Operator's Reference Manual

for

MQUEST

(Matlab-based Quality Evaluation of Subsurface Temperatures)

QUEST Version 2.0 featuring platform independence
(formerly restricted to Silicon Graphics platforms)

by

Ann Gronell Thresher

CSIRO Marine Research
GPO Box 1538, Hobart, Tasmania, 7001, Australia.

1. Introduction **3**
2. Scientific Quality Control Procedures **5**
Quality control of the temperature profile: **6**
3. Statistical QC procedures (QUOTA QC) **8**
4. Mquest System **8**
5. Getting Started **9**
Hardware Requirements: **9**
Setting up the data: **9**
Data Input: **10**
Data Export: **11**
Setting up the program - Defining Display parameters: **14**
Defining Quality Control Flags: **17**
6. Gui Control: **18**
Program Control and QC **21**
Functional keys **21**
Quality Control keys: ..... Error! Bookmark not defined.
7. Troubleshooting: **23**
8. Acknowledgements: **23**
9. References: **24**
10. Appendices **25**

Note: A quick start guide can be found at:

http://www.marine.csiro.au/~gronell/Mquest/Mquest_Quick_Start_Manual.doc

1. Introduction (back to TOC)

Scientific quality control is the process of combining statistical analyses of data (in this case, subsurface temperature profiles) with knowledge of historical means (climatology) and of the relevant environment (regional oceanography) to make a scientifically based decision about the validity of each data point. It is a vital step to be taken with any data set before scientific analysis can proceed. Knowledge of the recording instrument's performance and characteristics must also be used when evaluating the data.

The Commonwealth Scientific and Industrial Research Organisation (CSIRO) Division of Oceanography and the Bureau of Meteorology Research Centre (BMRC) have developed a comprehensive software system for the scientific quality control of sub-surface ocean temperature data. This system, called Mquest (which stands for Matlab-based Quality Evaluation of Subsurface Temperatures) is based on 16 years of experience with a system called QUEST (Quality Evaluation of Subsurface Temperatures).

QUEST combined the subsurface ocean temperature statistical analysis scheme developed by BMRC and CSIRO (Smith et al., 1991; Meyers et al., 1991), with the quality control procedures of CSIRO as outlined in Bailey et al. (1994) (henceforth referred to a the CSIRO Cookbook). This was the basis for all upper ocean temperature QC done at CSIRO and BOM. Mquest now uses the CSIRO Atlas of Regional Seas (CARS) to provide a sophisticated background for data assessment instead of the statistical analysis used by QUEST.

QUEST was coded in Fortran and would run only on Silicon Graphics computers. It was felt that Matlab would provide a platform-independent means of performing the same tasks in a simpler way, thereby allowing data managers who did not have access to SGI machines to use the Mquest system.

Mquest enables individual temperature profiles to be compared to the CARS climatology and to other profiles in the immediate area in order to identify real features of a given region and to help distinguish between such features and erroneous data. The scientific QC procedures are summarised in Figure 1.

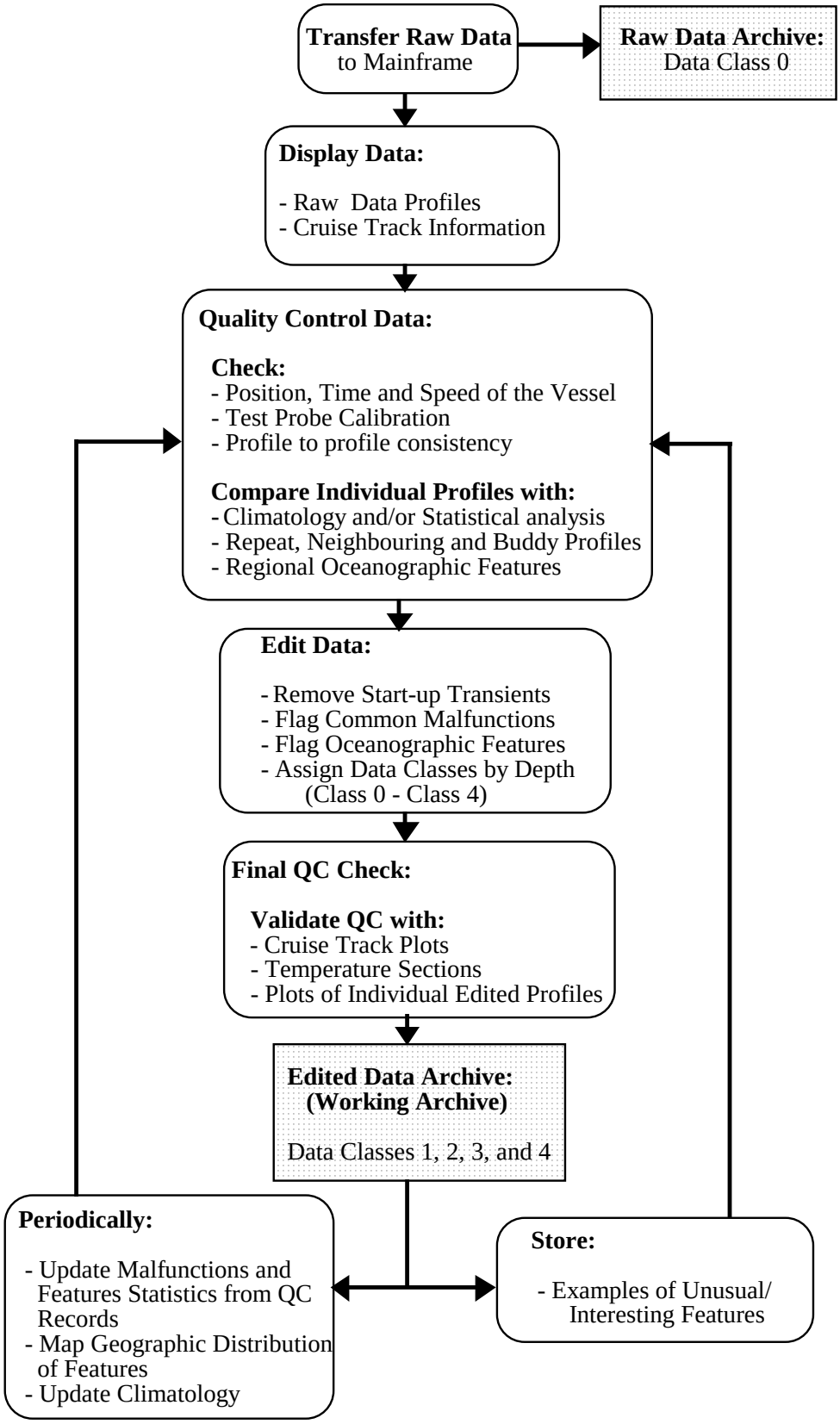


Figure 1. Scientific Quality Control Procedures

2. Scientific Quality Control Procedures (back to TOC)

Not all data can be scientifically quality controlled. Very large data sets with very old data are particularly difficult to QC. Where possible, we recommend full scientific quality control of every profile as described here. Where it is not possible, we apply statistical quality control with additional scientific quality control, using Mquest, of any profile that fails the statistical check. This process is described in full in "IOTA paper" and summarized in section 3.

Scientific quality control is necessary because the recording of temperature, by the many instruments used over time, can be subject to malfunctions which mimic real oceanographic features. For example, a temperature inversion (which is a feature of ocean temperature) and a wire stretch (which is a fault seen in XBT data) both exhibit an increase in temperature with depth. A compression of depth due to a badly calibrated depth sensor on an MBT can mimic upwelling in coastal regions.

If you can be sure that the feature is real, or if a malfunction is thought to result in minimal impact on the quality of the data, then the data is accepted and the appropriate flag and quality code is applied. If, however, there is doubt about the feature, or the impact to the quality of the data is severe, then the data is rejected. The operator takes a conservative approach to QC and uses the flag which results in the lower class of data if there is any uncertainty as to which code to use. The classes have been constructed to be sufficiently flexible so that good data will rarely be discarded if this method is followed.

Bathy-message (real-time) data is quality controlled with a certain amount of caution because this data is of lower resolution. Many of the subtle faults, which are identified for XBTs in the CSIRO Cookbook, will not be encoded in a bathy message and therefore may be difficult to detect. These messages appear in the data (particularly from WOCE) if the higher resolution XBT has not been submitted to the data centre of if they have slight variations in position from their delayed mode counterparts so they escape detection by duplicate checking routines. In the past, only "good" data was submitted to the data centres and so the corresponding bathy messages remained.

Other recorders have their own issues with quality and so it is necessary to use the data type to guide your decisions.

Though the primary aim of this software is to allow elimination of "bad" data so it doesn't affect subsequent analyses, we have found it also useful to flag features. It eliminates the need to reconsider a real feature you have previously assessed (but perhaps not flagged) and also allows mapping of regions where certain features are prevalent, thus increasing your knowledge of ocean dynamics.

The Scientific QC process uses a visual approach combined with an interactive editor (Mquest) to check and flag the upper ocean temperature data. The header information is tested for correct time, position and other operator errors. The vertical profiles are inspected for

common malfunctions, regional oceanographic features, profile to profile consistency along the cruise track, and repeat profiles of unusual features. All unedited/raw data is also stored so it is available if required.

Quality control of the temperature profile: **(back to TOC)**

Once the header information has been validated, the individual profiles are inspected. Any unusual features are checked by comparison with repeat or neighbouring profiles, and by using an archive of other profiles in the region. A repeat profile is a measurement which has been done within 15 minutes of another and a neighbouring profile is a measurement which is taken approximately every 100km along the route. This so-called "buddy" checking is one of the most powerful tools of the Mquest program.

Doubtful features can appear and be flagged at any point in the profile. Care must be taken in labelling features as doubtful, since a high proportion of the observed temperature structure is real.

The profile data is edited to remove start-up transients (see CSIRO Cookbook) and other malfunctions, and to flag real oceanographic features. If an anomaly is considered the result of a malfunction, the data quality is downgraded. The temperature values are classed (0-5 – Table 1) based on the QC flag applied and this value is stored with each temperature datum. For a detailed list and description of the QC codes Mquest can apply, see the CSIRO Cookbook and the summary in Appendix A.

Class	Quality	Description
Class 0	No QC Done	Class 0 data is the level at which all data enters the working archive, and has not yet been quality controlled.
Class 1	Good Data	Class 1 data is top quality data in which no malfunctions are identified and all real features have been verified during the quality control process.
Class 2	"Probably" Good Data	Class 2 data is good data in which some unusual but probably real features which are unconfirmed, and/or malfunction errors which can be corrected or are small enough to be ignored without seriously affecting the overall quality of the data, are observed. Data is downgraded to Class 2 from the depth of anomalous (probably real) features.
Class 3	"Probably" Bad Data	Class 3 data is suspect data in which some unusual and probably erroneous features are observed. Data is downgraded to Class 3 and rejected (may be retrieved) from the working archive from the depth of anomalous (probably erroneous) features.
Class 4	Bad Data	Class 4 data is bad data in which obviously erroneous values are observed. Data is downgraded to Class 4 and deleted from the working archive from the depth of erroneous features.
Class 5	Value changed	Class 5 data indicates that the original data value has been altered by Mquest; the original value can be found in the history record.

Table 1. Data Quality Classes as defined by WOCE for all upper ocean temperature data.

3. Statistical QC procedures (QUOTA QC) (back to TOC)

CSIRO has been active in developing statistical screening processes for the large body of un-QCd data in the oceans around Australia. Partly funded by the International Pacific Regional Center (IPRC), we have developed statistical procedures which allow us to subset the data that needs further examination, passing as much as 70% without further QC. In the Indian Ocean, this has resulted in considerable time savings, allowing us to create an Indian Ocean Temperature Archive of highly reliable data by directly QCing only a fraction of the dataset.

Eleven parameters are calculated from each profile in the archive. Means and standard deviations are calculated for regional boxes and the parameters are compared to these statistics. Where a parameter deviates by more than a predetermined number of standard deviations from the mean (this critical value varies with the inherent distribution of each parameter) that profile is flagged for hand-QC.

The profiles that fail the statistical screening are then viewed using Mquest against a background of highly QCd data. The Mquest operator must make a decision about whether or not the data in the profile is good; bad data is flagged and is not used in any further analyses. Real features such as temperature inversions are not flagged in this level of QC.

A full description of this process along with a discussion of some complications found during the development of the system can be found in “the IOTA paper” (ref).

4. Mquest System (back to TOC)

Mquest is a powerful, interactive, screen editing system based on Matlab7. It allows you to visualize, interrogate, and manipulate the data and it is the tool by which the scientific quality procedures are implemented. Through a master GUI with interactive controls and pop-up menus, it displays the individual temperature profiles and neighbouring or buddy data, actual data values, station position locations, and a waterfall of profiles adjacent in the database. Both real features and bad data observed on individual temperature profiles can be flagged using the CSIRO QC flags as defined in the CSIRO Cookbook (Appendix A). Also listed in Appendix A are additional flags that have been added since publication of the cookbook.

Entry of some flags requires the data itself to be edited, and this is done automatically whenever possible. In some cases, it is necessary to enter information directly using the keyboard and/or mouse as prompted. Secondary GUIs are occasionally used for input of instructions for particular flags or actions.

After a profile has been edited, the CSIRO QC flags and the data quality classes, are also displayed. All flags associated with the data are permanently stored with each profile, including the original value if, for instance, the program has modified a datum.

5. Getting Started (back to TOC)

Hardware Requirements:

Mquest runs on any computer system that has Matlab7.0 or higher installed with a valid license. It runs fastest if the data directories are "local" rather than accessed over a network but running over a network is feasible. It can run on a PC using a linked drive to a network or using a local directory. Rendering of the GUI can be slow when run on a secondary computer (such as using X-win or remote login).

A new data structure has been designed and all databases must be in this format before Mquest can access the profile data. This is based on netcdf and has been designated MQNC (Mquest NetCDF) to distinguish it from other netcdf versions of temperature datafiles. This format is a hybrid of the Argo format and the WOCE format used by NODC. Multiple profiles of a single station (e.g., temperature, salinity and conductivity) can all be held in one netcdf file and each station is held in a separate file.

Setting up the data: (back to TOC)

The underlying file structure is based on the MEDS-ASCII GTSPP format used by the United States National Oceanographic Data Center (NODC) for distribution of some sub-surface ocean temperature data sets (see Appendix B for format details). All MEDS-ASCII fields are present so the original record can be reconstructed if required and creation of files for data exchange is easier.

Each database is structured as a directory tree. The database has a name or 'prefix' identifying the top directory with subdirectories that lead to the profile files. The files are found using a unique number associated with the data. This number is broken into 2 character bits that make up the path to the data. The last character or two become the file name with a suffix of either "ed.nc" (edited data) or "raw.nc" (raw, untouched data).

For example, the profile with the unique ID of 5479823 in database IOTAeast would be found in IOTAeast/54/79/82/3ed.nc and in IOTAeast/54/79/82/3raw.nc. The creation of this directory tree is automatically handled when the database is created. Each directory will contain at most only 200 profiles (0-99, .ed plus .raw) which overcomes limitations of some computers when you try to list too many files in a directory.

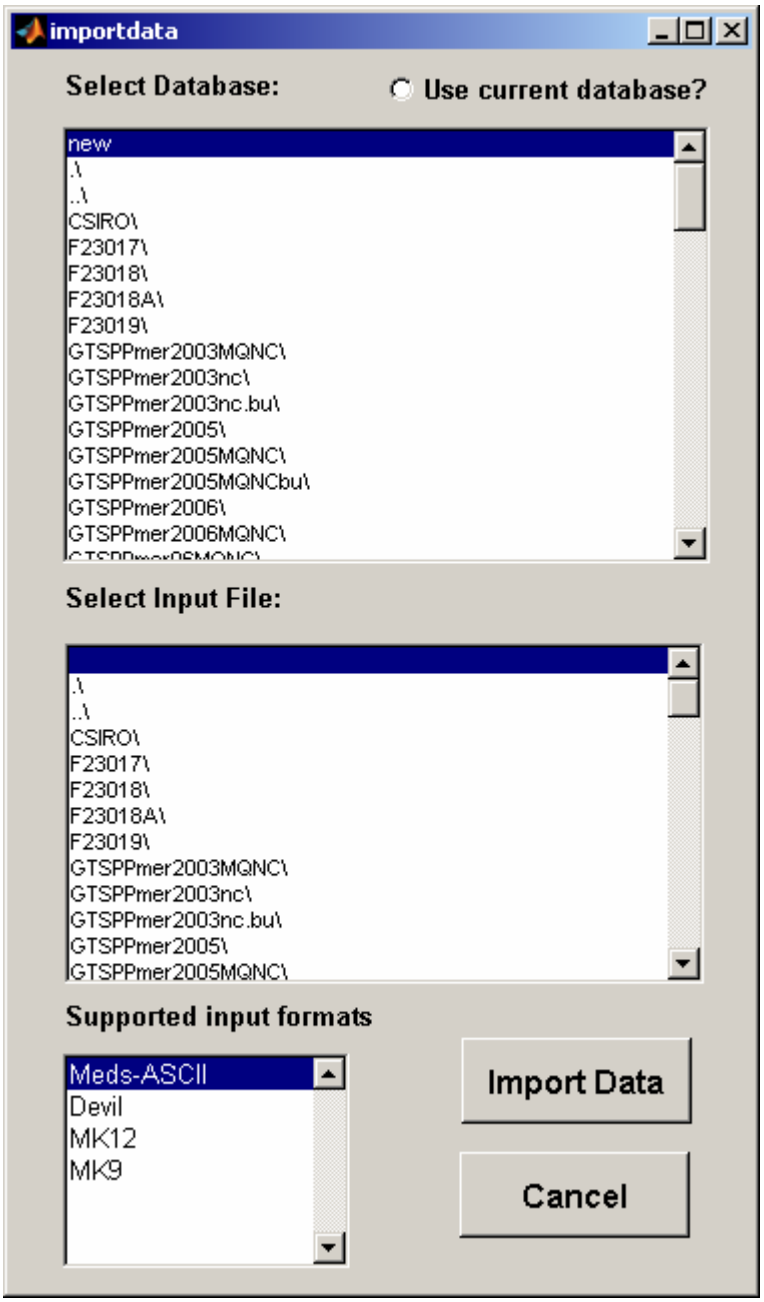
The primary requirement for this structure is a unique identifier for each profile and this will have to be provided if it is not already associated with the data. NODC supplies data with this identifier already assigned; we supply a CSIRO identifying number, or "CSID", when we add our own data to the database. This number is retained by NODC when it assigns its own database identifying number, or "DBID", after data submission. The identifying number is stored in the Surface_CODE record of each profile (see Appendix B).

The unique ID is also kept in a netcdf file that contains all the metadata required to retrieve any profile from the database – the 'keys' file (see the structure of his file in Appendix C). A 'keys' file is associated with each database and is named "databaseprefix"_keys.nc. Variables included are: Latitude, longitude, date, time, unique ID or station number, ship callsign if known, data type (XBT, CTD, etc), and whether the profile has failed one or more of the statistical screening tests (used only for QUOTA QC).

The order of the profiles within the keys file is irrelevant; Mquest allows you to select profiles based on month or year of the profile and they can then be sorted (if required) by callsign, time and date (to allow QC by ship voyage), or latitude (to associate data from similar regions), before processing begins. This means you can enter data at any time, regardless of its time sequence and profiles within a cruise need not be sorted before addition to the database. These sorting and selection criteria are chosen when Mquest begins.

Data Input: **(back to TOC)**

Mquest can build a database directly but file IO is relatively slow using Matlab. We therefore recommend you use another language (we use Fortran) for input of large numbers of profiles. For smaller additions, Mquest can be used and will allow you to begin QC of the data immediately.



At present, only Devil data and MEDS-ASCII can be directly input to Mquest. If required, other formats can be easily added.

The Fortran programs currently used to convert or append MEDS-ASCII data to a database are called **convertM2MQNC** and **convertDEVIL2MQNC**. These are easily converted to input other formats. The conversion programs prompt for the output database *prefix* and the input file name or, in the case of Devil data, the directory name. If the database doesn't exist, the input programs create it and all necessary directories and files. The profile file structure can be found in Appendix D.

Data Export: **(back to TOC)**

Data output is controlled by Mquest which can create either MEDS-ASCII data files or 2m (so-called “reformatted”) data files. The latter is our primary format for data archiving and our secondary format for data exchange. For export of large databases, we recommend Fortran. Currently, only **convertMQNCtoM** exists. We then use **convertM2newref** to create the 2m files.

Variables stored in the *_keys* file: (all field references can be found in Appendix B)

- ***Stn_number*** is the unique number assigned to each temperature profile. It is contained in the *SRFC_Parm* record of the MEDS-ASCII format identified by *SRFC_Code* = "??ID". Some profiles have an identifying number assigned by CSIRO, denoted by *SRFC_Code* = "CSID". The NODC unique identifying number is denoted by SRFC_Code = "DBID".

- ***Callsign*** is the callsign of the ship or platform taking the measurement (if known). This is contained in the *SRFC_Parm* record of the MEDS-ASCII format identified by *SRFC_Code* = "GCLL" and (with the date and time) is used for sorting the observations to allow viewing of the data by cruise-track.

- ***Obs_Year***, ***Obs_month***, ***Obs_Day***, and ***Obs_Time*** identify the date and time at which the measurement was made. Occasionally, we see a missing value for time which appears as 99.

- ***Data_Type*** identifies the source or method of acquisition of the data (e.g., "XB" = XBT, "BA" = Bathy message - see detailed code tables at - http://www.meds-sdmm.dfo-mpo.gc.ca/Meds/Databases/OCEAN/GTSPPcodes_e.htm

- ***obslat = Latitude***, ***obslng = Longitude*** = ***c360long*** (the raw longitude which is usually +/- 180 degrees converted to 360 degrees). Originally we recorded the original longitude in the obslng field but, because different data originators use different protocols (some us + for east, others use + for west longitudes), it is less confusing to retain only the 360 degree value in this file.

- ***source*** identifies the source of the data which can provide information about the level of QC applied and the reliability of the data.

- ***priority*** is a number that helps us prioritize the data based on the data source – this helps us determine which profile to keep when we detect duplicate profiles in the database. A high priority would be 1 (assigned for data that has been scientifically QCd); a low priority would be greater than 5 (assigned for data that has undergone no QC or QC that is not trusted).

- ***autoqc*** is 0 if the profile has either passed the statistical screening process or has not yet been screened and is 1, 2 or 3 if it has failed one or more of the statistical screening steps. This allows quick selection of data that has failed a statistical test so you can examine only those statistical failures.

A full description of the netcdf file structure for the keys data can be found in Appendix C and for the profile data can be found in Appendix D. Variables are generally named in accordance with the MEDS-ASCII format (Appendix B) though there are some additional

variables to make processing of the data easier. These variables are defined and communicated between Fortran programs and subroutines using the commons in the include file **newGTSPP2** (Appendix B).

Setting up the program - Defining Display parameters: **(back to TOC)**

When you first start Mquest, you must tell it how you want it to sort the data, which data subset you want to view (by month or year, choosing only certain QC codes or viewing only data that has failed a statistical screening). You must also specify which database to display and you can, at this point, add or modify the files you wish to use as "buddies". These are the databases you want displayed in the background to help you decide whether the data being QCd is reasonable.

This is made easier by assigning a 'username' to each configuration that is commonly used. When you start Mquest, you type:

quest('username')

This identifies the version of the program you wish to activate. Usernames cannot contain spaces, must be supplied in single quotes as shown and can contain any legal string characters. It is useful to name each version according to its application – so the version used to QC the IOTA east data is invoked by typing:

quest('iotaeast')

and the program then retrieves the settings you used the last time this version was run. When required, these settings can easily be changed before Mquest is invoked. The user can also be changed at this point using the list box at the top of the GUI. If the chosen user name has not been used before, default values for all parameters are set and these can be changed before you begin.

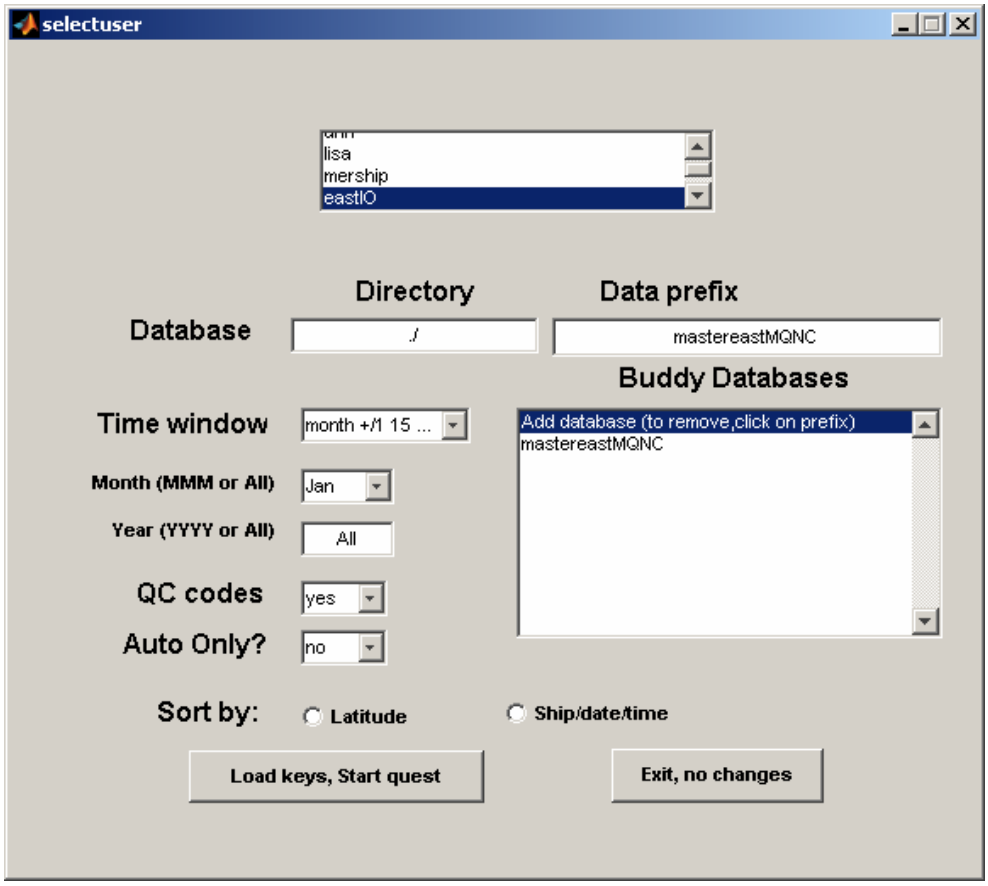


Figure 2. GUI started by "Mquest" command

Within this GUI, you can alter the time window for display (all months, single month or month +/- 2 weeks), the month to be displayed (pull down menu) or the year to be selected (text box).

You can specify whether you want to use a pre-selected subset of the database (created with the Fortran program **extractqcprofiles)** that allows you to display only those profiles with specific QC flags already applied. This program reads every profile in the database and then creates a "databaseprefix"_keysQC.nc containing the keys information for profiles that have been given the specific QC flags required; this keys file is then read instead of the full set of keys. You can also ask the program to use only those profiles that have failed the statistical screening (the variable 'autoqc' in the keys file).

Sorting of the profiles can also be controlled. If you click in the latitude box, then the profiles are sorted by latitude; if in the Ship/date/time box, then the profiles are sorted by callsign, date and time before display. If neither box is checked, then the profiles are not sorted and appear in the same order as they are stored in the keys file.

To change the database, simply type the new name and path into the edit boxes, though Mquest is usually run in the directory containing the database.

Viewing a cast in the context of its near neighbours (or “buddies) is probably the most powerful feature of Mquest. Buddy profiles can be viewed from files other than the working database. The buddy profiles for each QC session will be selected according to the rules established in the opening GUI so if you specify one month, then the buddies will all be from that month.

To add buddy files, click on the first line of the buddy list. This will start a GUI that will allow you to browse for additional buddy databases. Clicking on a directory will take you to that directory. Clicking on the “keys.nc” file of the required database will add it to the buddy list (see Figure 3). If you wish to remove a database from the buddy list, click on that database in the ‘quest’ GUI and it will be removed.

Figure 3. GUI to select BUDDY databases

Defining Quality Control Flags: **(back to TOC)**

The quality control flags have been defined in the CSIRO Cookbook (Bailey, et al., 1994). In order to communicate these flags to the program, they have been placed in two files read when the Mquest GUI starts. These populate the pull down menus that can be used to apply all flags. The files are called **questAflags**.txt (containing flags that are applied to features and suspected features – 'accept' flags) and **questRflags**.txt (containing the severe flags that result in the data being rejected – 'reject' flags). These files also contain information required by the program when applying these flags (i.e., the quality class assigned by the flag, and the depth at which to apply the flag).

The flags are sorted alphabetically with commonly used flags which don't have a keystroke shortcut first (if the flag you need is not at the beginning of this list and is used regularly, it probably has a keystroke shortcut – see appendix F). Then all other cookbook flags are listed, followed by the flags specific to the statistical QC process. This makes the flags easy to find in the list and simplifies the addition of new flags or the redefinition of old flags should this be necessary.

Additional flags have been defined for the statistical screening process and these can be found in Appendix E. These are usually applied by the screening programs; they will only be applied using Mquest if you have removed them during QC and want to retain the codes for future reference. They are found at the bottom of the pull down menus.

An additional parameter has been defined in the history record. This is "flag_severity" which defines the quality of the data applied by the flag. This helps distinguish between "accept" and "reject" versions of the same two character code and speeds up processing.

Once you have finished tailoring the GUI to your database, click on "load keys, start Quest" and the Mquest GUI will launch.

6. Gui Control: (back to TOC)

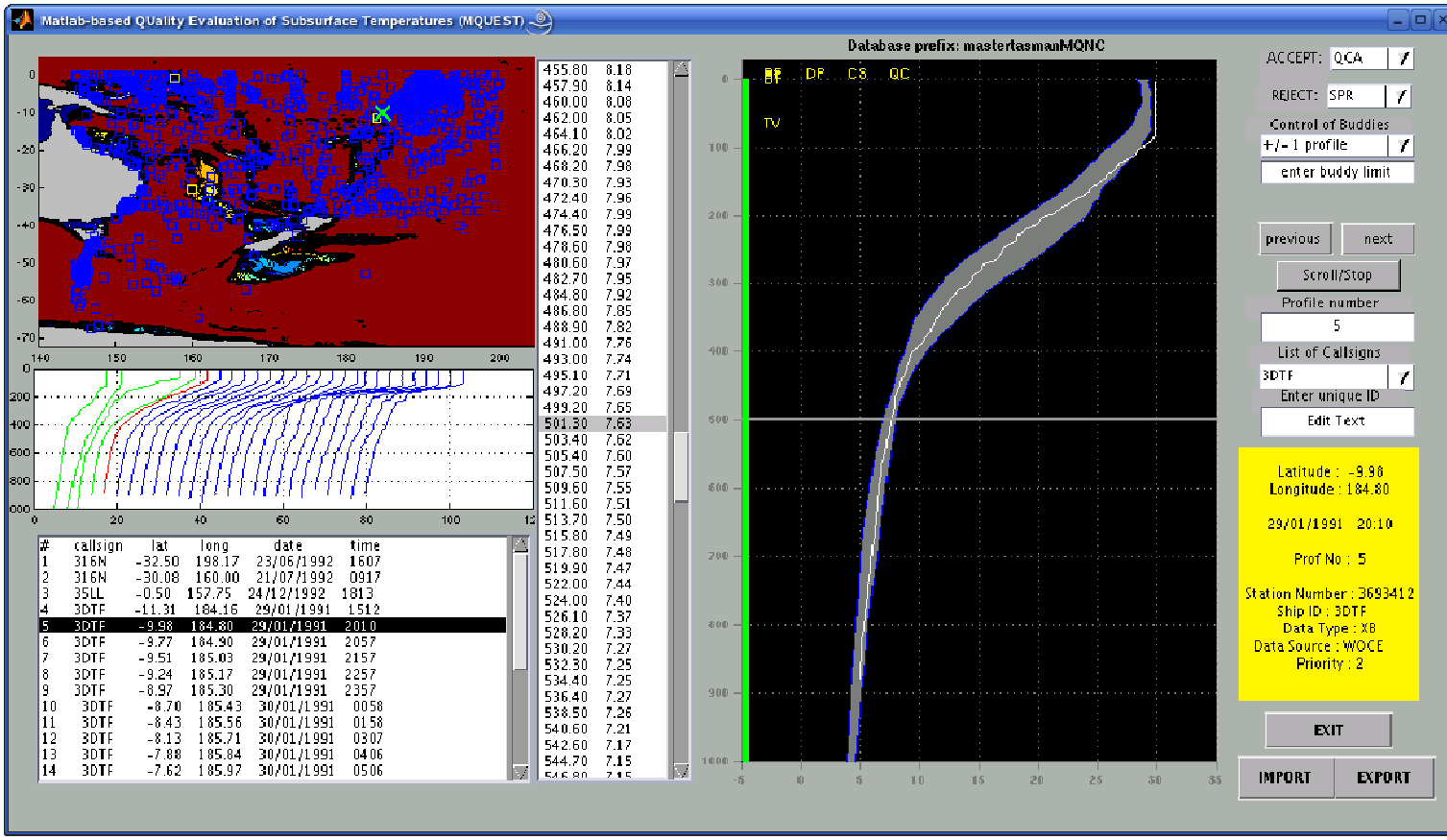


Figure 4 The Mquest GUI window

The GUI consists of several panels, pull down menus and buttons. The profile window displays the first profile in the data subset chosen. If QC has been applied, there will be a coloured band at the left of this window showing the quality of the data (Table 1) at each depth. If there is no coloured band, then the quality of all the data is '0'. Green indicates data of the highest quality, class 1; blue is class 2, yellow is class 3 and red indicates class 4 data. This colour is determined by the flags applied. It is easy to see where any change in data quality occurs. In addition, the two-character codes for the QC flags applied appear on the left-hand side of the plot, at the depth where they apply.

Behind the profile plot is the grey 3-sigma error band for the CARS climatology which helps you decide whether the data is good or bad.

Buddy comparisons (looking at a profile in the context of its near neighbours) are an integral part of deciding whether an apparent anomaly is real or the result of a malfunction. Buddy profiles can be displayed on demand either by using the "Page Up" button or by

changing your buddy selection criteria (see “Buddy Checking” – pg 19). You can toggle between the buddy display and the single profile display using the “Page Down” button (single profile) and the "Page Up" button (buddy profiles).

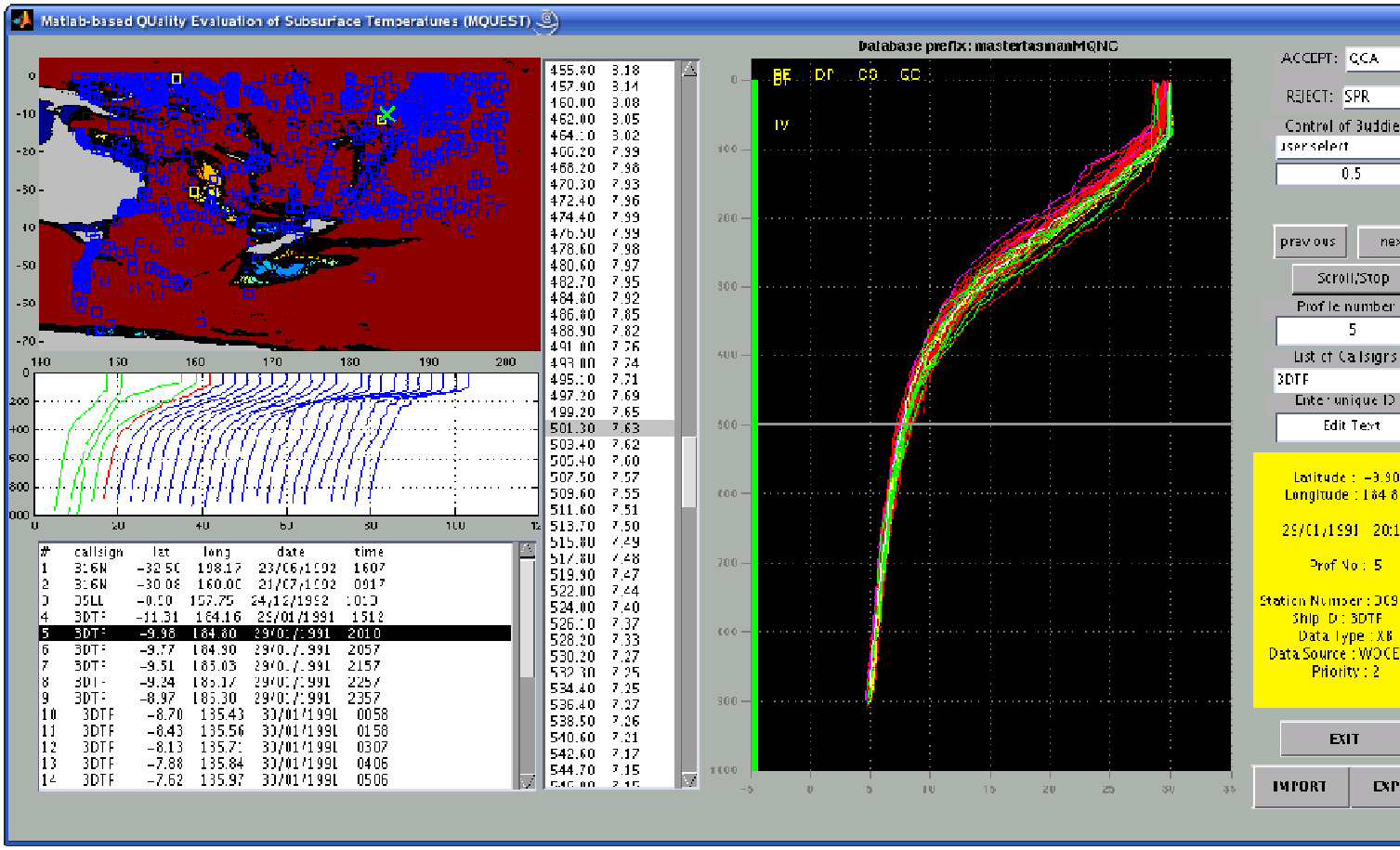


Figure 5. Buddy window - red profiles are from another dataset; magenta profiles are from the same year as the primary profile shown in white. The magenta profiles are clearer on-screen.

To the right are buttons that control how the profiles are displayed and menus that allow you to add the flags defined for Mquest. There is also a panel that contains the metadata for the current profile.

Moving between profiles can be done with the “**previous profile**”, “**next profile**”, and “**Scroll/Stop**” buttons. Once Scroll has started, you can only stop it by clicking on the button again (now labelled “**Stop Scroll**”). The **uparrow** and **downarrow** keys also move forward and back one profile in the list.

If the profiles are sorted by ship/date/time, profiles from a specific ship can be displayed by selecting that ship’s callsign from the pull down menu labeled “**List of Callsigns**”. This

automatically displays the first profile collected by that ship in the current subset of the database.

You can move between profiles using the "**profile number**" box but note that the number of each profile is specific to a particular subset – if a profile occurs in more than one subset (for instance if you select "All" months and in a later session select "Jun") it will have different numbers within the two subsets depending on its order in the sorted list.

If you know the unique ID of the profile you wish to see, it can be entered in the "**Enter unique ID**" box. This must be in the current dataset or it cannot be displayed.

You can also specify the way you want your buddies to be displayed using the two boxes under "**Control of Buddies**" – switching between all buddies in a user-selected area or buddies from the same cruise within 1 or 2 profiles of the current profile. This can be changed either by using the pull-down menu, by defining a radius in the edit box below or with keystrokes which are mapped to the most useful buddy areas. Further instructions and details can be found in the "Buddy Checking" section (pg 19).

To the left of the profile plot is a list box that shows you the temperature/depth pairs of the current profile, centred on either 500m or the bottom of the profile. There are two ways to move the line. You can click in the profile window which will move the line to the cursor point and also move the selected depth/temperature to match, or you can click on a depth/temperature pair and this will move the line to the appropriate depth in the profile window. When you click in the profile window, you need to be outside of the grey climatology plot or matlab doesn't respond. Many of the QC flags require a depth and this is usually provided by selecting the depth in this list box.

Further to the left are three panels displaying (TOP) the mapped locations of all profiles in this data subset (with bathymetry), (MIDDLE) a waterfall of the 25 profiles before and/or after the current profile (this moves only when you reach the end of the profiles being displayed since it takes significant time to re-display), and (BOTTOM) a metadata list of the profiles shown in the waterfall plot. Clicking on one of these metadata entries will display that profile in the profile window. In the map plot, positions of all plots in the data subset are marked with blue 'x's; the position of the current profile is marked by a green 'x'; the positions of profiles already examined (in this QC session only) are marked by yellow boxes. The current profile is shown in the waterfall window in red; profiles already examined are green; the rest are blue. When you QC a profile and reject some data, that part of the profile remains red when you move to the next profile. This is a good way to see if you have removed all bad data.

At the bottom of the profile plot, you will occasionally see a red message:

"THIS IS A REPEAT PROFILE"

indicating that another profile was taken within 15 minutes. This is useful only when the data has been sorted by ship/date/time.

If the data has been sorted by ship/date/time, the program also checks forexcessive speeds and displays the message:

"EXCESSIVE SPEED!!"

if the speed is greater than 25 knots.

Program Control and QC **(back to TOC)**

Mquest is designed to be controlled using keystrokes. These are significantly faster and easier to apply than using menus. As much as possible, keystrokes are defined logically (so 'q' means the following key will indicate a qc flag and 'i' means the qc flag to be applied is IVA). All keys are defined in lower case. In the descriptions below, keystrokes mapped to the description are shown in [].

Key mapping is simplified by use of a text file containing all the mapped keys which require a leading 'q' and their meanings ('**qkeyflags.txt**'). If you wish to redefine a key, this file can be edited and the changes will take place the next time you start Mquest. Some keys do not require pressing the 'q' key first – these usually perform more complicated flag processing and are defined in the Matlab code. These cannot be modified without editing the Matlab script 'getkeystroke.m'. Some QC flags do not have keys mapped and these must be chosen by the pull-down menus for either 'reject' or 'accept' flags (see above).

Functional keys (keys used to change program behaviour, not to apply QC codes)

Buddy checking - Buddies appear in the profile plot panel and are colour coded to give you additional guidance. Red and magenta profiles are from other datasets. Magenta profiles are from the same year as the current profile; red profiles are from different years. Profiles from the current dataset are coloured either yellow or green. If the current profile is north of -35 latitude, then profiles to the east of the current profile are yellow and west are green. Otherwise, profiles to the north are green and south are yellow. This sounds confusing but is designed to give you more detailed information about the ocean structure. For example, if the profile is in a frontal area, buddies to the east might all be warmer and west might all be colder so knowing which is which is useful. Similarly, buddies from the south would be expected to be colder than buddies from the north. The current profile is always white and plotted last so it is visible. When buddies are displayed, the climatology disappears.

The default buddy display is set to show profiles from the same database and same ship as the current profile. You can choose to display the profiles 1 [default - **F5**], 2 [**F6**], or 3 profiles (pull down buddy menu list to the right) either side of the current profile along the ship track. This is useful for larger scale features which may appear in several profiles along the cruise track, i.e., an inversion which can be seen developing in one profile, intensifying in the next and fading in the third. Without buddy checking, it would be difficult to determine whether or not the apparent inversion was real or the result of a wire stretch (Bailey et al., 1994).

You can also display all profiles within a specified radius (+/- latitude and longitude) of the main observation (1/2 [**F7**], 1 [**F8**], or .1 degrees [**F10**], 1.5 [**pad 5**], 2.0 [**4**], 4.0 [**pad 6**] or .25 degrees [**pad 3**]. For even more flexibility, you can specify any radius you wish by entering the number into the buddy text box. These buddies come from all the files defined in the opening GUI.

When the buddy region is changed, Mquest immediately displays the new buddies.

NOTE: to prevent excessively long pauses while Mquest draws the buddies in the window, there is a limit of around 100 to 150 buddies. If there are more than 100 buddies, Matlab chooses and plots an even distribution of the profiles; this prevents the 100 buddies coming from the beginning of the data subset.

To further speed the buddy plotting, you can use [**F9**] to restrict the buddies to the 10 closest to the current profile. The distance from the current profile is calculated and only the closest 10 from each buddy dataset are displayed. If you have selected "All" months, then these buddies come only from the month of the current profile so they are closest both in space AND time. This is denoted in the buddy list box with the buddy area in parentheses as (eg.):

10 buddies max (0.25)

where 10 is the number of buddies shown and 0.25 is the area (+/- latitude and longitude) from which they are drawn. To disable this limitation, press [**F9**] again – this acts as a toggle and will re-plot the buddies using all available in the current area.

Buddy profiles can be further restricted by choosing to display buddies only from the same year as the current profile. This can only be changed by using either [**pad 1**] (single year for buddies) of [**pad 2**] (all year buddies). In case no number pad is available, these functions have also been mapped to the [**7**] and [**8**] keys.

By default, Mquest displays only the 'good' data from buddy profiles. You can also choose to display "all" of the buddy data [**F12**] and toggle back to displaying only the "good" data [**F11**]; this can be changed only by use of the relevant keys (**F11** or **F12**).

Exit function - [or **ESC] -** Save the QC (if necessary) and exit the program. NOTE: DO NOT EXIT THE PROGRAM BY CLOSING THE INDIVIDUAL WINDOWS!!! THIS COULD RESULT IN CHANGES NOT BEING PROPERLY SAVED!!! Use only the exit button or the escape key to quit Mquest.

Zoom profile window and reset functions - [**right arrow** key - zooms the window to display 100 metres above and below the depth line; **left arrow** key – returns the window to it's original aspect ratio] - The zoom command changes the display to give better visual resolution of fine scale structure. The focal depth can be changed by moving the cursor within the window (clicking the left mouse button) but this does not reset the x-limit of the plot and so the profile might move off scale. If this happens, use the left arrow key to undo the zoom, then use the right arrow to zoom in again. This is a limitation of the current version of Mquest. The scale is retained in the Buddy Window, which helps in comparisons of small features.

Home function – [**Home]** displays the first profile in the current subset (profile number 1). If you press this by mistake, the previous profile is listed in the Matlab main command window and this number can be entered into the **profile number** box.

Zoom map window and reset functions – [**Delete** and **Insert**] – zooms [**Delete**] the map plot to focus on the position of the current profile / resets [**Insert**] it to show the original view of the entire data subset. This can help when you need more information about the local bathymetry to determine whether the probe has hit the bottom or not. While the bathymetry is good, it is not perfect and so caution should still be used when applying the hit bottom (HBR) flag.

For a Summary of Functional Keystrokes and Quality Control Keystrokes, please see:

http://www.marine.csiro.au/~gronell/Mquest/Mquest_keystroke_summary.html

7. Troubleshooting: (back to TOC)

As with any new system, there are still a few bugs. Most can be handled with a bit of patience. Please report all bugs to:

Dr. Ann Gronell Thresher
CSIRO Marine and Atmospheric Research
GPO Box 1538
Hobart TAS 7001
Australia
Phone +61-3-6232-5419 or 03-6232-5419
Email: ann.thresher@csiro.au

8. Acknowledgements: (back to TOC)

Neil Smith and Neville Smith were instrumental in moving program development forward and their contribution to the final result was invaluable. Neville Smith and Gary Meyers developed the original analysis techniques used by a primitive version of this system and Rick Bailey, Helen Phillips and Ann Gronell developed the Quality Control procedures implemented with this system. Further help was provided by Edwinna Tanner. Funding was provided by the Bureau of Meteorology Research Centre (BMRC), CSIRO Division of Oceanography, and the Department of the Environment (DEST). This work would not have been possible without support from the CSIRO Climate Change Research Program.

9. References: (back to TOC)

Bailey, R., A. Gronell, H. Phillips, G. Meyers and E. Tanner. 1994. *CSIRO Cookbook for Quality Control of Expendable Bathythermograph (XBT) Data.* CSIRO Marine Laboratories Report: 221. 75pp.

Gronell, A., 1995: Operator's Reference Manual for QUEST (Quality Evaluation of Subsurface Temperatures) (unpublished manuscript).

Levitus, S. 1982. *Climatological Atlas of the World Ocean;* NOAA Professional Paper 13, 173pp.

Meyers, G., H. Philips, N. Smith and J. Sprintall. 1991 Space and time scales for optimal interpolation of temperature - Tropical Pacific Ocean. *Prog. Oceanog.* 28:189-218.

Phillips, H., R. Bailey, and G. Meyers. 1990. Design of an Ocean Temperature Observing Network in the Seas North of Australia. Part II; Tropical Indian Ocean: Statistics. CSIRO Marine Laboratories Report: 211. 58 pp.

Smith, N.R. 1991. Objective Quality Control and Performance Diagnostics of an Oceanic SubsurfaceThermalAnalysisScheme. *J.GeophysicalRes.* 96:3279-3287.

Smith, N.R. 1993. The BMRC Objective Analysis System. WOCE Upper Ocean Thermal Data Assembly Centre Coordinating Group, Report of the Fourth meeting (UOT/DAC *IV). AppendixE.*

Smith, N., J.E. Bloomley and G. Meyers. 1991. A univariate statistical interpolation scheme for subsurface thermal analyses in the tropical oceans. *Prog. Oceanog.* 28:219-256.

WOCE. 1990. WOCE VOS/DMC Upper Ocean Thermal Data Assembly Centre Meeting report. 12pp..

10. Appendices (back to TOC)

Appendix A. Summary of CSIRO Quality Control Codes

Ref. No	Category		Accept Code	Action	Quality Class		Reject Code	Action	Quality Class
1. Header Information Flags									
1.1	Position Error		PEA	Manually correct.	Class 1 from the surface.		PER	Reject data from working archive.	Class 3 from the surface.
1.2	Time Error		TEA	Manually correct.	Class 1 from the surface.		TER	Reject data from working archive.	Class 3 from the surface.
1.3	Other / Probe Error		OPA	Manually correct.	Class 1 from the surface.		OPR	Reject data from working archive.	Class 3 from the surface.
1.4	Repeat Drop		REA	Flag as a repeat drop.	Class 1 from the surface.		RER	Bad repeat drop to be rejected from working archive.	Class 3 from the surface.
1.5	Duplicate Drop		DUA	Flag as a duplicate drop.	Class 1 from the surface		DUR	Low resolution duplicate to be rejected from working archive.	Class 3 from the surface.
1.6	ID change		IDA	The unique ID has been changed	Class 0 (no change)				

Ref. No	Category		Accept Code	Action	Quality Class		Reject Code	Action	Quality Class
2. Recorder Flags									
2.1	Surface Spikes (Start-up Transient)		CSA	Remove all surface data to 3.9m depth.	Class 1 from the surface.		CSR	Reject data from working archive.	Class 3 from the surface.
2.2	Test Probe		N/A	—	—		TPR	Reject test data on all occasions from working archive.	Class 4 from the surface.
2.3	Bathy System Software Fault (Modulo 10 Spikes)		MOA	Replace Spikes with linearly interpolated values.	Class 1 from the surface.		MOR	Reject data from working archive.	Class 3 from the surface.
2.4	PROTECNO Systems Leakage (PET Fault)		PFA	Downgrade data from depth of anomaly.	Class 2 from depth of anomaly.		PFR	Delete data from depth of anomaly from working archive.	Class 3 from depth of PET fault.
2.5	Bathy Systems Leakage (Cusping)		CUA	Downgrade data from depth of anomaly.	Class 2 from depth of cusping.		CUR	Reject data from depth of anomaly from working archive.	Class 3 from depth of cusping.
2.6	Bathy System Bowing (Bowed Mixed Layer)		BOA	Downgrade data from the surface.	Class 2 from the surface.		BOR	Reject data from working archive.	Class 3 from the surface.
2.7	SippicanMK9 Processor Malfunction (Sticking Bit Problem)		SBA	Apply a 19 point filter with coefficients of 0.0562 and downgrade.	Class 2 from the surface.		SBR	Reject data from working archive.	Class 3 from the surface.
2.8	SippicanMK9 Timing Delay Problem (Driver Error)		DRA	Downgrade data from the surface.	Class 2 from the surface.		DRR	Reject data from working archive.	Class 3 from the surface.

Appendix A. Summary of CSIRO Quality Control Codes (cont.)

Ref No.	Category		Accept Code	Action	Quality Class		Reject Code	Action	Quality Class
3. General Profile Flags									
3.1	Hit Bottom		HBA	Flag data from depth of anomaly from working archive.	Class 2 from depth of possible isothermal boundary layer.		HBR	Delete data from depth of anomaly from working archive.	Class 3 from depth of hit bottom event.
3.2	Wire Break		N/A	—	—		WBR	Delete data from depth of anomaly from working archive.	Class 4 from depth of wire break.
3.3	Spike		SPA	Remove erroneous data and linearly interpolate. Downgraded from depth of anomaly.	Class 2 from depth of spike.		SPR	Reject data from depth of anomaly from working archive.	Class 3 from depth of spike.
3.4	High Frequency Interference		HFA	Filter noisy data. Downgrade from depth of anomaly.	Class 2 from depth of high frequency interference		HFR	Reject data from start depth of anomaly and reject from working archive.	Class 3 from start depth of interference
3.5	Insulation Penetration		IPA	Replace spike with linearly interpolated data. Downgrade from depth of anomaly.	Class 2 from depth of spike.		IPR	Reject data from depth of anomaly from working archive.	Class 3 from depth of spike.
3.6	Constant Temperature Profile		CTA	Keep profile to 10 metres depth and flag CTR below 10 metres depth.	Class 1 to 10 metres, Class 3 below.		CTR	Reject data from the depth of the anomaly from working archive.	Class 3 from the depth of the anomaly.
3.7	No Trace		—	—	—		NTR	Delete data from the surface from working archive.	Class 4 from the surface.
3.8	No Good Profile		—	—	—		NGR	Delete data from depth of anomaly from working archive.	Class 4 from depth of anomaly.

Appendix A. Summary of CSIRO Quality Control Codes (cont.)

Ref No.	Category		Accept Code	Action	Quality Class		Reject Code	Action	Quality Class
4. Inversion / Wire Stretch Flags									
4.1	Inversion (Confirmed)		IVA	Verify inversion in neighbours and repeat drops.	Class 1 from the surface.		—	—	—
4.2	Inversion in mixed layer (Nub Confirmed)		NUA	Verify nub in neighbours and repeat drops.	Class 1 from the surface.		—	—	—
4.3	Inversion (Probable)		PIA	Check for similar features in neighbouring drops.	Class 2 from depth of probable inversion.		—	—	—
4.4	Wire Stretch (Possible)		WSA	Check if similar features are observed in neighbouring drops. Downgrade data.	Class 2 from depth of possible wire stretch.		—	—	—
4.5	Wire Stretch		—	—	—		WSR	Reject data from depth of anomaly from working archive.	Class 3 below depth of wire stretch.

Ref No.	Category		Accept Code	Action	Quality Class		Reject Code	Action	Quality Class

5. Structure / Signal Leakage Flags									
5.1a	Fine Structure Step-like (Confirmed)		STA	Verify step-like fine structure in neighbours usually repeat drops.	Class 1 from the surface.		—	—	—
5.1b	Fine Structure Step-like (Probable)		PSA	Check for step-like fine structure in neighbours and downgrade.	Class 2 from the surface.		—	—	—
5.2	Surface anomaly (Fine Structure Special Case)		SAA	Check for evidence of surface anomalies in the region and downgrade.	Class 2 from the surface.				
5.3	Fine Structure (Probable)		FSA	Check for fine structure in neighbours and downgrade.	Class 2 from the surface.		—	—	—
5.4	Leakage (Possible)		LEA	Check if similar anomalies are observed in neighbours and downgrade.	Class 2 from depth of possible leakage.		—	—	—
5.5	Leakage		—	—	—		LER	Reject data from depth of anomaly from working archive.	Class 3 below depth of leakage.

Appendix A. Summary of CSIRO Quality Control Codes (cont.)

Ref No.	Category		Accept Code	Action	Quality Class		Reject Code	Action	Quality Class
6. Eddy-Front / Temperature Offset Flags									
6.1	Eddy / Front		EFA	Verify eddy / front in repeat or neighbouring drops.	Class 1 from the surface.		—	—	—
6.2	Temperature Difference at Depth		TDA	Check for divergence of temperature at depth in comparison with neighbouring drops.	Class 2 from the depth of the temperature divergence.		—	—	—
6.3	Temperature Offset		—	—			TOR	Check neighbouring profiles for temperature differences. Reject data from working archive.	Class 3 from the depth of the temperature offset.
7. Quality Control Check Flag									
7.1	Quality Control Check		QCA	Indicates that Quality Control has been performed on a particular drop.	Class 1 from the surface.		—	—	—

Appendix B: MEDS-ASCII GTSPP file format (file newGTSPP2) – NOTE – modified from the original to accommodate MQNC format changes.

```
      c  For character-by-character input of GTSPP files in MEDS-ASCII format.

      integer   NOS_MAX,NPROF_MAX,NDP_MAX,NSEG_MAX,Nodepthmax

      parameter (NOS_MAX=100,NPROF_MAX=30,NDP_MAX=1500,NSEG_MAX=100)

      character*1 Stream_source,Uflag,Q_Pos,Q_Date_Time,Q_Record,
     &            Data_Avail,Dup_flag(NPROF_MAX),Digit_Code(NPROF_MAX),
     &            Standard(NPROF_MAX),Q_Parm(NPROF_MAX),
     &              SRFC_Q_Parm(NPROF_MAX),
     &            D_P_Code(NSEG_MAX),Depres_Q(NSEG_MAX,NDP_MAX),
     &            Prof_Q_Parm(NSEG_MAX,NDP_MAX)

      character*2  Obs_Month,Obs_day,Data_Type,A_No_Prof,A_Nparms,
     &            A_No_Seg(NPROF_MAX),Ident_Code(NOS_MAX),Act_Code(NOS_MAX),
     &            Profile_Seg(NSEG_MAX),A_Nsurfc

      character*3  A_Num_Hists

      character*4  Obs_Year,Obs_Time,Source_ID,Stream_Ident,QC_Version,
     &            Pcode(NPROF_MAX),SRFC_Code(NPROF_MAX),
     &            PRC_Code(NOS_MAX),Version(NOS_MAX),Act_Parm(NOS_MAX),
     &            Profile_Type(NSEG_MAX),A_No_depths(NSEG_MAX)

      character*5  Deep_Depth(NPROF_MAX)
      real  Deep_Depthr(NPROF_MAX)

      character*6  Bul_Header,A_Depth_Press(NSEG_MAX,NDP_MAX)

      character*8  MKy,One_deg_sq,MEDS_Sta,A_Latitude,Up_Date,
     &            PRC_Date(NOS_MAX),Aux_ID(NOS_MAX)

      character*9  A_Longitude,A_Prof_Parm(NSEG_MAX,NDP_MAX)

      character*10  Cruise_ID,Parm(NPROF_MAX),SRFC_Parm(NPROF_MAX),
     &            Previous_Val(NOS_MAX)

      character*12  Iumsgno,Bul_time

      character*4  oldProf_Type(NPROF_MAX)
      character*16 Prof_Type(NPROF_MAX)

      integer  nos_seg,No_Prof,Nparms,Nsurfc,Num_Hists
      integer  No_Depths(NSEG_MAX),No_Seg(NPROF_MAX)
      integer Flag_severity(NOS_MAX)

      real  Latitude,Longitude,Depth_Press(NSEG_MAX,NDP_MAX)
      real  Prof_Parm(NSEG_MAX,NDP_MAX),Aux_IDr(NOS_MAX)
c     real  Profparm(Nodepthmax,NPROF_MAX)
c     real  Depthpress(Nodepthmax,NPROF_MAX)
```

```
      common / ocean_nos / Nos_Seg

      common / oceanstn /
     &    Latitude, Longitude,
     &    No_Prof, NParms, Nsurfc, Num_Hists, No_Seg,
     &    Parm, Aux_ID, Previous_Val,
     &    One_Deg_Sq, PRC_Date, MEDS_Sta, Deep_Depth,
     &    Cruise_ID, Obs_Year, Obs_Month, Obs_Day, Obs_Time,Data_Type,
     &    Q_Pos, Q_Date_Time, Q_Record, Up_date, Bul_Time, Bul_Header,
     &    Source_ID, Stream_Ident, QC_Version, Data_Avail,
     &    Prof_Type, Dup_Flag, Digit_Code, oldProf_Type,
     &    Standard, Pcode, Q_Parm,
     &    SRFC_Code, SRFC_Parm,
     &    SRFC_Q_Parm,Ident_Code,
     &    PRC_Code, Version, Act_Code,
     &    Act_Parm, Flag_severity, Aux_IDr,Deep_Depthr

      common / oceanprf /
     &    Depth_Press, Prof_parm, No_Depths, Profile_Type,
     &    Profile_Seg, D_P_Code, Prof_Q_Parm, Depres_Q

      common /extras / Mky,Iumsgno,Stream_Source,Uflag
      common /extras2 / A_Latitude,A_Longitude,A_Num_Hists,A_No_Prof,
     &        A_Nparms,A_Nsurfc,A_No_Seg,A_No_Depths,A_Depth_Press,
     &    A_Prof_Parm

c                       MEDS Character Format                       2/4/93
c                        Master Record
c     Field                 Field                    Starting
c     Name                  Size                     Position
c     Mky                  char*08                     01
c     One_deg_sq            char*08                     09
c     Cruise_ID             char*10                     17
c     Obs_Year              char*04                     27
c     Obs_Month             char*02                     31
c     Obs_Day               char*02                     33
c     Obs_Time              char*04                     35
c     Data_Type             char*02                     39
c     Iumsgno               char*12                     41
c     Stream_Source         char*01                     53
c     Uflag                 char*01                     54
c     MEDS_Sta              char*08                     55
c     Latitude              char*08                     63
c     Longitude             char*09                     71
c     Q_Pos                 char*01                     80
c     Q_Date_Time           char*01                     81
c     Q_Record              char*01                     82
c     Up_Date               char*08                     83
c     Bul_Time              char*12                     91
c     Bul_Header            char*06                    103
c     Source_ID             char*04                    109
c     Stream_Ident          char*04                    113
c     QC_Version            char*04                    117
c     Data_Avail            char*01                    121
c     No_Prof               char*02                    122
c     Nparms                char*02                    124
c     Nsurfc                char*02                    126
c     Num_Hists             char*03                    128
```

```
c         Profile Information; repeats No_prof (1-30) times

c     No_Seg                 char*02                     01
c     Prof_Type              char*04                     03
c     Dup_flag               char*01                     07
c     Digit_Code             char*01                     08
c     Standard               char*01                     09
c     Deep_Depth             char*05                     10

c         Surface Parameter Group; repeats Nparms (0-30) times

c     Pcode                  char*04                     01
c     Parm                   char*10                     05
c     Q_Parm                 char*01                     15

c         Surface Codes Group; repeats Nsurfc (0-30) times

c     SRFC_Code              char*04                     01
c     SRFC_Parm              char*10                     05
c     SRFC_Q_Parm            char*01                     15

c         History Group; repeats Num_Hists (0-100) times

c     Ident_Code             char*02                     01
c     PRC_Code               char*04                     03
c     Version                char*04                     07
c     PRC_Date               char*08                     11
c     Act_Code               char*02                     19
c     Act_Parm               char*04                     21
c     Aux_ID                 char*08                     25
c     Previous_Val           char*10                     33

c                        Profile Record (1,100 times)

c     Mky                   char*08                     01
c     One_Deg_Sq             char*08                     09
c     Cruise_ID              char*10                     17
c     Obs_Year               char*04                     27
c     Obs_Month              char*02                     31
c     Obs_Day                char*02                     33
c     Obs_time               char*04                     35
c     Data_Type              char*02                     39
c     Iumsgno                char*12                     41
c     Profile_Type           char*04                     53
c     Profile_Seg            char*02                     57
c     No_Depths              char*04                     59
c     D_P_Code               char*01                     63

c                Parameter Group; occurs No_Depth (1-1500) times

c     Depth_Press            char*06                     01
c     Depres_Q               char*01                     07
c     Prof_Parm              char*09                     08
c     Prof_Q_Parm            char*01                     17
```

Appendix C. Structure of the “keys” file:

```
netcdf mastereastMQNC_keys {
dimensions:
      N_Casts = UNLIMITED ; // (126959 currently)
      String_1 = 1 ;
      Single = 1 ;
      String_2 = 2 ;
      String_4 = 4 ;
      String_5 = 5 ;
      String_8 = 8 ;
      String_10 = 10 ;
variables:
      float obslat(N_Casts) ;
      float obslng(N_Casts) ;
      float c360long(N_Casts) ;
      int autoqc(N_Casts) ;
            autoqc:conventions = "0=OK,1=fail aut1, 2=fail aut2, 3=fail 1&2,
9=unk" ;
            autoqc:_FillValue = 9 ;
      char stn_num(N_Casts, String_10) ;
      char callsign(N_Casts, String_10) ;
      char obs_y(N_Casts, String_4) ;
      char obs_t(N_Casts, String_4) ;
      char obs_m(N_Casts, String_2) ;
      char obs_d(N_Casts, String_2) ;
      char data_t(N_Casts, String_2) ;
      char d_flag(N_Casts, String_1) ;
            d_flag:conventions = "D=yes, N=no" ;
            d_flag:_FillValue = "N" ;
      char data_source(N_Casts, String_10) ;
      int priority(N_Casts) ;
}
```

Appendix D. Structure of the "profile" file:

```
netcdf 10ed {
dimensions:
      N_Prof = UNLIMITED ; // (1 currently)
      Nparms = 30 ;
      Nsurfc = 30 ;
      Num_Hists = 100 ;
      time = 1 ;
      latitude = 1 ;
      longitude = 1 ;
      depth = 1600 ;
      String_1 = 1 ;
      Single = 1 ;
      String_2 = 2 ;
      String_4 = 4 ;
      String_5 = 5 ;
      String_8 = 8 ;
      String_10 = 10 ;
      String_12 = 12 ;
      String_6 = 6 ;
      String_16 = 16 ;
variables:
      int woce_date(Single) ;
            woce_date:long_name = "date" ;
            woce_date:units = "yyyymmdd UTC" ;
            woce_date:data_min = 19980504 ;
            woce_date:data_max = 19980504 ;
      int woce_time(time) ;
            woce_time:long_name = "time of day" ;
            woce_time:units = "hhmmss" ;
            woce_time:data_min = 131200 ;
            woce_time:data_max = 131200 ;
      double time(time) ;
            time:long_name = "time" ;
            time:units = "days since 1900-01-01 00:00:00" ;
            time:data_min = 35917.5499999998 ;
            time:data_max = 35917.5499999998 ;
      float latitude(latitude) ;
            latitude:long_name = "latitude" ;
            latitude:units = "degrees_N" ;
            latitude:valid_min = -90.f ;
            latitude:valid_max = 90.f ;
            latitude:C_format = "%8.4f" ;
            latitude:FORTRAN_format = "F8.4" ;
            latitude:data_min = -16.2667f ;
            latitude:data_max = -16.2667f ;
      float longitude(longitude) ;
            longitude:long_name = "longitude" ;
            longitude:units = "degrees_E" ;
            longitude:scale_factor = -1.f ;
            longitude:valid_min = -180.f ;
            longitude:valid_max = 180.f ;
            longitude:C_format = "%9.4f" ;
            longitude:FORTRAN_format = "F9.4" ;
            longitude:data_min = -108.6667f ;
            longitude:data_max = -108.6667f ;
      int Num_Hists(Single) ;
      int No_Prof(Single) ;
```

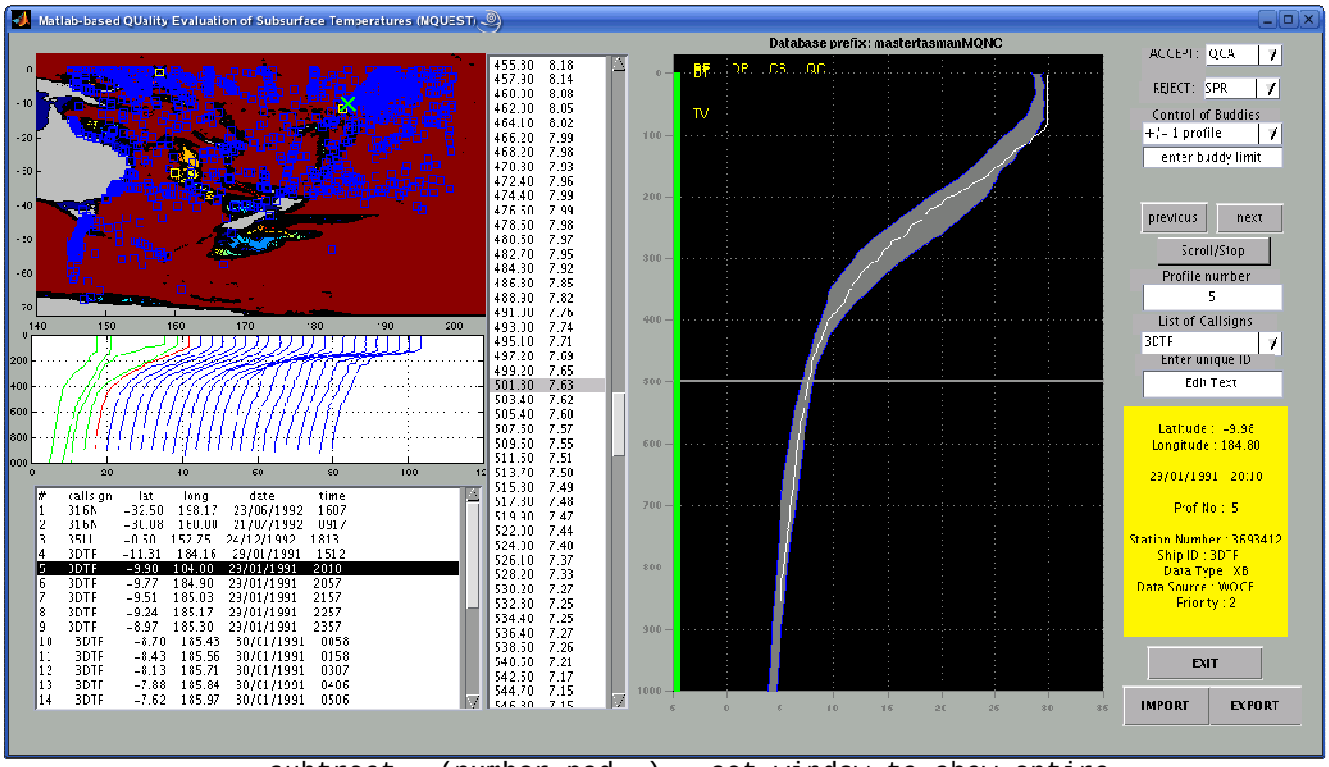
```
      int Nparms(Single) ;
      int Nsurfc(Single) ;
      char Mky(String_8) ;
      char One_Deg_Sq(String_8) ;
      char Cruise_ID(String_10) ;
      char Data_Type(String_2) ;
      char Iumsgno(String_12) ;
      char Stream_Source(String_1) ;
      char Uflag(String_1) ;
      char MEDS_Sta(String_8) ;
      char Q_Pos(String_1) ;
      char Q_Date_Time(String_1) ;
      char Q_Record(String_1) ;
      char Up_date(String_8) ;
      char Bul_Time(String_12) ;
      char Bul_Header(String_6) ;
      char Source_ID(String_4) ;
      char Stream_Ident(String_4) ;
      char QC_Version(String_4) ;
      char Data_Avail(String_1) ;
      char Prof_Type(N_Prof, String_16) ;
      char Dup_Flag(N_Prof, String_1) ;
      char Digit_Code(N_Prof, String_1) ;
      char Standard(N_Prof, String_1) ;
      float Deep_Depth(N_Prof) ;
      char Pcode(Nparms, String_4) ;
      char Parm(Nparms, String_10) ;
      char Q_Parm(Nparms, String_1) ;
      char SRFC_Code(Nsurfc, String_4) ;
      char SRFC_Parm(Nsurfc, String_10) ;
      char SRFC_Q_Parm(Nsurfc, String_1) ;
      char Ident_Code(Num_Hists, String_2) ;
      char PRC_Code(Num_Hists, String_4) ;
      char Version(Num_Hists, String_4) ;
      char PRC_Date(Num_Hists, String_8) ;
      char Act_Code(Num_Hists, String_2) ;
      char Act_Parm(Num_Hists, String_4) ;
      float Aux_ID(Num_Hists) ;
      int Flag_severity(Num_Hists) ;
      char Previous_Val(Num_Hists, String_10) ;
      char D_P_Code(N_Prof, String_1) ;
      int No_Depths(N_Prof) ;
      float Depthpress(N_Prof, depth) ;
            Depthpress:_FillValue = -99.99f ;
      float Profparm(N_Prof, time, depth, latitude, longitude) ;
            Profparm:_FillValue = -99.99f ;
      char DepresQ(N_Prof, depth, String_1) ;
      char ProfQP(N_Prof, time, depth, latitude, longitude, String_1) ;

// global attributes:
            :title = "Meds-ASCII UOT data" ;
}
```

Appendix E. Flags applied by Statistical Screening

QC flag	statistical test assigning the flag
GSA	{bad single point gradient flag }
GLA	{bad long-inversion gradient flag }
NAA	{not assessed - not enough data in the region...}
SOA	{surface temperature offset questionable}
M1A	{T difference MDL - sst-0.5 deg}
M2A	{Gradient MLD - gradient > 0.25}
T1A	{T100 error}
T2A	{T250 error}
DZA	{DZDT error}
DDA	{DTDZ error}
TZA	{T(Z) - temperature of a depth surface}
TVA	{Ken's integrated raw temp}
ZTA	{Z(T) - depth of a temperature surface}
RTA	{integrated raw T (heat content)}
BTA	{integrated binned T}

Appendix F. A summary of the shortcut keys useful in entering QC and moving within a data set:

key	description/action:
esc	exit program after saving any QC
f1	apply QCA flag
f2	show QC
f3	delete QC
f4	'kill' profile (replace with unedited, raw data)
f5	set buddies to +/- 1 profile
f6	set buddies to +/- 1 profile
f7	set buddy region to +/- 0.5 deg
f8	set buddy region to +/- 1.0 deg
f9	display only 10 buddies from the current selection
f10	set buddy region to +/- 0.1 deg
f11	show good buddy data only
f12	show all buddy data including rejected data
delete	zoom map focussed on the current profiles position
insert	return map to original axes
pagedown	profile display
pageup	buddy display
leftarrow	zoom profile
rightarrow	reset zoom to original axes
uparrow	go to next profile
downarrow	goto previous profile
home	goto drop 1
numpad1	set buddies for only one year - display buddy profiles only if they are from the same year as the current profile
numpad2	set buddies for all years - display all buddies, regardless of year
numpad3	set buddies to +/- 0.25 deg
numpad5	set buddies to +/- 1.5 deg
numpad6	set buddies to +/- 4 deg
numpad9	set buddies to +/- 3 deg
multiply	(number pad *) - set temp scale to show surface values that are offscale



subtract (number pad -) - set window to show entire trace -used to see bottom of deep traces

These require a leading ‘q’ key:

key	**flag**	**description (see Cookbook for detailed descriptions):**
1	SPA	%spikes - SPA - chop only one point!
2	TOR	%temperature offset - reject from surface SEE ALSO STANDALONE "O" KEY
3	TPR	%test probe - reject from the surface
5	HFA	%high frequency filter THE ENTIRE TRACE SEE ALSO STANDALONE "H" KEY
w	WSA	%wire stretch
e	EFA	%eddy-front region
r	REA	%repeat profile (within 15 minutes)
t	TOR	%TOR from cursor SEE ALSO STANDALONE "O" KEY
y	BBA	%bad bathy (replace with missing value)
u	URA	%under resolved profile
i	IVA	%inversion - confirmed
o	TOA	%temperature offset - accept!!! %reject from cursor
p	PIA	%probable inversion
leftbracket	CTA	%constant temperature - accept to 10m and reject below
rightbracket	STA	%steps - confirmed
backslash	NON	%not used
a	NON	%not used
s	SAA	%surface anomaly
d	DUR	%duplicate profile - reject
f	FSA	%fine structure
g	NON	%not used
h	HFA	%high frequency noise - filter from the current cursor point to the bottom of the trace SEE ALSO STANDALONE "H" KEY
j	PSA	%probable steps
k	LER	%leakage - reject from cursor position
l	LEA	%leakage - accept and flag from surface
z	HBR	%hit bottom SEE ALSO STANDALONE "Z" KEY
x	NGR	%no good - reject from cursor
c	CSA	%chop surface spikes
v	IPR	%insulation penetration - reject from cursor SEE ALSO STANDALONE "V" KEY
b	WBR	%wire break - reject from cursor SEE ALSO STANDALONE "B" KEY
n	NUA	%nub
comma	IPA	%insulation penetration - interpolate over spike (launches GUI so you can select the end point of the interpolation. The start point is the current cursor position).
period	SPA	%spikes - interpolate over spike(s) (launches the GUI described above for IPA).

The following keys are stand-alone - they are used WITHOUT the leading 'q' key:

key description (see Cookbook for detailed descriptions):

```
case 'f1'        %add QCA flag
case '1'         %force update of QC flags on the data
case '3'         %HBA - Hit bottom - automatically puts on HBA at current
                 cursor position - make sure you pick this from the
                 depth/temperature list.
case '4'         %display buddies +/1 2.0 degrees lat and long
case '5'         %get rid of duplicate history records...
case '6'         %get cursor position from map and retrieve profiles from
                 coordinates
case '7'         %set buddies for only one year - display buddy profiles
                 only if they are from the same year as the current
                 profile
case '8'         % set buddies for all years - display all buddies,
                 regardless of year - used with '7' and '9' to control
                 buddy selection - all year, single month, single year,
                 single month AND single year…
case '9'         %set buddies for single month - display buddy profiles
                 only if they are from the same month as the current
                 profile
case '0'         %redraw map window...
case 'equal'     %reapply the last REJECT menu item selected - saves using
                 cursor
case 'b'         %add WBR automatically
case 'c'         %add CSA and QCA, and display buddy profiles
case 'd'         %DUR profile, save, and send to the next profile
case 'f'         %add FSA, CSA and QCA flags and display buddy profiles
case 'h'         %HFA the entire profile from the surface
case 'j'         %add PSA, CSA and QCA flags and display buddy profiles
case 'k'         %add LER at surface
case 'o'         %TOR entire profile
case 'p'         %change position (lat or long) - new position is entered
                 via GUI
case 't'         %TEA - change date and/or time - new date and time are
                 entered into a GUI
case 'v'         %add IPV from surface
case 'x'         %NGR entire profile
case 'z'         %add HBR and NGR automatically at the selected depth -
                 removes any HBR or NGR already present
case 'backslash'%plot raw data underneath edited data in buddy window.
case 'delete'    %zoom map focussed on the current profile's position
case 'insert'    %return map to original axes
case 'numpad1'   %set buddies for only one year - display buddy profiles
                 only if they are from the same year as the current
                 profile - superseded by '7', '8' and '9' keys
case 'numpad2'   %set buddies for all years - display all buddies,
                 regardless of year - superseded by '7', '8' and '9' keys
case 'numpad3'   %set buddies to +/- 0.25 deg
case 'numpad5'   %set buddies to +/- 1.5 deg
case 'numpad6'   %set buddies to +/- 4 deg
case 'numpad9'   %set buddies to +/- 3 deg
case 'multiply'  %(number pad *) - set temp scale to show surface values
                 that are offscale
case 'subtract'  %(number pad -) - set window to show entire trace -used
                 to see bottom of deep traces
```

```
case 'home'      %goto drop 1
```