please select the version appropriate for your version of windows (32-bit or 64-bit).
2. When installing Python select "Install for all Users" and use the default installation directory C:\Python27.
3. When installing please select "add python.exe to Path" see Figure 1.



Python 2.7.11 (64-bit) Setup

Customize Python 2.7.11 (64-bit)

Select the way you want features to be installed.
Click on the icons in the tree below to change the way features will be installed.

- Register Extensions
- Tcl/Tk
- Documentation
- Utility Scripts
- pip
- Test suite
- Add python.exe to Path

Prepend C:\Python27\ to the system Path variable. This allows you to type 'python' into a command prompt without needing the full path.

This feature requires 0KB on your hard drive.

Figure 1: Adding Python.exe to Path

4. After successfully install Python open command prompt and navigate to the directory containing the GoldQC files. Once in the directory run "pip install -r requirements.txt"

2.2 Installing IPhreeqc

This version of GoldQC was developed and tested using the Windows COM version of IPhreeqc specifically IPhreeqcCOM-3.3.12-12704. Installation of IPhreeqc are below:

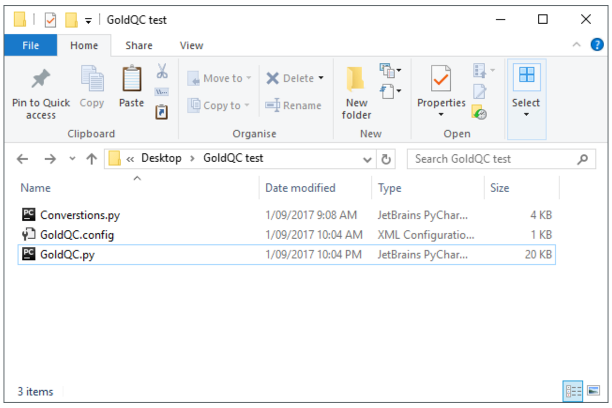
1. Download the Windows COM version of IPhreeqc that matches your computer, notice both 32-bit and 64-bit COM versions should be installed on 64-bit versions of Windows. IPhreeqc is available to download at https://wwwbrr.cr.usgs.gov/projects/GWC_coupled/phreeqc/.

2. Install the downloaded IPhreeqcCOM version leaving the installation parameters as default.

2.3 Testing GoldQC

GoldQC.py has a simple check built in to be able to test that the installation of Python, comtypes and IPhreeqc are running correctly. To test that all the required components are installed correctly please follow the below steps:

1. Copy Conversions.py, GoldQC.py and GoldQC.config to a new directory on your computer, in this example I will use a folder on my Desktop called GoldQC test. Your directory should now contain three files as in figure 2.

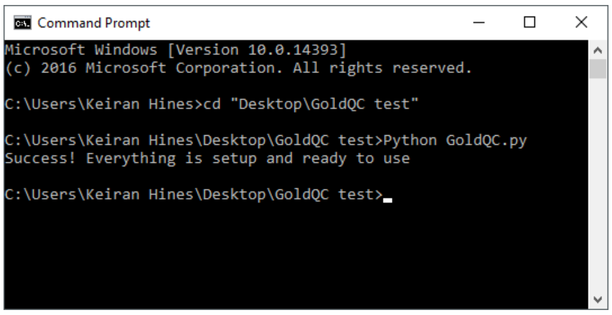


Name	Date modified	Type	Size
Converstions.py	1/09/2017 9:08 AM	JetBrains PyChar...	4 KB
GoldQC.config	1/09/2017 10:04 AM	XML Configuratio...	1 KB
GoldQC.py	1/09/2017 10:04 PM	JetBrains PyChar...	20 KB

Figure 2: Testing GoldQC

2. Open the GoldQC.config file and confirm that the database file listed under phreeqc>database exists, depending on your version of windows or the version of IPhreeqcCOM installed you may need to update this file path.

3. Open Command Prompt and navigate to your test directory and run "Python GoldQC.py. If everything is configured correctly you should see a message saying "Success! Everything is setup and ready to use". See Figure 3.



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Keiran Hines>cd "Desktop\GoldQC test"

C:\Users\Keiran Hines\Desktop\GoldQC test>Python GoldQC.py
Success! Everything is setup and ready to use

C:\Users\Keiran Hines\Desktop\GoldQC test>_
```

Figure 3:Running GoldQC test

3 Using GoldQC with GoldSim

After successfully installing the required components of GoldQC you are now ready to put it to work in GoldSim. To allow for ease of use when changing between different simulation models it is advised to create a new working directory for the model you will be working on and place the GoldQC.py, Converstion.py and GoldQC.config in the working directory along with the GoldSim gsm file and the relevant GoldQC.dll version (GoldQC.dll for 32-bit and GoldQC_x64.dll for 64bit) In this example I will be using the GoldQC_x64.dll version. The below instructions will guide you through adding and configuring GoldQC in GoldSim version 12.

3.1 Configuring GoldSim

1. Create the initial conditions of the model, including the input data you are wishing to pass into GoldQC. This example uses a very basic model that loads seven elements into a solution of water. See Figure 4.

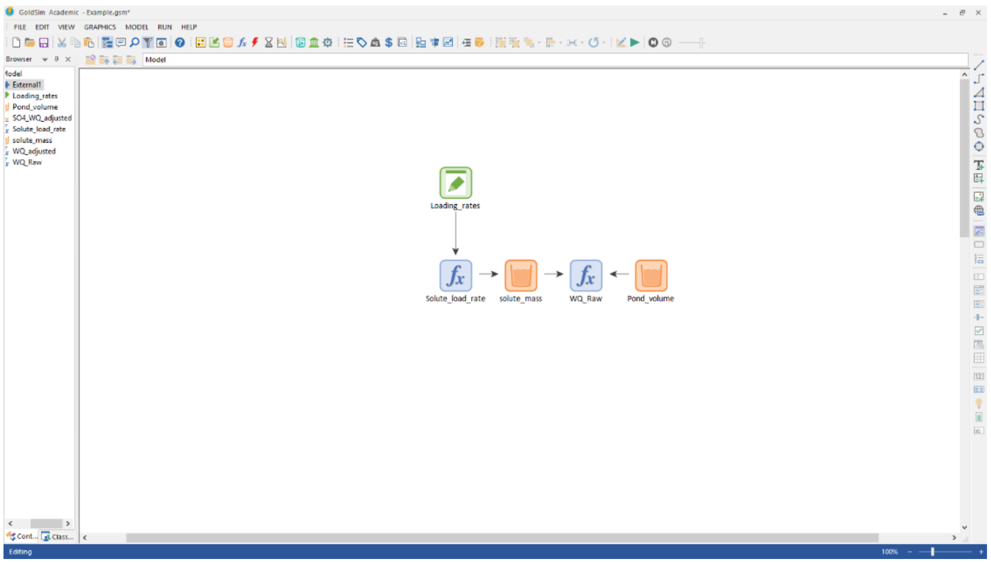


Figure 4: Initial condition of GoldSim Simulation

2. Right click and select Insert Element, Functions and External.

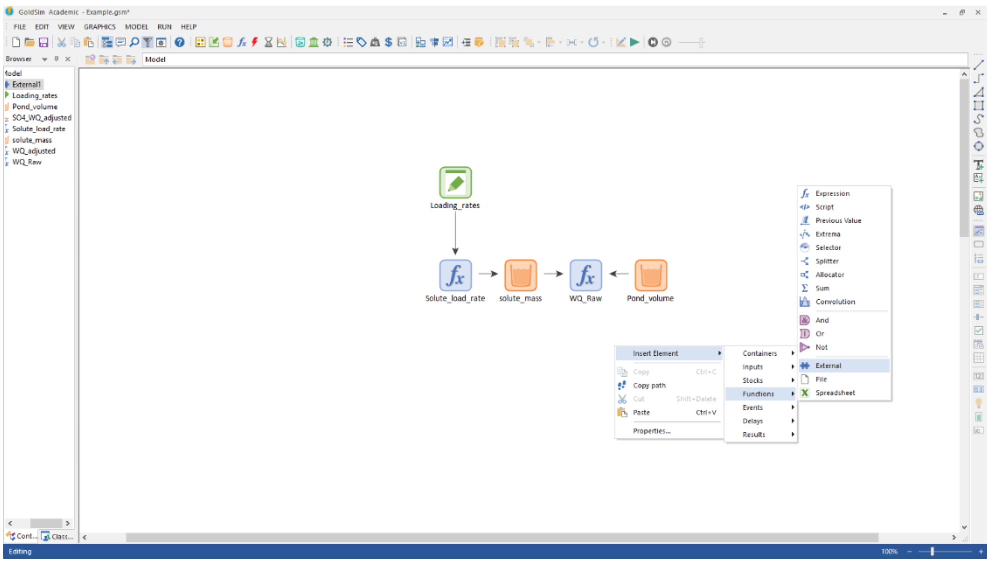
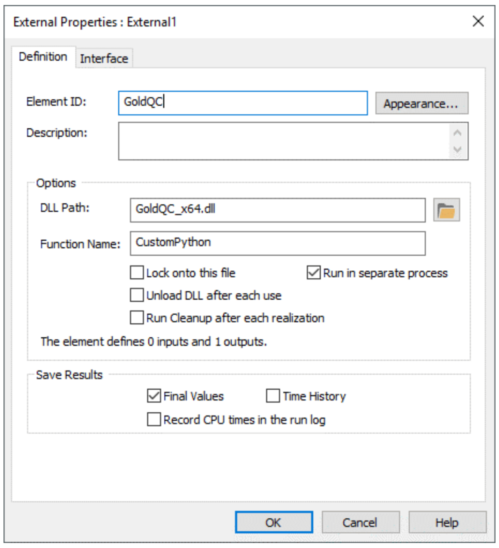


Figure 5: Adding an External Element

3. The External Properties dialog will pop up. In the DLL path specify the path to the needed GoldQC.dll file, in this example I am using the GoldQC_x64.dll file. In the Function Name field type “CustomPython”. Finally tick the “Run in separate process” dialog box. See Figure 6.


External Properties : External1

Definition | Interface

Element ID: GoldQC

Description:

Options

DLL Path: GoldQC_x64.dll

Function Name: CustomPython

☐ Lock onto this file ☑ Run in separate process

☐ Unload DLL after each use

☐ Run Cleanup after each realization

The element defines 0 inputs and 1 outputs.

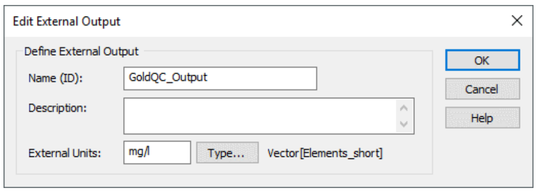
Save Results

☑ Final Values ☐ Time History

☐ Record CPU times in the run log

Figure 6: Configuring External Element Part 1

4. Next switch to the interface tab of the External Properties Dialog, in the top section labelled Input Interface Definition, click the green plus and select the input Vector you wish to use with GoldQC. Secondly click the green plus in the Output Interface Definition section, an Edit External Output dialog will appear. In the Edit External Interface dialog specify the name followed by the type as Vector, the Vector Row Labels option should be set to the same row labels as the input. Finally set the units to mg/l. See figure 7. Figure 8 shows the final view of the external element interface dialog.


Edit External Output

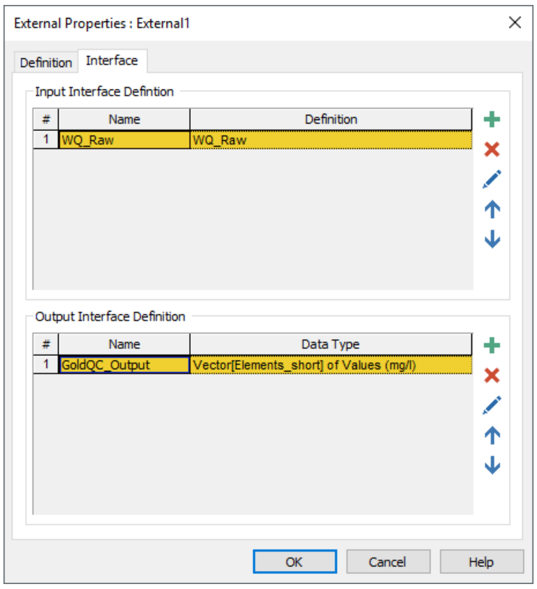
Define External Output

Name (ID): GoldQC_Output

Description:

External Units: mg/l Vector[Elements_short]

Figure 7: Configuring External Element Part 2


External Properties : External1

Definition | Interface

Input Interface Defintion

#	Name	Definition
1	WQ_Raw	WQ_Raw

Output Interface Definition

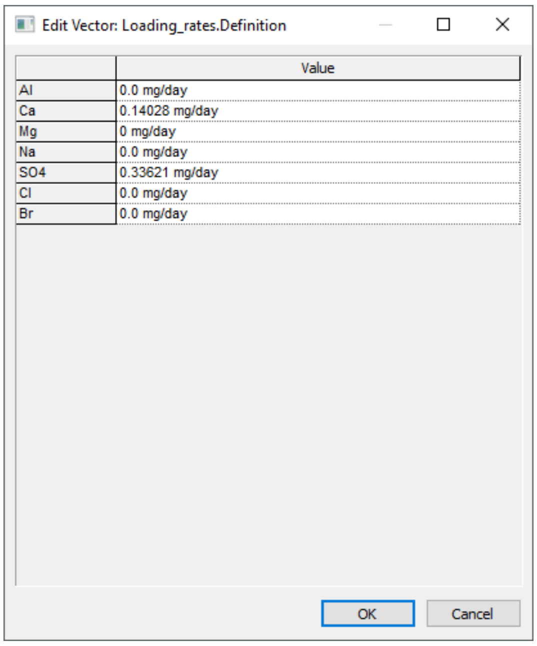
#	Name	Data Type
1	GoldQC_Output	Vector[Elements_short] of Values (mg/l)

Figure 8: Configuring External Element Part 3

5. You are now ready to use the GoldQC output as required in your model by connecting it to other GoldSim elements.

3.2 Configuring GoldQC

Now that the GoldQC external element is configured in GoldSim the next step is to setup the configuration file of GoldQC to match the element symbols listed in GoldSim and select the required PHREEQC database. In this example they are listed in the Loading_rates data field see Figure 9.

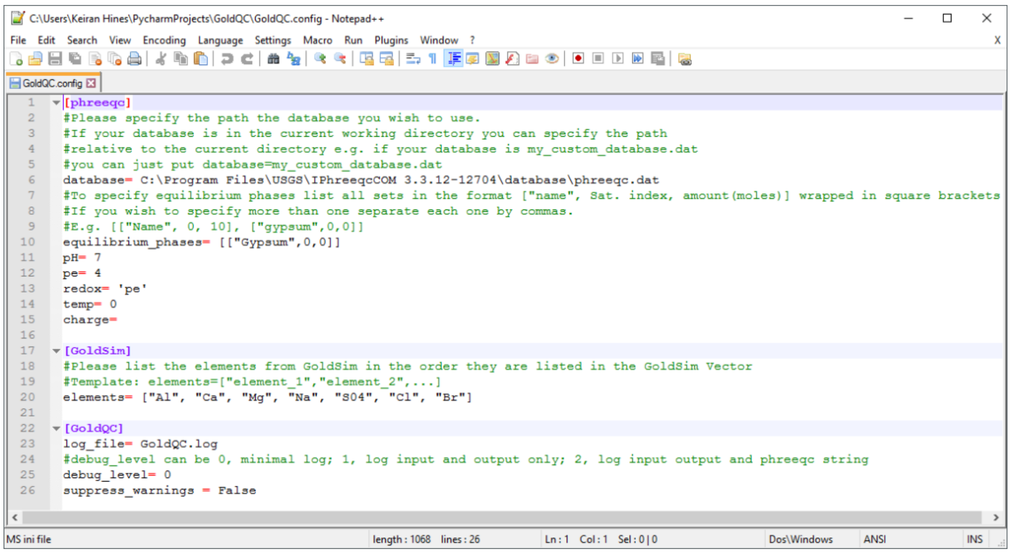


	Value
Al	0.0 mg/day
Ca	0.14028 mg/day
Mg	0 mg/day
Na	0.0 mg/day
SO4	0.33621 mg/day
Cl	0.0 mg/day
Br	0.0 mg/day

Figure 9: Element Symbols as defined in GoldSim

1. Open the GoldQC.config file in your preferred text editor.

2. In the config file locate the line in the GoldSim section defining the elements, here you should list all the elements as specified in GoldSim in between square braces, separated by commas and wrapped in single or double quotes. See figure 9 for the elements shown in Figure 10 transcribed to GoldQC required format.


C:\Users\Keiran Hines\PycharmProjects\GoldQC\GoldQC.config - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
GoldQC.config
MS ini file
length : 1068 lines : 26
Ln : 1 Col : 1 Sel : 0 | 0
Dos\Windows
ANSI
INS

```
[phreeqc]
#Please specify the path the database you wish to use.
#If your database is in the current working directory you can specify the path
#relative to the current directory e.g. if your database is my_custom_database.dat
#you can just put database=my_custom_database.dat
database= C:\Program Files\USGS\IPhreeqcCOM 3.3.12-12704\database\phreeqc.dat
#To specify equilibrium phases list all sets in the format ["name", Sat. index, amount(moles)] wrapped in square brackets
#If you wish to specify more than one separate each one by commas.
#E.g. [["Name", 0, 10], ["gypsum",0,0]]
equilibrium_phases= [["Gypsum",0,0]]
pH= 7
pe= 4
redox= 'pe'
temp= 0
charge=

[GoldSim]
#Please list the elements from GoldSim in the order they are listed in the GoldSim Vector
#Template: elements=["element_1","element_2",...]
elements= ["Al", "Ca", "Mg", "Na", "S04", "Cl", "Br"]

[GoldQC]
log_file= GoldQC.log
#debug_level can be 0, minimal log; 1, log input and output only; 2, log input output and phreeqc string
debug_level= 0
suppress_warnings = False
```

Figure 10: Elements listed in GoldQC required Format

3. Next, if required update the database file path under the PHREEQC section. This will only be needed if you want to use a database other than the default phreeqc.dat database, your databases are not in the default installation directory or you are not using version 3.3.12-12704 of IPhreeqcCOM. If you want to use a custom database it is recommended to put it in the same working directory as the GoldQC files, doing so will allow you to specify the database by name only, not requiring the full file path.

4. Optionally if you would like you can specify values for pH, pe, redox, temperature(temp), charge balance and one or more equilibrium phases in the phreeqc section.

5. Finally, in the GoldQC section, if you choose you can also change the log file by specifying an alternate log_file, turn on debug by setting debug to 1 or 2 or suppress warnings by setting suppress_warings to True.
   Notice: Turning on debug will increase the running time and log size significantly but will give you increase details of how the simulation ran. See table 1 for debug level details.

3.3 Using multiple instances of GoldQC in one model

Using multiple instances of GoldQC in one model is possible, to do this simply place a copy of the GoldQC dll, GoldQC.py, GoldQC.config and Conversions.py into a new directory, one for each instance of GoldQC you would like in your model. This will allow you to specify separate configs for each instance of GoldQC and receive separate log outputs from each instance.

3.4 Using pH or Alkalinity with GoldQC

When using pH with GoldSim and GoldQC keep in mind that the values will be expected as a H+ value and will be returned to GoldSim as h+ as well not standard pH. In your config file elements list, pH can be specified as either H or pH, GoldQC will treat these the same.

For Alkalinity, in the config element list, it can be specified as Alk, alk, or Alkalinity, GoldQC will handle all these inputs the same.

4 References

GoldSim Technology Group. (2017). *GoldSim*. Retrieved from GoldSim Website: https://www.goldsim.com/Home/

GoldSim Technology Group. (2017). *Linking GoldSim to Python.* Retrieved from GoldSim: https://www.goldsim.com/library/models/featurescapabilities/dllscripts/pythondll/

Heller, T. (n.d.). *comtypes*. Retrieved from Python Software Foundation: https://pypi.python.org/pypi/comtypes

Maurits, L. (n.d.). *PrettyTable*. Retrieved from Python Software Foundation: https://pypi.python.org/pypi/PrettyTable

USGS. (n.d.). *PHREEQC Version 3*. Retrieved from PHREEQC: https://wwwbrr.cr.usgs.gov/projects/GWC_coupled/phreeqc/

Appendix A: Configuration File Template

```
[phreeqc]
#Please specify the path the database you wish to use.
#If your database is in the current working directory you can specify the path
#relative to the current directory e.g. if your database is my_custom_database.dat
#you can just put database=my_custom_database.dat
database=C:\Program Files\USGS\IPhreeqcCOM 3.3.12-12704\database\phreeqc.dat
#To specify equilibrium phases list all sets in the format ["name", Sat. index, amount(moles)]
wrapped in square brackets.
#If you wish to specify more than one separate each one by commas.
#E.g. [["Name", 0, 10], ["gypsum",0,0]]
equilibrium_phases= [["Gypsum",0,0], ["halite", 0, 0]]
pH= 7
pe= 4
redox= 'pe'
temp=
charge=

[GoldSim]
#Please list the elements from GoldSim in the order they are listed in the GoldSim Vector
#Template: elements=["element_1","element_2",...]
elements= ["Al", "Ca", "Mg", "Na", "S04", "Cl", "Br"]

[GoldQC]
log_file= GoldQC.log
#debug_level can be 0, minimal log; 1, log input and output only; 2, log input output and phreeqc
string
debug_level=  0
suppress_warnings= False
use_Config_pH= True
```