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 Maximium threshold. Not verified in raw data | N | 2Q |
| Exceeds Maximium 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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Parameter Name** | **Units** | **Media** | **Low Threshold** | **High Threshold** | **Monitoring Type** | **Sensor Type** |
| 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 | 6600 Series |
| Dissolved Oxygen Saturation | % | Water | 0 | 500 | Continuous | 6600 Series |
| Dissolved Oxygen Saturation | % | Water | 0 | 500 | Continuous | YSI EXOs |
| Dissolved Oxygen Saturation | % | Water | 0 | 270 | Discrete | NULL |
| pH |  | Water | 0 | 14 | Discrete | NULL |
| pH |  | Water | 2 | 14 | Continuous | YSI EXOs |
| pH |  | Water | 2 | 14 | Continuous | 6600 Series |
| Salinity | ppt | Water | 0 | 70 | Discrete | NULL |
| Salinity | ppt | Water | 0 | 70 | Continuous | YSI EXOs |
| Salinity | ppt | Water | 0 | 70 | Continuous | 6600 Series |
| Secchi Depth | m | Water | 0 | 50 | Discrete | NULL |
| 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 | 110 | Discrete | NULL |
| Turbidity | NTU | Water | 0 | 4000 | Continuous | YSI EXOs |
| Turbidity | NTU | Water | 0 | 1000 | Continuous | 6600 Series |
| Water Temperature | Degrees C | Water | -5 | 45 | Continuous | YSI EXOs |
| Water Temperature | Degrees C | Water | -2 | 50 | Discrete | NULL |
| Water Temperature | Degrees C | Water | -5 | 45 | Continuous | 6600 Series |

# 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.

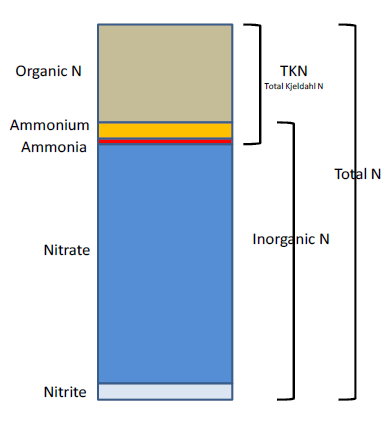
|  |  |  |  |
| --- | --- | --- | --- |
| **Value Qualifier** | **Include YN/10** | **MDL YN/10** | **Qualifier Source** |
| H | 0 | 0 | STORET\_WIN |
| J | 0 | 0 | STORET\_WIN |
| V | 0 | 0 | STORET\_WIN |
| Y | 0 | 0 | STORET\_WIN |

# 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 + NO3O2;
2. TN = TKN + NO3 + NO2;
3. TN = ORGN + NH4 + NO3O2;
4. TN = ORGN + NH4 + NO2 + NO3;
5. TN = TKN + NO3;
6. TN = ORGN + NH4 + NO3;

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 Maximium threshold. Not verified in raw data | N | 2Q |
| Exceeds Maximium 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 wq

LEFT JOIN Combined\_ValueQualifier vq 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 + NO3O2 - ' + 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 + NO3O2 - ' + 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