

# Data Acquisition Standard Operating Procedures

## St. Andrews Aquatic Preserve Water Quality Monitoring (ID# 470)

Last Updated: 5/6/2023

### Program Summary

---

Water quality

### URLs

---

- Program - <https://floridadep.gov/rcp/aquatic-preserve/locations/st-andrews-aquatic-preserve>
- DDI - <https://data.florida-seacar.org/programs/details/470>

### Contacts

---

Contact Name	Organization	Email	Phone
Jonathan Brucker		Jonathan.Brucker@dep.state.fl.us	850-670-7723

### Data Tables

---

- Data\_470A\_Final
- Data\_470A\_Load
- Data\_470B\_Load

### Data Stored Procedures

---

- usp\_combined\_wq\_wc\_nut\_insert\_470

# Data Acquisition Standard Operating Procedures: ProgramID 470

Date Created: 02/07/2019

Created By: *Claude Kershaw*

Data File Path:

1. STORET and WIN Data Tables already in the SEACAR database.
2. Spatial Data: In the "Notes" Section of the DDI.

DDI URL: <http://dev.seacar.waterinstitute.usf.edu/datadiscovery/programs/details/470>

Contact Information:

Contact Name: Jonathan Brucker

Contact Organization: Program - St. Andrews Aquatic Preserve Water Quality Monitoring

Contact Email: Jonathan.Brucker@dep.state.fl.us

Contact Phone:

Procedure Overview:

1. Use SQL Server Import Export Wizard to load the Stations from the "Notes" in the DDI into table **Locations\_470A**.
2. Use SQL Server Import Export Wizard to load the data from **Data\_STORET\_Load** into table **Data\_470A\_Load** where StationIDs are found in **Locations\_470A**.
3. Use SQL Server Import Export Wizard to load the data from **Data\_WIN\_Load** into table **Data\_470B\_Load** where StationIDs are found in **Locations\_470A**.
4. Execute procedure usp\_Data\_470A\_Load\_insert to load the data into table **Data\_470A\_Final**.
5. Execute procedure usp\_Data\_470B\_Load\_insert to load the data into table **Data\_470B\_Final**.
6. Add the Monitoring Locations from table **SampleLocation\_STORET** and **SampleLocation\_WIN** to the **SampleLocation\_Point** table if they are found in **Locations\_470A**.
7. Add new Monitoring Locations into the **SampleLocation** table. This will generate a LocationID for each Monitoring Location.
8. Update the **SampleLocation\_Point** table with the LocationID generated in the **SampleLocation** table. Run procedure usp\_SampleLocation\_Point\_update to do this.
9. Update the LocationID column in table **Data\_470A\_Final** with the LocationID in the **SampleLocation** table. Join on the ['STORET\_'+Station\_ID] column in **Data\_470A\_Final** and the ProgramLocationID column in **SampleLocation**.
10. Update the LocationID column in table **Data\_470B\_Final** with the LocationID in the **SampleLocation** table. Join on the ['WIN\_'+Station\_ID] column in **Data\_470B\_Final** and the ProgramLocationID column in **SampleLocation**.

Data Tables

1. Data\_470A\_Load

2. Data\_470A\_Final
3. Data\_470B\_Load
4. Data\_470B\_Final

#### Data Stored Procedures

1. usp\_Data\_470A\_Load\_insert
2. usp\_Data\_470B\_Load\_insert
3. usp\_SampleLocation\_Point\_update

#### GIS Procedures

1. The Monitoring Location information is found in the table **Locations\_470A**.
2. Complete steps 6 through 10 in the "Procedure Overview" section of this document.

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [dbo].[usp_combined_wq_wc_nut_insert_470]
```

```
AS
BEGIN
SET NOCOUNT ON;
SET XACT_ABORT ON;
```

```
-- THIS IS JUST FOR STORET/WIN DATA MANUAL LOADS
```

```
-- Constants - PLEASE SET NOW!!
DECLARE @combinedTable varchar(50) = 'Combined_WQ_WC_NUT'
DECLARE @dataLoadCode varchar(10)
```

```
-- Temporary
SET @dataLoadCode = '470A'
```

```
-- Setup data load
DECLARE @runBy varchar(50) = SYSTEM_USER;
DECLARE @programID varchar(10);
DECLARE @dataStreamID varchar(10);
```

```
SELECT @dataStreamID = DataStreamID,
@programID = ProgramID
FROM DataStreamProcedure
WHERE DataLoadCode = @dataLoadCode;
```

```
-- Delete Existing Data
exec usp_delete_combined @dataStreamID, 'Combined_WQ_WC_NUT'
```

```

INSERT INTO Combined_WQ_WC_NUT (ProgramID, DataStreamID, ParameterID, LocationID, SampleDate,
ActivityDepth_m, TotalDepth_m, RELATIVEDEPTH, ResultValue, DateAdded, SampleFraction,
ValueQualifierID, ActivityType)
SELECT @programID, @dataStreamID, c.parameterID, LocationID, a.ACTIVITY_START_DATE,
[dbo].[udf_convert_units](a.DEPTH_UNIT, 'm', ACTIVITY_DEPTH),
[dbo].[udf_convert_units](d.result_UNIT, 'm', d.RESULT_VALUE), NULL, dbo.udf_convert_units(b.Original
c.Units, a.RESULT_VALUE) as ResultValue, getdate(), a.SAMPLE_FRACTION, vq.ValueQualifierID,
a.ACTIVITY_TYPE
```

```
FROM Data_470A_Final
INNER JOIN Combined_Conversion_Parameters b ON a.CHARACTERISTIC = b.OriginalParameter and
a.RESULT_UNIT = b.OriginalUnits and b.DataStreamID = @dataStreamID
```

```
INNER JOIN Combined_Parameters c ON b.TargetParameterID = c.ParameterID
LEFT JOIN Combined_ValueQualifier vq on vq.ValueQualifier = a.VALUE_QUALIFIER
LEFT JOIN (SELECT RESULT_VALUE, RESULT_UNIT, ACTIVITY_ID
```

```
FROM Data_470A_Final
WHERE CHARACTERISTIC = 'Depth, bottom') d ON a.ACTIVITY_ID = d.ACTIVITY_ID
```

```
WHERE (ISNUMERIC(a.RESULT_VALUE) = 1)
AND a.ACTIVITY_START_DATE not like '0%'
```

```
AND vq.QualifierSource = 'STORET_WIN'
UNION ALL
```

```
-- INSERT MDL when result is '*Non-detect'
SELECT @programID, @dataStreamID, c.parameterID, LocationID, a.ACTIVITY_START_DATE,
[dbo].[udf_convert_units](a.DEPTH_UNIT, 'm', ACTIVITY_DEPTH),
[dbo].[udf_convert_units](d.result_UNIT, 'm', d.RESULT_VALUE), NULL, CASE a.RESULT_VALUE WHEN
'*Non-detect' THEN dbo.udf_convert_units(b.OriginalUnits, c.Units, a.MDL) WHEN 'ON BOTTOM' THEN
[dbo].[udf_convert_units](d.result_UNIT, 'm', d.RESULT_VALUE) END as ResultValue, getdate(),
a.SAMPLE_FRACTION, vq.ValueQualifierID, a.ActivityType
```

```
FROM Data_470A_Final
INNER JOIN Combined_Conversion_Parameters b ON a.CHARACTERISTIC = b.OriginalParameter and
a.DETECTION_UNIT = b.OriginalUnits and b.DataStreamID = @dataStreamID
```

```
INNER JOIN Combined_Parameters c ON b.TargetParameterID = c.ParameterID
LEFT JOIN Combined_ValueQualifier vq on vq.ValueQualifier = a.VALUE_QUALIFIER
LEFT JOIN (SELECT RESULT_VALUE, RESULT_UNIT, ACTIVITY_ID
```

```
FROM Data_470A_Final
```

```
WHERE CHARACTERISTIC = 'Depth, bottom') d ON a.ACTIVITY_ID = d.ACTIVITY_ID
WHERE a.RESULT_VALUE IN ('*Non-detect','ON BOTTOM')
AND a.ACTIVITY_START_DATE not like '0%'
AND v.QualifierSource = 'STORET_WIN'
```

```
DELETE Combined_Data_Tracking
WHERE DataStreamID = @dataStreamID
AND @combinedTable = @combinedTable
```

```
INSERT INTO Combined_Data_Tracking (ProgramID, IndicatorID, DataStreamID, ParameterID,
CombinedTableName, NumRowsFinal, NumRowsCombined, LastUpdateDate, LastUpdateBy)
SELECT @ProgramID, b.IndicatorID, a.DataStreamID, a.ParameterID, @CombinedTable,
COUNT(a.ResultValue), COUNT(a.ResultValue), GETDATE(), @runBy
FROM Combined_WQ_WC_NUTa
INNER JOIN Combined_Parameters b on a.ParameterID = b.ParameterID
WHERE a.ProgramID = @ProgramID
AND a.DataStreamID = @dataStreamID
GROUP BY a.ProgramID, b.IndicatorID, a.DataStreamID, a.ParameterID
```

```
/*
SELECT *
FROM DataStreamProcedure
WHERE ProgramID = 470
```

```
exec usp_delete_combined 144, 'Combined_WQ_WC_NUT'
exec usp_delete_combined 182, 'Combined_WQ_WC_NUT'
```

```
*/
```

```
END
```

```
GO
```