Data Acquisition Standard Operating Procedures

St. Andrew Bay Aquatic Preserve Seagrass Monitoring (ID# 556)

Last Updated: 2/17/2024

Program Summary

Linda Fitzhugh from Gulf Coast State College (GCSC) and the St. Andrew Bay Resource Monitoring Association (SABRMA) and volunteers from the community monitored almost every fall in St. Andrew Bay behind Shell Island (SAB), West Bay Bowl (WB-BOWL) and West Bay Arm (WB-ARM) from 2000–2009.

URLs

- Program - https://myfwc.com/media/17704/simm3-standrewbay.pdf
- DDI - https://data.florida-seacar.org/programs/details/556

Contacts

Contact Name	Organization	Email	Phone

Data Tables

- DATA_556A_Final
- DATA_556A_Load

Data Stored Procedures

- usp_Data_556A_Load_insert
- usp_combined_sav_insert_556A

Data Acquisition Standard Operating Procedures: ProgramID *556*

Date Created: *04/20/2020*

Created By: *Mrudhula Murali*

Data File Path:

1. "\\forest.usf.edu\data\PDrive\CAS-WI\Misc Projects\SEACAR_FDEP\Data\ ID_0556_SABAP_SAV\"

DDI URL: http://dev.seacar.waterinstitute.usf.edu/datadiscovery/programs/details/556

Contact Information:

Contact Name:

Contact Organization:

Contact Email:

Contact Phone:

Procedure Overview:

1. Use SQL Server Import Export Wizard to load the file "ID_556_dataCompiled.xlsx" sheet "SiteLocations" into table **Locations_556A.**
2. Use SQL Server Import Export Wizard to load the file "ID_556_dataCompiled.xlsx" sheet "ID_556 "in to table **Data_556A_Load.**
3. Convert the [Lat,Long] columns from degree to decimal before loading to the **[Data_556A_Load]** Table.
4. Increase the column size of [comments] to nvarchar(300) to avoid truncation error during the data load process.
5. Execute procedure usp_Data_556*_Load_insert to load the data into table **Data_556*_Final.**
6. Add the Monitoring Locations from tables **Locations_556A** with [Station] column as [Monitoring_Location] to the **SampleLocation_Point** table.
7. Add new Monitoring Locations into the **SampleLocation** table. This will generate a LocationID for each Monitoring Location.
8. Update the **SampleLocation_Point** and **SampleLocation_Line** 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_556A_Final** with the LocationID in the **SampleLocation** table. Join on the [Transect] column in **Data_556A_Final** and the ProgramLocationID column in **SampleLocation.**

Data Tables

1. Data_556*_Load
2. Data_556*_Final

Data Stored Procedures

1. usp_Data_556*_Load_insert
2. usp_SampleLocation_Point_update

GIS Procedures

1. The Monitoring Location information is found in the tables **Locations_556A**
2. Complete steps 8 through 11 in the “Procedure Overview” section of this document.

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [dbo].[usp_combined_sav_insert_556A]
AS
BEGIN
SET NOCOUNT ON;
SET XACT_ABORT ON;

-- Add Species Scientific Name
/*
alter table DATA_556A_Final add ScientificName varchar(50)
ÿ
update DATA_556A_Final
SET ScientificName = 'Halodule wrightii'
where Species = 's'
update DATA_556A_Final
SET ScientificName = 'Thalassia testudinum'
where Species = 't'
update DATA_556A_Final
SET ScientificName = 'Syringodium filiforme'
where Species = 'm'
*/

-- Constants - PLEASE SET NOW!!
DECLARE @dataLoadCode varchar(10) = '556A';
DECLARE @combinedTable varchar(50) = 'Combined_SAV'
DECLARE @parameterID int

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

SELECT@dataStreamID = DataStreamID,
@programID = ProgramID
FROMDataStreamProcedure
WHEREDataLoadCode = @dataLoadCode;


--delete existing data
exec usp_delete_combined @datastreamID, @combinedTable
ÿ
-- Insert data
SET @parameterID = 24
ÿ
--PER THE DEP 10/16/2023, a.[%_Cover] == '' should be replaced with 0
UPDATE a
SET a.[%_Cover] = 0
FROM Data_556A_Final a
WHERE a.[%_Cover] = ''

ÿ
INSERT INTO Combined_SAV (ProgramID, DataStreamID, ParameterID, LocationID,  SampleDate, SpeciesID
SamplingMethod1, SamplingMethod2, ReportingLevel, QuadSize_m2, Grid, ResultValue, Depth_M,  DateAd
QuadIdentifier,  SiteIdentifier)
SELECT@programID, @dataStreamID, @parameterID, a.LocationID,  a.Date, c.SpeciesID,  'Fixed',
'Natural', 'Quadrat', 1.00, 100, a.[%_Cover], NULLIF(a.Depth_m,''), GETDATE(),  Distance_m, Transe
FROMData_556A_Finala
LEFT JOINref_conversion_species b on a.Species = b.OriginalCommonIdentifier AND b.DataStreamID =
@dataStreamID
INNER JOIN  ref_species c on b.SpeciesID = c.SpeciesID
WHEREa.[%_Cover] <> ''
ANDa.LocationID is not null
ÿ
```

```
/*
SELECT *
FROM Combined_SAV

SELECT Distinct ProgramID, b.IndicatorName, c.ParameterName, NumRowsCombined, LastUpdateDate
FROM Combined_Data_Tracking a
INNER JOIN Indicator b on a.IndicatorID = b.IndicatorID
INNER JOIN Combined_Parameters c on a.ParameterID = c.ParameterID
WHERE b.Habitat = 'Submerged Aquatic Vegetation'

SELECT *
FROM Data_556A_Final

SELECT *
FROM ref_conversion_species
where ProgramID = 556

SELECT *
FROM Combined_Parameters a
INNER JOIN Indicator b on a.IndicatorID = b.IndicatorID
where b.Habitat = 'Submerged Aquatic Vegetation'

SELECT *
FROM DataStreamProcedure
WHERE ProgramID = 556

SELECT *
FROM ref_species
where habitat = 'Submerged Aquatic Vegetation'
and scientificname like '%drif%'

SELECT *
FROM ref_conversion_species
WHERE ProgramID = 556

exec usp_delete_combined 1299, 'Combined_SAV'
*/

END




GO
```

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC usp_Data_556A_Load_insert
AS
BEGIN
SET NOCOUNT ON
SET XACT_ABORT ON

INSERT INTO [dbo].[DATA_556A_Final]([Location]
,[Transect],[Staff],[Date],[Distance_m],[Depth_m],[Species]
           ,[%_Cover],[Blade_length_cm],[T_width_mm],[Comments])
SELECT [Location] ,[Transect],[Staff],[Date],[Distance_m],[Depth_m],[Species]
           ,[%_Cover],[Blade_length_cm],[T_width_mm],[Comments] FROM DATA_556A_Load
END
GO
```