

# Contributing to the IQuOD AutoQC project

Simon Good | simon.good@metoffice.gov.uk | December 2015 | Version 0.1

## Introduction

The AutoQC project is a collaborative effort to code up the automatic quality control (QC) tests in use in marine observation processing systems around the world in a common software system. Data with known quality are used to assess which combination of tests best finds bad data without rejecting a lot of good data. An open license is attached to the code so anyone is free to make use of the QC checking code in their own systems. Help is needed to code up QC checks in the system. By incorporating the widest possible selection of QC checks in AutoQC, we can find the best possible combined QC.

#### **Finding the code**

The AutoQC code is hosted in the IQuOD Github repository. It can be found at https://github.com/IquOD.

There are two projects to note:

- 1. The AutoQC project contains the QC checking code and software to run each check on data and analyse results.
- 2. The wodpy project a Python reader for World Ocean Database (WOD; http://www.nodc.noaa.gov/OC5/WOD/pr\_wod.html) ASCII file format.

AutoQC is currently set up to use data in WOD ASCII format so the wodpy code is needed when working on AutoQC. There is a description of the way to use it at <a href="https://github.com/IQuOD/wodpy">https://github.com/IQuOD/wodpy</a>. If the functions provided by wodpy do not meet your needs for your QC checks, please feel free to add what you need, or get in contact and we'll try to help. If modifying wodpy a key piece of information to know is that the data are stored in structures that match the description of the file format in the WOD documentation: <a href="http://data.nodc.noaa.gov/woa/WOD/DOC/wodreadme.pdf">http://data.nodc.noaa.gov/woa/WOD/DOC/wodreadme.pdf</a>. Again, please do get in touch if you need help.

## **Getting set up**

Below are the basic steps to getting a copy of the code to work with. This is based on instructions written for the Met Office Iris software: <u>http://scitools.org.uk/iris/docs/latest/developers\_guide</u>

The first thing that is needed is to set up a Github account if you do not already have one. This is done by following the instructions found at <a href="https://github.com/">https://github.com/</a>.

Development of code is done in your own version of the code repository. You get your own version by 'forking' IQuOD's version of the repository. To do this go to the Github page for the IQuOD repository (https://github.com/IQuOD/AutoQC) and click on the fork button at the top right of the screen.

You now have a fork of the repository in your user account, but to work on it you need the code on your desktop machine. This is achieved by cloning your online repository on your desktop. Note that you need to have git installed on your machine (<u>http://www.git-scm.com/</u>). Instructions below were written for Linux but should be similar or identical on other operating systems.

git clone git@github.com:your-user-name/AutoQC.git cd AutoQC git remote add upstream git://github.com/IQuOD/AutoQC.git

The first line above makes the clone of your repository. The second line changes directory into the clone. The third line links our clone to the IQuOD version. This allows us to download any changes to the IQuOD version of the repository to keep our version in sync with the IQuOD version. This is not covered further here, but there is information on this at <a href="https://help.github.com/articles/syncing-a-fork/">https://help.github.com/articles/syncing-a-fork/</a>.

We also need to download some other software, in particular the wodpy software. The easiest way to do this is using 'pip' (<u>https://pypi.python.org/pypi/pip/</u>). See the instructions at <u>https://github.com/IQuOD/wodpy</u>.

There are also other requirements for running AutoQC, which are described at <u>https://github.com/IQuOD/AutoQC</u>. If problems are encountered then please get in contact.

#### Adding a new QC check

Once the above setting up is done, it is straightforward to add new code to AutoQC:

1. **Open an issue.** To let everyone know what you are doing and avoid duplication of effort it is a good idea to open an issue about the QC check you are adding.

Go to <u>https://github.com/IQuOD/AutoQC/issues</u> and open an issue. Note that there may already be an issue open about the QC check you want to add so check for this first. Either way, you can assign yourself to the issue to show that you are working on it and stop other people doing the same thing.

2. Adding a QC check. This is done by adding a new source code file to the qctests directory. The easiest way to get started is to copy an existing check such as the Argo global range check. Each QC check file is structured the same way. It has to contain a 'test' function that receives as its argument a profile (which is a wodpy profile object) and returns the QC results. Note that, optionally, other arguments can be passed in as keywords. The EN background check uses this functionality and should be looked at if an example is needed of this.

The test function performs the QC check and must return to the main program an array of QC results, ideally one per level (although this is not appropriate in all cases). The array should contain True if the test rejected a level and False if the level passed the check.

- 3. Add tests for the QC code. Ideally there should be a set of tests also implemented for the new check. These run the QC check with a set of predefined inputs and confirms that the expected outputs are returned. This allows us to check that future code changes have not broken the functionality of the check. These are stored in the tests directory and, again, it is easiest to follow an existing example. Please ask for advice if required.
- 4. **Run AutoQC.** To try running AutoQC with your new routine, go back to the AutoQC directory and type the line below. This will run AutoQC on a test set of 150 profiles.

python AutoQC.py

5. Save your changes to the repository. To do this you first need to add your new source code files to the repository, e.g.

git add qctests/my\_test.py git add tests/my\_test\_validation.py

Then, we need to commit the changes to the repository:

git commit

It is a good idea to write some text to describe what you have done when requested. To transfer the code on your local machine to the online repository, we need to do a 'push'.

git push

6. **Request that your code be put in the IQuOD repository.** This is done using a pull request. Go to your repository webpage and click on the green button above the code listing, next to where it says 'branch'. This will bring up a dialogue where you can tell us about your new code and create the pull request.

## **Getting help**

If you get stuck at any stage, please get in contact. An easy way is via the issues page on the IQuOD github website. By including our account names e.g. @s-good in your messages, we'll receive the message. Or, using email: <u>simon.good@metoffice.gov.uk</u>. Finally, if your code is in another language it may be possible to include it without rewriting – please email us about this if you are interested.