3.9 Miscellaneous Programs

Some programs that can be used with QWDATA are available command line prompts, but are not available from within the QWDATA menus. The three programs discussed in this section can be used to manipulate tab-delimited batch files and to edit precision codes using a batch process.

3.9.1 Program to Concatenate Tab-Delimited Batch Files (qwcat)

A program to concatenate multiple ‘qwsample’ and qwresult’ file pairs into a single file pair can be initiated by typing the following at the command line prompt:

```
qwcat [-q] file1 [ file2 .. fileN ]
```

Where:
"-q" is an optional argument that causes suppression of some of the messages written by the program,

"file1 file2 fileN" is a mandatory list of filenames for NWIS-QW tab-delimited sample files, or a shell-interpreted wildcard string. The list might contain only one filename, which is useful for validating the contents of the batch. The program will determine the filename of each matching NWIS-QW result file by replacing the text "sample" in the sample filename with the text "result".

An example of this command is:

```
qwcat rejected.sample*
```

Where: "rejected.sample*" is a shell-interpreted wild-card string that matches all files in the current directory that begin with the text: "rejected.sample".

Output is written to the current directory in files named "qwsample.cat" and "qwresult.cat". Any existing files of these names are renamed using a current date-time suffix.

The following errors can cause the program to stop:

- No write access to the current directory.
- Inability to rename pre-existing output files: "qwsample" & "qwresult".
- Inability to open files in temporary directory (/tmp or $TMPDIR).
- Invalid optional argument flag.
- No valid qwsample filename specified.

The following errors can cause the program to stop processing the batch files:

- qwsample filename does not contain the text "sample".
- qwsample or qwresult files do not exist in current directory.
- qwsample or qwresult files are not readable by current userid.
- qwsample or qwresult files do not contain text in 1st 10 bytes.
- qwsample records with duplicative sample integers.
- qwresult records with duplicative combination of sample integer + pcode.
- qwresult record with sample integer not present in qwsample file.

The following can cause unexpected and undesirable behavior:

- Duplicative command-line specification of the same qwsample filename (eg. with wildcards) is silently reduced to one inclusion.
- Batch-file pairs may not be processed in the order entered; they are processed in ASCII sort order by qwsample filenames.
- qwsample records are not sorted within each file; they are added in order of the qwsample file.
- qwresult records are processed in the same order as the qwsample records and secondarily sorted using ASCII sort order by parameter code.
- qwsample and qwresult zero-length records are silently ignored.
- If records exist in qwsample or qwresult files without tab delimiters they are counted, but ignored. The count of these records is printed.
- qwsample record with sample integer not matching qwresult records is included. Count of such records printed.
- qwsample and qwresult records with too few tabs are padded (to the right) with empty fields delimited with tabs. Count of such records printed.
- Program does not track orphaned sample/result messages and they are ignored.

3.9.2 Program to Subset Tab-Delimited Batch Files (qwsplit)

A program to subset one or more tab-delimited batch-file pairs using project code and (or) station-identification number can be initiated using the following:

```
qwsplit [-p fileP] [-s fileS] file1 [ file2 .. fileN ]
```

Where: -p fileP specifies an optional filename that provides data on how to split the data by project,

-s fileS specifies an optional filename that provide data on how to split the data by station, and

"file1 file2 .. fileN" is a mandatory list of one or more tab-delimited batch filenames of sample data. The matching result filenames are assumed to be the same, except the filename text: "sample" is altered to "result".

An example of this command is:

```
qwsplit rejected.sample
```

If neither a "-p file" nor an "-s file" is supplied, then the program will use all the project codes found in the qwsample files to subset the data. One pair of subset-data files will be created for each unique project code found in the data. Output files will be named according to the pattern: "qwsample.PROJECT" and "qwresult.PROJECT", where "PROJECT" is a project code found in the data. Data found in the input file(s) without a project code will be automatically associated with the project code "misc"; and thus will be written to files named "qwsample.misc" and "qwresult.misc".

More control over how the data are split can be obtained by using one or both of the optional split-specification files. Project-split specifications may be supplied in a file named after "-p" on the command line, as in:

```
qwsplit -p qwsplit.projects rejected.sample
```

Likewise, station-split specifications may be supplied in a file named after "-s" on the command line. Both the "-p" and "-s" command-line options may be used, however the project-split specifications have precedence over the station-split specifications when both apply to the same sample.

The split-specification files consist of records for which projects or stations belong in separate output files, and how those files should be named. In a project-split file, a project-identification number is specified, followed by one or more blank (or whitespace) characters, followed by a filename component. Additional projects may be specified on subsequent lines in the file. The same project should not be specified more than once in the file, however multiple projects may be assigned to the same filename component. Filename components should not contain blanks (or shell metacharacters, such as "*" or "?").

The following is an example project-split file:

```
404000300   bds
00300       bds
404039000   jim
39000       jim
```

This file would split any "qwsample" records with project code of "404000300" or project code "00300" into a file named "qwsample.bds". New data will be concatenated onto the end of the file, if it already exists. Similarly, "qwresult" records associated with either of these project codes will be concatenated to the end of a file named “qwresult.bds". All sample and result records with a project code not identified in the project-split file are appended to the files named "qwsample.misc" and "qwresult.misc".

The station-split file is prepared in the same manner, except that station-identification numbers are presented where project codes appear in the above example.

The following errors can cause the program to stop:

- No write access to the current directory.
- No filename specified afterwards when "-p" or "-s" on the command line.
- An invalid (unknown) command-line option flag.
- No qwsample filename is given on the command line.
- Station-split or project-split file specified cannot be opened.
- Inability to concatenate data to a temporary qwsample or qwresult file.

The following errors can cause the program to stop processing the batch files:

- qwsample filename does not contain the text "sample".
- qwsample or qwresult files do not exist in current directory.
- qwsample or qwresult files are not readable by current userid.
- qwsample or qwresult files do not contain text in first10 bytes.
- qwsample records with duplicative sample integer exist.
- qwresult records with duplicative combination of sample integer + pcode.
- qwresult record with sample integer not present in qwsample file.

The following can cause unexpected and undesirable behavior:

- Duplicative command-line specification of the same qwsample filename (eg. with wildcards) is silently reduced to one inclusion.
- Warning printed if station-split or project-split file contains records where split data are duplicative or incomplete.
- Zero-length qwsample, qwresult, or split-file records are silently ignored.
- Warning printed of the count of ignored untabbed sample & result records.
- All blanks are removed from project and station-id found in sample records prior to applying subsetting rules.
- Because the program qwcat is used by qwsplit, errors that may cause qwcat to operate incorrectly, also apply to qwsplit.

3.9.3 Program to Edit Existing Precision Codes in Field Values (qwfix)

A program to edit existing precision codes in field values can be initiated using the following command:

```
qwfix
```

The user is prompted for three items:

1. file name of input file (1 and *-cards format)
2. list of parameters to check and desired precision codes
3. name of output file

The file of 1&*-cards should contain the data that you want to edit. The file can be created using option 8 in the Batch Processing menu within QWDATA (*see Section 3.8.8*). Extra data, including parameters and methods, can be included in this file and not be affected, and the extra data will not be copied to the output file.

The format for the list of the rounding editions to be made is: parameter code in columns 1-5, method code in column 6 (field parameters will not have method codes), and rounding specifications in columns 8-18, with a decimal point in column 15.

Here is an example parameter file:

```
00095  3333321.000
00400  0000032.100
00950B 2222222.100
00950D 2222222.220
```

Note on rounding specifications: As used above, the numbers indicate the number of significant figures for the specified range. 00095 3333321.000 will result in the following:

```
        1234567.000  will display as 1230000
         123456.000  will display as  123000
          12345.000  will display as   12300
           1234.000  will display as    1230
            123.000  will display as     123
             23.000  will display as      23
              2.000  will display as       2
               .123  will display as       0
               .12   will display as       0
               .1     will display as       0
```

An example of the program dialog follows:

```
/srv2/owq/home$ qwfix
PLEASE ENTER NAME OF INPUT QW_CARD FILE: qwcards
PLEASE ENTER NAME OF INPUT PRECISION-DATA FILE: list.field
PLEASE ENTER NAME OF FILE TO HOLD THE OUTPUT: out
```

Only changes to the input data will be written to the output file.

NOTES*:* Use the qwcardsin program to enter the corrected data into NWIS. To output tables of the values with the edited precision codes, you will need to use the "user-defined" rounding option. This program will not change the precision code for values that are less than 0.000009.

3.9.4 Program to Convert 1 and * Card Batch Files to Tab-Delimited Batch Files (star2pair)

This program is not part of a usual NWIS installation, and this documentation is included for users' convenience. The program and up-to-date documentation are available from a Web page maintained by Phoenix:

http://phoenix.cr.usgs.gov/reload/reload.html

This program will convert a 1-and-*-format batch file to tab-delimited batch files when the following command is entered:

```
star2pair < input file
```

where ***input file*** is the name of the 1-and-* card batch file to convert.

The program creates tab-delimited files named 'qwsample' and 'qwresult'. If these files exist in the working directory, they are overwritten. This program might be useful to add information to a sample that cannot be added using the 1-and* card format, such as value qualifiers.

Some things you should know about this program:

- The program does NOT VERIFY the input data.
- The program does syntactically verify the *-record and the X-record.
- When a syntax failure occurs, the program prints an error message, the line number, and contents of the offensive input record; then stops.
- Date conversion, case translation, and blank removal are provided.
- Syntactical conversion of result deletions, and null-value qualifiers is provided.
- Special handling of the non-standard syntax used by the NWQL for end-of-file and sample laboratory-identification number (such as P99998).

3.9.5 Program to Convert RDB Output Files to Excel Files (rdb2excel)

This program is not part of a usual NWIS installation and this documentation is included for users' convenience. The program and up-to-date documentation are available from the Web page:

http://okwaterweb.cr.usgs.gov/nawqa/excel/install.html

This program would be helpful for preparation of data retrievals for internal or external users. The program will convert an RDB file to an Excel-workbook file when the following command is entered:

```
rdb2excel rdb_file_name excel_file_name
```

where ***rdb_file_name*** is the input file and ***excel_file_name*** is the output file.

Functions of this program are:

- Include RDB comments (in blue) above the column headings.
- Include column names (in red) above the first line of data.
- Convert RDB dates to Excel dates, inferred from: column-definition specification of [Dd] and actual field length of 10 or less.
- Convert RDB date-time to Excel date-time, inferred from: column-definition specification of [Dd] and actual field length greater than 10.
- Convert RDB text fields to Excel text, inferred from: column-definition of [Ss] or "" (null).
- Convert RDB numeric fields to Excel numeric data, inferred from: column-definition spec. of [Nn].
- Convert RDB missing values to Excel missing values, note that one or more blanks is -not- missing.
- Set column widths using the wider of: column-definition specification or the column-name length.
- Enforce the Excel limits on the number of rows/sheet, columns/sheet, and characters/cell.
- Name the worksheet within the workbook with the basename of the input RDB file.
- Create an output binary file when run under UNIX, that is compatible with Excel versions: 5, 95, 97, 2000, and 2002.

Functions not included in this program (intentionally):

- Convert the RDB data type of Month ("M") to anything other than text.
- Ensure that a consistent number of fields is present on each line.
- Include the column-definition row in the output.
- Read/write STDIN/STDOUT (though STDERR is used).
- Append to an existing Excel file.
- Write Excel formula.
- Convert an Excel file to an RDB file (some day, perhaps).
- Automatically append an ".xls" suffix to the output filename.

The following conditions will cause errors when running this program:

- Failure to specify exactly two command-line arguments
- Too many rows or columns for Excel
- Any RDB column definition other than: [dmns].
- Output file size exceeding 7-MB.

3.9.6 Program to Adjust Stored Sample Times (qwtimeshift)

The program *qwtimeshift* can be used to modify the sample start and end times stored in the database. Beginning in NWIS 4.3, two new fields are available to compute the stored UTC time in the database. See sections *2.4.6, 2.4.7, and 2.1.9* for more detailed information about these time-related fields. This program is provided so that users can fix those samples that may have been changed prior to the release of NWIS 4.3. This program to adjust stored sample times can be initiated using the following command at the UNIX prompt:

```
qwtimeshift
```

The first query will be for the input file:

Enter filename**? (<CR> to quit): ________________________________**

The input file format has the following conditions:

- Lines starting with '#' are ignored and are treated as comments.
- Each line in the input file contains two fields separated by white-space. White-space characters can be either spaces or tabs.
- The first field is a sequence of 10 digits--the first 8 digits are the record number of the sample to be updated and the last 2 digits represent the database number used to locate the sample.
- The second field is one or more digits preceded by an optional sign-- "+" or "-". This field is the offset in hours by which the sample time in the database is adjusted.

Example of an input file:

```
0020000201      1      #  Begin date
0020000301      -1     #  Begin and end date
```

Possible error messages:

- **Invalid format:**
  A line in the input file was not in the correct format.
- **Problem parsing the filename, please reenter**.
  This probably indicates that some control characters or a space are in the filename that was entered.
- **QW_SAMPLE_## does not exist in database <db_no>**
  This probably indicates that the wrong database number was specified in the input file.

- **Sample <record_no> does not exist in database <db_no>**
  The sample listed in the message could not be found in the specified database. Update the input file with the correct record_no and/or database number.   All other records in the input file will still be processed.

  - **SQL Error:**
    This indicates there was a problem executing the SQL statement.

  - **Unable to compile <regex name> regex**:
    This is an internal program error. The regular expression named in the error message could not be compiled.

  - **Unable to determine if QW_SAMPLE_## exists in database <db_no>**
    This probably indicates that the wrong database number was specified in the input file. It may also indicate an RDBMS problem.

  - **Unable to open <file>: <msg>**
    The specified file could not be opened for the reason stated in the message.

  - **You do not have write access to the QW_SAMPLE_## table in ...**
    You do not have sufficient privileges to update the specified database.

Additional information about the application of UTC time and QW-related fields and issues can be found at: *http://phoenix.cr.usgs.gov/www/utc.html.*