# **NOAA ERD SCAT Data Management Tools**

# **Description and User Manual**

Prepared for:

National Oceanic and Atmospheric Administration

Office of Response and Restoration

Emergency Response Division

Prepared By:

Research Planning, Inc.

1121 Park St. Columbia SC 29201

# NOAA SCAT Data Management Tools

This document describes the components of the files, scripts, and executables that have been developed for managing Shoreline Cleanup Assessment Technique (SCAT) data by the NOAA Emergency Response Division (ERD). This toolbox includes five major components:

1. Microsoft Access database and interface to store and enter tabular SCAT data
2. ESRI shapefile templates to store spatial SCAT data
3. ESRI ArcGIS toolboxes and python scripts to facilitate the entry and analysis of spatial and tabular SCAT data in ArcGIS,
4. A stand-alone report generator program to perform rapid tabular data analysis and generate summary products, and
5. A stand-alone spatial snapping program to conduct rapid spatial analysis and generate summary products.

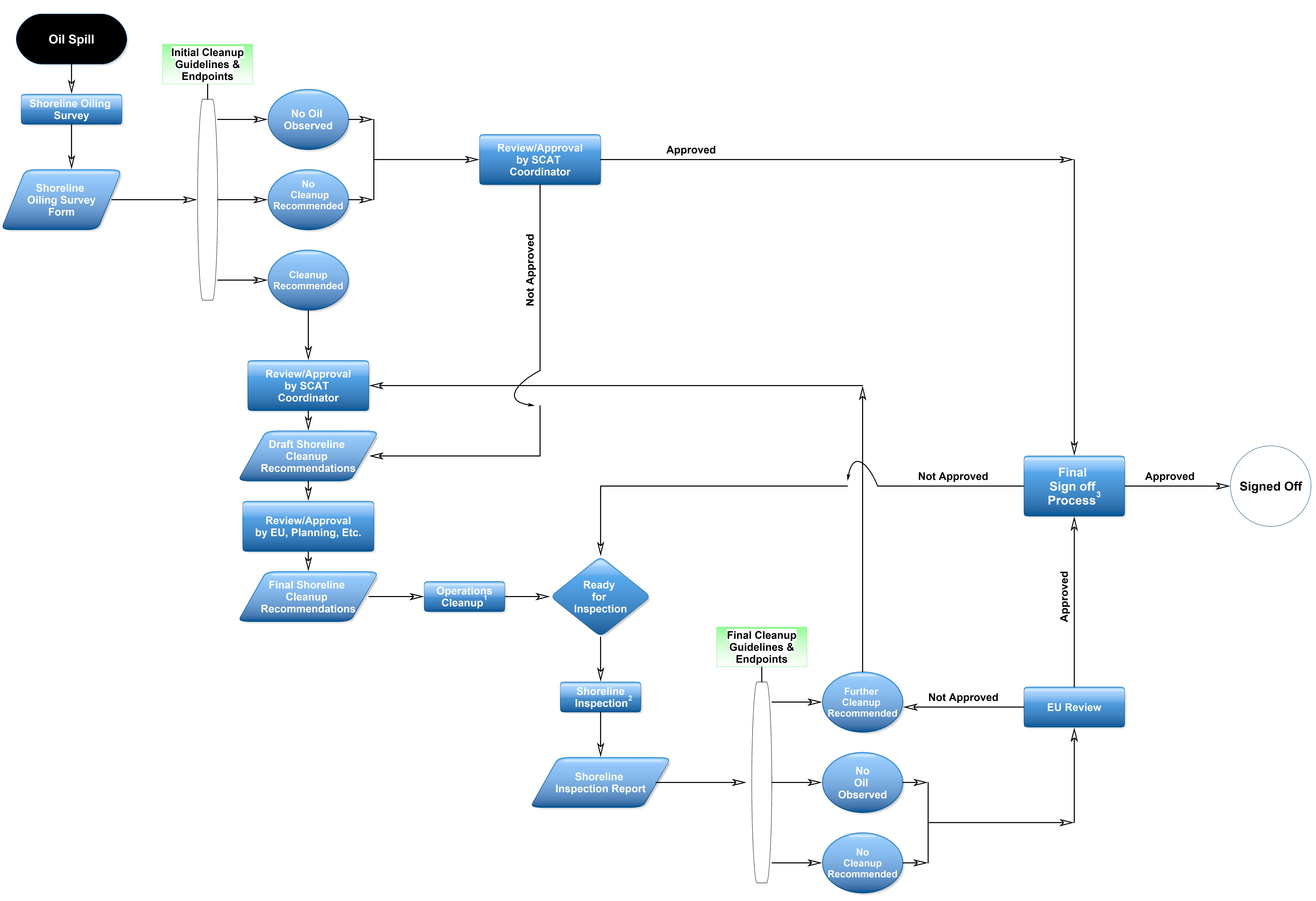
The following sections provide an overview of these tools, detail the overall data management context, provide specific details and user guides for each specific tool, and provide workflow guidelines for use of these tools as part of data management for a typical incident.

# Toolbox Overview

COMPLETE SECTION

# SCAT Process Review

For incidents involving spilled oil along shorelines, Shoreline Cleanup Assessment Technique (SCAT) program is often established to collect data on shoreline habitats, oil type, degree of contamination, physical processed, ecological and cultural resources, as well as to assign cleanup methods and endpoints, and track cleanup progress. Very often multiple repeated surveys are needed to monitor the effectiveness and effects of ongoing treatment method. A full description of shoreline assessment methods, techniques is beyond the scope of this document. Users are referred to the NOAA Shoreline Assessment Manual (NOAA, 2013) for a more detailed description of SCAT program requirements and specific methods. However, we briefly outline the needs of a data management program to track data as it relates to SCAT to provide a framework of use of these tools.



**Figure X.** General flow diagram for the SCAT process for a typical incident. Field data are collected during the Shoreline Oiling Survey and Shoreline Inspection stages, but summaries and data management are required at multiple points in the process.

# **Microsoft Access Database**

## **Overview**

This database is the primary tool used to store attribute information describing the results of SCAT surveys. The database generally consists of two separate Microsoft Access .mdb files. One file contains the user interface including forms, queries, and VBA code that allows automation of multiple tasks. The second file contains the actual data. Separating the Access files that contain the interface and associated logic from the data allows greater data security during updates or modifications to the database interface and tools during use in an actual incident. Also, interface-data separation allows easier use of these tools in incidents that require temporary child databases to be created to permit multiple analysts to conduct parallel data entry.

These databases are intended to be used to enter SCAT survey attribute data, track treatment and cleanup recommendations, and store administrative status data for shoreline segments daily throughout an incident response.

## Contents and Requirements

The database an interface consist of two separate Microsoft Access .mdb files, and use of the database requires Microsoft Access 2007, 2010, or 2013. This tool was developed by NOAA and is designed to run on a Microsoft Windows 32-bit operating system and has not been tested on other environments or with other versions of Microsoft Access.

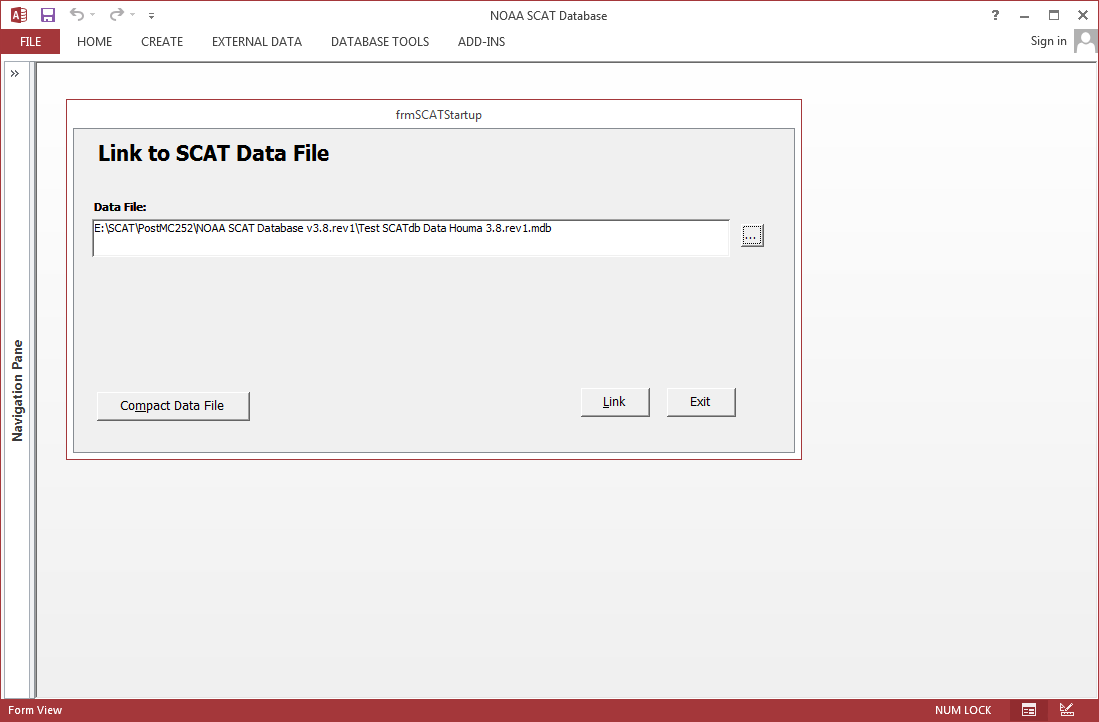
## Usage

The database, interface, and associated tools are designed to accommodate multiple different data storage and analysis tasks conducted as part of a variety of tasks. Usage guidelines are provided separately for each of these tasks below. The separate tasks are as follows:

1. Opening a database for use
2. Adding incident-specific information
3. Adding shoreline segment and operational division information
4. Adding field survey data
5. Reviewing survey data
6. Adding Shoreline Treatment Recommendation (STR) information
7. Adding people and organizations
8. Creating and importing helper child databases
9. Modifying database setup
10. Checking survey data in database for errors

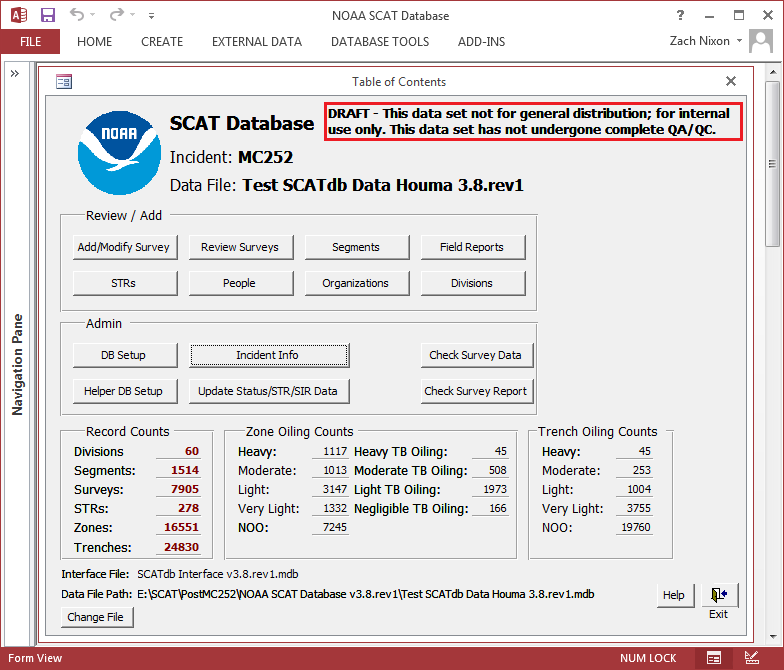
**Opening a database for use**

The initial startup screen visible upon opening the interface database prompts the user to link the interface database to a backend data storage database. The startup screen is shown below:



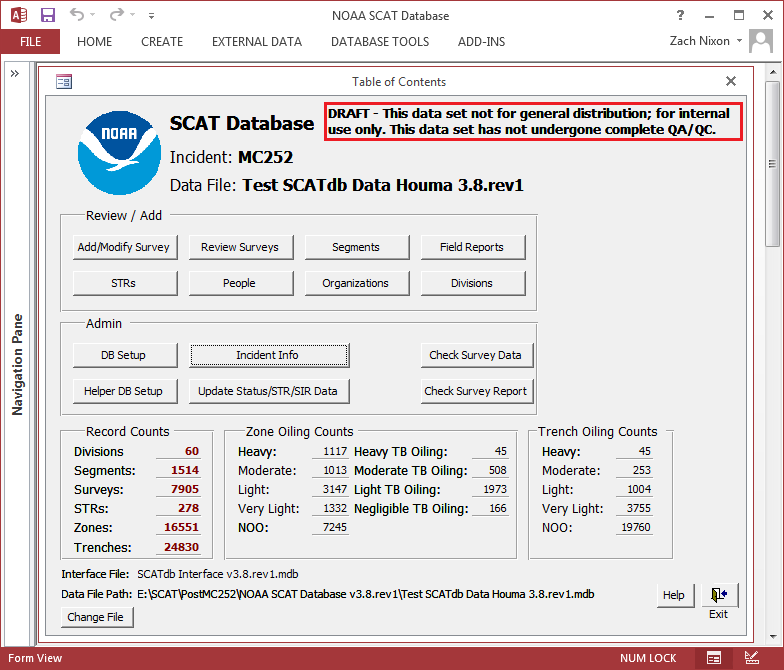
Users should enter the location of the data storage database on their computer or network in the text box, or click on the file browser button to the right of the text box, and navigate to the location of the data storage database. The data storage database selected is stored in the interface database so users will not need to enter the location of this file again unless its name or location have changed.

After users identify the data storage database to link to, the interface dashboard will open. The dashboard is the primary method by which users interact with the forms, tools, and scripts that are stored in the interface database. The interface dashboard is shown below:



**Adding Incident-specific information**

When preparing a blank database for use with a specific incident, users should begin by adding incident specific information to the database. These functions are

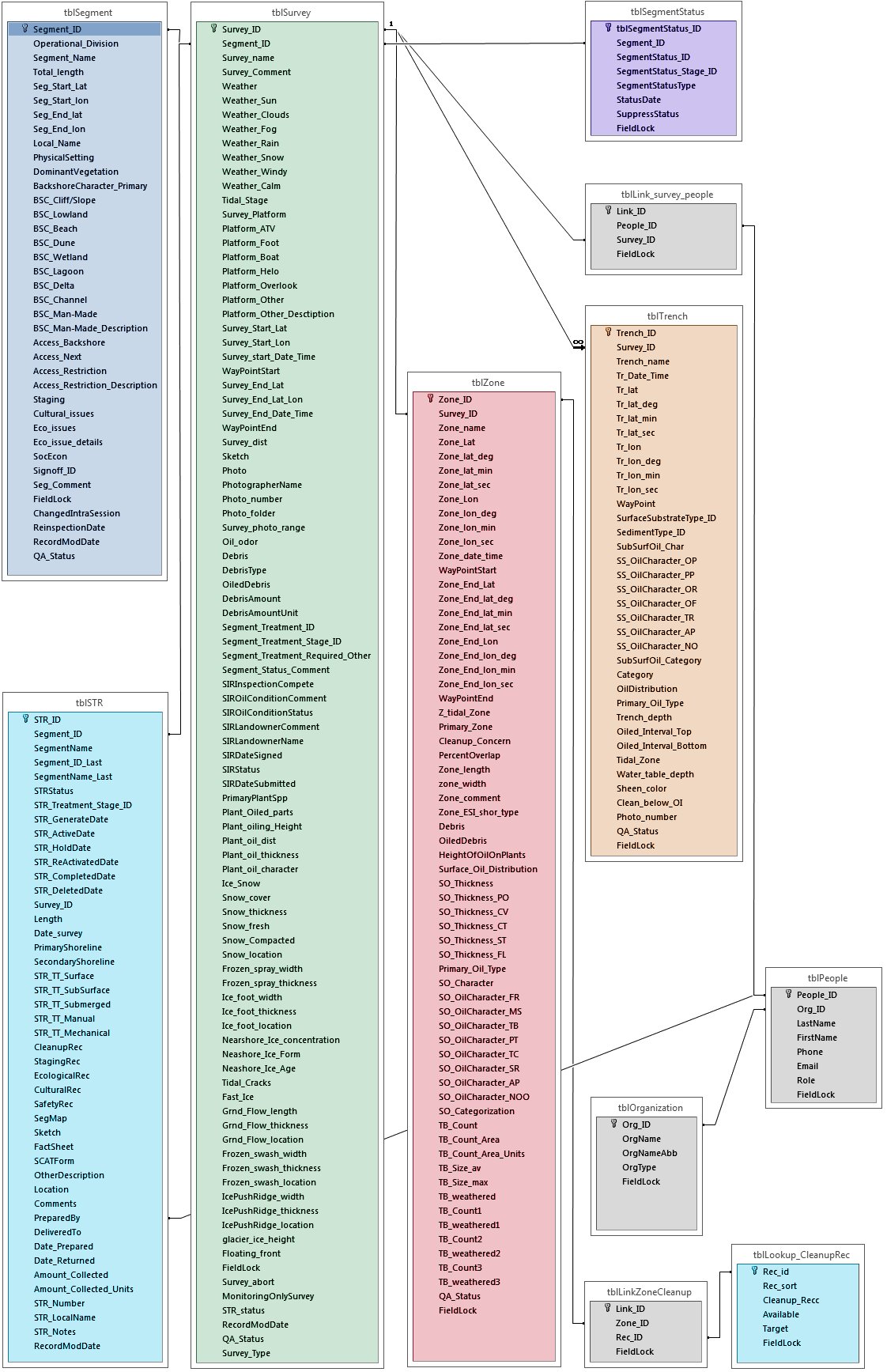


COMPLETE DOCUMENTATION FOR REMAINING ACCESS DB TASKS:

1. Adding incident-specific information
2. Adding shoreline segment and operational division information
3. Adding field survey data
4. Reviewing survey data
5. Adding Shoreline Treatment Recommendation (STR) information
6. Adding people and organizations
7. Creating and importing helper child databases
8. Modifying database setup
9. Checking survey data in database for errors

## Data Structure

Following is a simplified Entity-Relationship (ER) diagram depicting the core tables in the database that store mission critical SCAT and related status data. Note that the data storage .mdb file contains nearly 50 separate tables, so a complete representation of the database structure is not possible in a single ER diagram. Some relationships have been visually simplified to eliminate unnecessary depiction of link tables that contain no mission-critical data.



# ESRI Data Templates

## Overview

Included with these tools are a variety of template ESRI shapefiles intended to be used to store the primary spatial data types. Below is a short description of each of the template.

1. Segments\_CURRENT – Contains all of the SCAT shoreline segments in a spatial format. Links to the database via the SEGNUM field on the shapefile and SEGMENT\_ID from the database.
2. OilZones\_CURRENT – Contains all of the SCAT oil zone information in a spatial format. Links to the database via the ZONESURVID field in both the shapefile and database.
3. SCATGrids\_CURRENT – Contains all of the SCAT grid cells. This shapefile is not applicable to all response events.

## Contents and Requirements

The folder containing these templates includes three shapefiles with predefined attributes. The shapefiles are intended for use with ArcGIS 10.0, 10.1, and 10.2, but could be used with any software package or version that makes use of the ESRI shapefile standard.

# ESRI ArcGIS Tools

## Overview

This toolbox contains a set of tools intended for use in ESRIs ArcGIS software that perform various tasks that are repeated daily. These tools were all developed using Python scripts. Below is a short description of each of the tools within the toolbox.

1. Make Cumulative and Daily Zones. This tool joins OilZones\_CURRENT with AllZonesAsOf\_YYYY-MM-DD.dbf file to generate the cumulative oil zones and daily oil zones shapefiles delivered to NOAA each evening. Note that this tool requires reports generated from the ReportTool mentioned above in order to attach the degree of oiling category for each oil zone from the backend database to the oil zone shapefile.
2. Make Snapping Zones – This tool joins the OilZones\_CURRENT shapefiles with the AllZones\_AsOf\_YYYY-MM-DD.dbf file to generate a oil zone shapefile with the items suitable for running the Snapping Tool mentioned below. Note that this tool requires reports generated from the ReportTool mentioned above.
3. Import Snapping Tool -This tool takes the single shapefile output of the snapping report and generates maximum oiling, most recent oiling, and surveyed (clipped) segment shapefiles. The script also re-orders the features of the most recent oiling shapefile to draw correctly for the current oiling map deliverable. Note that the re-ordering of the features in the shapefile utilizes a executable called, shpsort.exe. This executable is designed by California Department of Fish and Game to run on a Microsoft Windows 32-bit operating system and has not been tested on other environments.
4. Make Status Output – SCCP – This tool utilizes the clipped segment shapefile and an exported query called, ZoneStatusQuery in the backend database to generate the most recent status shapefile. The tool does create some other files that were used for planning purposes during earlier stages of the response, but these files are no longer being used.

## Contents and Requirements

The folder containing these tools includes an ArcGIS toolbox and a set of Python scripts associated with each tool in the toolbox. These tools are intended to be used within ArcGIS 10.0, 10.1, and 10.2. These tools require Python version 2.7 or more recent to operate.

## Usage

ADD USAGE GUIDE FOR ARCGIS TOOLS

## Outputs

ADD OUTPUT GUIDE FOR ARCGIS TOOLS

# NOAA Snapping Tool

## Overview

This tool is a stand-alone executable intended to be run whenever spatial outputs are required that summarize current or historical oiling conditions on a linear shoreline over the course of the incident such as GIS deliverables depicting maximum oiling or current oiling. This typically occurs daily, or whenever additional SCAT survey data is collected.

Many data summary and reporting requirements require summarization of surface oiling results from multiple zones recorded during multiple surveys which may overlap along the same unique length of shoreline. This tool generates outputs for this purpose by snapping the polylines contained in the ESRI shapefile representing oil zones to their parent segment contained in the ESRI shapefile representing shoreline segments. The tool generates ESRI shapefiles containing the spatial results of this operation, as well as calculates the sum lengths covered within specific summary categories.

Only zones which indicate oiling are used in this process. Zones which have an oiling category of NOO, Negligible Tar Balls, or Undefined are not used. Zones are matched to their segment by using the segment name. In the case of duplicate segment names in the segments shapefile, the segment with the shortest length is used. Length is calculated by the python script assuming projection when the earth is a sphere. This is slightly off from the length calculated by ArcGIS using projected coordinate systems. Multi-geometries are allowed. The set of start and stop points is considered when determining if the zone has potential island components.

The snapping executable operates in the following manner:

* The segments are split so that no piece of a segment is over 20 meters long. (In theory, this set should no longer be necessary now that we are inserting the projection points as described below)
* The potential island zones (defined as zones where the start and stop points of the zone are within 20 meters) are broken into 8 pieces and “opened” by 1 meter.
* The start and stop points of the zones are projected onto the matching segment and those points are inserted into the segments. These projections are the closest points on the segment to the zone start and stop points.
* For each oil zone and each start-stop point pair, the closest points are found on the matching segment.
* The keys for the linear portions of the segment between this start-stop point pair are added to a dictionary, and the accumulated dictionary is then read to determine the oiled length.

Known weaknesses of the algorithms used by the tool:

* Zones which are far from their segment. (E.g. Incorrectly assigned segment) may be erroneously handled.
* Zones which have their start-stop points in different sub-geometries of the segment. (E.g. zones do not match their segment well, zones that span open areas of their matching segment) may be erroneously handled.
* Zones that cross the start-stop point of an island segment may be erroneously handled.

## Contents and Requirements

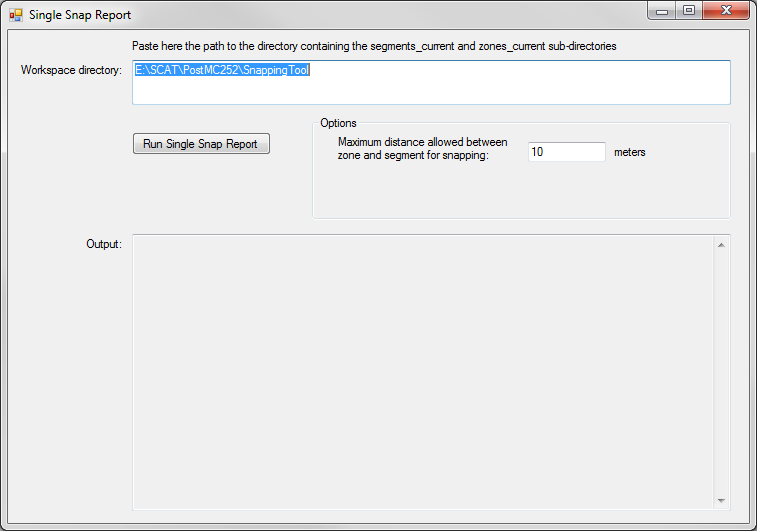
The folder containing this tool includes a single executable file: SingleSnapZones.exe. This tool was developed by NOAA and is designed to run on a Microsoft Windows 32-bit operating system and has not been tested in other environments.

## Usage

The tool snaps all the oil zones in the OilZones\_CURRENT shapefile to the Segments\_CURRENT shapefile in order to help determine the maximum oiling and most recent oiling conditions at each position along the shoreline. The snapping tool outputs four different shapefiles, but the OiledLineSegments\_D.shp is the file generally used by the ArcMap tools mentioned above to generate the most recent oiling and maximum oiling shapefiles.

The tool interface is shown in here. Users should:

* Enter the location of the workspace directory containing the folder structure and files in the text box
* Modify the maximum snapping distance allowed between geometry representing surface oiling zones and shoreline segments (default is 10 meters)
* Click the “Run Single Snap Report” button to begin the process
* Output will be displayed in the “Output” window as the executable runs



## Outputs

ADD OUTPUT GUIDE FOR SNAPPING TOOL

# NOAA Report Generator Tool

## Overview

This tool is a stand-alone executable intended to be run whenever tabular outputs are required that summarize oiling conditions, segment statuses, etc. over the course of the incident such as tabular deliverables reporting segment statuses for all segments. The report generator tool reads the MS Access database and generates a folder of various reports in multiple output formats, including DBF, CSV, MS Excel, and HTML.

The most utilized reports from the report tool at this point are as follows:

* SegmentStatusCurrent\_YYYY-MM-DD.xls – most recent status of a shoreline segment
* SegmentStatusFullHistory\_YYYY-MM-DD.xls – full status history for each shoreline segment.
* DBF/AllZonesAsOf\_YYYY-MM-DD.dbf – list of all relevant oil zone information from the backend database (e.g. degree of oiling category).

## Contents and Requirements

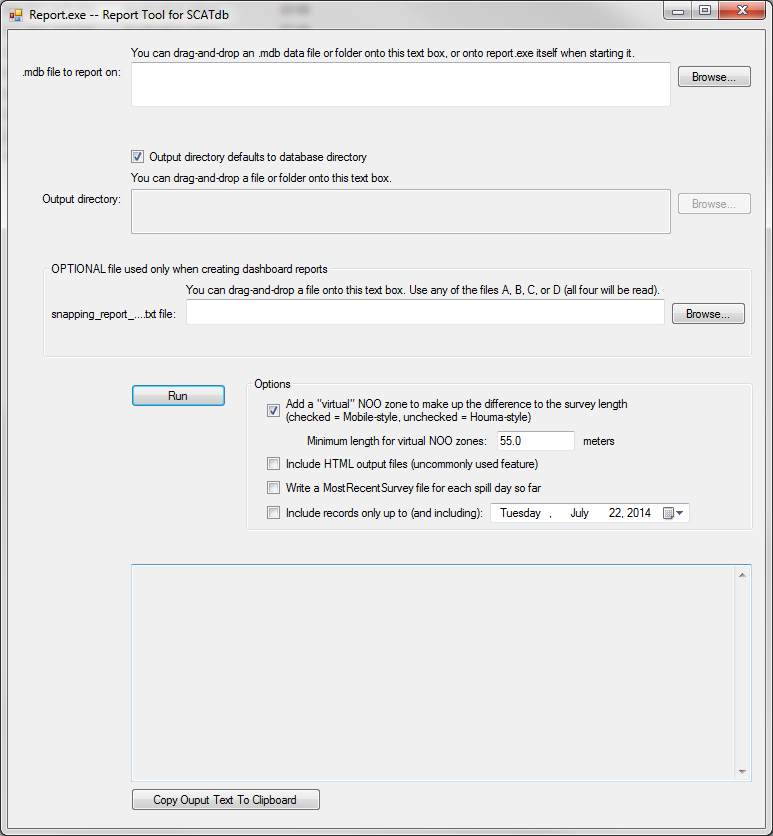
The folder containing this tool includes a single executable file: Report vX.X.exe and multiple additional DLLs packaged with the executable file. These additional files are required to be located in the same folder so that the tool can run. These DLL files are located in the zip archive containing the tool in the proper location within the file structure relative to the executable file.

This tool was developed by NOAA and is designed to run on a Microsoft Windows 32-bit operating system and has not been tested in other environments.

## Usage

The tool interface is shown below. Users should:

* Enter the path to the Microsoft Access data storage database in the text box, drag the file into the text box, or click on the browse button to the right of the text box to specify the location of the data storage database in the file system.
* Specify whether outputs should be located the same directory as the database using the checkbox
* If this checkbox is unchecked, the user should enter the path to the output directory in the text box, drag the directory into the text box, or click on the browse button to the right of the text box to specify the location of the directory in the file system
* If the user wishes specific sub-segment geometries to be accounted for when computing length totals for some of the reports produced, the user may enter the path to the snapping output text file in the text box, drag the text file into the text box, or click on the browse button to the right of the text box to specify the location of the text file in the file system. Note that any of the text files accompanying the ESRI shapefiles output (A”, “B”, “C”, or “D”) by the snapping tool are suitable.



Report.exe is a utility for generating various .csv, .xml, .dbf, and .html reports from a SCATdb Access database.

**Definitions and computations**

* A *zone letter* is the letter part of a zone name [tblZone, field Zone\_name], which is allowed to be two characters to support more than 26 zones. The zone letter of zone “A” is “A”. The zone letter of zones “A1”, “A2”, and “A3” is “A”. The zone letter of zones “AA” and “AA1” is “AA”. The zone letter of the zone “” (empty string) is “” (empty string). The letters “S” and “V” are treated specially when they occur as the non-initial letter because they are used to represent “sediment” and “vegetation”. The zone letter of zones “AS”, “AS1”, “A1S”, “AV”, “AV1”, and “A1V” is “A”. The zone letter of zones “S” and “S1” is “S”. The zone letter of zones “V” and “V1” is “V”.
* A *sub-zone number* is the numeric part of a zone name [tblZone, field Zone\_name]. The sub-zone number of “A1”, AB1”, “AS1”, and “A1S” is “1”. The sub-zone number of “A123” is 123.
* The set of *sub-zones* of a zone is the set of zones that share the zone’s zone letter. “A1”, “A2”, and “A3” are the sub-zones of zone “A”. A *non-atomic zone* is a zone that is comprised of sub-zones. An *atomic zone* is a zone that is not comprised of sub-zones.
* The *length of a survey* is the length specified in that survey’s record [tblSurvey, field Survey\_dist]. But if this length exceeds the segment length [tblSegment, field Total\_length] and the segment length is non-zero, the length of the survey is taken to be the segment length.
* The *length of an atomic zone* is the length specified in the zone’s record [tbl\_Zone, field Zone\_length]. However, if that length is zero, the length of the zone is taken to be the survey length as defined above.
* The *length of a non-atomic zone* (a zone comprised of sub-zones) is the sum of the lengths of its sub-zones, minus the overlap for each sub-zone. That is, for each sub-zone we multiply the sub-zone’s length by the percent overlap [tbl\_Zone, field PercentOverlap] and then sum the results.
* The *oiling categorization of an atomic-zone or a sub-zone* is the value (“Heavy”, “Moderate”, “NOO”, etc.) as computed by the database code described in a separate section.
* The *oiling categorization of a non-atomic zone* is the maximum oiling categorization of its subzones, regardless of the lengths or overlap percentages of the sub-zones.
* The *primary oiling categorization of a survey* is the oiling categorization of its primary zone [tbl\_Zone, field Primary\_Zone].
* The *table of oiling categorizations for a survey* is a mapping from oiling categorizations to lengths. To make this table, we keep a running length total and consider these categorization types in order from highest (Heavy) on down. For each categorization, we consider the zones (atomic zones and non-atomic zones) that have that categorization (using the *oiling categorization of a non-atomic zone* as defined above if the zone is a non-atomic zone). For each such zone, we take the length of the zone (using the *length of a non-atomic zone* as defined above if the zone is a non-atomic zone) and add that category’s row in the table. If, as we loop through the categories in this way, the total of all the columns (not counting the NOO column) ever exceeds the segment length, we stop the loop at that point. Thus we include the heavier categories first, and cut off the lighter categories in the case that the zone lengths total more than the survey length (which should not happen once all the data has been QA’d). Finally, we either do or do not add a “virtual” NOO zone to bring the sum of the lengths in the table up to the length of the survey. In Mobile, the rule is that if the zone lengths do not total the survey length, this means the difference implicitly represents an NOO zone. That is, instead of recording a final NOO zone, the surveyors and SCATdb data entry person can leave it off knowing that it will be included implicitly at report time. This includes the case where there are no zones in the survey at all, with the implicit meaning that the entire survey length was NOO. In Houma, things are done differently, and a case where the zone lengths don’t add up to the survey length would be a problem with the data, and not something we want to try to correct automatically with the virtual NOO. Thus, there is a check box on the report.exe interface where you can choose to use the virtual NOO zones (Mobile) or not (Houma).
* When asking what survey of a segment is the one with the worst-case scenario, that is, the survey where the most oiling was found, we have two methods that can be used:
  + The *single-highest-oiling-zone method* finds the survey that contains the single highest oiling zone among all the surveys of the segment, where the *single highest oiling zone* is the zone or sub-zone (for this purpose we treat all zone and sub-zones as siblings and don’t use the concept of a non-atomic zone) whose oiling categorization is the highest. In the case of two zones or sub-zones that tie for the highest categorization, the length of the zone or sub-zone is used (in this particular case, overlaps are *not* considered when taking the length of subzones). In the case of a length tie, the zone from the more recent survey [tblSurvey, field Survey\_start\_Date\_Time] wins.
  + The *longest-oiling-length method* takes the table of oiling categorizations for the survey and sums the length column, not including the NOO row.
* The *majority ESI type* of a survey is the ESI type that appears most commonly among the ESI types of the zones [tbl\_Zone, field Zone\_ESI\_shor\_type] of that survey, for those zones that have a non-empty ESI type. This does not count sub-zones repeatedly, so if zones “A1” and “A2” both have ESI type 10A, they only count as one “vote” for type 10A.
* The *primary ESI type* of a survey is the ESI type of the zone that is marked as the primary zone.

## Output

This tool outputs a large number of reports in multiple formats. These are detailed below.

1. **BigFiveSummary\_DATE.xls** This report is a single table with one line per county/parish, giving five values for each:

* (A) Total Shoreline Surveyed - This is the sum of the survey lengths for all surveys of the county/parish.
* (B) Currently Oiled - This finds the *table of oiling categorizations* for each most-recent [based on tblSurvey, field Survey\_start\_Date\_Time] survey of each segment in the county/parish and totals their combined lengths, not counting the NOO lines.
* (C) Currently No Oil Observed - This is a simple subtraction of (A) minus (B). Note that this is not necessarily the same as summing the NOO lines from the *table of oiling categorizations* for each survey of the county/parish, because the oiling lengths plus the NOO lengths do not necessarily add to the survey length, either because of a data problem or because of differences in how virtual NOO zones are treated. In particular, note that this (C) value is essentially behaving as if virtual NOO zones had been added, since it assumes that anything that any length that wasn’t oiled can be considered NOO.
* (D) Total Shoreline Oiled - This is the total mileage of shoreline that has ever seen oil (e.g., if the segment was 1000 meters and survey 1 saw oiling over the first 500 meters but none over the second 500 meters, and survey 2 saw oiling over the second 500 meters but none over the first 500 meters, we would say that all 1000 meters had seen oil, and we take the sum of such computations across all segments in the county for this column. This number is currently computed by the special-purpose “snapping” code, which takes all the zones from all the surveys and “snaps” them to their segments (so that overlaps can be disregarded) and then finds the total shoreline length covered.
* (E) SCAT Assessed to Date - This is the total of the lengths of all surveys of segments in the county/parish.

1. **SurveySummaryForCoastGuard\_DATE.**xls. This report is a single table with one line per county/parish, giving three values for each:

* Current Oiled (Miles) - For each segment in the parish/county, this finds the most recent survey of that segment [based on tblSurvey, field Survey\_start\_Date\_Time], computes the *table of oiling categorizations* for each such survey, and then totals the length columns, not counting the NOO column.
* All Oiled (Max) (Unique Miles) - This is similar to Current Oiled (Miles), except instead of the most recent survey of each segment, we use the survey of the segment with the most oil found, according to the *longest-oiling-length method*.
* Assessed To Date (Unique Miles) - For each segment in the parish/county, this finds the most recent survey of that segment, and then totals the survey lengths of all such surveys.

1. **SOBySeg\_MostRecentSurvey\_DATE.xls** - This report has several tabs, which are all variations on a theme. All of the data in this file is based on the most recent survey of a segment [based on tblSurvey, field Survey\_start\_Date\_Time]. Each tab contains tables that break down the lengths surveyed into the various oiling categories (“Heavy”, “Moderate”, “NOO”, etc.). The main tab is named “ShorelineOilingBySegment” and contains several tables.

* The first table includes one row per segment, giving the date of the most recent survey of the segment, the category breakdown of the surveyed lengths, and the majority ESI type of the survey. This table also includes a Segment Length column, which is the length of the segment and not necessarily the length of the survey (e.g., if a survey only covered part of a segment). This is in contrast to the remaining tables on the tab, which include the *survey length*, not the segment length.
* The second table gives the same data, but instead of listing each segment separately it gives a “rolled up” total for each county/parish.
* The third table gives the same data, but instead of listing each segment separately it gives a “rolled up” total for each of three high-level “habitat” categories based on the majority ESI types of the surveys.
* The fourth table gives the same data, but instead of listing each segment separately it gives a “rolled up” total for each operational division.
* The second, third, and fourth tables also include a Currently Oiled column, which is just a sum of the oiling columns.

1. **DailyTeamCoverage\_DATE.xls** - This report has one row per day since the start of SCAT operations. For each day it gives the number of unique teams that did surveys on that day, and several statistics about how much surveying those teams did and what they found. This includes:

* the total miles surveyed and average miles surveyed by a team
* the total and average miles when restricted to certain habitat types
* the total and average miles when restricted to oiled or un-oiled shoreline
* the total number of segments and average number of segments surveyed that day
* the miles of various oiling categories found that day
* a flag if there were no surveys that day due to weather

1. **SegmentStatusCurrent\_DATE.xls** - This report is a single table that gives the most recent status of each segment, where the status is the SIR/STR status of the most recent survey [based on tblSurvey, field Survey\_start\_Date\_Time], or the status of the most recent status record of the segment[tblSegmentStatus, fields SegmentStatusID and StatusDate], in the case the status record is more recent than the survey. It also gives the primary oiling categorization of the most recent survey.
2. **SegmentStatusFullHistory\_DATE.xls** - This report is a single table with the rows grouped by segment. For each segment it contains an entry for each survey of the segment, and each segment status record for the segment, interleaved and sorted by date. For a survey it gives the date of the survey, the SIR/STR status result of the survey [tblSurvey, fields SIR\_Treatment\_Required and SIR\_Treatment\_Required\_Other], and the primary oiling categorization of the survey. For a segment status record it gives the status.

ADD ADDITIONAL OUTPUT GUIDES

# Intended Workflow

This section describes in general the typical flow of operations for day-to-day operational use of these tools for a specific incident.

COMPLETE SECTION

# Frequently Asked Questions

ADD FAQ SECTION