

SEACAR Analysis Filters: TN, Depth, Value Qualifiers and Thresholds

Date: March 4, 2022

Threshold Filtering

Threshold filters, following FDEP NEAR (insert full name) are used to exclude specific results values from the SEACAR Analysis. Based on the threshold filters, the following QAQC Flags are inserted into the SEACAR_QAQCFlagCode and SEACAR_QAQC_Description columns of the export data. The Include_YN column indicates whether the QAQC Flag will also indicate that data are excluded from analysis. No data are excluded from the data export, but the analysis scripts can use the Include_YN column to exclude data.

Table 1. QA Flags inserted based on threshold checks.

SEACAR_QAQC_Description	Include_YN	SEACAR_QAQCFlagCode
Exceeds Maximum threshold. Not verified in raw data	N	2Q
Exceeds Maximum threshold. Verified in raw data	N	3Q
Below Minimum threshold. Not verified in raw data	N	4Q
Below Minimum threshold. Verified in raw data	N	5Q
Within threshold tolerance	Y	6Q
No defined thresholds for this parameter	Y	7Q

Table 2. Threshold values based on parameter and sensor type.

ParameterName	Units	Media	LowThreshold	HighThreshold	MonitoringType	SensorType
Dissolved Oxygen	mg/L	Water	0	22	Discrete	NULL
Dissolved Oxygen	mg/L	Water	0	50	Continuous	YSI EXOs
Dissolved Oxygen	mg/L	Water	0	50	Continuous	Analysis Only - 2022-04-04
Dissolved Oxygen	mg/L	Water	0	50	Continuous	6600 Series
Salinity	ppt	Water	0	70	Continuous	6600 Series
Salinity	ppt	Water	0	70	Continuous	YSI EXOs
Salinity	ppt	Water	0	70	Continuous	Analysis Only - 2022-04-04
Salinity	ppt	Water	0	70	Discrete	NULL
Water Temperature	Degrees C	Water	-5	45	Continuous	YSI EXOs
Water Temperature	Degrees C	Water	0	40	Discrete	NULL
Water Temperature	Degrees C	Water	-5	45	Continuous	Analysis Only - 2022-04-04
Water Temperature	Degrees C	Water	-5	45	Continuous	6600 Series
pH	NULL	Water	2	14	Continuous	Analysis Only - 2022-04-04
pH	NULL	Water	2	14	Continuous	6600 Series

pH	NULL	Water	0	13	Discrete	NULL
pH	NULL	Water	2	14	Continuous	YSI EXOs
Dissolved Oxygen Saturation	%	Water	0	310	Discrete	NULL
Dissolved Oxygen Saturation	%	Water	0	500	Continuous	YSI EXOs
Dissolved Oxygen Saturation	%	Water	0	500	Continuous	6600 Series
Dissolved Oxygen Saturation	%	Water	0	500	Continuous	Analysis Only - 2022-04-04
Specific Conductivity	mS/cm	Water	0	100	Continuous	6600 Series
Specific Conductivity	mS/cm	Water	0	200	Continuous	YSI EXOs
Specific Conductivity	mS/cm	Water	0	100	Discrete	NULL
Turbidity	NTU	Water	0	4000	Continuous	YSI EXOs
Turbidity	NTU	Water	0	1000	Continuous	6600 Series
Turbidity	NTU	Water	0	4000	Continuous	Analysis Only - 2022-04-04
Turbidity	NTU	Water	0	NULL	Discrete	NULL
Total Suspended Solids, TSS	mg/L	Water	0	NULL	Discrete	NULL
Chlorophyll a uncorrected for pheophytin	ug/L	Water	0	NULL	Discrete	NULL
Chlorophyll a corrected for pheophytin	ug/L	Water	0	NULL	Discrete	NULL
Secchi Depth	m	Water	0	50	Discrete	NULL
Light Extinction Coefficient	m ⁻¹	Water	0	NULL	Discrete	NULL
Colored dissolved organic matter, CDOM	PCU	Water	0	NULL	Discrete	NULL
Fluorescent dissolved organic matter, FDOM	QSE	Water	0	NULL	Discrete	NULL
Total Nitrogen	mg/L	Water	0	NULL	Discrete	NULL
Total Kjeldahl Nitrogen TKN	mg/L	Water	0	NULL	Discrete	NULL
NO ₂ +3 Filtered	mg/L	Water	0	NULL	Discrete	NULL
NH ₄ Filtered	mg/L	Water	0	NULL	Discrete	NULL
Total Phosphorus	mg/L	Water	0	NULL	Discrete	NULL
PO ₄ Filtered	mg/L	Water	0	NULL	Discrete	NULL
Ammonia- Un-ionized (NH ₃)	mg/L	Water	0	NULL	Discrete	NULL
Nitrate (N)	mg/L	Water	0	NULL	Discrete	NULL
Nitrite (N)	mg/L	Water	0	NULL	Discrete	NULL
Nitrogen, organic	mg/L	Water	0	NULL	Discrete	NULL

Value Qualifier Filtering

Value qualifier codes included within the data are used to exclude certain results from the analysis. The data are retained in the data export files, but the analysis uses the “Include” column to filter the results.

Table 3. Value qualifier codes excluded from analysis.

Value Qualifier	Include YN/10	MDL YN/10	Qualifier Source
H	0	0	STORET_WIN
J	0	0	STORET_WIN
V	0	0	STORET_WIN
Y	0	0	STORET_WIN

Activity Type Filtering

Activity type is used to determine whether data are included in the SEACAR analysis. The following table shows which activity types are included.

ActivityType	Include_YN
Equipment Blank	N
Field	Y
Field Blank	N
Field Msr/Obs	Y
Field Replicate	Y
Sample	Y
Sample-Composite	Y
Sample/Field	Y

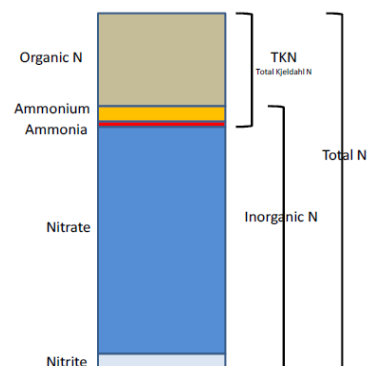
Total Nitrogen Calculation

The logic for calculated Total Nitrogen was provided by Kevin O'Donnell and colleagues at FDEP (with the help of Jay Silvanima, Watershed Monitoring Section). The following logic is used, in this order, based on the availability of specific nitrogen components. The actual SQL code used to produce the calculation is provided in Appendix A: SQL Code for Total Nitrogen Calculation.

- 1) $TN = TKN + NO_3O_2$;
- 2) $TN = TKN + NO_3 + NO_2$;
- 3) $TN = ORGN + NH_4 + NO_3O_2$;
- 4) $TN = ORGN + NH_4 + NO_2 + NO_3$;
- 5) $TN = TKN + NO_3$;
- 6) $TN = ORGN + NH_4 + NO_3$;

Additional Information:

- Rules for use of sample fraction:



- FDEP report that if both “Total” and “Dissolved” are reported, only “Total” is used. If the total is not reported, they do use dissolved as a best available replacement.
- An analysis of all SEACAR data shows that 90% of all possible TN calculations can be done using nitrogen components with the same sample fraction, rather than use nitrogen components with mixed total/dissolved sample fractions. In other words, TN can be calculated when TKN and NO3O2 are both total sample fraction, or when both are dissolved sample fraction. This is important, because then the calculated TN value is not based on components with mixed sample fractions.
- Values inserted into data:
 - ParameterName = “Total Nitrogen”
 - SEACAR_QAQCFlagCode = “1Q”
 - SEACAR_QAQC_Description = “SEACAR Calculated”

Determination of Surface and Bottom Sampling Data

The logic for implementing Surface and Bottom analysis follows the rules of the EPA/NOAA/USGS joint program, National Aquatic Resource Surveys, National Coastal Condition Assessment (<https://data.florida-seacar.org/datadiscovery/programs/details/118>)

The following logic will be used for identifying data as “Surface” and “Bottom”. These are done in order, and the value “Surface” or “Bottom” is inserted into the RelativeDepth column of the SEACAR data export file, with appropriate QAQC Flags added so that users know the logic used to determine surface, bottom, or both. It is important to note that a single sample could be used for both surface and bottom analysis, when located in very shallow coastal waters. The actual SQL code used to determine surface and bottom samples is provided in Appendix B: SQL Code for Depth Determination.

Surface Analysis

- If “Activity_Depth” <= 1 meter, then Analysis=Surface
 - *Explanation: Surface samples are to be taken at 0.5 meter below the surface (but need to allow some wiggle room for samples taken at slightly more than 0.5m to be “surface” – same rationale as for “bottom” cutoff of 1m)*
 - Values inserted into data:
 - RelativeDepth export column = “Surface”
 - SEACAR_QAQCFlagCode = “9Q”
 - SEACAR_QAQC_Description = “9Q - Surface Analysis: Activity_Depth <= 1 meter”
- If “Relative_Depth” = “Surface” and “Activity_Depth” is NULL, then Analysis=Surface
 - No values are inserted into data: RelativeDepth export column = “Surface”
- If “Relative_Depth” and “Activity_Depth” are NULL, then Analysis=Surface
 - *Explanation: We assume “surface” for all programs that do not report depth, based on the assumption that anyone collecting samples throughout the water column, at varying depths, would be responsible enough to report the actual depth.*
 - Values inserted into data:

- RelativeDepth export column = “Surface”
 - SEACAR_QAQCFlagCode = “11Q”
 - SEACAR_QAQC_Description = “11Q - Surface Analysis: Relative_Depth and Activity_Depth are NULL”
- This leaves everything else (e.g., mid-depths and below) excluded from the surface analysis.

Bottom Analysis

- If “Total_Depth” minus “Activity_Depth” <= 1 meters, then Analysis=Bottom
 - *Explanation: Bottom samples are to be taken at 0.5 meter above the bottom, but 1m is used for a buffer.*
 - Values inserted into data only when the RelativeDepth export column was not already populated with “Surface” from the steps above.
 - RelativeDepth export column = “Bottom”
 - SEACAR_QAQCFlagCode = “12Q”
 - SEACAR_QAQC_Description = “12Q - Bottom Analysis: Total_Depth minus Activity_Depth <= 1 meters”
- If “Relative_Depth” = “Bottom” and “Activity_Depth” is NULL, then Analysis=Bottom
 - No values are inserted into data: RelativeDepth export column = “Surface”
- This leaves everything else (e.g., mid-depths and above) excluded from the analysis.

Filter Logic for SEACAR Analysis

The following logic can be used by the Analysis Scripts to include surface and bottom data.

- Surface Analysis
 - RelativeDepth = “Surface”
- Bottom Analysis
 - RelativeDepth = “Bottom”, or
 - RelativeDepth = “Surface” AND SEACAR_QAQCFlagCode contains “12Q”

Note: IWR Analysis - Response from Phil Homann (FDEP): We use all surface water data regardless of depth (with the exception of Dissolved Oxygen in lakes, which must be sampled at a depth of 2.0 meters or less).

SEACAR QAQC Flags

The following is a list of existing QAQC Flags inserted into the SEACAR_QAQCFlagCode and SEACAR_QAQC_Description columns of the export data. The Include_YN column indicates whether the QAQC Flag will also indicate that data are excluded from analysis. No data are excluded from the data export, but the analysis scripts can use the Include_YN column to exclude data.

SEACAR_QAQC_Description	Include_YN	SEACAR_QAQCFlagCode
SEACAR Calculated	Y	1Q
Exceeds Maximum threshold. Not verified in raw data	N	2Q

Exceeds Maximum threshold. Verified in raw data	N	3Q
Below Minimum threshold. Not verified in raw data	N	4Q
Below Minimum threshold. Verified in raw data	N	5Q
Within threshold tolerance	Y	6Q
No defined thresholds for this parameter	Y	7Q
Original Value = "ON BOTTOM". Replaced with Total Depth	Y	8Q
Surface Analysis: Activity_Depth <= 1 meter	Y	9Q
Surface Analysis: Relative_Depth = "Surface" and Activity_Depth is NULL	Y	10Q
Surface Analysis: Relative_Depth and Activity_Depth are NULL	Y	11Q
Bottom Analysis: Total_Depth minus Activity_Depth <= 1 meters	Y	12Q
Bottom Analysis: Relative_Depth = "Bottom" and Activity_Depth is NULL	Y	13Q
Surface and Bottom Analysis: Total Depth is less than 1 meter	Y	14Q

Appendix A: SQL Code for Total Nitrogen Calculation

```

CREATE PROC [dbo].[usp_combined_wq_wc_nut_calculateParameters]
AS
    SET NOCOUNT ON

    /* -----
       Determine events that likely have necessary components to calculate TN, but which are missing TN
    ----- */
    SELECT
        SEACAR_EventID
    INTO
        #calculationsNeeded
    FROM
        Combined_WQ_WC_NUT
    WHERE
        ParameterID IN (
            16,          -- Total Kjeldahl Nitrogen TKN
            73,          -- Nitrate (N)
            75,          -- Nitrite (N)
            17,          -- NO2+3 Filtered
            18,          -- NH4 Filtered
            78           -- Nitrogen, organic
        )
        AND SEACAR_EventID NOT IN (
            SELECT
                SEACAR_EventID
            FROM
                Combined_WQ_WC_NUT
            WHERE
                ParameterID = 15           -- Total Nitrogen
            --GROUP BY
            SEACAR_EventID
        )
        --AND ISNUMERIC(ResultValue + 'e0') = 1
    GROUP BY SEACAR_EventID
    HAVING
        COUNT(DISTINCT ParameterID) > 1

    CREATE TABLE #calculationsTN (
        SEACAR_EventID          uniqueidentifier          NOT NULL,
        MaxRowID                int                        NOT NULL,
        TN_mgl                  numeric(25, 8)            NOT NULL,
        PRIMARY KEY (SEACAR_EventID)
    )

    ;WITH calc
    AS
    (
        SELECT
            SEACAR_EventID,
            TKN_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 16 THEN ResultValue END)),
            NOx_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 17 THEN ResultValue END)),
            MAX(RowID) AS MaxRowID
        FROM
            Combined_WQ_WC_NUT
        LEFT JOIN
            Combined_ValueQualifier
            ON vq.ValueQualifierID = wq.ValueQualifierID
    )

```

```

WHERE          SEACAR_EventID IN (SELECT SEACAR_EventID FROM #calculationsNeeded)
                                AND ParameterID IN (16, 17)
                                AND (vq.ValueQualifierID IS NULL OR vq.Include_YN = 1)
                                --AND ISNUMERIC(ResultValue + 'e0') = 1

GROUP BY SEACAR_EventID
)
INSERT INTO #calculationsTN
SELECT          SEACAR_EventID, MaxRowID, TKN_mgl + NOx_mgl AS TN_mgl
FROM            calc
WHERE           SEACAR_EventID NOT IN (SELECT SEACAR_EventID FROM #calculationsTN)
                                AND TKN_mgl IS NOT NULL
                                AND NOx_mgl IS NOT NULL

PRINT 'TKN + NO302 - ' + CONVERT(varchar, @@ROWCOUNT) + ' records calculated'

;WITH calc
AS
(
    SELECT          SEACAR_EventID,
                    TKN_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 16 THEN ResultValue END)),
                    NO3_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 73 THEN ResultValue END)),
                    NO2_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 75 THEN ResultValue END)),
                    MAX(RowID) AS MaxRowID

    FROM            Combined_WQ_WC_NUT                                wq
    LEFT JOIN       Combined_ValueQualifier                        vq ON vq.ValueQualifierID = wq.ValueQualifierID
    WHERE           SEACAR_EventID IN (SELECT SEACAR_EventID FROM #calculationsNeeded)
                                AND ParameterID IN (16, 73, 75)
                                AND (vq.ValueQualifierID IS NULL OR vq.Include_YN = 1)
                                --AND ISNUMERIC(ResultValue + 'e0') = 1

    GROUP BY SEACAR_EventID
)
INSERT INTO #calculationsTN
SELECT          SEACAR_EventID, MaxRowID, TKN_mgl + NO2_mgl + NO3_mgl AS TN_mgl
FROM            calc
WHERE           SEACAR_EventID NOT IN (SELECT SEACAR_EventID FROM #calculationsTN)
                                AND TKN_mgl IS NOT NULL
                                AND NO2_mgl IS NOT NULL
                                AND NO3_mgl IS NOT NULL

PRINT 'TKN + NO3 + NO2 - ' + CONVERT(varchar, @@ROWCOUNT) + ' records calculated'

;WITH calc
AS
(
    SELECT          SEACAR_EventID,
                    NORG_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 78 THEN ResultValue END)),
                    NOx_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 17 THEN ResultValue END)),
                    NH4_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 18 THEN ResultValue END)),
                    MAX(RowID) AS MaxRowID

```



```

FROM          Combined_WQ_WC_NUT                                wq
LEFT JOIN      Combined_ValueQualifier                        vq ON vq.ValueQualifierID = wq.ValueQualifierID
WHERE          SEACAR_EventID IN (SELECT SEACAR_EventID FROM #calculationsNeeded)
              AND ParameterID IN (78, 17, 18)
              AND (vq.ValueQualifierID IS NULL OR vq.Include_YN = 1)
              --AND ISNUMERIC(ResultValue + 'e0') = 1

GROUP BY SEACAR_EventID
)
INSERT INTO #calculationsTN
SELECT          SEACAR_EventID, MaxRowID, NORG_mgl + NH4_mgl + NOx_mgl AS TN_mgl
FROM            calc
WHERE          SEACAR_EventID NOT IN (SELECT SEACAR_EventID FROM #calculationsTN)
              AND NORG_mgl IS NOT NULL
              AND NH4_mgl IS NOT NULL
              AND NOx_mgl IS NOT NULL

PRINT 'NORG + NH4 + NO302 - ' + CONVERT(varchar, @@ROWCOUNT) + ' records calculated'

;WITH calc
AS
(
    SELECT          SEACAR_EventID,
                    NORG_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 78 THEN ResultValue END)),
                    NO3_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 73 THEN ResultValue END)),
                    NO2_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 75 THEN ResultValue END)),
                    NH4_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 18 THEN ResultValue END)),
                    MAX(RowID) AS MaxRowID
    FROM            Combined_WQ_WC_NUT                                wq
    LEFT JOIN      Combined_ValueQualifier                        vq ON vq.ValueQualifierID = wq.ValueQualifierID
    WHERE          SEACAR_EventID IN (SELECT SEACAR_EventID FROM #calculationsNeeded)
                  AND ParameterID IN (78, 73, 75, 18)
                  AND (vq.ValueQualifierID IS NULL OR vq.Include_YN = 1)
                  --AND ISNUMERIC(ResultValue + 'e0') = 1

    GROUP BY SEACAR_EventID
)
INSERT INTO #calculationsTN
SELECT          SEACAR_EventID, MaxRowID, NORG_mgl + NH4_mgl + NO2_mgl + NO3_mgl AS TN_mgl
FROM            calc
WHERE          SEACAR_EventID NOT IN (SELECT SEACAR_EventID FROM #calculationsTN)
              AND NORG_mgl IS NOT NULL
              AND NH4_mgl IS NOT NULL
              AND NO2_mgl IS NOT NULL
              AND NO3_mgl IS NOT NULL

PRINT 'NORG + NH4 + NO3 + NO2 - ' + CONVERT(varchar, @@ROWCOUNT) + ' records calculated'

;WITH calc
AS
(

```

```

SELECT          SEACAR_EventID,
                TKN_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 16 THEN ResultValue END)),
                NO3_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 73 THEN ResultValue END)),
                MAX(RowID) AS MaxRowID
FROM            Combined_WQ_WC_NUT          wq
LEFT JOIN      Combined_ValueQualifier      vq ON vq.ValueQualifierID = wq.ValueQualifierID
WHERE          SEACAR_EventID IN (SELECT SEACAR_EventID FROM #calculationsNeeded)
                AND ParameterID IN (16, 73)
                AND (vq.ValueQualifierID IS NULL OR vq.Include_YN = 1)
                --AND ISNUMERIC(ResultValue + 'e0') = 1

GROUP BY SEACAR_EventID
)
INSERT INTO #calculationsTN
SELECT          SEACAR_EventID, MaxRowID, TKN_mgl + NO3_mgl AS TN_mgl
FROM            calc
WHERE          SEACAR_EventID NOT IN (SELECT SEACAR_EventID FROM #calculationsTN)
                AND TKN_mgl IS NOT NULL
                AND NO3_mgl IS NOT NULL

PRINT 'TKN + NO3 - ' + CONVERT(varchar, @@ROWCOUNT) + ' records calculated'

;WITH calc
AS
(
    SELECT          SEACAR_EventID,
                    NORG_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 78 THEN ResultValue END)),
                    NO3_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 73 THEN ResultValue END)),
                    NH4_mgl = CONVERT(numeric(25,8), MAX(CASE WHEN ParameterID = 18 THEN ResultValue END)),
                    MAX(RowID) AS MaxRowID
    FROM            Combined_WQ_WC_NUT          wq
    LEFT JOIN      Combined_ValueQualifier      vq ON vq.ValueQualifierID = wq.ValueQualifierID
    WHERE          SEACAR_EventID IN (SELECT SEACAR_EventID FROM #calculationsNeeded)
                    AND ParameterID IN (78, 73, 18)
                    AND (vq.ValueQualifierID IS NULL OR vq.Include_YN = 1)
                    --AND ISNUMERIC(ResultValue + 'e0') = 1

    GROUP BY SEACAR_EventID
)
INSERT INTO #calculationsTN
SELECT          SEACAR_EventID, MaxRowID, NORG_mgl + NH4_mgl + NO3_mgl AS TN_mgl
FROM            calc
WHERE          SEACAR_EventID NOT IN (SELECT SEACAR_EventID FROM #calculationsTN)
                AND NORG_mgl IS NOT NULL
                AND NH4_mgl IS NOT NULL
                AND NO3_mgl IS NOT NULL

PRINT 'NORG + NH4 + NO3 - ' + CONVERT(varchar, @@ROWCOUNT) + ' records calculated'

```

```

/* -----
Insert calculated TN data
----- */
INSERT INTO Combined_WQ_WC_NUT (ProgramID, DataStreamID, ParameterID, LocationID, ActivityID, ActivityType, SampleDate,
ActivityDepth_m, RelativeDepth, TotalDepth_m,
ResultValue, MDL, PQL, DetectionUnit,
ValueQualifierID, SampleFraction, ResultComments, DateAdded, SEACAR_EventID)
SELECT
ProgramID, DataStreamID, p.ParameterID, LocationID,
ActivityID, ActivityType, SampleDate, ActivityDepth_m, RelativeDepth, TotalDepth_m,
c.TN_mgl AS ResultValue, MDL, PQL, p.Units AS
DetectionUnit, ValueQualifierID, SampleFraction, ResultComments, GETDATE() AS DateAdded, wq.SEACAR_EventID
FROM
Combined_WQ_WC_NUT wq
INNER JOIN #calculationsTN c ON wq.RowID = c.MaxRowID
INNER JOIN Combined_Parameters p ON p.ParameterID = 15 -- Total Nitrogen
WHERE
c.TN_mgl IS NOT NULL

INSERT INTO Combined_QAQCFlagID (CombinedRowID, SEACAR_QAQCFlagID, CombinedTable)
SELECT
a.RowID, 1, 'Combined_WQ_WC_NUT'
FROM
(SELECT SEACAR_EventID, MAX(RowID) as RowID
FROM Combined_WQ_WC_NUT
WHERE ParameterID = 15
GROUP BY SEACAR_EventID
) a
INNER JOIN #calculationsTN b ON a.SEACAR_EventID = b.SEACAR_EventID
AND NOT EXISTS (SELECT *
FROM Combined_QAQCFlagID c
WHERE a.RowID = c.CombinedRowID
AND c.SEACAR_QAQCFlagID = 1)

/*
INSERT INTO Combined_QAQCFlagID (CombinedRowID, SEACAR_QAQCFlagID, CombinedTable)
SELECT
a.RowID, 1, 'Combined_WQ_WC_NUT'
FROM
Combined_WQ_WC_NUT a
INNER JOIN #calculationsTN b ON a.RowID = b.MaxRowID
WHERE
a.ParameterID = 15
AND NOT EXISTS (SELECT *
FROM Combined_QAQCFlagID c
WHERE a.RowID = c.CombinedRowID
AND c.SEACAR_QAQCFlagID = 1)
*/

```

GO

Appendix B: SQL Code for Depth Determination

```
CREATE PROC [dbo].[usp_combined_wq_wc_nut_update_RelativeDepth]
```

```
AS
```

```
    SET NOCOUNT ON;  
    SET XACT_ABORT ON;
```

```
/* -----  
    Set Relative Depths according to Shawn's confirmed logic  
----- */
```

```
CREATE TABLE #relDepth
```

```
(  
    CombinedRowID int,  
    RelativeDepth varchar(20),  
    SEACAR_QAQCFlagID int  
)
```

```
INSERT INTO #relDepth(CombinedRowID, RelativeDepth, SEACAR_QAQCFlagID)  
SELECT a.RowID, 'Surface', 9 -- Surface Analysis: Activity_Depth <= 1 meter  
FROM Combined_WQ_WC_NUT a  
WHERE a.ActivityDepth_m <= 1.0  
UNION ALL  
SELECT a.RowID, 'Surface', 11 -- Surface Analysis: Relative_Depth and Activity_Depth are NULL  
FROM Combined_WQ_WC_NUT a  
WHERE a.RelativeDepth IS NULL  
AND a.ActivityDepth_m IS NULL  
UNION ALL  
SELECT a.RowID, 'Bottom', 12 -- Bottom Analysis: Total_Depth minus Activity_Depth <= 1 meters  
FROM Combined_WQ_WC_NUT a  
WHERE a.TotalDepth_m - a.ActivityDepth_m <= 1.0  
AND a.TotalDepth_m IS NOT NULL  
AND a.ActivityDepth_m IS NOT NULL
```

```
INSERT INTO Combined_QAQCFlagID(CombinedRowID, SEACAR_QAQCFlagID, CombinedTable)  
SELECT a.CombinedRowID, a.SEACAR_QAQCFlagID, 'Combined_WQ_WC_NUT'  
FROM #relDepth a  
WHERE NOT EXISTS (SELECT *  
                  FROM Combined_QAQCFlagID b  
                   WHERE a.CombinedRowID = b.CombinedRowID  
                   AND a.SEACAR_QAQCFlagID = b.SEACAR_QAQCFlagID)
```

```
-- UPDATE Combined_WQ_WC_NUT with Relative Depth WHERE Missing
UPDATE a
SET a.RelativeDepth = b.RelativeDepth
FROM Combined_WQ_WC_NUT a
INNER JOIN #relDepth b on a.RowID = b.CombinedRowID

-- Drop temp table
DROP TABLE #relDepth
```

GO