STANDARD OPERATING PROCEDURE (SOP) – Water Quality Planning Division

Title: Validation of Continuous Water Quality Monitoring Data Collected by a Greenspan Aqualab Auto-Analyzer

Team Leader: _____ Date: _____

Quality Control Review_____ Date:_____

Section Manager:_____ Date:_____

Effective Date: 12/19/06

1.0 PURPOSE 1

2.0 SCOPE AND APPLICABILITY 1

3.0 METHOD OR PROCEDURAL SUMMARY 2

4.0 LIMITATIONS 2

5.0 SAFETY 2

6.0 EQUIPMENT, SUPPLIES, REAGANTS, AND STANDARDS. 2

7.0 PROCEDURES 3

8.0 CALCULATIONS. 8

9.0 QUALITY CONTROL. 8

10.0 REFERENCES 8

11.0 DEFINITIONS 9

12.0 POLLUTION PREVENTION AND WASTE MANAGEMENT. 9

13.0 SHORTHAND PROCEDURE. 9

1.0 PURPOSE

This SOP details the procedure for validating real-time, continuous ambient surface water quality data collected by Greenspan Aqualab Auto-Analyzers, based on quality control (QC) limits specified in the Continuous Water Quality Monitoring Network (CWQMN) Quality Assurance Project Plan (QAPP) or otherwise directed to be used by the Quality Assurance (QA) officer.

2.0 SCOPE AND APPLICABILITY

This SOP is used by Data Management & Quality Assurance (DM&QA) staff to provide guidance on validation procedures. Other Texas Commission on Environmental Quality (TCEQ) program areas and TCEQ contractors also use this SOP if the need arises for data validation to be performed by staff other than those within DM&QA. Water quality parameters for validation

may include, but are not limited to, the following: temperature, specific conductivity, pH, dissolved oxygen (DO), turbidity, nitrate (NO_3), ammonia (NH_3), and total reactive phosphorus (TRP). Some automated validation procedures are performed by the Leading Environmental Analysis and Display System (LEADS) data processing software, and the water data validator performs manual validation procedures.

3.0 METHOD OR PROCEDURAL SUMMARY

Water data validators are responsible for the collection and processing of Aqualab QC data for validation purposes. The data validator downloads data directly from Aqualab instruments and examines it for record completeness and accuracy in reporting. Data are then formatted for processing with SAS™ Software. The validator uses SAS™ codes to perform automated QC calculations. These calculations and acceptable operating limits are provided by the various SOPs for each parameter and are based on laboratory control samples (LCS) of on-board Aqualab chemical standards (for NO_3, NH_3, and TRP) and actual measurements (for conductivity, pH, and turbidity). Any data that fall outside of QAPP limits or limits established by the CWQMN QA Officer are manually flagged or qualified. LEADS automatically flags data values that fall outside of established critical limits (see Attachment A). The data validator investigates data losses, reviews operator logs for information about equipment maintenance or unusual, corresponds with site operators as needed, and documents all validation procedures in a validator's log.

4.0 LIMITATIONS

4.1 Data validation is dependent upon the ability to connect via modem to the Aqualab analyzers and download data. In the event communications problems arise, data may not be accessible for downloading.

4.2 Some Aqualab components are susceptible to fouling when sediment, nutrient, or algae levels are high in ambient surface waters. Data invalidation may occur because of this fouling.

4.3 Unforeseen events such as equipment malfunctions, floods, or vandalism could result in lost data or data in need of invalidation.

4.4 In the event that data are reloaded or reprocessed in LEADS after validation, previously flagged and/or invalidated data will default to their original status. Data must be validated again by referring to validator's notes and operator logs.

4.5 LEADS is a developing system, and the software tools used to validate data could contain defects that may or may not have been identified.

4.6 User must have access to the LEADS Manual Validation Software. This access can be obtained by contacting the TCEQ LEADS Administrator.

5.0 SAFETY

Usual office and computer safety practices apply. For additional information about the TCEQ Safety Program, see **http://home.tnrcc.state.tx.us/internal/admin/opp/docs/06-08_opp.pdf**

6.0 EQUIPMENT

Computer Hardware**:**
- Pentium IV PC, 5.2 MB RAM, 10 GB Hard Drive
- Data logger and data communication hardware
- External modem (US Robotics Sportster 28,800 or equivalent)

- Serial cable
- RJ11 standard phone cord
- Power adapter
- Analog telephone jack

Computer Software:
- HP UNIX
- HP Common Desktop Environment
- Exceed for PC
- LEADS user interface
- Manual validation data retrieval and validation software
- Greenspan software Analyser32 and Aquagraph
- Microsoft Excel or similar spreadsheet application
- Microsoft Word or similar word processing application
- SAS™ Software version 8 or greater

7.0 PROCEDURE(S)

7.1 Daily Procedures

7.1.1 Verify each morning the operation of Aqualab analyzers and the transmission of data.

7.1.1.1 Using an internet browser access the TCEQ internal server at the following address: **http://dsr/.** Select the Continuous Ambient Monitoring Station (CAMS) Data Printout option from the Data Reporting Pages Section.

7.1.1.2 From the CAMS Data Printout Page select the link for water data (http://dsr.tceq.state.tx.us/water_dataout). Once at the CAMS Water Data Printout Page select a location, start date, number of days worth of data to report, type of data (5-minute, 15- minute or 1-hour), time format, output format, and parameters of interest. *Note: Although you have the option to download 5- minute, 15-minute, or 1-hour data, the Aqualab analyzer only measures field parameters every hour and nutrients every six hours. Confirm data retrieval for all parameters measured by the Aqualab at the site of interest.

7.1.1.3 In the event that data are missing, access the Comms Report Page within the Status Pages Section. This page provides a check of communications between remote sites and the central computer site.

7.1.1.4 If necessary, access the Operator Logs Page from the Status Pages Section. Contained in the operator logs are notes about site maintenance activities and/or unusual site conditions or events.

7.1.1.5 Contact site operators and project lead as needed for possible site investigations or repairs.

7.1.1.6 Contact the LEADS Administrator as needed to resolve communications problems and/or to recover data.

7.2 Equipment Set Up Procedures

7.2.1 An external modem is necessary for communicating with the Aqualabs and downloading data from the Aqualab sites. A serial cable, a power adapter, and a RJ11 standard phone cord are necessary for proper operation/installation of the modem.

7.2.1.1 Before a modem installation, turn off your computer and attached devices.

7.2.1.2 Connect the serial cable between the modem and computer using the COM, MODEM, RS-232, or SERIAL port on the back of your computer.

7.2.1.3 Plug the power adapter into the power jack on the back of the modem and into a standard wall outlet.

7.2.1.4 Plug one end of the phone cord into the modem telephone jack and the other end into a phone wall jack.

7.2.1.5 Set the DIP switches on the back of the modem accordingly: 1 = up, 2 = up, 3 = down, 4 = up, 5 = up, 6 = up, 7 = up, 8 = down.

7.2.1.6 Turn the modem power on and from the computer **Start Menu** proceed through these steps: **Settings** ⇨ **Control Panel** ⇨ **Phone and Modem Options**. From the **Modems** tab select **Add.** and follow the steps in the Add/Remove Hardware Wizard to install the modem.

7.2.2 Greenspan Analyser32 Software is also necessary for communicating with the Aqualabs and downloading data from the Aqualab sites.

7.2.2.1 Upon opening Analyser32, make sure that the "Modem Connect" button is selected and proceed to the **Setup** tab.

7.2.2.2 From the Communication and Access Setup Page, make sure that **the following buttons are selected: "COM1, 19200 Baud, Tone Dial, and Log File Off**." Select the **New** tab to enter phone contact information for Aqualab sites.

7.2.2.3 Enter phone numbers and site names as needed for all Aqualab sites with which you will be communicating. This SOP provides contact information for current Aqualab sites:

Phone Number	**Site Name**
9,12548656928,,,*3,,*3	Gatesville
9,12547960937,,,*3,,*3	Green Creek
9,12546753571,,,*3,,*3	Clifton
9,12547852736,,,*3,,*3	Resley Creek

7.3 Weekly Data Download Procedures

7.3.1 Format data for validation.

7.3.1.1 Data files must be opened using AquaGraph Software and exported in a comma delimited (*.csv) format.

7.3.1.1.1 Open the file at WaterData\AqualabData\[sitename]AutoPoll.dat.

7.3.1.1.2 Once the data file is open in Aquagraph, follow these steps: File ⇨ Export Text File. Select the destination for the file to be saved, enter a name for the file, and make sure that the Save As Type is set to Comma Delimited Format (*.csv).

7.3.1.1.3 Open the comma delimited data file in Microsoft Excel and follow these steps for formatting: Delete the first fifteen (15) rows of header information, leaving only data in the file. Search and replace the month in the date to be a three character textual month (for example, you would replace /06/ with –jun- and /01/ with –jan-). Do this for each month in the file. Reformat the date column as a custom format of dd-mmm-yyyy hh:mm). Save this newly formatted file as a comma delimited file.

7.4 Weekly Data Processing with SAS™ Software

7.4.1 Use SAS™ Software and programs to perform data processing.

7.4.1.1 First run the program ReadAqualab.SAS to read the data file. Within the first data statement, assign a name for the output file and the location of the input file. Make sure to use quotes for the in file name.

7.4.1.2 Start up the program AqualabErrorsLCS.sas to determine which data, if any, are in need of flagging. Fill in all the needed information in the %let statements. This code will produce several output tables in an HTML format.

7.4.1.3 Any entry in the tables with the word ‘FAIL’ next to it indicates data that did not pass quality assurance calculations and are in need of flagging. The validator will flag all data from the last ‘Passing’ data point to the last ‘FAIL’ data point.

7.5 Weekly Data Flagging with Manual Validation Tools

7.5.1 Once data have been identified for flagging, use a Unix emulator (such as Exceed) to navigate through Manual Validation.

7.5.1.1 Enter user name and password as prompted, and select the graph icon located at the bottom of the page.

7.5.1.2 Select your name from the users list, enter your password again, and select the “OK” button.

7.5.1.3 Select the proper scale from the Scale Table and select “OK.” There are two water scales available (Water_scale-east and Water_scale-west), and these account for differences in conductivity between east Texas and west Texas monitoring sites. Water_scale-west is most appropriate for sites in the panhandle and far west Texas (TCEQ Regions 1, 2, 3, 6, 7, 8).

7.5.1.4 You will now be able to make selections based on the data you wish to view. You can select the month from which you wish

to view data, the start and end time of the data record, and the time code. * Note: You can only view one month of data at a time in Manual Validation.

7.5.1.5 Select your region from the "Limit to" list, make sure that the "Sort by" is set to CAMS, and select the "Show Sites" button. This will allow you to select the specific monitoring site. Once you have selected your specific site from the list, select the "Show Available" button in the middle of the page.

7.5.1.6 From the list at the bottom of the page, select the parameters you would like to see graphically and press "OK." Select up to four parameters at one time by pressing the "Ctrl" key to make multiple selections. The system defaults to graph all parameters on one page, each with its own graph. The graphs appear in the same order as the list from which you selected parameters.

7.5.1.7 Use the buttons at the bottom of the page (Top, Bottom, Min, Beginning, End, left/right arrows, up/down arrows) to navigate through the data record. To zoom in on the data, press the "Shift" key as you click your mouse on the data record. To zoom out, press the "Ctrl" key as you click your mouse on the data record. The Top and Bottom buttons allow you to view the topmost portion of the graphical output. The "Min" button shows the minimum scale values for each of the parameters graphed (notice the negative numbers on the scales). The "Beginning" button will return you to the start of the data record from anywhere within the record, and the "End" button will take you to the end of the data record you are viewing. The arrows allow you to move up, down, left (back) and right (forward) through the data record. As you move the pointer over the data, the Time, Value, and Flag values at the bottom of the page will change according to the data.

7.5.1.8 Manually select any data in need of flagging based on SAS data files. From the Edit menu select "Flags" and choose the appropriate flag for data qualification (Attachment A). From the edit menu, choose "Select All" to return to the data record and/or select more data for flagging.

7.5.1.9 Investigate irregular data patterns by referring to the operator log, contacting the site operator, and/or notifying the Surface Water Quality Monitoring (SWQM) Team Project Lead.

7.5.1.10 Once the data have been marked with the appropriate flags, proceed to enter validator notes. From the file menu, select "Validate Data." In the 'Description (New) portion, enter a detailed explanation for any changes or flags applied to the data. Initial all note entries and complete the validation procedure by selecting the "Validate" button at the bottom of the page.

7.6 Monthly Data Processing with SAS™ Software

7.6.1 Use SAS™ Software and programs to perform data comparison to find data anomalies not readily apparent in LEADS Manual Validation Software.

7.6.1.1 Run CAMS Data Printout from DSR webpage (http://dsr/cgibin/lsr/water_dataout) for the period since the last analysis.

7.6.1.2 Choose the following parameters:

- Surface Water Temperature (10010)
- Surface Specific Conductance (10095)
- Surface Dissolved Oxygen (10300)
- Surface Water pH (10400)
- Water Ammonia (10100)
- Water Nitrate (10101)
- Turbidity (10104)
- Water Total Reactive Phosphorus (10105)

7.6.1.3 Choose the 15-minute average, Local Standard Time (LST), and Comma-Delimited Listing options.

7.6.2 When report appears in browser window, go to File > Save As… > and save the file as a Text File (*.txt). at MO/Waterdata/Aqualabdata.

7.6.3 Open the SAS Program at WaterData\Continuous Water Monitoring\SASValidation\CompareDateTime.sas to evaluate the data.

7.6.3.1 The report will identify POSSIBLE data anomalies, mainly data time shifts and dissimilar data records between LEADS and the Aqualab that need to be checked.

7.6.4 Investigate any true data inaccuracies between data downloaded from the CAMS data printout and data displayed in LEADS. Notify the project lead of possible data overwriting.

7.6.5 If data inaccuracies are found, refer to Section 7.5 and qualify the data in Manual Validation accordingly.

7.7 Quarterly Data Processing with SAS™ Software

7.7.1 Use SAS™ Software and programs to perform data visualization to find anomalies including, but not limited to, data dropouts, invalid data points, missing data, etc.

7.7.1.1 Run CAMS Data Printout from DSR webpage (http://dsr/cgibin/lsr/water_dataout).

7.7.1.2 Choose the following parameters:

- Surface Water Temperature (10010)
- Surface Specific Conductance (10095)
- Surface Dissolved Oxygen (10300)
- Surface Water pH (10400)
- Water Ammonia (10100)
- Water Nitrate (10101)
- Turbidity (10104)
- Water Total Reactive Phosphorus (10105)

7.7.1.3 Choose the 15-minute average, Local Standard Time (LST), and Comma-Delimited Listing options.

7.7.2 When report appears in browser window, go to File > Save As… > and save the file as a Text File (*.txt) at MO/Waterdata/Aqualabdata.

7.7.3 Open the SAS Program at WaterData\Continuous Water Monitoring\SASValidation\AQLFocusVal.sas to evaluate the data.

7.7.4 Fill in the necessary information in the header of the program.

7.7.5 Submit the program. It is not necessary to make any modifications to the text file.

7.7.6 View graphs and change scales as necessary in header of program to enable viewing of the entire data record on each graph.

7.7.7 If data is in need of qualification, validate data as necessary. See Section 7.5 for a detailed procedure.

8.0 CALCULATIONS

All calculations performed for Aqualab data validation are automated using SAS™ codes and software. These calculations are explained in the SAS notes. A copy of the SAS routines are attached (see Attachment B). The engineering notes distributed by Greenspan Technology provides calculation specifications and acceptable operating. *CWQMN QAPP* provides validation limits in Section A.7.

9.0 QC

Each data validator is responsible for the review, validation, and verification of data from assigned ambient stations. Data outliers and unusual data will be investigated by the validator. Validators are also responsible for maintaining detailed records of all validation activities and follow-up actions related to the invalidation/flagging of ambient water quality data. Records should be sufficient to reconstruct the data validation event later if necessary.

10.0 DEFINITIONS

CAMS	Continuous Ambient Monitoring Station
CWQMN	Continuous Water Quality Monitoring Network
DM&QA	Data Management and Quality Assurance Section
DO	Dissolved Oxygen
LCS	Laboratory Control Sample
LEADS	Leading Environmental Analysis and Display System
NH_3	Ammonia
NO_3	Nitrate
QA	Quality Assurance
QAPP	Quality Assurance Project Plan
QC	Quality Control
SAS™	Trademark name, Data Analysis and Transformation Software
SOP	Standard Operating Procedure
SWQM	Surface Water Quality Monitoring
TCEQ	Texas Commission on Environmental Quality
TRP	Total Reactive Phosphorus

11.0 REFERENCES

Continuous Water Quality Monitoring Network Quality Assurance Project Plan
Greenspan Technology Engineering Notes
SAS Code with comments (Attachment B)
TCEQ Operating Policies and Procedures, Chapter 6.13
Monitoring Operations Hazardous Waste Disposal Plan

12.0 POLLUTION PREVENTION AND WASTE MANAGEMENT

Supervisors, sampling personnel, and laboratory analysts should identify and implement innovative and cost-saving waste reduction procedures as part of the method development, and review and revision of standard operating procedures. Wastes that do result from these procedures are managed and disposed in accordance with appropriate state and federal regulations.

Refer to Chapter 6.13 of the TCEQ Operating Policies and Procedures for guidelines on general recycling, waste reduction, and water and energy conservation. Review these procedures for specific employee responsibilities and mechanisms for office related waste prevention and management. Consult the *Monitoring Operations Hazardous Waste Disposal Plan* for laboratory specific waste minimization recommendations and requirements for proper handling of hazardous waste that result from laboratory procedures.

13.0 SHORTHAND PROCEDURE

13.1 Daily Procedures

- Confirm collection and transmission of data.
- Access operator logs and/or contact site operators for possible site investigations if necessary.
- Contact LEADS administrator if necessary for possible recovery of lost data.

13.2 Weekly Procedures

- Use external modem to dial into Aqualab sites and download data.
- Format data for processing.
- Process data using SAS™ Software and codes.
- Flag data as needed using Manual Validation. Investigate irregular data patterns.
- Note any changes in validator's notes within Manual Validation and in a hard copy validator's log.

13.3 Monthly Procedures

- Run CAMS Data Printout with appropriate parameters.
- Save the file as a Text File (*.txt) at MO/Waterdata/Aqualabdata.
- Open the SAS Program to evaluate the data.
- Fill in the necessary information in the header of the program.
- Submit the program.
- View several of the potential anomalies.
- Investigate any actual anomalies and notify project lead.

13.4 Quarterly Procedure

- Run CAMS Data Printout with appropriate parameters.
- Save the file as a Text File (*.txt) at MO/Waterdata/Aqualabdata.
- Open the SAS Program to evaluate the data.

- Fill in the necessary information in the header of the program.
- Submit the program.
- View graphs and change scales as necessary in header of program.
- Validate data as necessary.

COPY

ATTACHMENT A

Below are the validation flags used by MeteoStar LEADS. If there are less than nine valid 5-minute samples available within an hour, the priorities listed below are consulted to determine which flag to assign to the hourly average. Flags with higher priorities overwrite flags of lower priority.

Flag	Description	Manually Set	Automatic	Priority
AQI	Data invalid	X		12
FEW	Not enough samples to create an hourly average		X	4
LIM	Data failed one or more automatic quality checks		X	5
LST	Lost or missing data - indicates that data for this sample period is not stored in the MeteoStar database		X	1
MAL	Instrument malfunction		X	8
NEG	Data failed NEG test - indicates data values are too negative; normally low negative values are set to zero by the MeteoStar software		X	3
NOL	Instrument not on line; seen before the first successful calibration of a parameter		X	13
PMA	Instrument in preventative maintenance mode set by field operators	X		6
VAL	Valid data	X	X	0

COPY

ATTACHMENT B

ReadAqualab.sas

```
%let dataset = gates;
%let csvfile = 'c:\documents and settings\nragland\my documents\gates.csv';

*The infile was a tab separated export from the Ingres driver;
data &dataset (drop = a b c d e f g h i j k l m n o p q r s t u v w x po4lcs1)  ;
        infile &csvfile dlm=',' truncover DSD;

*input the following fields.  Because there is a delimiter, no need to tell SAS how long each field
is;
        input
                a $
        datetime datetime17.
                b
                c
                d
                e
                v
                ecchk
                f
                g
                h
                phchk
                i
                turbchk
                j
                k
                nh4chk1
                nh4chk2
                l
                m
                no3chk1
                no3chk2
                n
                o
                po4chk1
                po4chk2
                p
                po4chk0
                q
                r
                s
                t
                dochk
                nh4lcs
```

COPY

```
                w
                no3lcs
                x
                po4lcs1
                u
                ;
        nh4std = 0.5;
        no3std = 0.5;
        po4std = 0.04;
        po4lcs = po4lcs1/1000;
    format datetime datetime17.;

run;
```

AqualabErrorLCS.sas

```
    %let sitename = 'Gatesville - July 17 to July 28';
    %let file = gates;
    %let lib = work.;
    %let htmlpth = 'c:\documents and settings\nragland\my documents\GreensSummary.html';

    data nh3fail;
                length result $ 6;
                set &file;
                bias = (nh4lcs/nh4std)*100;
                date1 = datepart(datetime);
                time = timepart(datetime);
                if bias < 70 then result = 'FAIL' ;
                else if bias > 130 then result = 'FAIL';
                else result = 'Passed';
                if nh4lcs ne '.';
                format
                date1 mmddyy10.
                time time5.;

    run;

    data nh3fail2 (keep = datetime date time nh4lcs) ;
                set &file;
                if nh4std ne '.';
                date = datepart(datetime);
                time = timepart(datetime);
                format
                date mmddyy10.
                time time5.;

    run;

    data nh3fail3 (drop = nh4lcs datetime date1);
```

```
        set nh3fail2;
        date = intnx('day', date, 1) ;
        lcsd = nh4lcs;

run;

proc sort data = nh3fail2;
        by date time;

run;

proc sort data = nh3fail3;
        by date time;

run;

data nh3merge;
        merge nh3fail2 nh3fail3;
        by date time;

run;

data nh3mergefail;
        length result $ 6;
        set nh3merge;
        rpd =(abs(nh4lcs - lcsd)/((nh4lcs+lcsd)/2))*100;
        if rpd > 30 then result = 'FAIL';
        else result = 'Passed';
        if rpd ne '.';

run;

data no3fail;
                length result $ 6;
                set &file;
                bias = (no3lcs/no3std)*100;
                date1 = datepart(datetime);
                time = timepart(datetime);
                if bias < 70 then result = 'FAIL' ;
                else if bias > 130 then result = 'FAIL';
                else result = 'Passed';
                if no3lcs ne '.';
                format
                date1 mmddyy10.
                time time5.;

run;

data no3fail2 (keep = datetime date time no3lcs) ;
                set &file;
                if no3std ne '.';
                date = datepart(datetime);
```

```
                time = timepart(datetime);
        format
                date mmddyy10.
                time time5.;

run;

data no3fail3 (drop = no3lcs datetime date1);
        set no3fail2;
        date1 = datepart(datetime);
        time = timepart(datetime);
        date = intnx('day', date1, 1) ;
        lcsd = no3lcs;
                format
                date mmddyy10.
                time time5.;

run;

proc sort data = no3fail2;
        by date time;

run;

proc sort data = no3fail3;
        by date time;

run;

data no3merge;
        merge no3fail2 no3fail3;
        by date time;

run;

data no3mergefail;
        length result $ 6;
        set no3merge;
        rpd =(abs(no3lcs - lcsd)/((no3lcs+lcsd)/2))*100;
        if rpd > 30 then result = 'FAIL';
        else result = 'Passed';
        if rpd ne '.';

run;

data po4fail;
                length result $ 6;
                set &file;
                bias = (po4lcs/po4std)*100;
                date1 = datepart(datetime);
                time = timepart(datetime);
                if bias < 75 then result = 'FAIL' ;
                else if bias > 125 then result = 'FAIL';
```

```
                else result = 'Passed';
                if po4lcs ne '.';
                format
                date1 mmddyy10.
                time time5.;

run;

data po4fail2 (keep = datetime date time po4lcs) ;
                set &file;
                if po4std ne '.';
                date = datepart(datetime);
                time = timepart(datetime);
                format
                date mmddyy10.
                time time5.;

run;

data po4fail3 (drop = po4lcs datetime date1);
        set po4fail2;
        date1 = datepart(datetime);
        time = timepart(datetime);
        date = intnx('day', date1, 1) ;
        lcsd = po4lcs;
        format
                date mmddyy10.
                time time5.;

run;

proc sort data = po4fail2;
        by date time;

run;

proc sort data = po4fail3;
        by date time;

run;

data po4merge;
        merge po4fail2 po4fail3;
        by date time;

run;

data po4mergefail;
        length result $ 6;
        set po4merge;
        rpd =(abs(po4lcs - lcsd)/((po4lcs+lcsd)/2))*100;
        if rpd > 30 then result = 'FAIL';
        else result = 'Passed';
```

```
        if rpd ne '.';
run;
data &lib.ecerror;
        length result $ 6;
        set &file;
        if ecchk < 1700 then result = 'FAIL';
        else if  ecchk > 2140 then result = 'FAIL';
        else result = 'Passed';
        if ecchk ne '.';
run;
data &lib.no3error;
        length result $ 6;
        set &file;
        no3chk = no3chk1 - no3chk2;
        if no3chk < 3000 then result = 'FAIL';
        else if no3chk > 4500 then result = 'FAIL';
        else result = 'Passed';
        if no3chk ne '.';
run;
data &lib.pherror;
        length result $ 6;
        set &file;
        if phchk < 3.8 then result = 'FAIL';
        else if phchk > 4.5 then result = 'FAIL';
        else result = 'Passed';
        if phchk ne '.';
run;
data &lib.doerror;
        length result $ 6;
        set &file;
        if dochk < 19000 then result = 'FAIL';
        else if dochk > 28000 then result = 'FAIL';
        else result = 'Passed';
        if dochk ne '.';
run;
data &lib.turberror;
        length result $ 6;
        set &file;
        if turbchk < -30 then result = 'FAIL';
        else if turbchk > 50 then result = 'FAIL';
        else result = 'Passed';
        if turbchk ne '.';
```

```
run;

data &lib.nh4error;
        length result $ 6;
        set &file;
        nhchk = nh4chk1 - nh4chk2;
        if nhchk < 5000 then result = 'FAIL' ;
        else if nhchk > 6500 then result = 'FAIL';
        else result = 'Passed';
        if nhchk ne '.';

run;

data &lib.po4error;
        length result $ 6;
        set &file;
        po4chka = po4chk1 - po4chk0;
        po4chkb = po4chk2 - po4chk1;
        if po4chka < 9500 then result = 'FAIL';
        else if po4chka > 13500 then result = 'FAIL';
        else if po4chkb < 7500 then result = 'FAIL';
        else if po4chkb > 11500 then result = 'FAIL';
        else result = 'Passed';
        if po4chk1 ne '.';
        if po4chk2 ne '.';
        if po4chk0 ne '.';

run;

ods html body = 'c:\documents and settings\nragland\desktop\myhtml.html';

proc print data = nh3fail label noobs;
        var date1 time bias result;
        label
                date1 = 'Date'
                time = 'Time'
                bias = 'Bias'
                result = 'Result';
        title1 &sitename ' Validation Summary';
        title2 ' ';
        title3 'Ammonia Bias Results';

run;

proc print data = no3fail label noobs;
        var date1 time bias result;
        label
                date = 'Date'
                time = 'Time'
                bias = 'Bias'
                result = 'Result';
```

```
        title 'Nitrate Bias Results';
run;
proc print data = po4fail label noobs;
        var date1 time bias result;
        label
                date1 = 'Date'
                time = 'Time'
                bias = 'Bias'
                result = 'Result';
        title 'Phosphate Bias Results';
run;
proc print data =  nh3mergefail label noobs;
        var date time rpd result;
        label
                date = 'Date'
                time = 'Time'
                rpd = 'RPD'
                result = 'Result';
        title 'Ammonia RPD Results';
run;
proc print data =  no3mergefail label noobs;
        var date time rpd result;
        label
                date = 'Date'
                time = 'Time'
                rpd = 'RPD'
                result = 'Result';
        title 'Nitrate RPD Results';
run;
proc print data =  po4mergefail label noobs;
        var date time rpd result;
        label
                date1 = 'Date'
                time = 'Time'
                rpd = 'RPD'
                result = 'Result';
        title 'Phosphate RPD Results';
run;
proc print data = &lib.ecerror label noobs;
        var datetime ecchk result;
        label
                datetime = 'Date/Time'
```

```
                ecchk = 'EC Check Count'
                result = 'Result';
        title 'EC Check Count Results';

run;

proc print data = &lib.no3error label noobs;
        var datetime no3chk result;
        label
                datetime = 'Date/Time'
                no3chk = 'Nitrate Check Count'
                result = 'Result';
        title 'Nitrate Check Count Results';

run;

proc print data = &lib.pherror label noobs;
        var datetime phchk result;
        label
                datetime = 'Date/Time'
                phchk = 'pH Check Count'
                result = 'Result';
        title 'pH Check Count Results';

run;

proc print data = &lib.doerror label noobs;
        var datetime dochk result;
        label
                datetime = 'Date/Time'
                dochk = 'Dissolved Oxygen Check Count'
                result = 'Result';
        title 'Dissolved Oxygen Check Count Results';

run;

proc print data = &lib.turberror label noobs;
        var datetime turbchk result;
        label
                datetime = 'Date/Time'
                turbchk = 'Turbidity Check Count'
                result = 'Result';
        title 'Turbidity Check Count Results';

run;

proc print data = &lib.nh4error label noobs;
        var datetime nhchk result;
        label
                datetime = 'Date/Time'
                nhchk = 'Ammonia Check Count'
```

```
                result = 'Result';
        title 'Ammonia Check Count Results';

run;

proc print data = &lib.po4error label noobs;
        var datetime po4chka po4chkb result;
        label
                datetime = 'Date/Time'
                po4chka = 'Phosphate Check Count 1'
                po4chkb = 'Phosphate Check Count 2'
                result = 'Result';
        title 'Phosphate Check Count Results';

run;

ods html close;
```

CompareDateTime.sas

```
*input the file path for the Aqualab file;
%let csvfile = 'c:\documents and settings\nragland\my documents\gates.csv';
*input the file path for the CAMS Data Printout file (This program reads in water quality data
from the CAMS Data Printout on the DSR webpage.  When
downloading using the CAMS Data Printout tool, you must choose the following parameters:
Surface Water Temperature (10010) in deg C, Surface Specific Conductance (10095) in uS/cm,
Surface Dissolved Oxygen (10300) in mg/l, Surface Water pH (10400) in pH,
Water Ammonia (10100) in mg/l, Water Nitrate (10101) in mg/l, Turbidity (10104) in NTU, and
Water Total Reactive Phosphate (10105) in mg/l.  Download as CSV and save as text;
%let infile = 'c:\documents and settings\nragland\my documents\cdptest.txt';

*Reading in Aqualab data;
data aqldata (drop = a b c d e f g h i j k l m n o p q r s t u v w x po4lcs1 y z aa bb cc dd ee)
        ;
        infile &csvfile dlm=',' truncover DSD;

*inputting the fields from the infile.  Many of them are not kept, but must be inputted for format
integrity, so they are named nonsensically.;
        input
                a
                datetime datetime17.
                b
                c
                d
                v
                spc
                a
                h
                do1
                pH1
                i
```

```
            turb1
            j
            nh3
            k
            f
            ff
            no3
            l
            m
            g
            trp1
            o
            y
            z
            p
            q
            temp
            r
            s
            t
            bb
            cc
            w
            dd
            x
            ee
            u;
        trp2 = trp1 * .001;
*Rounding values to better match what is found in the CAMS Data Printout dataset;
        do = round(do1, .1);
        ph = round(ph1, .1);
        turb = round(turb1, 1);
        trp = round(trp2, .01);
        date1 = datetime - (45 * 60);
        date = datepart(date1);
        timenew2 = timepart(date1) ;
*separating the time so that manipulations can be made the minutes that are needed to match
with
CAMS Data Printout data;
        minutes = minute(timenew2);
        hour = hour(timenew2);
*Manipulating minutes in the time (using the rule of 7s) to better match the CAMS Data Printout
data;
        if 53 <= minutes
```

```
            then minutes = 00;
        if 07 >= minutes
            then minutes = 00;
        if 08 <= minutes <= 22
            then minutes = 15;
        if 23 <= minutes <= 37
            then minutes = 30;
        if 38 <= minutes <= 52
            then minutes = 45;

        timenew = hms(hour,minutes,0);
        format datetime datetime17.
            date mmddyy10.
            timenew time5.
            timenew2 time5.;

run;

*Manipulating data from the Aqualab file to retain values;
data aql;

*Retaining values to match with the repeating values in the CAMS Data Printout.  Once the
LEADS
data goes to discrete values, this step will probably not be necessary.;
        retain lastspc;
        retain lastdo;
        retain lastph;
        retain lastturb;
        retain lastnh3;
        retain lastno3;
        retain lasttrp;
        retain lasttemp;
        set aqldata;

*Setting the values to retain;
        if spc ne '.' then lastspc = spc;
        spc = lastspc;
        if do ne '.' then lastdo = do;
        do = lastdo;
        if ph ne '.' then lastph = ph;
        ph = lastph;
        if turb ne '.' then lastturb = turb;
        turb = lastturb;
        if nh3 ne '.' then lastnh3 = nh3;
        nh3 = lastnh3;
        if no3 ne '.' then lastno3 = no3;
        no3 = lastno3;
        if trp ne '.' then lasttrp = trp;
        trp = lasttrp;
```

```
        if temp ne '.' then lasttemp = temp;
        temp = lasttemp;

run;

data cdp;
        infile &infile dlm = ',' truncover dsd ;

*inputting the fields from the CAMS Data Printout;
        input
                date mmddyy10.
                dummy
                time time6.
                dummy2
                temp
                spc
                nh3
                no3
                turb1
                trp
                do
                pH;

        timenew = time;

*eliminating missing values.  Missing values are a result of data that has been invalidated in
LEADS,
and is not necessarily non-matching...the CAMS Data Printout does not retain the actual data
value, so
it is better to ignore these missing values.  If there is actually an overwrite of the data, it will
come
out in the other parameters' values;
        if pH ne '.';
        if do ne '.';
        if trp ne '.';
        if nh3 ne '.';
        if no3 ne '.';
        if spc ne '.';
        if temp ne '.';

        turb = round(turb1, 1);

*formatting the values to human readable format for troubleshooting purposes;
        format
                date mmddyy10.
                time time5.
                datetime datetime15.
                timenew time5.;

run;
```

```
*Sorting for the merge;
proc sort data = cdp;
        by date timenew pH do trp nh3 no3 turb spc temp;

run;

*Sorting for the merge;
proc sort data = aql;
        by date timenew pH do trp nh3 no3 turb spc temp;

run;

data combo (keep = date timenew temp spc do pH turb nh3 no3 trp);
        merge aql (in = inc) cdp (in = ind);

*must merge by the actual parameters in order to get non-matching parameters to fall out;
        by date timenew pH do trp nh3 no3 turb spc temp;

        if inc = 0;

*eliminating missing values.  Missing values are a result of data that has been invalidated in
LEADS,
and is not necessarily non-matching...the CAMS Data Printout does not retain the actual data
value, so
it is better to ignore these missing values.  If there is actually an overwrite of the data, it will
come
out in the other parameters' values;
        if date ne '.';
        if timenew ne '.';
        if pH ne '.';
        if do ne '.';
        if trp ne '.';
        if nh3 ne '.';
        if no3 ne '.';
        if turb ne '.';
        if spc ne '.';
        if temp ne '.';

run;
```