

Methods Used to Create Wood-Pawcatuck Watershed Association Database and Reports

Elise Torello, elise.torello@gmail.com

1. Introduction

The Wood-Pawcatuck Watershed Association (WPWA) has over 20 years of valuable water quality monitoring data collected from sites throughout the 300+ square mile watershed. Water samples have been collected over the years through a volunteer sampling effort run by the University of Rhode Island Watershed Watch (URI WW) program. URI WW also analyzed the water samples. The data currently in the WPWA database span the years 1988 through 2009 and will continue to expand as new data are collected yearly. Water monitoring data for the Coastal (West) and Saugatucket watersheds are stored in the database with WPWA's data for geographical clarity and continuity.

In the past, all of the WPWA data had been stored in many MS Excel spreadsheets in various formats. The goal of the project described in this document was to: 1) compile all of the 20+ years of data into a MS Access database; 2) design simple, clear PDF reports showing multi-year tables and plots of the data for use by scientists and the public; 3) automate the production of these reports for easy yearly update; 4) create a Google Map showing all of the 150+ current and legacy WPWA sampling stations as placemarks on the map; 5) link the data reports to the Google Map placemarks via sampling station "portal" pages; and 6) achieve all of the previous goals in a way that makes maintenance and updates in the future as easy and painless as possible.

Having the WPWA data in an Access database has several advantages: the data are stored in one place in a consistent format; data can be readily sorted, sub-setted, reformatted, joined, plotted, summarized, updated, and reported; production of yearly updated reports can be automated to a large extent; data can be output back to Excel spreadsheets or other formats if desired. Linking the PDF data reports to sampling site markers in a Google map also has several advantages: adding spatial context to sampling data makes it easier to use and visualize; having an easy-to-use way for the public and scientists to access the data can help increase visibility and support for WPWA; people can use the aerial imagery in Google Maps to view the sampling sites and data relative to where they live and other surface features; many people are already familiar with using Google Maps and like using the interface; Google Maps is powerful, easy to use, fun, and free!

This document describes the journey from goal (1) through goal (5), and will be updated as the WPWA database evolves.

2. Creating the MS Access Database

This section describes the steps taken to create the WPWA MS Access database. Table 1 contains a brief glossary of terms for those unacquainted with database jargon.

Term	Definition
Table	Object defined for storing data
Query	Object used for browsing, sorting, filtering, or joining data, producing a <i>recordset</i>
Recordset	Results from a query—users can treat them like a table, but they do not permanently exist in the database and are generated on the fly
Record	Also called row—all of the data related to a specific subject within a table, such as total phosphorus data for a particular sample
Field	Also called attribute or column, a specific piece of data within a row such as site number, concentration, etc.
Join	Linking tables or queries via their related data values (usually primary keys), such as site number and date
Primary Key	One or more fields in a table which, together, uniquely identify a data record
Form	Object designed mostly for data input or display, or to control application execution
Report	Object designed for formatting, printing, summarizing selected data
SQL	Structured Query Language—language used to write a query (needed in UNION queries)
Macro	Object defining actions to perform in response to a defined event (like clicking a button on a form)
Visual Basic	Used to write functions or subroutines to perform custom procedures in the database
Function, subroutine	Visual Basic code that calculates and returns a value (function) or performs other actions (subroutine). All data necessary to run the code are passed in as arguments.
Argument	Information provided to a Visual Basic function or subroutine for use by (within) the function or subroutine

Table 1. Database and programming terms used in this document.

2.1. Gather Data and Create the Data Tables

Microsoft Excel data files containing all of the WPWA data were gathered from Denise Poyer (WPWA), Linda Green, and Elizabeth Herron (both URI Watershed Watch). These data were in a variety of formats and consisted of “raw” field and laboratory data as well as yearly data summaries. For each year and sampling parameter, data were reformatted in Excel to allow them to be loaded into Access data tables. *ParamNum* (parameter number, Table 2), *SiteNum* (sampling site number), and *SamplingYear* fields were added as primary keys that uniquely identify each data record. *SiteNums* are always four digits. Sampling sites with both shallow and deep samples have two *SiteNums*: the first three digits are the same, but shallow samples are identified with a *SiteNum* ending in 0 while deep samples are identified with a *SiteNum* ending in 1.

Once all of the data for a sampling parameter were ready, they were imported into a temporary Access table (one table per sampling parameter). The format of these tables closely resembled the original Excel spreadsheets in a “horizontal,” or “flat,” format. That is, for each *ParamNum*, *SiteNum* and *SamplingYear* there is a *May*, *June*, *July*, *August*, *September*, and *October* field (Figure 1). No calculated fields (means, for example) were imported.

ParamNum	Parameter	Units
1	Chlorophyll a (digital)	ug/L
2	Dissolved Oxygen	mg/L
3	Secchi Depth	M
4	<i>Nitrogen, Total</i>	<i>ug/L</i>
5	Phosphorus, Total	ug/L
6	<i>Phosphorus, Dissolved</i>	<i>ug/L (ppb)</i>
7	<i>Chloride</i>	<i>mg/L</i>
8	<i>Nitrogen, Ammonia Dissolved</i>	<i>ug/L</i>
9	<i>Nitrate + Nitrite, Dissolved</i>	<i>ug/L (ppb)</i>
10	<i>Ph</i>	<i>S.U.</i>
11	<i>Alkalinity</i>	<i>mg/L</i>
<u>12</u>	<u>Bacteria – Enterococci</u>	<u>MPN/100 mL</u>
<u>13</u>	<u>Bacteria - Fecal coliform</u>	<u>CFU</u>
<u>14</u>	<u>Bacteria - E. coli</u>	<u>CFU</u>
15	<i>Calcium</i>	<i>mg/L</i>
16	<i>Magnesium</i>	<i>mg/L</i>
17	<i>Sodium</i>	<i>mg/L</i>

Table 2. *ParamNum* values for parameters sampled by WPWA. Parameters in bold, italic font are the “chemistry” parameters and are all processed the same way. The three bacteria parameters (underlined) are all processed the same way. Secchi depth, chlorophyll a, and total phosphorus are processed separately. There are no dissolved oxygen data currently in the database.

WPWA	Samp	Original Station Name	Sample	Param	May	June	July	August	September	October
1010	1988	100 Acre		8	15		170			110
1010	1997	Hundred Acre Pond		8			60			130
1010	1998	HUNDRED ACRE POND		8			270			130
1010	1999	Hundred Acre Pond		8	230		50			90
1010	2000	HUNDRED ACRE POND		8	170		65			50
1010	2001	Hundred Acre Pond		8			15			15
1010	2002	Hundred Acre Pond		8	190					
1010	2003	Hundred Acre Pond		8		80				
1010	2004	Hundred Acre Pond		8	120					50
1010	2005	Hundred Acre Pond	1	8	15		15			290
1010	2006	Hundred Acre Pond	1	8	15					260
1010	2007	HUNDRED ACRE POND	1	8	40		15			90
1010	2008	Hundred Acre Pond	1	8		40	10	20		190

Figure 1. Example of “flat” format table.

2.2. Create “Vertical” MS Access Data Tables

The Secchi disk and chlorophyll a data were already aggregated by season rather than being stored by month; therefore, they are permanently stored in their original “flat” format tables (named *Secchi* and *Chlorophyll*, respectively). However, the rest of the data tables were reformatted and final “vertical-format” Access data tables were created. A “vertical” format of storing data is preferable to the “flat” format for several reasons. A vertical arrangement is more efficient since the horizontal format has a very large number of empty fields for months in which a sample was not collected for a *SiteNum* and

SamplingYear (the field exists whether or not there is a data value in it—note the *June*, *August*, and *September* columns in Figure 1). These empty fields are eliminated in the vertical format, since only *ParamNum/ SiteNum/ SamplingYear/ SamplingMonth* combinations with a data value are included. In addition, the most effective way to create report pages containing calculated values (means, etc.), data tables, and charts in Access is using crosstab queries (crosstab queries are discussed further later on). However, to create a crosstab query, the data need to be in a vertical format. Therefore, it was desirable to create new “vertical” tables from the “horizontal” tables for consistency amongst parameters and efficiency/accuracy in data storage and reporting.

The vertical tables were created from the flat tables using an append query called *MakeVertical*. An append query selects data from one table or recordset and appends them to the end of another table. The query was run six times per flat table, each time extracting a different month of data, assigning the appropriate *SamplingMonth* value (5-10), and appending the resultant recordset to the vertical table. The resulting vertical table for each sampling parameter had a data record for each *ParamNum*, *SiteNum*, *SamplingYear*, and *SamplingMonth* (these four fields collectively creating the primary key), plus the original station name for QA/QC, a sample depth, and a data value (Figure 2). Once the vertical tables were created, generating crosstab queries with summary data and pivotcharts of the data for inclusion in reports was much more straightforward, and thus less prone to error.

WPWA Site	Sampling Year	Sampling Month	Parameter N	Original Station Name	Sample Dep	Concentration
1010	2004	5	8	Hundred Acre Pond		120
1010	2004	10	8	Hundred Acre Pond		50
1010	2005	5	8	Hundred Acre Pond	1	15
1010	2005	7	8	Hundred Acre Pond	1	15
1010	2005	10	8	Hundred Acre Pond	1	290
1010	2006	5	8	Hundred Acre Pond	1	15
1010	2006	10	8	Hundred Acre Pond	1	260
1010	2007	5	8	HUNDRED ACRE POND	1	40
1010	2007	7	8	HUNDRED ACRE POND	1	15
1010	2007	10	8	HUNDRED ACRE POND	1	90
1010	2008	6	8	Hundred Acre Pond	1	40
1010	2008	7	8	Hundred Acre Pond	1	10
1010	2008	8	8	Hundred Acre Pond	1	30
1010	2008	10	8	Hundred Acre Pond	1	190
1011	1998	7	8	Hundred Acre Pond		260
1011	1999	7	8	Hundred Acre Pond		10
1011	1999	10	8	Hundred Acre Pond		80
1011	2001	7	8	Hundred Acre Pond		160

Figure 2. Example of a “vertical” table.

Once all of the vertical data tables were created and quality-checked, the tables for individual chemistry parameters were concatenated into one table for all of the chemistry parameter data. This table shares the same structure as the tables for the individual parameters and is named *TBL_Chemistry_FINAL* (Figure 3). Storing all of the data with the same structure in the same data table is better database design, and allows the data reports for all of the parameters in this table to be generated using the same set of queries, forms, and reports.

Site	Year	Month	Parameter	Original Station Name	Depth	Concentration
1011	2008	10	6	Hundred Acre Pond	6	11
1011	2008	10	7	Hundred Acre Pond	6	24
1011	2008	10	8	Hundred Acre Pond	6	300
1011	2008	10	9	Hundred Acre Pond	6	140
1011	2008	10	10	Hundred Acre Pond	6	6.49
1030	1991	5	5	Chipuxet R @Yawgoo Mil		13
1030	1991	5	6	Chipuxet R @Yawgoo Mil		2
1030	1991	5	7	Chipuxet R @Yawgoo Mil		11
1030	1991	5	9	Chipuxet R @Yawgoo Mil		390

Figure 3. Final table containing chemistry data: *TBL_Chemistry_FINAL*.

Similarly, the three vertical bacteria data tables were concatenated into one table named *TBL_Bacteria_Vertical_NUMERIC*. The bacteria data are stored separately from the chemistry data due to the different format of the data. Bacteria concentrations often contain special characters such as "<", ">", "*", "~", and "TNTC", which made it necessary to initially store the bacteria concentration data as text fields. Unfortunately, this made plotting of the data impossible. To allow plotting, the data were processed to create a numeric bacteria concentration field in the table *in addition to* the original text-format field (as documentation for the numeric data values). Further, since numeric values ranged from below detection ("<1") to 4800+ or "TNTC" (too numerous to count), the decision was made to set a maximum concentration for plotting (display) purposes only. The maximum plotted concentration was decided to be twice the EPA or RI DEM limit for recreational contact (Denise Poyer, Program Director of WPWA, personal communication). The maximum plotted value and EPA/DEM limit are prominently displayed in bold, red letters in the data report above the plot. Placing an upper limit on the data in the plots made the bacteria plots much more readable and informative. Other special characters in the data were handled for plotting purposes as follows:

- "<1", "<10", etc: Plotted as ½ the numeric part of the field. This is how other "below detection" values were handled in the database.
- "Dry" or "Lake dry": Omitted since no sample was collected.
- "~1200" etc: Plotted using the numeric part of the field as the data value.
- Two values separated by "," or "/": The mean of the two values was plotted.

The original concentrations, including text and special characters, are reported in tabular format on the bacteria reports along with the plots.

The queries and temporary data tables used in generating the final vertical Access data tables were archived in a separate database (*WPWA_Sampling_Data-ARCHIVE.accdb*). Keeping multiple copies of data is not generally a good idea, as updates made to data in one table will not be reflected in the other table(s); however, I was not comfortable deleting the raw data tables. Currently, only the active, final data tables are in the master WPWA database (*WPWA_Sampling_Data-MASTER.accdb*)--these are the tables with which all future data work should be done.

2.3. Additional MS Access Data Tables

The overall list of data tables in WPWA's database is in Table 3. The data tables (*Chlorophyll*, *Secchi*, *TBL_Bacteria_Vertical_NUMERIC*, and *TBL_Chemistry_FINAL*) have already been described above. Two

tables, *MasterSiteList* and *Watershed_Watch_Sites*, contain important information about the sampling sites and are used in generating the data and sampling site reports (discussed later). Two more tables, *Param_Metadata* and *Param_Yearly_Metadata*, contain information about the sampling parameters in the database. Information in these two tables is incorporated into the data reports and will also be discussed later.

Four tables, *TBL_All_Data_Files_For_Site_Pages*, *TBL_One_Bacteria_For_Report*, *TBL_One_Bacteria_For_Report_Thresholds*, and *TBL_One_Param_For_Report*, exist in an empty state except when they are being used during active report generation. Some of the queries used during report production are very slow-running but would need to execute for each report page. To speed up report output, these queries are run once and the results stored in tables for use during report generation. Thus, a temporary table is filled by a query, the reports are generated, and then the table is emptied by another query when the reports are complete. These temporary tables will be described in more detail later in this paper.

The last table used in report generation is *TBL_Thresholds*, which contains important threshold values for some parameters. These may be values that are transition points for trophic status (oligotrophic, mesotrophic, and eutrophic), US Environmental Protection Agency (US EPA) recreational contact limits for bacteria, Rhode Island Department of Environmental Management (RI DEM) limits for bacteria, etc. The values in this table are appended to the actual sampling data during report production in order to display threshold lines on the data plots. This was a tricky process and is described in detail later.

MS Access Table	Description
<i>Chlorophyll</i>	Chlorophyll a data (flat format).
<i>Secchi</i>	Secchi disk data (flat format).
<i>TBL_Bacteria_Vertical_NUMERIC</i>	Bacteria (E. coli, Enterococci, and fecal coliform) final data
<i>TBL_Chemistry_FINAL</i>	Chemistry (nitrogen, phosphorus, pH/alkalinity, etc) data
<i>MasterSiteList</i>	Master list of all sampling sites and their attributes, including other adjacent watersheds.
<i>Watershed_Watch_Sites</i>	List of sampling sites provided by URI Watershed Watch. This list is not complete relative to <i>MasterSiteList</i> .
<i>Param_Metadata</i>	Metadata related to sampling parameters including units, descriptions, links to further info, etc.
<i>Param_Yearly_Metadata</i>	Year-specific metadata, such as detection limits, etc.
<i>TBL_All_Data_Files_For_Site_Pages</i>	Temporary table filled for use during sampling site report generation; otherwise, is empty.
<i>TBL_One_Bacteria_For_Report</i> , <i>TBL_One_Bacteria_For_Report_Thresholds</i>	Temporary tables filled for use during bacteria data report generation; otherwise, are empty.
<i>TBL_One_Param_For_Report</i>	Temporary table filled for use during chemistry data report generation; otherwise, is empty.
<i>Site_LatLon</i>	Contains original latitude/longitude data for sampling sites. These lat/longs may not be accurate and are not used!
<i>TBL_Thresholds</i>	Contains important data threshold values (for example, trophic status, EPA/DEM limits) for some parameters

Table 3. Tables in the final WPPWA database.

3. Creating the Data Reports

There were many steps involved in creating the WPGA data reports—they are described in this section. In order to make the generation of reports easy and automated for the end user of the database, some complicated database work and programming was necessary behind the scene. I have done my best to describe the nasty guts of the system as clearly as possible, but I invite the reader to please email me (elise.torello@gmail.com) with any questions.

3.1. Selecting Which Reports to Generate

Chlorophyll a, Secchi disk, chemistry, bacteria, and sampling site data were in different formats, and thus their reports were generated using distinct (but similar) procedures which are described later. Regardless of whether the same behind-the-scenes database procedures are used for multiple sampling parameters, reports for each sampling parameter are generated separately for end-user clarity.

The first step in creating data reports is to launch the form designed for this purpose, *FRM_Report_Options* (Figure 4). Choosing a report option by clicking a radio button, then clicking the “Click Here to Generate Reports” button, activates a macro named *DoReport.ReportItA* (see the “On Click” field in the Property Sheet). The *DoReport.ReportItA* macro performs one action-- “OpenForm”, which launches a second form named *FRM_Report_Site_Options* (Figure 5).

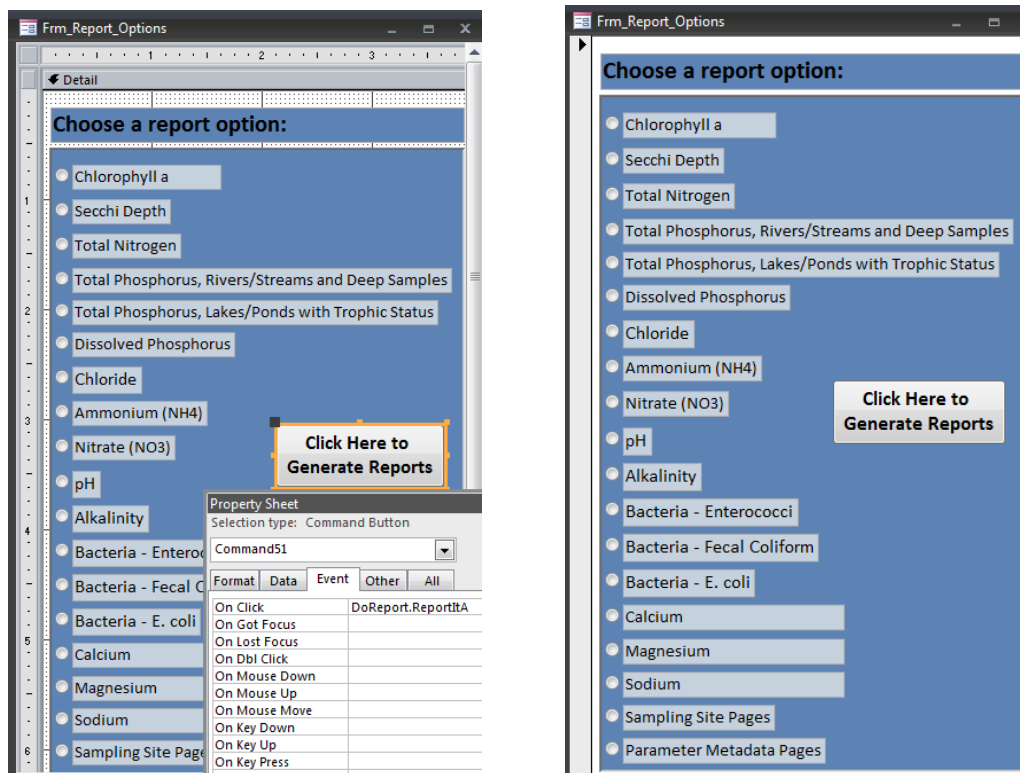


Figure 4. Part of the form (*FRM_Report_Options*) in design view (left) and form view (right). Note the “On Click” field in the Property Sheet which activates the macro named *DoReport.ReportItA* when the “Click Here to Generate Reports” button (with yellow border) is clicked.

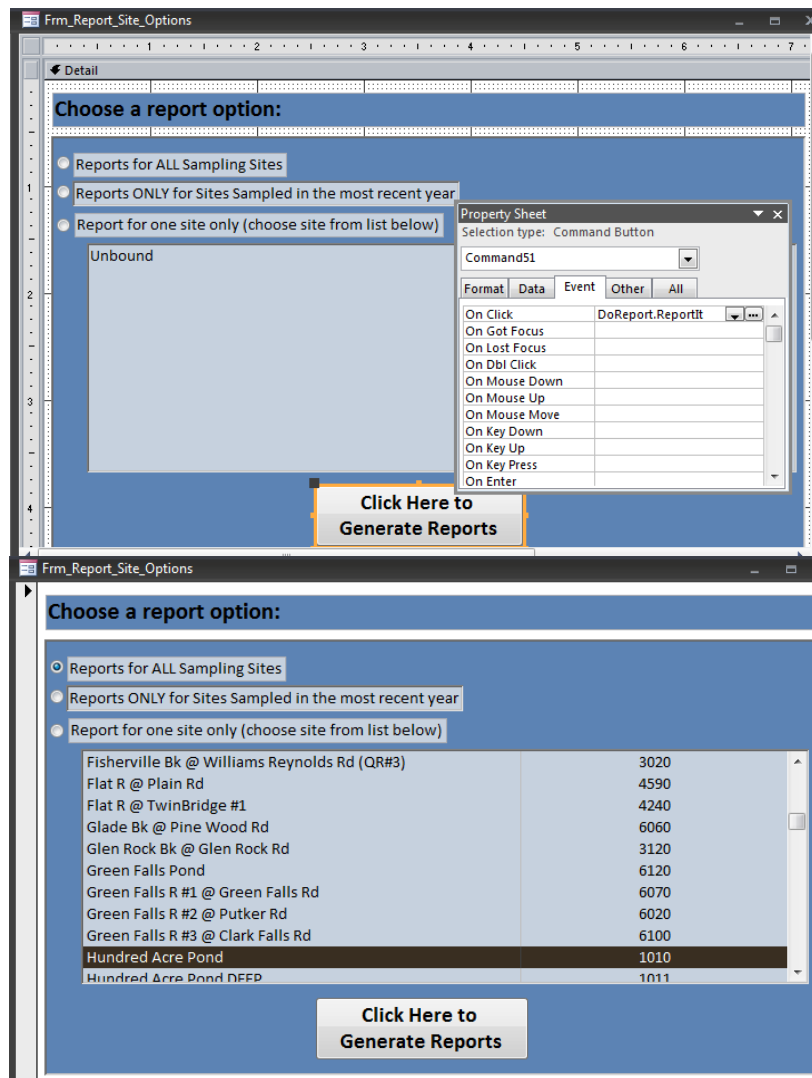


Figure 5. Form *Frm_Report_Site_Options* in design view and form view. Note the action performed in the “On Click” field in the Property Sheet for the button.

The purpose of this second form is to allow the user to either generate the chosen parameter reports for ALL sampling sites, for sites sampled in the latest sampling year, or for one site only. Since the number of active sampling sites is about half of the number of total sites, the “latest year” option would be the preferred option after adding a new year of data to the database. However, if the report format for a parameter has been revised, it would be necessary to regenerate reports for all sites for consistency.

To produce reports for a single site, a list box named *SiteList* was created in the form with a query named *QRY_One_Site_Form* as its Row Source (Figure 6). The *QRY_One_Site_Form* query produces a simple list of all possible sampling sites using the *MasterSiteList* table. In the list box, the “Bound Column” is 2; this is the column containing the *SiteNum* (column 1 contains the site name). The selected *SiteNum* is accessed from the *ReportIt* macro using the following syntax: `[Forms]![Frm_Report_Site_Options]![SiteList]`. This value is passed into the report-generating functions

and is used by the functions when the third report option (one site only) is selected. Otherwise, the default value (1010) is ignored by the report-generating functions.

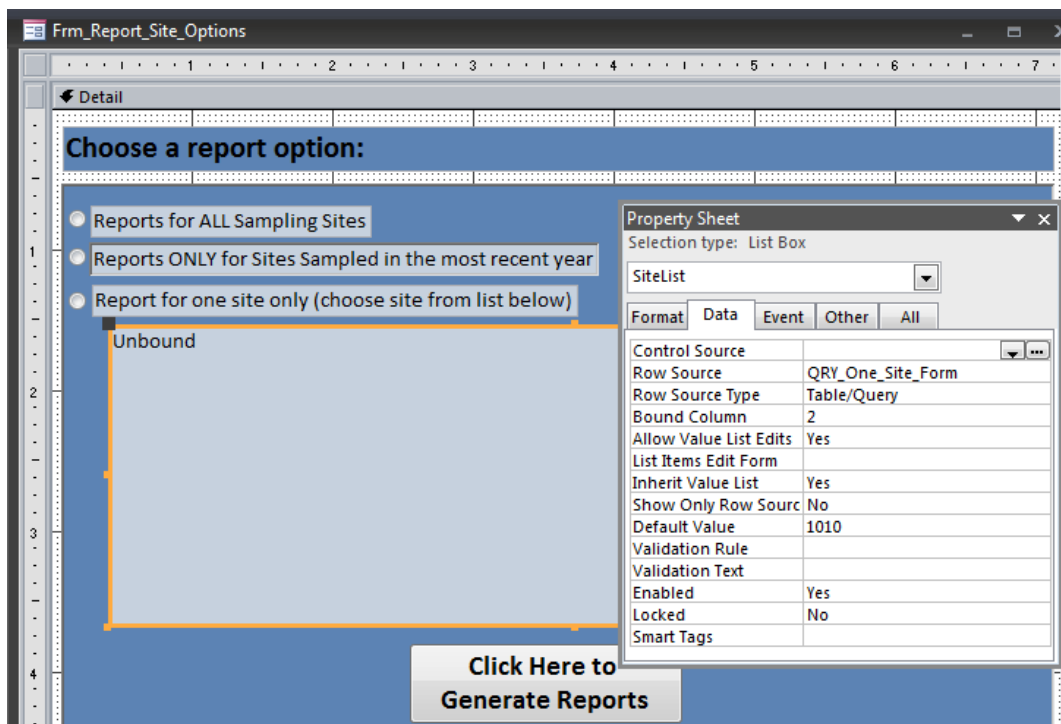


Figure 6. List box for choosing a sampling site for report generation. Note that the row source is the query *QRY_One_Site_Form*.

When the “Click Here to Generate Reports” button is clicked from *Frm_Report_Site_Options*, a macro named *DoReport.ReportIt* is launched. This macro performs actions based on the numeric values passed to it by the chosen radio buttons in the two forms above (Figure 7). Each radio button from *Frm_Report_Options* passes a unique number usually equal to the chosen parameter’s *ParamNum* to the macro, which calls one of the functions in the Visual Basic (VB) module to generate the reports for that *ParamNum*. The radio button numeric value is accessed from inside the macro using the following (ugly) syntax: `[Forms]![Frm_Report_Options]![Rpt_Opt]`. Similarly, each radio button from *Frm_Report_Site_Options* passes a value to the macro, which is accessed using this syntax: `[Forms]![Frm_Report_Site_Options]![Rpt_Opt]`. Note the name of the form after the first “!”. The combination of these two numbers determines which actions the macro performs, that is, which parameter reports are produced for which sites. Report production for each parameter (or group of related parameters) is described below.

ReportIt	1=[Forms]![Frm_Report_Options]![Rpt_Opt] And 1=[Forms]![Frm_Report_Site_Options]![Rpt_Opt]
...	...
...	...
...	1=[Forms]![Frm_Report_Options]![Rpt_Opt] And 2=[Forms]![Frm_Report_Site_Options]![Rpt_Opt]
...	...
...	...
...	3=[Forms]![Frm_Report_Options]![Rpt_Opt] And 1=[Forms]![Frm_Report_Site_Options]![Rpt_Opt]
...	...
...	...
...	3=[Forms]![Frm_Report_Options]![Rpt_Opt] And 2=[Forms]![Frm_Report_Site_Options]![Rpt_Opt]
...	...
...	...
...	4=[Forms]![Frm_Report_Options]![Rpt_Opt]
...	...
...	...

RunCode	Make_Ch1_Secchi_Reports("RPT_TOP_Chlorophyll", "QRY_Param_Sites_Chlorophyll", "Chlorophyll")
Close	Form, Frm_Report_Site_Options, Prompt
StopMacro	
RunCode	Make_Ch1_Secchi_Reports("RPT_TOP_Chlorophyll", "QRY_Param_Sites_New_Year_Chlorophyll", "Chlorophyll")
Close	Form, Frm_Report_Site_Options, Prompt
StopMacro	
RunCode	Make_Ch1_Secchi_Reports("RPT_TOP_Secchi", "QRY_Param_Sites_Secchi", "Secchi")
Close	Form, Frm_Report_Site_Options, Prompt
StopMacro	
RunCode	Make_Ch1_Secchi_Reports("RPT_TOP_Secchi", "QRY_Param_Sites_New_Year_Secchi", "Secchi")
Close	Form, Frm_Report_Site_Options, Prompt
StopMacro	
RunCode	Generate_Chemistry_Reports([Forms]![Frm_Report_Options]![Rpt_Opt], "Nitrogen", 0, [Forms]![Frm_Report_Site_Options]![Rpt_Opt])
Close	Form, Frm_Report_Site_Options, Prompt
StopMacro	

Figure 7. Part of the macro named *DoReport.ReportIt*, which calls VB functions to generate data reports. The bottom box should be to the right of the top box—space did not allow displaying the entire width of the macro window.

3.2. Generating the Chlorophyll a and Secchi Disk Reports

As was noted before, the Secchi depth and chlorophyll data are stored in different formats than the rest of the sampling data and thus are processed separately for reporting. The first step in generating the reports is to launch the *FRM_Report_Options* form (Figure 4) as described above. If the “Chlorophyll a” or “Secchi Depth” report option is chosen, the *DoReport.ReportIt* macro calls the Visual Basic (VB) function named *Make_Ch1_Secchi_Reports* (Figure 8). This function generates the chlorophyll or Secchi disk report pages and outputs one PDF file per site. The name of the report necessary for generating the PDF report files, the parameter number, the number representing the desired site report option (all=1, latest year=2, or single site=3), the site for which a report will be generated if the site report option is “3” (single site), and the name of the subfolder on the local computer where the reports will be written, are passed into the function as arguments.

```

'-----
' Generate Chlorophyll and Secchi reports. This procedure accepts the report name,
' query providing the station list, and subfolder name where the PDF files will be
' output as arguments.
'-----
Public Function Make_Ch1_Secchi_Reports(ByVal RPT_TOP_Name As String, Param As
Integer, WhichSites As Integer, RptSite As Integer, PathName As String)

' Error handling
On Error GoTo Make_Ch1_Secchi_Reports_Err

' Declare the database, recordset, and report filter string variables

```

```

Dim dbwpwa As DAO.Database
Dim rcdRecQuery As DAO.Recordset
Dim strFilter As String

' Set the current database and recordset variables for use in the do...until loop
Set dbwpwa = CurrentDb

If WhichSites = 3 And RptSite Like "###1" Then
' this is not a surface water site-kick out of function
    MsgBox ("Please select a SURFACE water site (Site Number ending in 0).")
    Exit Function
End If

If Param = 1 Then
    If WhichSites = 2 Then
' generate reports for the latest year only
        Set rcdRecQuery = dbwpwa.OpenRecordset("QRY_Param_Sites_New_Year_Chlorophyll")
    Else
' generate reports for all years
        Set rcdRecQuery = dbwpwa.OpenRecordset("QRY_Param_Sites_Chlorophyll")
    End If
Else
    If WhichSites = 2 Then
        Set rcdRecQuery = dbwpwa.OpenRecordset("QRY_Param_Sites_New_Year_Secchi")
    Else
        Set rcdRecQuery = dbwpwa.OpenRecordset("QRY_Param_Sites_Secchi")
    End If
End If

' Make the mouse an hourglass to indicate a wait
DoCmd.Hourglass True

' Start looping through the recordset opened above to generate a one page report for
' each SiteNum. This loop will process every SiteNum and terminate when the end of
' the file (EOF) is reached.

If WhichSites = 3 Then
' just report for a single site
    rcdRecQuery.FindFirst ("SiteNum=" & RptSite)

    If Not rcdRecQuery.NoMatch Then
' found the record-output the report
        strFilter = "[MstSiteNum]=" & RptSite
        ' Open the report
        DoCmd.OpenReport RPT_TOP_Name, acViewPreview, "", strFilter, acHidden
        ' Output the file
        DoCmd.OutputTo acOutputReport, RPT_TOP_Name, "PDFFormat (*.pdf)",
"C:\Users\Elise Torello\documents\WPWA\Field Data\Master Access Database\WPWASites\" &
PathName & "\" & rcdRecQuery![RptFileName], False, "", 0, acExportQualityPrint
        ' Close the report
        DoCmd.Close acReport, RPT_TOP_Name
    Else
        MsgBox ("No data exist for this site!")
    End If
Else
' loop through the recordset until the end of the file is reached
    Do Until rcdRecQuery.EOF
        ' Set the filter for the report to the current SiteNum
        strFilter = "[MstSiteNum]=" & rcdRecQuery![SiteNum]
        ' Open the report
        DoCmd.OpenReport RPT_TOP_Name, acViewPreview, "", strFilter, acHidden
        ' Output the file
    End If
End If

```

```

        DoCmd.OutputTo acOutputReport, RPT_TOP_Name, "PDFFormat(*.pdf)", "C:\Users\Elise
Torello\documents\WPWA\Field Data\Master Access Database\WPWASites\" & PathName & "\"
& rcdRecQuery![RptFileName], False, "", 0, acExportQualityPrint
        ' Close the report
        DoCmd.Close acReport, RPT_TOP_Name
        ' Move on to the next record in the query
        rcdRecQuery.MoveNext
        'and loop
    Loop
End If

' Turn off the hourglass
DoCmd.Hourglass False

Make_Ch1_Secchi_Reports_Exit:
    Exit Function

Make_Ch1_Secchi_Reports_Err:
    msgbox Error$
    Resume Make_Ch1_Secchi_Reports_Exit

End Function

```

Figure 8. *Make_Ch1_Secchi_Reports* VB function.

The Secchi and chlorophyll data were not converted to vertical format but were kept in their original, flat formats in Access tables named *Chlorophyll* and *Secchi*, respectively. WPWA's original Excel data files for these parameters contained summarized data, including trophic status (TSI) fields. The calculations for TSI evolved over the 20+ years of WPWA sampling, and it was decided after consultation with Elizabeth Herron of URI Watershed Watch that the latest formula should be used for all data. Therefore, TSI is recalculated on-the-fly in queries for all data using VB functions (Figure 9).

For calculated values, especially those that appear in multiple queries, it makes sense to create VB functions or subroutines to perform these calculations for several reasons. If the calculation of a value is complex, it is clearer and more convenient to write the code needed to perform the calculation in the VB programming environment. The VB programming environment provides more flexibility to the programmer, and there are debugging and other tools available to help troubleshoot problems. Also, if a calculated value is generated in more than one query, using a function allows you to create the code to perform this calculation in one place and simply call the function to return the calculated value inside the queries. If there is ever a need to modify the way the value is calculated, the code changes necessary only need to be made in one place. All of the functions and subroutines in a VB module are available to any query within the database.

```

'-----
'This function takes the median Chlorophyll number and generates the Trophic Status Index
'for Chlorophyll for the year.
'-----
Public Function Chl_TSI(Median As Single) As String

If Median < 2.6 Then
'Oligotrophic
Chl_TSI = "O"
Else
If Median < 7.2 Then
'Mesotrophic
Chl_TSI = "M"
Else
If Median <= 35 Then
'Eutrophic
Chl_TSI = "E"
Else
'Hypereutrophic
Chl_TSI = "H"
End If
End If
End If

End Function

```

Figure 9. Simple Visual Basic function for calculating chlorophyll a trophic status. Functions for calculating Secchi depth and total phosphorus trophic status are very similar.

The chlorophyll and Secchi disk data reports contain a data table and a data plot. The table and plot are each incorporated into the main report as subreports. The queries to extract data from the Access tables for use as record sources for both the tabular and PivotChart subreports were *QRY_rpt_Chlorophyll* and *QRY_rpt_Secchi* (Figure 10). The TSI calculations were performed within these queries, plus additional fields were added to allow plotting of TSI threshold values (low chlorophyll or high Secchi depth (oligotrophic), moderate chlorophyll/Secchi depth (mesotrophic), high chlorophyll or low Secchi depth (eutrophic)) along with the data values. The original *TSI* field contained some symbols indicating that data in that record should be excluded for some reason: # = naturally dark water color, * = insufficient data, ** = shallow site, f (phi) = river site. These symbols were imported into Access with the rest of the Secchi and chlorophyll data and used to exclude these records from the reports.

Field:	SiteNum	SamplingYear	ParamNum	Min	Max	Median	TSI	TSI_Calculated: Chl_TSI([Median])	CValue: "Season Maximum"	Concentration: Max Chlorophyll
Table:	Chlorophyll	Chlorophyll	Chlorophyll	Chlorophyll	Chlorophyll	Chlorophyll	Chlorophyll			
Sort:	Ascending	Ascending								
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:							In ("O","o","M","m","E","e")			

Figure 10. *QRY_rpt_Chlorophyll* query. The original *TSI* field is used to exclude inappropriate data from the report; the *TSI_Calculated* field stores the TSI calculated by the VB function *Chl_TSI*. *QRY_rpt_Secchi* is very similar.

The tabular subreports for the chlorophyll and Secchi data are named *RPT_Crosstab_Chlorophyll* and *RPT_Crosstab_Secchi*, respectively (Figure 11). These are very simple reports displaying fields directly from the queries described above.

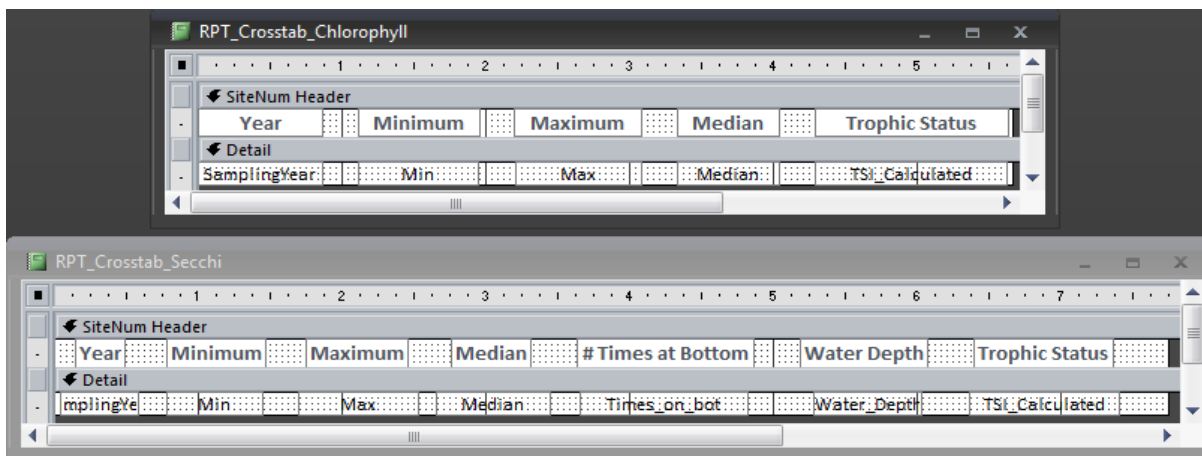


Figure 11. Tabular Chlorophyll and Secchi subreports (*RPT_Crosstab_Chlorophyll* and *RPT_Crosstab_Secchi*) in design view.

The subreports displaying Chlorophyll and Secchi data plots are much more complex. Both plots display TSI threshold lines, and displaying these lines in MS Access was not easy or straightforward. It was necessary to concatenate “dummy” data for each threshold value onto the real data for every sampling year for each site with each type of data. Threshold data for each sampling parameter are stored in the *TBL_Thresholds* table (Figure 12).

TBL_Thresholds				
ParamNum	MonthText	SamplingMonth	Concentration	
1	Low (Oligotrophic)	13	2.6	
1	Moderate (Mesotrophic)	14	7.2	
1	High (Eutrophic)	15	35	
3	Oligotrophic	13	4	
3	Mesotrophic	14	2	
3	Eutrophic	15	0.7	
4	Low	13	320	
4	Moderate	14	710	
4	High	15	1000	
5	Low (Oligotrophic)	13	12	
5	Moderate (Mesotrophic)	14	24	
5	High (Eutrophic)	15	66	
7	Low/Mod. Threshold	13	100	
8	Human Impact	13	1000	
10	Low normal	13	6.5	
10	High normal	14	9.5	
11	Critical	13	2	
11	Endangered	14	5	
11	Highly Sensitive	15	10	
12	RI DEM contact	13	61	
13	EPA rec. contact	13	200	
13	RI DEM contact	14	400	
14	EPA rec. contact	13	126	
17	Drinking Water Standard	13	250	

Figure 12. Entire contents of the *TBL_Thresholds* table.

Note names of the fields in this table: *MonthText* contains the name of the threshold for each parameter; *SamplingMonth* contains 13, 14, or 15; and *Concentration* contains the data value of the threshold. These fields allow the threshold numbers to be included with the real data as “data” for “dummy” sampling months (13-15) with “month names” identifying the meaning of the threshold values. These will be displayed in the plot legend.

These threshold data were incorporated into the real sampling data using UNION queries, which had to be written in structured query language (SQL) (Figure 13). The UNION queries for the chlorophyll and Secchi data are very similar, so only the chlorophyll will be described here. First, the real data values to be plotted were selected from the *QRY_rpt_Chlorophyll* query (the data field *Max* was given the alias *Concentration* in the *QRY_rpt_Chlorophyll* query). These data are concatenated with the threshold data using a UNION with another query. This second query extracts one record per *SamplingYear* and *SiteNum* from the *Chlorophyll* table (SELECT DISTINCT) and joins these records with the *TBL_Thresholds* table using the *ParamNum* field. An INNER JOIN is used to keep only those data existing in both tables (*ParamNums* other than Chlorophyll from the thresholds table are excluded). The *MonthText* field in *TBL_Thresholds* is renamed to *CValue* to match the field in the real chlorophyll data in the first SELECT query. This is because the chlorophyll (and Secchi depth) data represent the sampling year and not data for a particular month (the seasonal maximum (*Max*) or seasonal median (*Median*) is used in the plot). Thus, the second query produces a recordset with “dummy threshold data” for each *SamplingYear* and *SiteNum* combination that has real data. Performing a UNION of the real and dummy data allows for plotting of these data together.

```
SELECT QRY_rpt_Chlorophyll.ParamNum, QRY_rpt_Chlorophyll.SiteNum,
QRY_rpt_Chlorophyll.SamplingYear, QRY_rpt_Chlorophyll.CValue,
QRY_rpt_Chlorophyll.Concentration
FROM QRY_rpt_Chlorophyll
UNION
(SELECT DISTINCT Chlorophyll.ParamNum, Chlorophyll.SiteNum,
Chlorophyll.SamplingYear, TBL_Thresholds.MonthText as CValue,
TBL_Thresholds.Concentration
FROM (Chlorophyll INNER JOIN TBL_Thresholds ON Chlorophyll.ParamNum =
TBL_Thresholds.ParamNum)
WHERE (((Chlorophyll.TSI) In ("O", "o", "M", "m", "E", "e", "H", "h", ""))));
```

Figure 13. *QRY_PivotChart_Chlorophyll_Threshold* UNION query. The *QRY_PivotChart_Secchi_Threshold* UNION query is very similar.

The next step was to create the PivotChart subreport from this UNION recordset. These subreports are actually subFORMS—this is necessary since there are no PivotChart reports in Access, only PivotChart forms. However, Access forms behave perfectly well as subreports. Again, this procedure is very similar for the chlorophyll and Secchi data, so only the chlorophyll is described. The subreport for the Chlorophyll data is named *FRM_PivotChart_Chlorophyll_Threshold* (Figure 14, Figure 15), and the Secchi subreport is *FRM_PivotChart_Secchi_Threshold*.

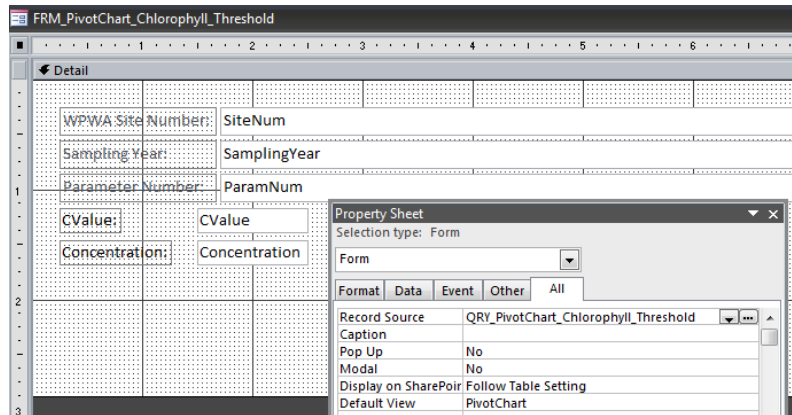


Figure 14. *FRM_PivotChart_Chlorophyll_Threshold* in design view.

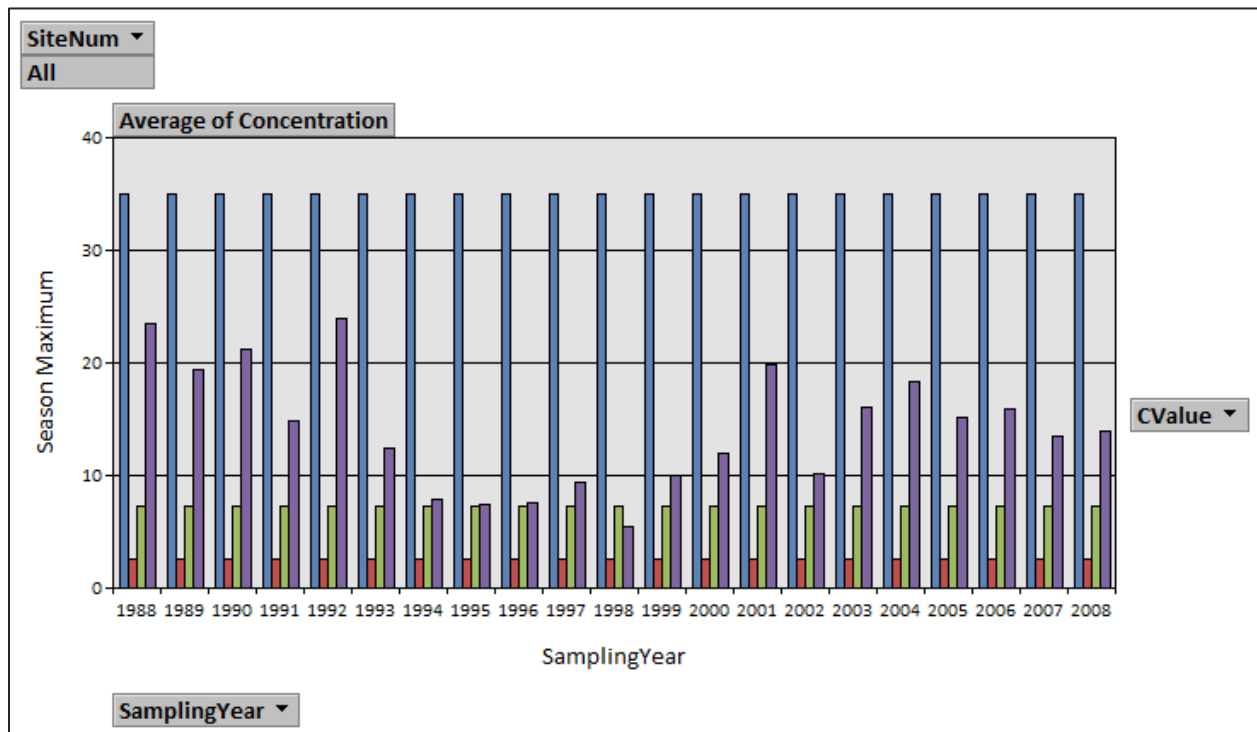


Figure 15. *FRM_PivotChart_Chlorophyll_Threshold* in PivotChart view.

Note in the PivotChart above: *SiteNum* is in the Filter Field drop zone, *CValue* is in the Series Field drop zone, *Concentration* is in the Data Field drop zone (with summary set to Average by right-clicking on the drop zone), and *SamplingYear* is in the Category Field drop zone. The threshold “dummy data” are visible as the tallest blue bars that are all the same height, shorter green bars that are all the same height, and shortest red bars that are all the same height. To convert these thresholds bars to threshold lines, click three times on one of the tallest bars to highlight ALL of the tall bars (one click to select one bar, then a double-click to select them all), then right click and select “Change Chart Type” (Figure 16).

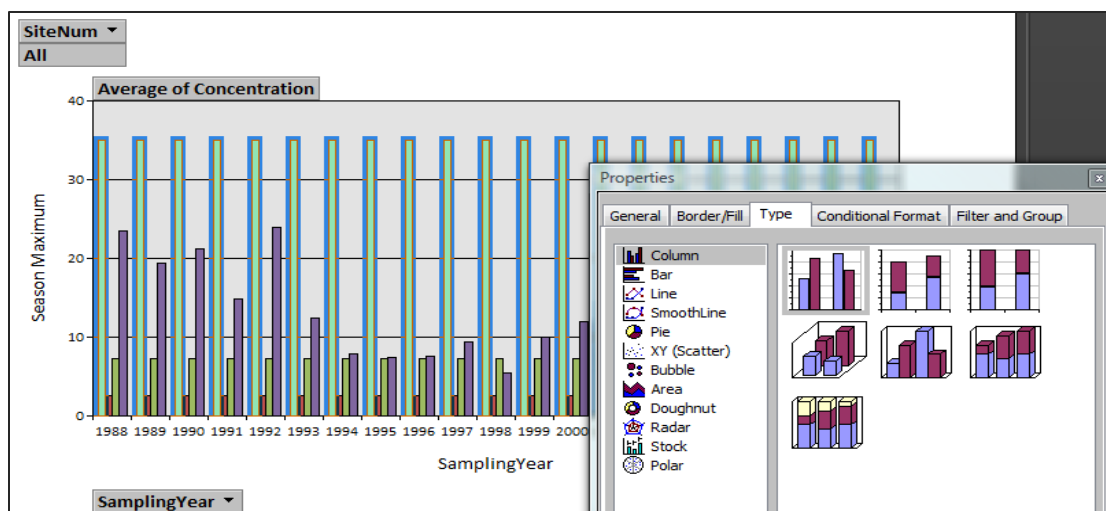


Figure 16. Change chart type.

Select “Line” and choose the upper-leftmost (simple) line chart (Figure 17). Then, click the “Line/Marker” tab and choose the desired line color and weight and close the Properties window. Now, the threshold should be represented as a line. Repeat this procedure for the rest of the threshold bars to create a plot as shown in Figure 18.

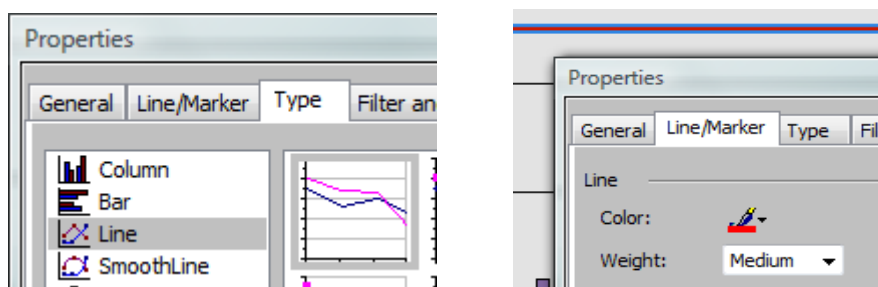


Figure 17. Changing the chart type for thresholds.

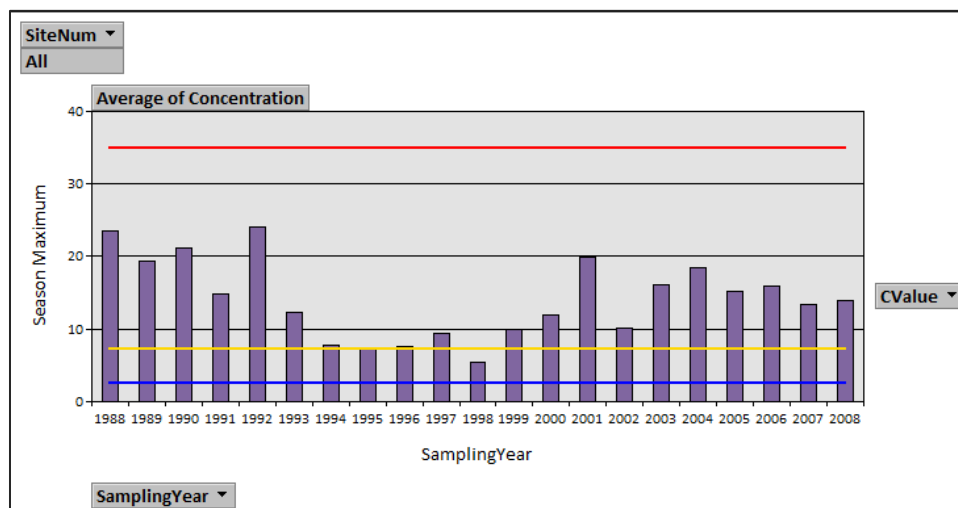


Figure 18. Plot with threshold lines.

To color-code the real chlorophyll data, the remaining bars for the real data were conditionally formatted. Click three times on any data bar to select ALL data bars, and then open the Property Sheet. Click on the “Conditional Format” tab, choose the style shown, choose the two colors (in this case, yellow for the low values and green for the high values), and enter “12” as the threshold number (Figure 19). This number was chosen by Elizabeth Herron and Linda Green for the maximum seasonal chlorophyll data to indicate an algae bloom.

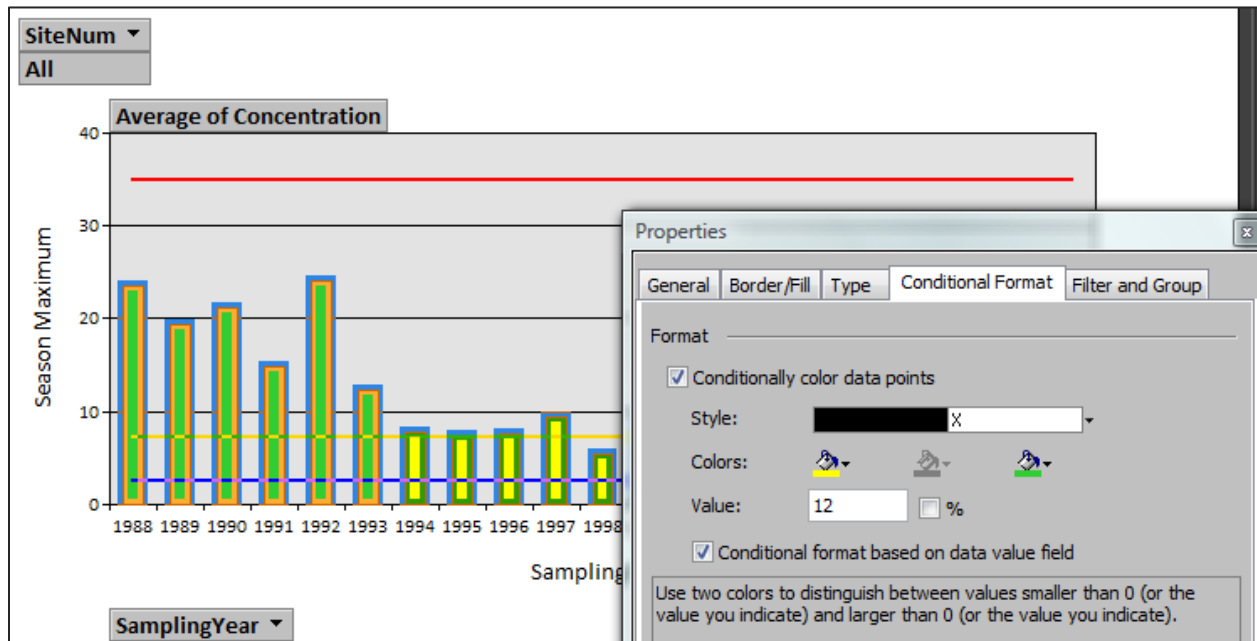


Figure 19. Conditionally formatting the data.

Finally, with the property sheet still open, click on the background of the chart, click on the “Show/Hide” tab, and un-check the “Field Buttons/Drop Zones” box. Close the Properties window, and you should have the finished chlorophyll PivotChart Form (subreport) (Figure 20).

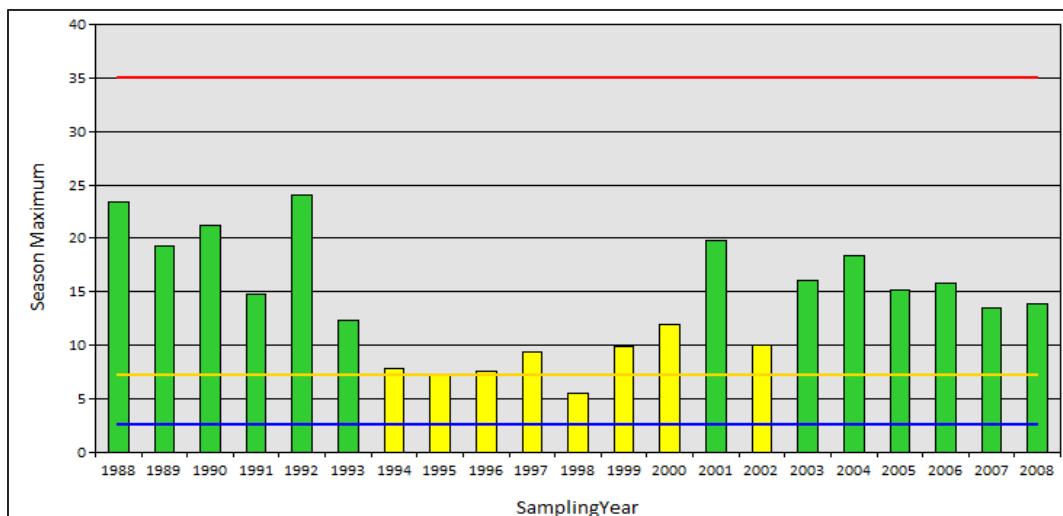


Figure 20. Final chlorophyll PivotChart form.

As mentioned before, similar procedures were used to create the Secchi subreport, although the plot format is different. The Plot Area is rotated 180 degrees; that is, the horizontal axis is at the top of the plot, with the bars extending downward toward the bottom of the page. This, combined with a plot background gradient from light blue (top) to dark blue (bottom), is designed to better represent Secchi depth below the water surface. Also, the color-coding of the bars is different for the Secchi data than it was for the chlorophyll data (Figure 22)--the bars are color coded to indicate trophic status on a gradient from red (eutrophic) to blue (oligotrophic).

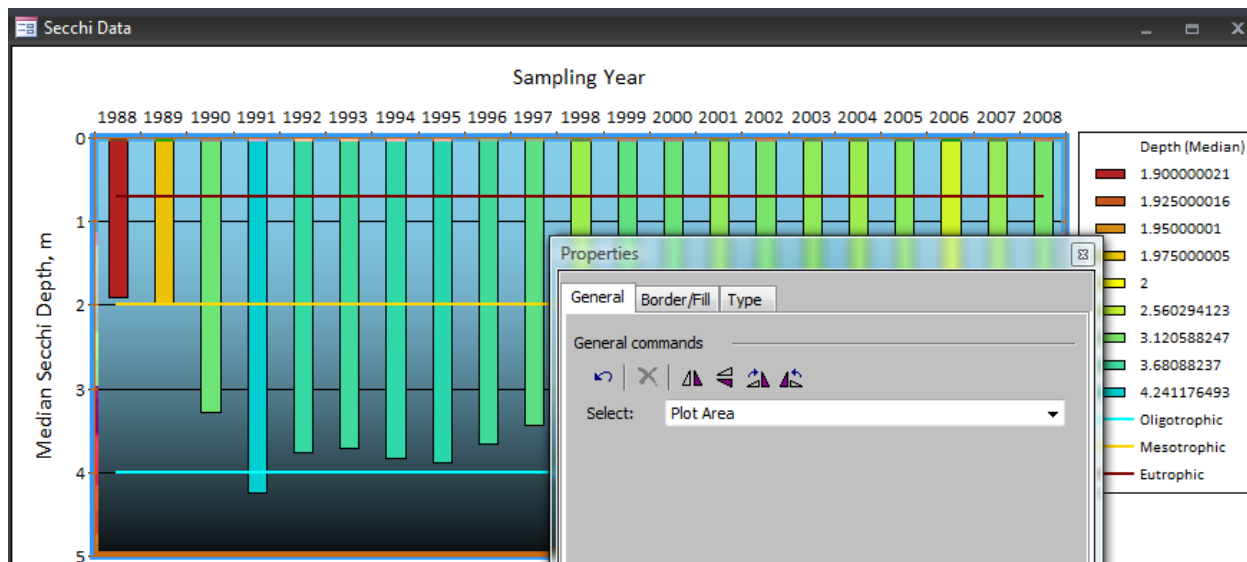


Figure 21. Secchi PivotChart showing rotated Plot Area and colored plot background.

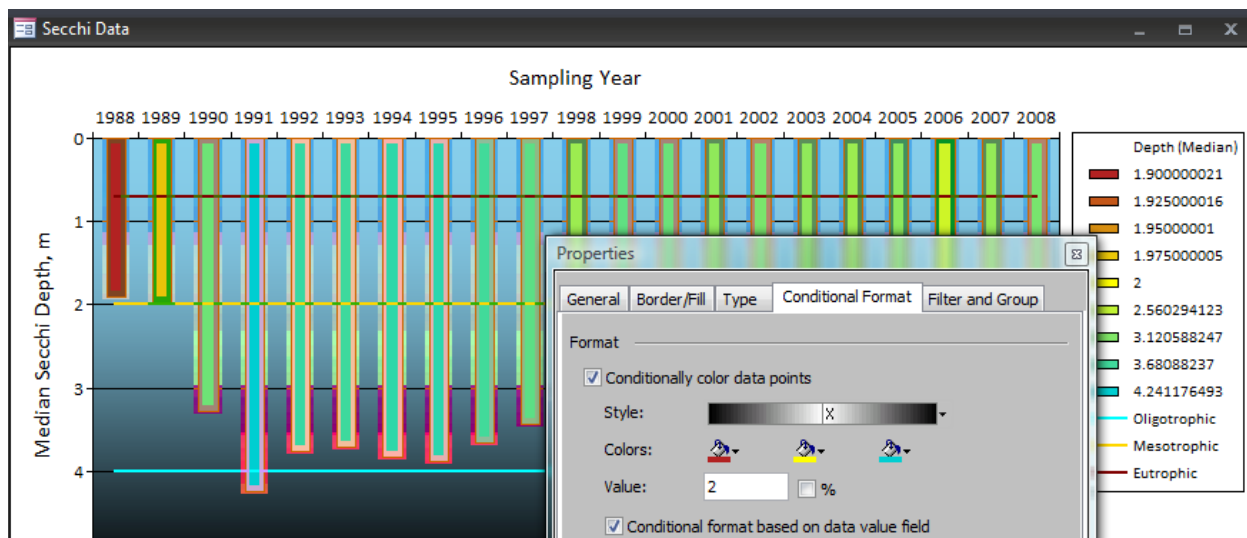


Figure 22. Secchi PivotChart showing color-coded data.

With the subreports created, it is time to create the top-level chlorophyll and Secchi reports. A top-level data report was needed for every *SiteNum* for which there were Secchi or chlorophyll data. First, queries named *QRY_Param_Sites_Secchi* and *QRY_Param_Sites_Chlorophyll* were created that contain one record per *SiteNum* with data by setting the "Unique Values" property for the query to "yes" (Figure

23). A field containing file names for the PDF-format reports (local machine) was generated using a VB function (Figure 24). In addition, information fields about the parameters were included from the *Param_Metadata* table—these fields will be displayed in the final data reports.

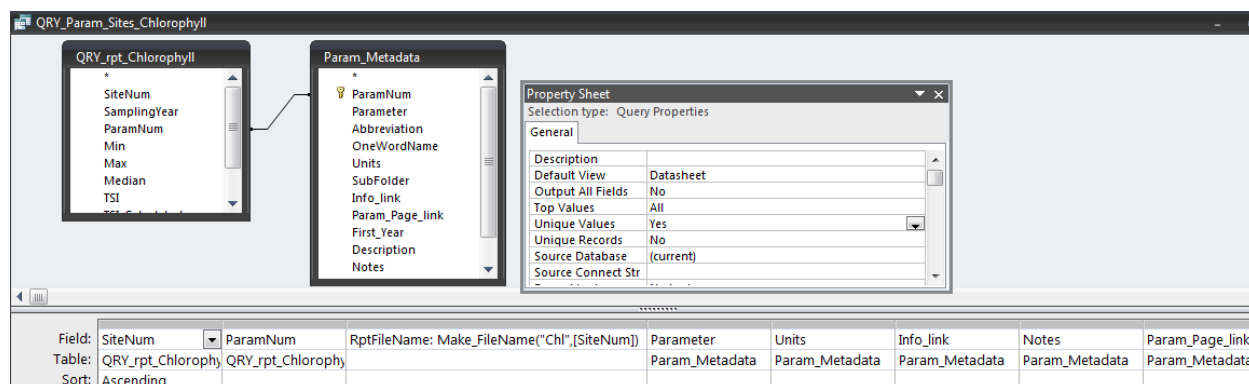


Figure 23. *QRY_Param_Sites_Chlorophyll* query showing Unique Values = “Yes”. Also note the VB function call *Make_Filename* to generate values in the *RptFileName* field. *QRY_Param_Sites_Secchi* is very similar.

```

'-----
'This function creates a local file name for the report containing data for
'the parameter abbreviation and SiteNum passed in as arguments
'-----
Public Function Make_Filename(Abbrev As String, SiteNum As Integer)
    Make_Filename = Abbrev & "_" & SiteNum & ".PDF"
End Function

```

Figure 24. Simple Visual Basic function for generating the local file names for data reports.

The *QRY_Param_Sites* queries above generate lists of ALL sites with chlorophyll or Secchi data. As was mentioned earlier, the user has the option to generate reports only for sites sampled in the latest sampling season. To allow this, two additional queries named *QRY_Param_Sites_New_Year_Chlorophyll* and *QRY_Param_Sites_New_Year_Secchi* were created (Figure 26). These queries look very similar to the previously described *QRY_Param_Sites* queries but contain a link to a TOTALS query named *QRY_Sites_With_Latest_Year_of_Data* (Figure 25). This query uses another query, *QRY_Latest_Year_in_DB*, to select only those sites for which there are any data from the latest sampling year in the chemistry data table, keeping only one record per *SiteNum*. The chemistry data table is used is this query since it is the most comprehensive list of sites and sampling years. The number passed into the *DoReport.ReportIt* macro by the *Frm_Report_Site_Options* form determines which of the two *QRY_Param_Sites* queries for chlorophyll and Secchi depth (for ALL sites or for only active sites) gets used by the *Make_Chlor_Secchi_Reports* function to create the recordset used in generating reports. Recall that this is the function that actually generates the reports for these parameters.

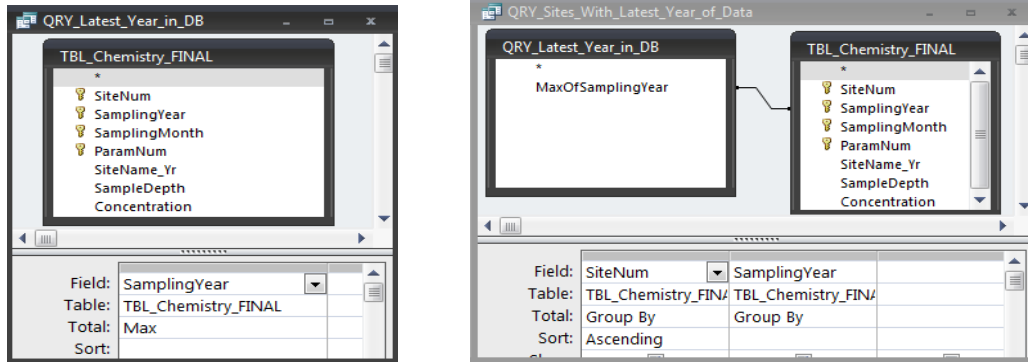


Figure 25. *QRY_Latest_Year_in_DB* and *QRY_Sites_With_Latest_Year_of_Data*.

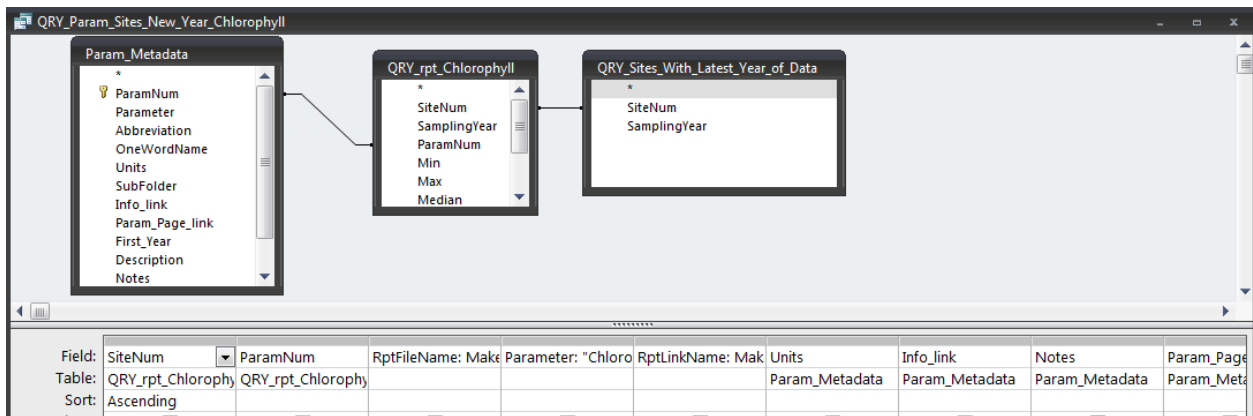


Figure 26. *QRY_Param_Sites_New_Year_Chlorophyll* (*QRY_Param_Sites_New_Year_Secchi* is similar).

The final queries in the report generation process for chlorophyll and Secchi data are named *QRY_For_Rpt_TOP_Chlorophyll* and *QRY_For_Rpt_TOP_Secchi* (Figure 27). These queries are the data sources for the top level parameter/site reports. The *QRY_Param_Sites_* queries are joined with the *MasterSiteList* table to access the sampling site name. Also in these queries, the *SiteNum* field was given an alias of *MstSiteNum* and the *ParamNum* field was given an alias of *MstParamNum* to allow the joining of the subreports to the top level reports.

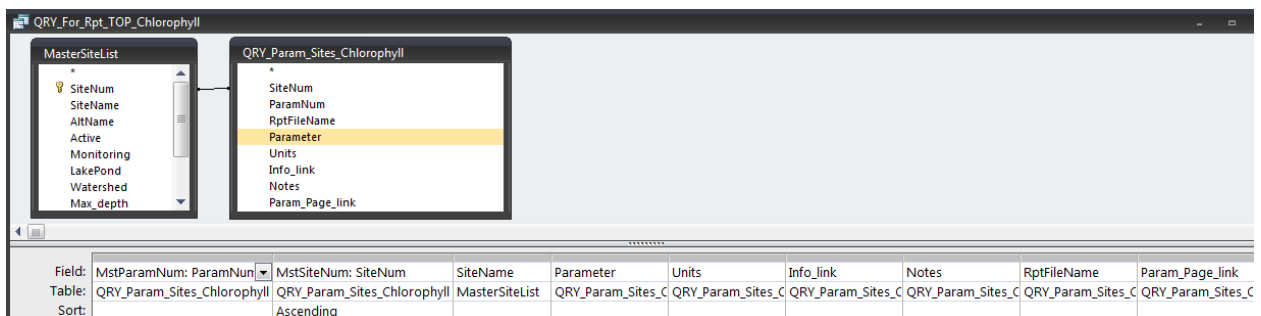


Figure 27. *QRY_For_Rpt_TOP_Chlorophyll*. *QRY_For_Rpt_TOP_Secchi* is similar.

Recall that the subreports for the Chlorophyll data are named *RPT_Crosstab_Chlorophyll* and *FRM_PivotChart_Chlorophyll_Threshold*; the subreports for the Secchi data are named

RPT_Crosstab_Secchi and *FRM_PivotChart_Secchi_Threshold*. These subreports were incorporated into the top-level reports (*RPT_TOP_Chlorophyll* and *RPT_TOP_Secchi*) by simply dragging and dropping them from the Access Objects sidebar into the top-level report detail section, then linking master/child fields (Figure 28, Figure 29). A new page was forced after each Detail section of the top-level report, which causes data for one *SiteNum* per parameter to be displayed on each page. Linking master/child fields forces ONLY the data for the current *SiteNum* to be displayed in the subreports on each page.

Thus, a multi-page report is generated for each parameter (one page per sampling site). Each page contains data for one *SiteNum/ParamNum*, with a table and plot of all of the data for that *SiteNum* and *ParamNum* ONLY thanks to the linking of master and child fields. Figure 30 and Figure 31 show examples of the final chlorophyll and Secchi depth report pages, respectively.

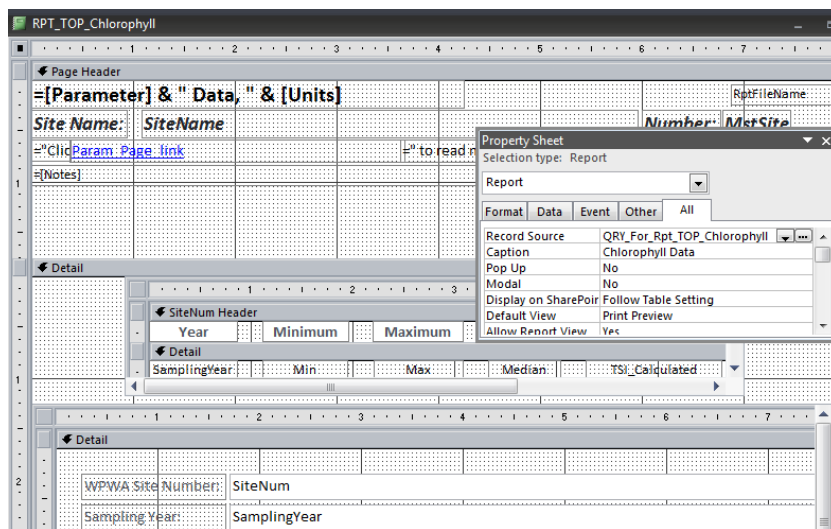


Figure 28. *RPT_TOP_Chlorophyll* in design view showing the Report Record Source (*QRY_For_Rpt_TOP_Chlorophyll*). *RPT_TOP_Secchi* is very similar.

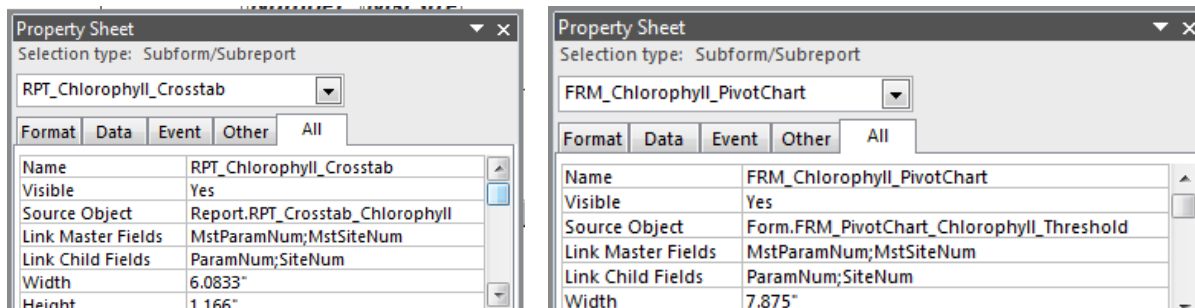


Figure 29. Property sheets showing Source Objects for the tabular (*Report.RPT_Crosstab_Chlorophyll*) and PivotChart (right, *Form.FRM_PivotChart_Chlorophyll_Threshold*) subreports. Note the Link Master Field and Link Child Field properties. These properties are similar to those in *RPT_TOP_Secchi*.

Chlorophyll Data, ug/L

Site Name: Tucker Pond

Number: 1070

Click <http://www.wpsa.org/waterdata/Chlorophyll.PDF> to read more information about Chlorophyll.

Low algae levels (oligotrophic): < 2.6 ug/L; Moderate algae levels (mesotrophic): 2.6 - 7.2 ug/L; Elevated algae levels (eutrophic): 7.3 - 35 ug/L; Extremely elevated algae level (hypereutrophic): > 35 ug/L. 1988-1991: mean was used to determine tro

Year	Minimum	Maximum	Median	Trophic Status
1991	0.3	15.5	5.3	M
1992	0.5	45.9	5.5	M
1993	1.4	22.2	3.1	M
1994	0.4	3.5	1.2	O
1995	1.8	3.2	2.4	O
1996	1.1	9.9	2.3	O
1997	2.0	15.3	3.6	M
1998	2.1	18.9	5.0	M
1999	0.9	16.4	5.6	M
2000	1.8	29.2	7.0	M
2001	1.9	8.2	3.8	M
2002	1.2	6.5	3.9	M
2003	4.0	35.2	13.4	E
2004	2.7	32.7	8.1	E
2005	3.2	25.0	9.0	E
2006	1.6	12.9	4.8	M
2007	0.4	37.5	6.1	M
2008	1.8	16.9	8.7	E

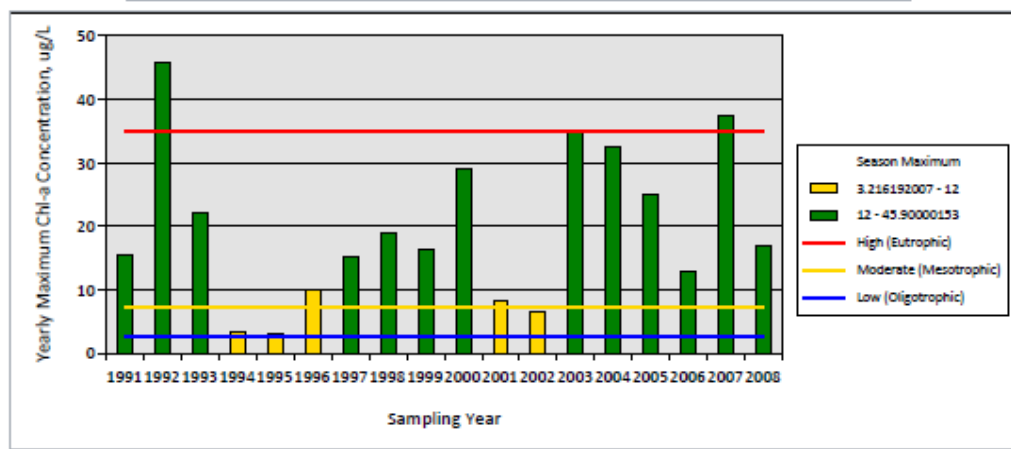


Figure 30. Example of a top-level Chlorophyll report (*RPT_TOP_Chlorophyll*).

Secchi Depth Data, M

Site Name: Alton Pond

Number: 4480

Click <http://www.wpwa.org/waterdata/Secchi.PDF> to read more information about Secchi Depth.

For 1988 and 1989, mean secchi depth was used to determine trophic status. 0 = Oligotrophic TSI < 40 (> 4.0 m); M = Mesotrophic TSI 40 - 50 (2.0 - 4.0 m); E = Eutrophic TSI > 50 (0.7 - 2.0 m); H = Hypereutrophic TSI > 65 (< 0.7 m). Data excluded from reports: * = Insufficient data to determine TSI/Median Trophic Status, ** = TSI/Median Trophic Status not classified because of shallowness of the site, f (phi) = As a river site, Secchi depth may not reflect trophic status, # = Water clarity not a good indicator of trophic status due to naturally dark color.

Year	Minimum	Maximum	Median	# Times at Bottom	Water Depth	Trophic Status
1989	1.3	3.9	2.5		3.9	M
1990	1.8	2.8	2.2			M
1992	1.7	3.5	2.8	4	4.2	M
1993	2.3	4.2	3.1	6	4.2	M
1994	2.5	4.0	3.4	3	4.2	M
1995	2.8	4.3	3.4	4	4.2	M
1996	1.7	3.9	3.0	1	4.2	M
1997	2.7	3.8	2.9	0	4.2	M
1998	1.0	2.1	1.5	0	4.2	E
1999	1.1	3.1	1.9	0	4.2	E
2000	1.3	2.5	1.9	0	4.2	E
2001	1.0	2.4	1.7	0	4.2	E
2002	1.1	2.6	1.7	0	4.2	E
2003	1.0	2.6	1.4	0	4.2	E
2004	1.6	3.2	2.4	0	4.2	M
2005	1.7	3.8	2.7	0	4.2	M
2006	1.3	3.2	2.0	0	4.2	M
2007	1.9	3.7	2.6	0	4.2	M
2008	1.7	3.2	2.5	0	4.35	M

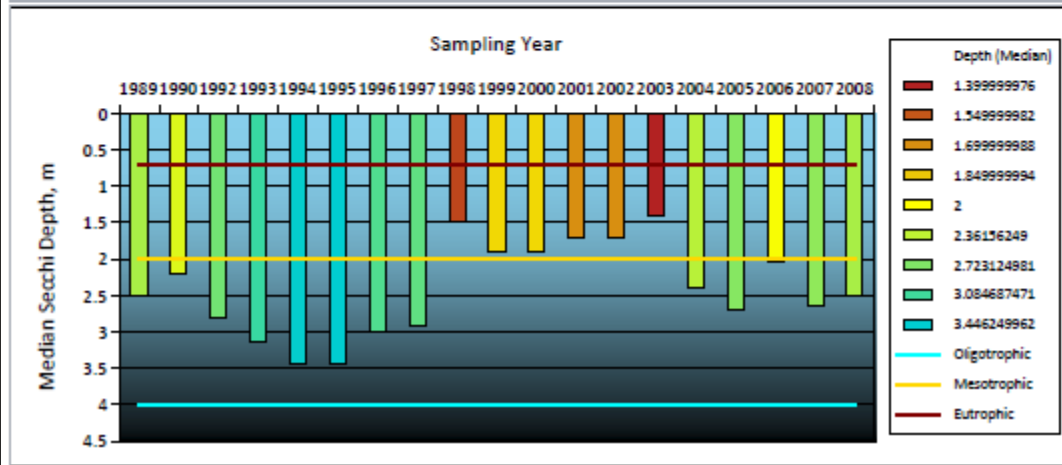


Figure 31. Example of a top-level Secchi depth report (*RPT_TOP_Secchi*).

3.3. Generating Chemistry Reports

Most of the reports for the chemistry parameters are generated using the same VB function, VB subroutine, queries, reports, and forms. These will be described in detail below. Chemistry parameters whose reports are generated differently (total phosphorus and chloride) will be discussed later.

3.3.1. Visual Basic Function and Subroutine

Selecting a chemistry parameter from the report options form causes the *Generate_Chemistry_Reports* function to be called by the *ReportIt* macro (Figure 7). This function takes five arguments: the *ParamNum* value (passed into the function as the numeric value of the radio button from the report

options form), the name of the local subfolder into which to write the reports, the number of decimal places to display in the data table, the report options number (all sites=1, most recently sampled sites =2, one site=3, passed into the function as the numeric value of the radio button from the *Frm_Report_Site_Options* form), and the *SiteNum* for report generation if the site option is "3". For example, if "Dissolved Phosphorus" is chosen from the report options form and "latest year only" is chosen from the site options form, the value passed to the macro from the radio button (equal to the *ParamNum*) is 6 and the function call is "*Generate_Chemistry_Reports* ([Forms]![Frm_Report_Options]![Rpt_Opt], "Phosphorus",0, [Forms]![Frm_Report_Site_Options]![Rpt_Opt],1010)" ("1010" is the default site number for the single-site reporting option).

The main job of the *Generate_Chemistry_Reports* function is to use the *ParamNum* passed in as an argument, along with the site report option, to fill a temporary table (*TBL_One_Param_For_Report*) with the data from *TBL_Chemistry_FINAL* that will be used in report generation. The temporary table has the same structure as the *TBL_Chemistry_FINAL* table. Once the table is filled, the *Generate_Chemistry_Reports* function calls another subroutine named *Make_Chemistry_Reports*.

```
'-----
' Generate chemistry reports by calling Make_Chemistry_Reports subroutine
'-----
Public Function Generate_Chemistry_Reports(Param As Integer, SubFolder As String,
NumDec As Integer, WhichSites As Integer, RptSite As Integer)

Dim SQL_string As String

On Error GoTo Generate_Chemistry_Reports_Err

msgbox "Please be patient while the reports are generated...", vbInformation, ""
' set the mouse to hourglass to indicate a wait
DoCmd.Hourglass True

' empty the temporary data table just in case...
DoCmd.RunSQL "DELETE TBL_One_Param_For_Report.* FROM TBL_One_Param_For_Report;", -1

If WhichSites = 3 Then
'just load data for one site into the temporary table
    SQL_string = "INSERT INTO TBL_One_Param_For_Report SELECT TBL_Chemistry_FINAL.*
FROM TBL_Chemistry_FINAL WHERE TBL_Chemistry_FINAL.ParamNum=" & Param & " AND
TBL_Chemistry_FINAL.SiteNum=" & RptSite & ";"
Else
'load all data for this parameter into the temporary table
    SQL_string = "INSERT INTO TBL_One_Param_For_Report SELECT TBL_Chemistry_FINAL.*
FROM TBL_Chemistry_FINAL WHERE TBL_Chemistry_FINAL.ParamNum=" & Param & ";"
End If

' insert desired records into temporary data table
DoCmd.RunSQL SQL_string, -1
' Run the Make_Chemistry_Reports subroutine to create reports

If Param = 7 Then
'this is the only chemistry parameter processed by this function that has a threshold
    If WhichSites = 2 Then
' make reports for sites with latest year only
        Call Make_Chemistry_Reports("RPT_TOP_Chemistry_Threshold",
"QRY_Param_Sites_New_Year_Chemistry", SubFolder, NumDec, WhichSites)
    Else
'make reports for all sites
```

```

        Call Make_Chemistry_Reports("RPT_TOP_Chemistry_Threshold",
"QRY_For_Rpt_TOP_Chemistry", SubFolder, NumDec, WhichSites)
    End If
Else
    If Param = 5 Then
        'total phosphorus data are processed slightly differently
        If WhichSites = 2 Then
            Call Make_Chemistry_Reports("RPT_TOP_TP", "QRY_Param_Sites_New_Year_TP",
SubFolder, NumDec, WhichSites)
        Else
            Call Make_Chemistry_Reports("RPT_TOP_TP", "QRY_For_Rpt_TOP_TP", SubFolder,
NumDec, WhichSites)
        End If
    Else
        If WhichSites = 2 Then
            Call Make_Chemistry_Reports("RPT_TOP_Chemistry",
"QRY_Param_Sites_New_Year_Chemistry", SubFolder, NumDec, WhichSites)
        Else
            Call Make_Chemistry_Reports("RPT_TOP_Chemistry",
"QRY_For_Rpt_TOP_Chemistry", SubFolder, NumDec, WhichSites)
        End If
    End If
End If

' empty the temporary data table...
DoCmd.RunSQL "DELETE TBL_One_Param_For_Report.* FROM TBL_One_Param_For_Report;", -1

' make the mouse normal again
DoCmd.Hourglass False

Generate_Chemistry_Reports_Exit:
    Exit Function

Generate_Chemistry_Reports_Err:
    msgbox Error$
    Resume Generate_Chemistry_Reports_Exit

End Function

```

Figure 32. *Generate_Chemistry_Reports* function.

All information needed by *Make_Chemistry_Reports* is passed into it by *Generate_Chemistry_Reports* as arguments. The *Make_Chemistry_Reports* subroutine (Figure 33) takes five arguments: *RPT_TOP_Name* (the name of the top-level report), *QRY_Name* (the name of the query that generates a list of sites with data for the chosen parameter), *PathName* (the subfolder name passed in via *Generate_Chemistry_Reports* from the macro), and *Decimals* (also passed in via *Generate_Chemistry_Reports* from the macro), and *WSites* (the site report option number: 1, 2, or 3). If the user wants to generate reports for ALL sites, the name of the query that generates the site list is *QRY_for_Rpt_TOP_Chemistry*; otherwise, the query is *QRY_Param_Sites_New_Year_Chemistry*. The report and query details will be discussed later.

```

'-----
' Generate Generic reports. This procedure accepts the report name,
' query providing the station list, and subfolder name where the
' PDF files will be output as arguments.
'-----
Public Sub Make_Chemistry_Reports(ByVal RPT_TOP_Name As String, ByVal QRY_Name As
String, ByVal PathName As String, ByVal Decimals As Integer, WSites As Integer)

```

```

' Error handling
On Error GoTo Make_Chemistry_Reports_Err

' Declare the database, recordset, and report filter string variables
Dim dbwpwa As DAO.Database
Dim rcdRecQuery As DAO.Recordset
Dim strFilter As String

' Set the current database and recordset variables for use in the do...until loop
Set dbwpwa = CurrentDb
Set rcdRecQuery = dbwpwa.OpenRecordset(QRY_Name)
' Make the mouse an hourglass to indicate a wait
DoCmd.Hourglass True

If rcdRecQuery.EOF Then
    If WSites = 3 And RPT_TOP_Name = "RPT_TOP_Phosphorus_Trophic" Then
        MsgBox ("No data for this site! Be sure you have chosen a lake/pond SURFACE
water site (Site Number ending in 0).")
    Else
        MsgBox ("No data for this site! Please be sure you have chosen a report and
site combination that is likely to exist. For example, there are only Alkalinity data
for lake/pond SURFACE samples.")
    End If
Else
    ' Start looping through the QRY_Param_Sites_Chemistry file to generate a one page
    ' report for each SiteNum. This loop will process every SiteNum and terminate
    ' when the end of the file (EOF) is reached.
    Do Until rcdRecQuery.EOF
        ' Set the filter for the report to the current SiteNum and ParamNum
        strFilter = "[MstSiteNum]=" & rcdRecQuery![MstSiteNum]
        ' Open the report
        DoCmd.OpenReport RPT_TOP_Name, acViewPreview, "", strFilter, acHidden

        ' Set the number of decimal places for the data table
        Reports(RPT_TOP_Name)![Chemistry Report].Report![May].DecimalPlaces = Decimals
        Reports(RPT_TOP_Name)![Chemistry Report].Report![June].DecimalPlaces = Decimals
        Reports(RPT_TOP_Name)![Chemistry Report].Report![July].DecimalPlaces = Decimals
        Reports(RPT_TOP_Name)![Chemistry Report].Report![August].DecimalPlaces = Decimals
        Reports(RPT_TOP_Name)![Chemistry Report].Report![September].DecimalPlaces =
Decimals
        Reports(RPT_TOP_Name)![Chemistry Report].Report![October].DecimalPlaces = Decimals
        Reports(RPT_TOP_Name)![Chemistry Report].Report![Average].DecimalPlaces = Decimals

        ' Output the file
        DoCmd.OutputTo acOutputReport, RPT_TOP_Name, "PDFFormat(*.pdf)", "C:\Users\Elise
Torello\documents\WPWA\Field Data\Master Access Database\WPWASites\" & PathName & "\"
& rcdRecQuery![RptFileName], False, "", 0, acExportQualityPrint
        ' Close the report
        DoCmd.Close acReport, RPT_TOP_Name
        ' Move on to the next record in the query
        rcdRecQuery.MoveNext
        'and loop
    Loop
End If

' Turn off the hourglass
DoCmd.Hourglass False

Make_Chemistry_Reports_Exit:
Exit Sub

Make_Chemistry_Reports_Err:

```

```

msgbox Error$
Resume Make_Chemistry_Reports_Exit

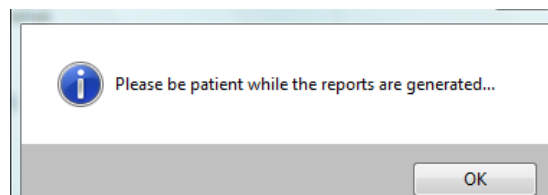
End Sub

```

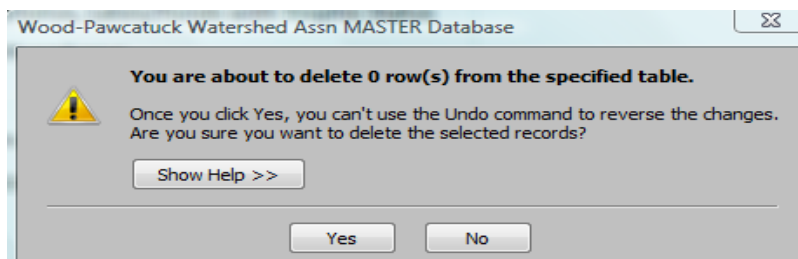
Figure 33. The *Make_Chemistry_Reports* subroutine.

First, the subroutine opens the query passed in as an argument to generate the list of sites (*SiteNums*) with data for the desired parameter. Next, the mouse pointer is set to an hourglass to indicate a wait. If there are no records for some reason (for example, alkalinity data were requested for a river site), an error message is output. Otherwise, for each *SiteNum* in the query list, the subroutine opens the data report page for that *SiteNum*, sets the appropriate number of decimal places to display in the data table, outputs the PDF file for that report page, then closes the report page and moves to the next *SiteNum* in the query. When the end of the query (or end of file, EOF) is reached, the mouse pointer is returned to normal, the subroutine is exited, and control returns to the *Generate_Chemistry_Reports* function. The function then empties the temporary data table, returns the mouse pointer to normal (just to make sure), and exits.

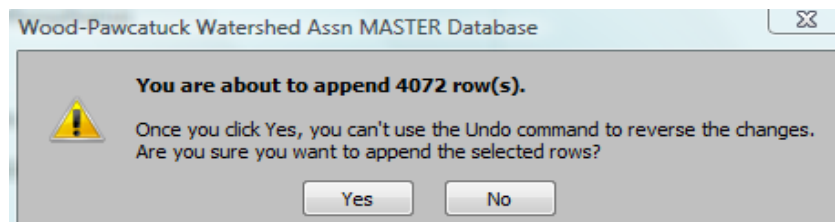
While all of this code is executing, the user will see a series of information boxes requiring input (Figure 34). In each case, click “OK” or “Yes” to continue with report production.



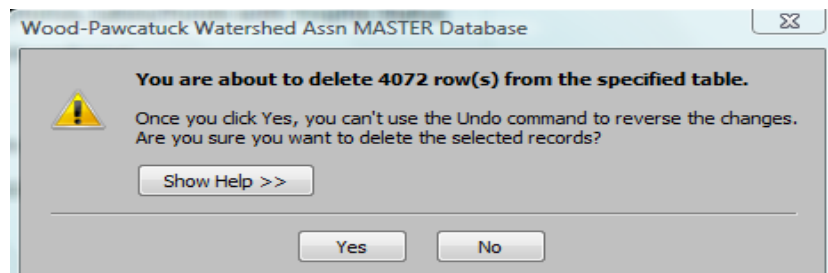
Click “OK” to continue.



Click “Yes” to continue, even if there are records in the table to be deleted.



Click “Yes” to load the desired data into the temporary table.



Click “Yes” to empty the temporary table for the next reporting session.

Figure 34. Information boxes requiring user input during chemistry report production.

3.3.2. Create CROSSTAB Query and Subreport

As with the chlorophyll and Secchi depth data, each data report page contains two sub-reports: one containing a data table and the other containing a data plot. The tables were generated using crosstab queries, which put the vertical data back into horizontal (spreadsheet-like) format. The main advantage with using crosstab queries in data reporting is their data summarization options, including the calculation of monthly and seasonal averages (see last column in design view below). The crosstab query for the chemistry data was named *QRY_Crosstab_Chemistry* (Figure 35).

Field:	ParamNum	SiteNum	SamplingYear	Expr1: MonthName([SamplingMonth])	Concentration	Average: Concentration
Table:	TBL_One_Param_Fo	TBL_One_Param_Fo	TBL_One_Param_Fo		TBL_One_Param_Fo	TBL_One_Param_Fo
Total:	Group By	Group By	Group By	Group By	Avg	Avg
Crosstab:	Row Heading	Row Heading	Row Heading	Column Heading	Value	Row Heading
Sort:	Ascending	Ascending	Ascending			

ParamNum	WPW	Sampl	May	June	July	August	Septeml	October	Average
4	1010	1993						510.0	510.0
4	1010	1994	820.0					350.0	585.0
4	1010	1995	580.0		920.0				750.0
4	1010	1996			565.0			615.0	590.0
4	1010	1997	620.0		440.0			490.0	516.7
4	1010	1998	580.0		650.0			560.0	596.7

Figure 35. *QRY_Crosstab_Chemistry* in design view (top) and datasheet view (bottom).

The tabular subreport created from this query is named *RPT_Crosstab_Chemistry* (Figure 36). The Record Source property for the report is set to *QRY_Crosstab_Chemistry*. You will soon see how this subreport was incorporated into the final data report.

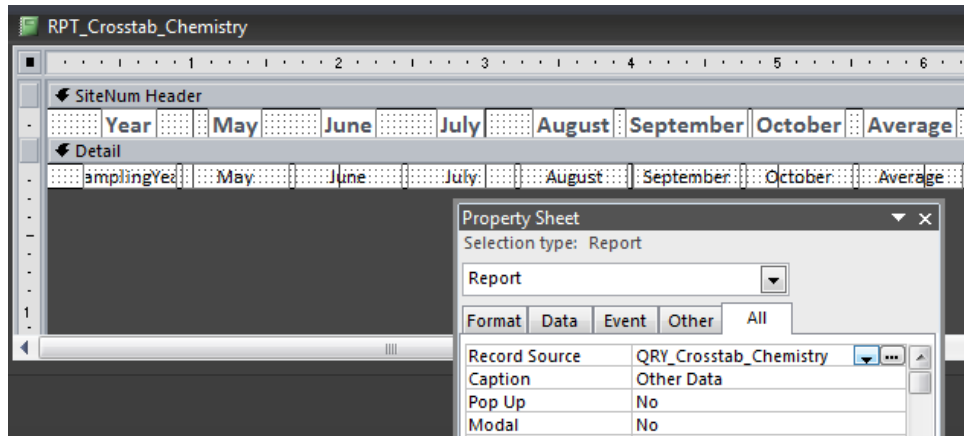


Figure 36. The *RPT_Crosstab_Chemistry* subreport showing the record source property.

3.3.3. Create PIVOTCHART Query and Subreport

The second subreport contains a PivotChart (data plot). To create this subreport, a query named *QRY_PivotChart_Chemistry* was created (Figure 37). During query design, the averages of all of the data for every *SiteNum* for each month and year are displayed in the chart; however, during report generation, the PivotChart subreport will only display data for the current *SiteNum* (will be discussed later).

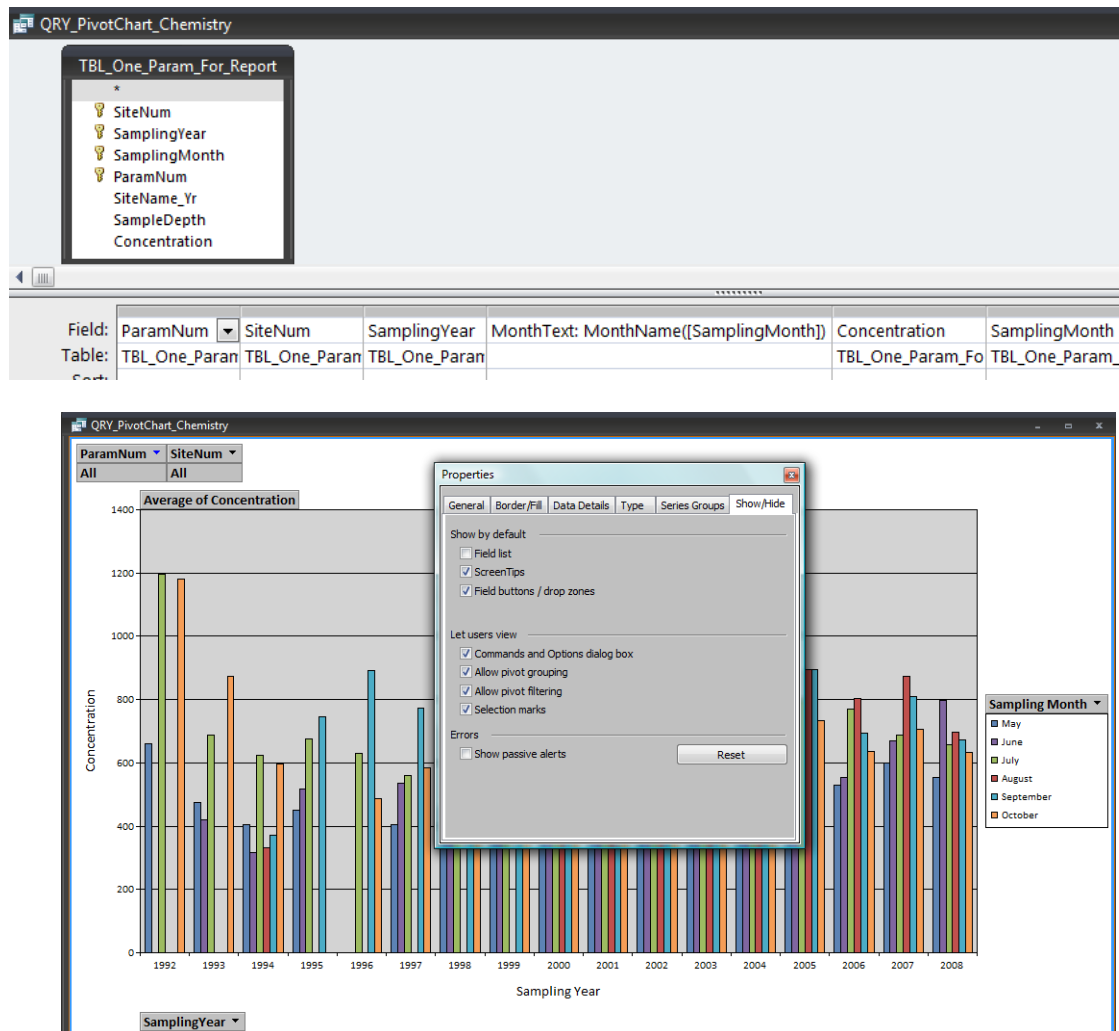


Figure 37. QRY_PivotChart_Chemistry query in design view (top) and PivotChart view (bottom). Note the fields displayed in the “drop zones” in PivotChart view. For the final report, the “Field buttons/drop zones” check box in the Properties window is cleared.

Next, a PivotChart form named FRM_PivotChart_Chemistry was created using the QRY_PivotChart_Chemistry query as the record source (Figure 38). Note the fields in the drop zones.

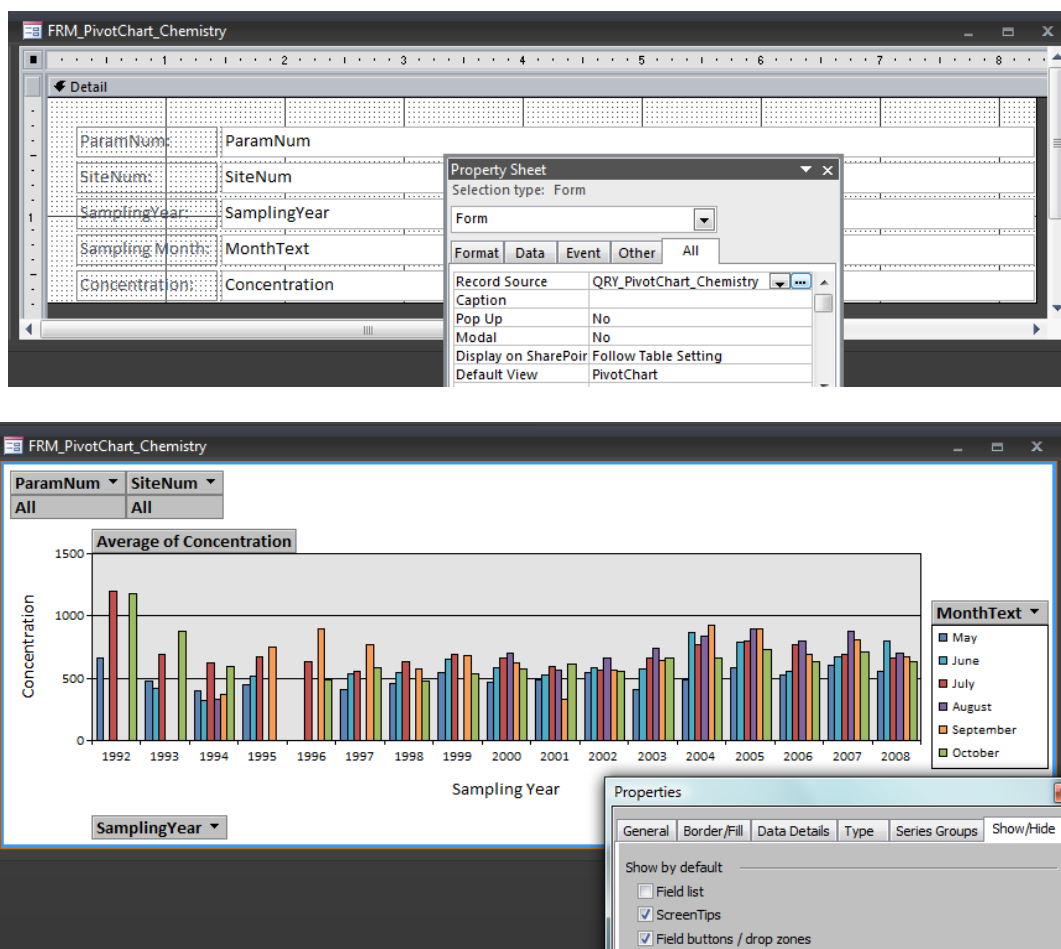


Figure 38. *FRM_PivotChart_Chemistry* in design view (top) and PivotChart view (bottom). Note the “Record Source” field in the Property Sheet in design view; also note the values in the “Field buttons /drop zones” in the Properties window in PivotChart view. The check box for “Field buttons /drop zones” is cleared in the final report.

3.3.4. Make Top-Level Report Pages

A top-level data report is needed for every *ParamNum* and *SiteNum* for which there are data. A query named *QRY_Param_Sites_Chemistry* was created that contains one record per *ParamNum/SiteNum* currently in the temporary (*TBL_One_Param_For_Report*) table (Figure 39). This was accomplished by setting the “Unique Values” property for the query to “yes”.

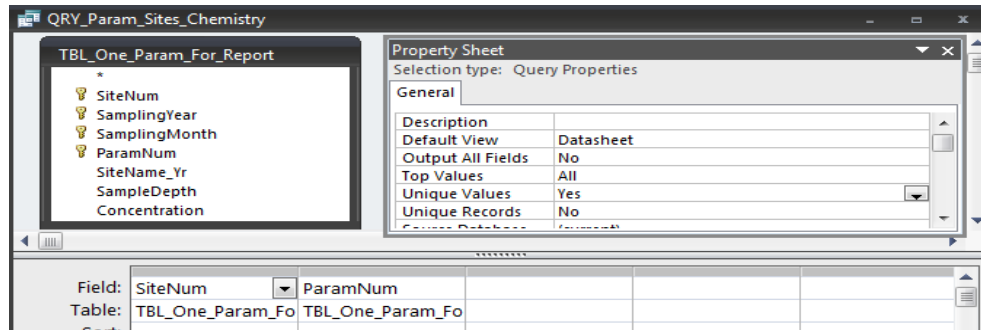


Figure 39. QRY_Param_Sites_Chemistry query showing Unique Values = "Yes".

A query named *QRY_For_Rpt_TOP_Chemistry* was created that joins the *QRY_Param_Sites_Chemistry* with the *Param_Metadata* and *MasterSiteList* tables to access informational fields for the reports (Figure 40). *QRY_For_Rpt_TOP_Chemistry* is used as the data source for the top level parameter/site reports, plus it is used by the *Make_Chemistry_Reports* VB function as the list of *SiteNums* for which to generate report files. Within in this query, the *SiteNum* field is given an alias of *MstSiteNum* and the *ParamNum* field is given an alias of *MstParamNum* to allow the joining of the subreports to the top level reports. Plus, a field named *RptFileName* containing file names for the PDF-format reports (local machine) was generated using the VB function in Figure 24.

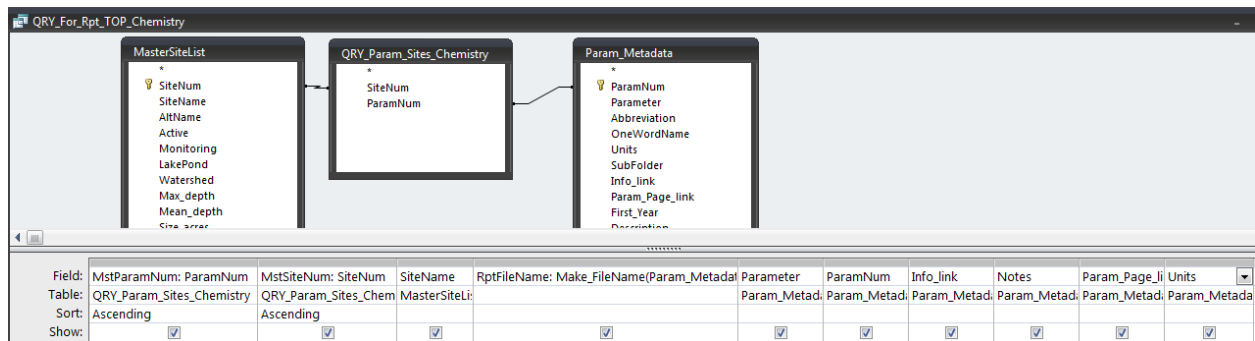


Figure 40. QRY_For_Rpt_TOP_Chemistry.

A query using *QRY_Param_Sites_Chemistry* was created to eliminate sites not sampled in the most recent sampling season. This second query is named *QRY_Param_Sites_New_Year_Chemistry* (Figure 41). It uses the *QRY_Sites_With_Latest_Year_of_Data*, linked to *QRY_Param_Sites_Chemistry*, to keep only those sites sampled in the most recent year. A *RptFileName* field containing file names for the PDF-format reports (local machine) was generated using the VB function in Figure 24, since this field is needed within the *Make_Chemistry_Reports* VB function.

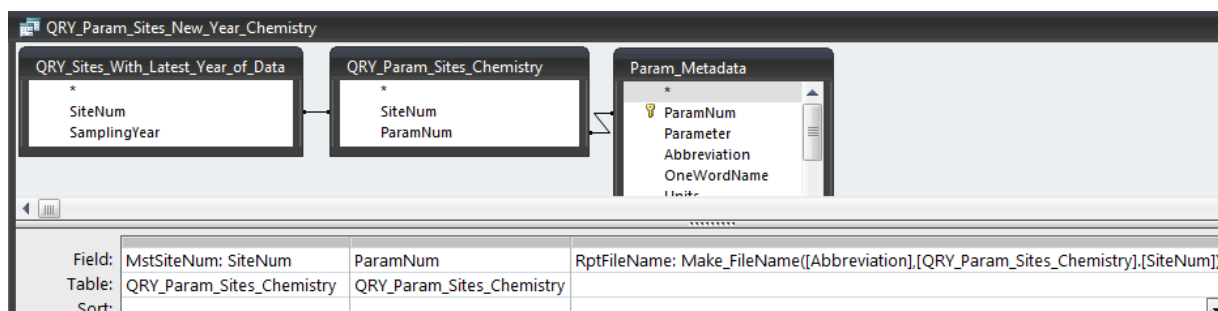


Figure 41. *QRY_Param_Sites_New_Year_Chemistry* query, which eliminates sites not sampled in the most recent sampling season. Note the VB function call *Make_FileName* to generate file names for the data reports on the local computer.

The top-level chemistry report is named *RPT_TOP_Chemistry*, with *QRY_For_Rpt_TOP_Chemistry* as the REPORT data source (Figure 42). The tabular (*RPT_Crosstab_Chemistry*) and PivotChart (*FRM_PivotChart_Chemistry*) subreports were added to the top-level report's Detail section by dragging and dropping from the Access object window. The tabular subreport Source Object was *RPT_Crosstab_Chemistry* (Figure 43), and the PivotChart Source Object is *FRM_PivotChart_Chemistry* (Figure 44). For both subreports, the Link Master Fields Properties were set to "MstParamNum;MstSiteNum" and the Link Child Fields were set to "ParamNum;SiteNum". As with the chlorophyll and Secchi reports, linking the master and child fields in the subreports forces each subreport to display only the data for the *SiteNum* on a particular page. The subreports were reformatted until they looked good within the main report (any changes made to the subreports from within the top-level report are automatically saved to the original subreports). An example of the finished product is shown in (Figure 45).

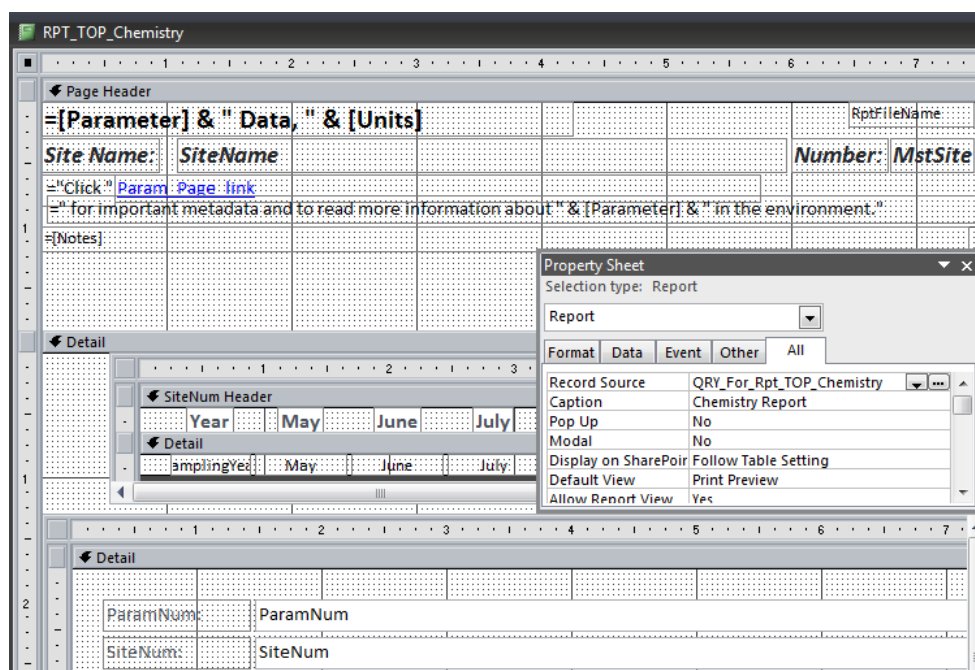


Figure 42. *RPT_TOP_Chemistry*, the top-level data report, showing *QRY_For_Rpt_TOP_Chemistry* as the Report record source.

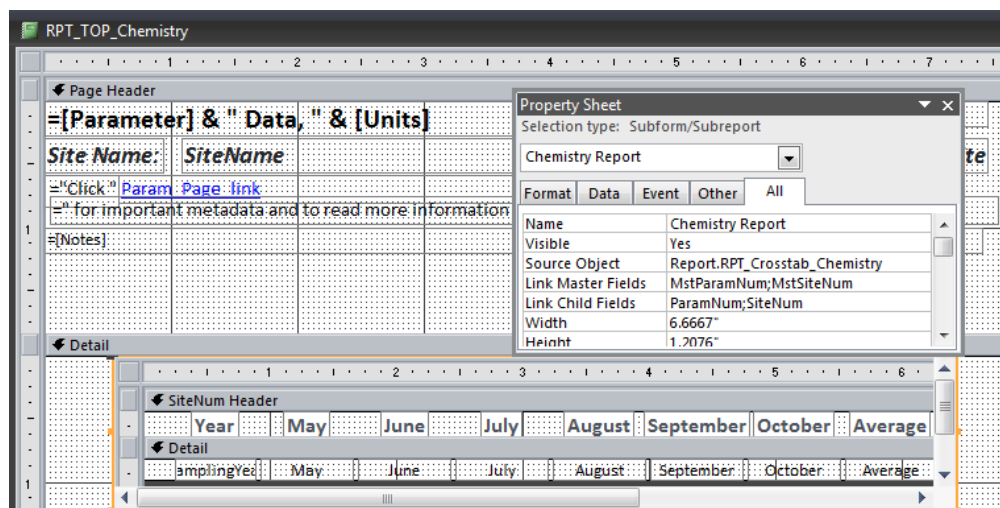


Figure 43. RPT_TOP_Chemistry showing RPT_Crosstab_Chemistry as the tabular subreport's Source Object. Note the Link Master/Child Fields.

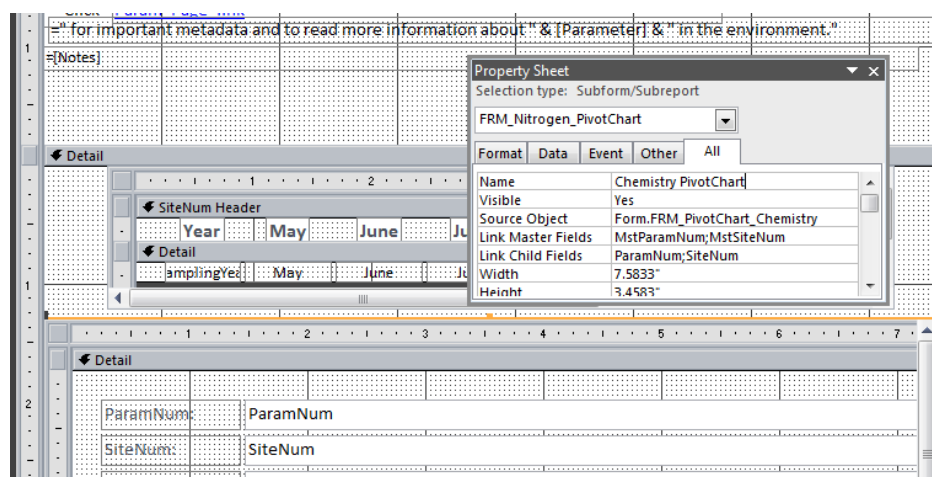


Figure 44. RPT_TOP_Chemistry showing FRM_PivotChart_Chemistry as the PivotChart subreport's Source Object. Note the Link Master/Child Fields.

Nitrogen, Ammonium Data, ug/L

Site Name: Falls R @ Twin Bridges (A) (#2)

Number: 4230

Click <http://www.edc.uri.edu/wpwa/WPWSites/Ammonium.PDF>

for important metadata and to read more information about Nitrogen, Ammonium in the environment.

No data for 1989 - 1996. >1000 ppb indicates human impact. Mean calculated using half the limit of detection for ND (Not Detected).

Year	May	June	July	August	September	October	Average
1997		10	40		5	30	21
1998	5	90			50	50	49
1999	40	40			10	40	33
2000	50	30	10			10	25
2001			50	40	15	15	30
2002	10	10	10	30	10	10	13
2003	40	80	10	50	20	40	40
2004	80		60	15	15		43
2005	15		15	15	15	60	24
2006	15				50	50	38
2007	35	85	60	13	16	13	37
2008	30	40	20	10	10	10	20

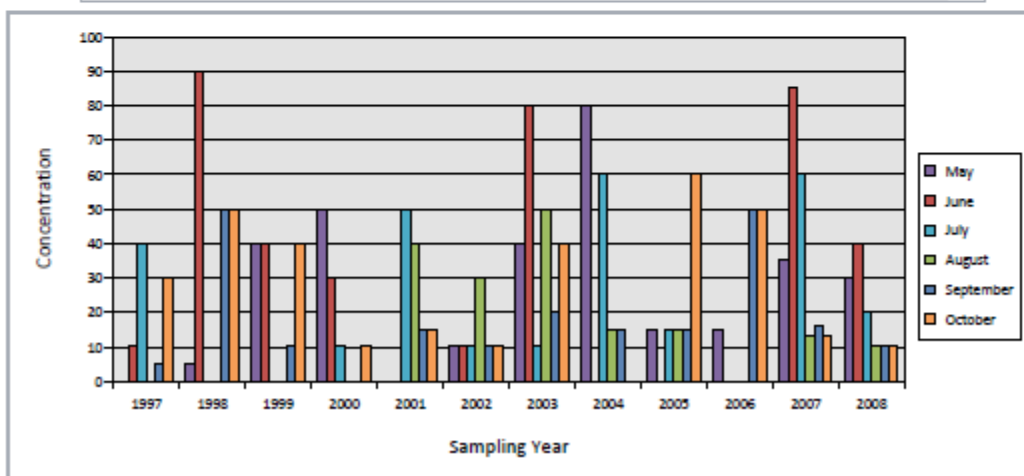


Figure 45. Example of a top-level chemistry data report.

3.3.5. Chemistry Reports with Threshold Lines

A special top-level report was created to allow plotting of threshold lines along with chemistry data. Currently, only chloride data are plotted using this form (total phosphorus with trophic status data are also plotted with threshold lines, but this used a different procedure and is described later). The top-level report name is *RPT_TOP_Chemistry_Threshold*, and it is very similar to *RPT_TOP_Chemistry*. Like *RPT_TOP_Chemistry*, *RPT_TOP_Chemistry_Threshold* uses the *QRY_For_Rpt_TOP_Chemistry* query as the data source for the main report page. Also like *RPT_TOP_Chemistry*, *RPT_TOP_Chemistry_Threshold* uses *RPT_Crosstab_Chemistry* as its tabular subreport. However, *RPT_TOP_Chemistry_Threshold* uses a special PivotChart form—*FRM_PivotChart_Chemistry_Threshold*--created to allow plotting of threshold lines. Like with the chlorophyll and Secchi data, this form uses a UNION query named *QRY_PivotChart_Chemistry_Threshold* to concatenate the appropriate threshold data from *TBL_Thresholds* onto the actual data (Figure 46). The threshold data are then reformatted as lines in the plot as described earlier. The threshold PivotChart query and form are generic enough that, in the

future, additional chemistry parameters can be plotted with threshold lines without needing to make any modifications (Figure 47).

```

SELECT QRY_PivotChart_Chemistry.ParamNum, QRY_PivotChart_Chemistry.SiteNum, QRY_PivotChart_Chemistry.SamplingYear, QRY_PivotChart_Chemistry.MonthText,
QRY_PivotChart_Chemistry.Concentration, QRY_PivotChart_Chemistry.SamplingMonth
FROM QRY_PivotChart_Chemistry
UNION (SELECT DISTINCT TBL_One_Param_For_Report.ParamNum, TBL_One_Param_For_Report.SiteNum, TBL_One_Param_For_Report.SamplingYear,
TBL_Thresholds.MonthText, TBL_Thresholds.Concentration, TBL_Thresholds.SamplingMonth
FROM (TBL_One_Param_For_Report INNER JOIN TBL_Thresholds ON TBL_One_Param_For_Report.ParamNum = TBL_Thresholds.ParamNum));

```

Figure 46. UNION query *QRY_PivotChart_Chemistry_Threshold*.

If chloride reports are being generated, the *Generate_Chemistry_Reports* VB function passes the *RPT_TOP_Chemistry_Threshold* report name to the *Make_Chemistry_Reports* VB subroutine (instead of *RPT_TOP_Chemistry*). Otherwise, the report generation procedures are the same as for the rest of the chemistry data.

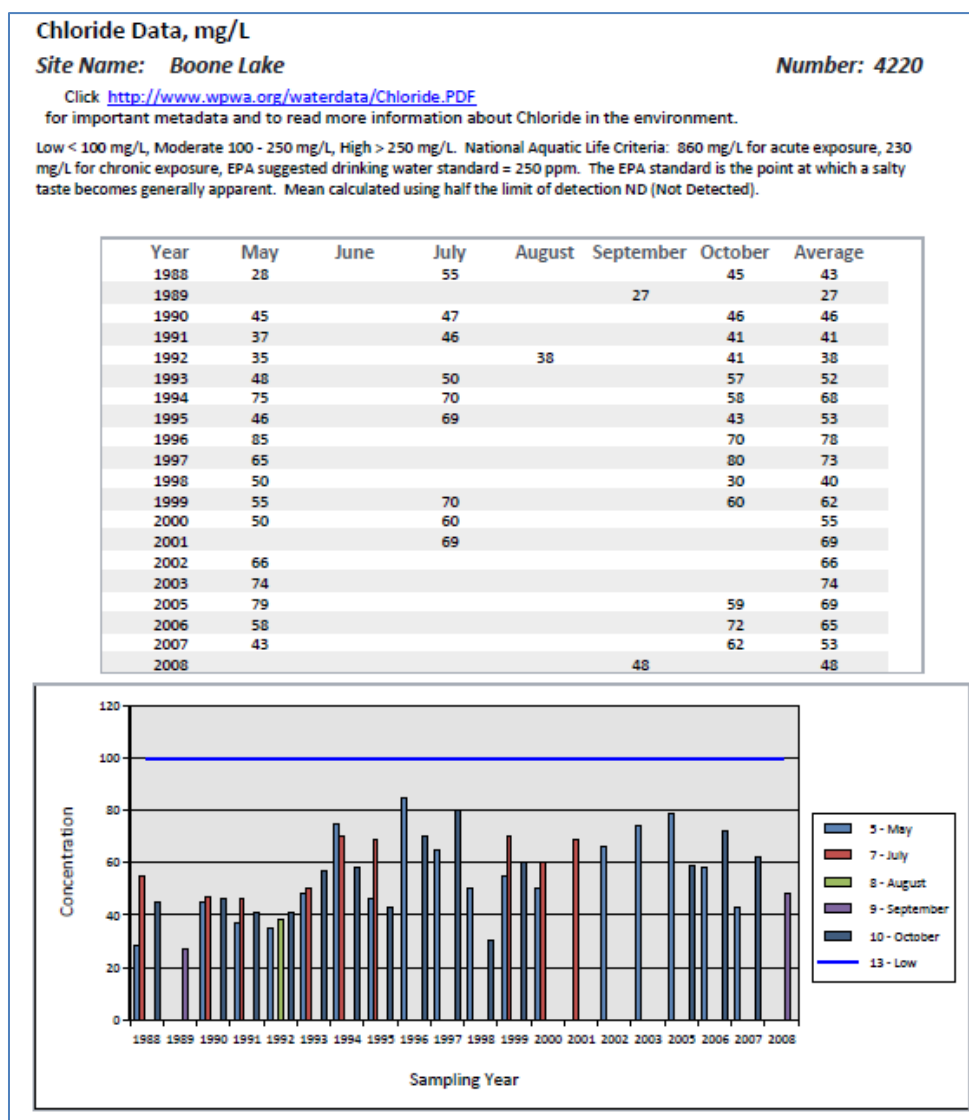


Figure 47. Example of a chloride report showing a threshold line.

3.4. Generating Total Phosphorus Reports With and Without Trophic Status

For lakes and ponds, total phosphorus surface water data were used to calculate trophic status (TSI), but it would be inappropriate to calculate/present TSI for deep water or river/stream samples. Therefore, it was necessary to process lake/pond surface water data separately in order to calculate and present TSI only where appropriate. The *QRY_Param_Sites_Phosphorus_Trophic* query uses the *LakePond* field in the *MasterSiteList* table to make a list of sites for which total phosphorus trophic status should be calculated (Figure 48). The *SiteNum* field was given the alias *MstSiteNum* for later linking to subreports.

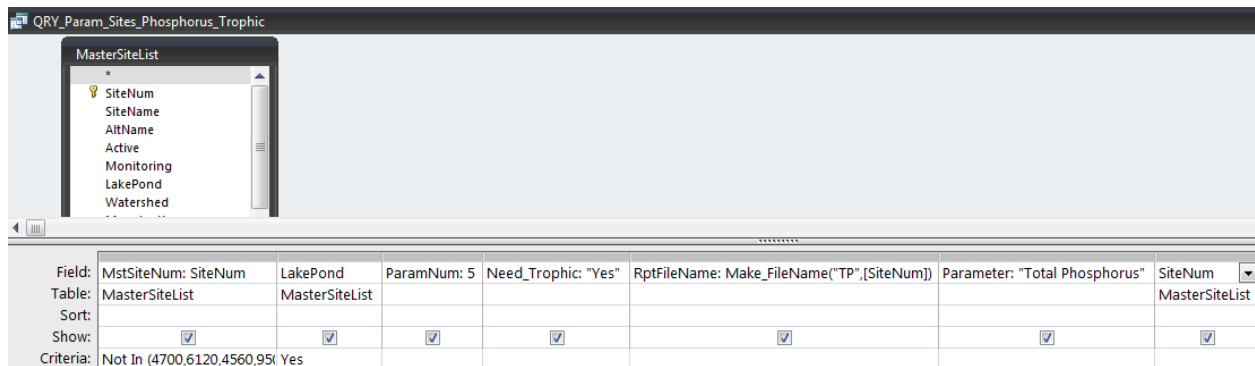


Figure 48. *QRY_Param_Sites_Phosphorus_Trophic* query.

The *QRY_Param_Sites_TP* query then uses the results of the *QRY_Param_Sites_Phosphorus_Trophic* query to exclude data from the lake/pond sites from the rest of the total phosphorus data in (Figure 49).

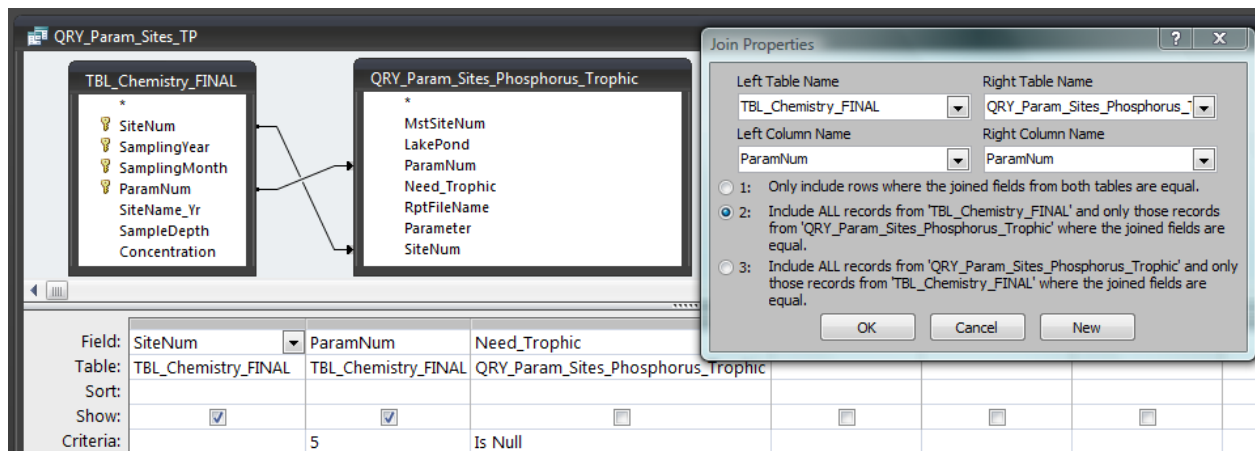


Figure 49. *QRY_Param_Sites_TP* query. All of the *Need_Trophic* fields in *QRY_Param_Sites_Phosphorus_Trophic* contain the value "Yes", so these records are excluded from *QRY_Param_Sites_TP* by only keeping total phosphorus records from *TBL_Chemistry_FINAL* without matching records in *QRY_Param_Sites_Phosphorus_Trophic*. The unmatched records have no data in the *Need_Trophic* field ("Is Null").

Both of these *QRY_Param_Sites_* queries have corresponding queries that exclude sites not sampled in the latest sampling year. These two queries are named *QRY_Param_Sites_New_Year_Phosphorus_Trophic* and *QRY_Param_Sites_New_Year_TP*, and were

constructed by linking the *QRY_Param_Sites_* queries with *QRY_Sites_With_Latest_Year_of_Data* (Figure 50). For total phosphorus WITHOUT trophic status, the *QRY_Param_Sites_* queries are used by the *Generate_Chemistry_Reports* function when it calls the *Make_Chemistry_Reports* subroutine to produce reports, just like with the rest of the chemistry data. In addition, the rest of the queries and reports used to create the total phosphorus WITHOUT TSI report pages are very similar to those described for the rest of the chemistry data and include: *QRY_Crosstab_TP* (also used as the basis for the total phosphorus with TSI crosstab query), *RPT_Crosstab_TP*, *QRY_PivotChart_TP* (also used as the basis for the total phosphorus with TSI PivotChart query), *FRM_PivotChart_TP*, *QRY_For_Rpt_TOP_TP*, and *RPT_TOP_TP* (Figure 51).

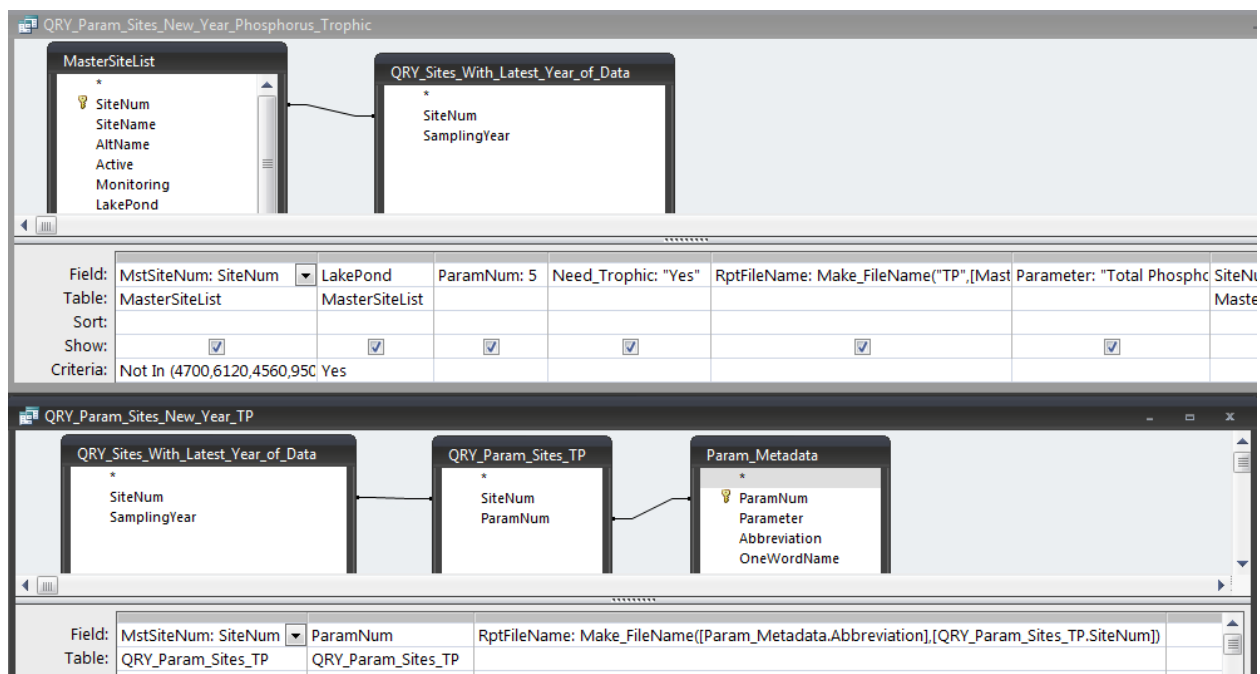


Figure 50. *QRY_Param_Sites_New_Year_Phosphorus_Trophic* and *QRY_Param_Sites_New_Year_TP*.

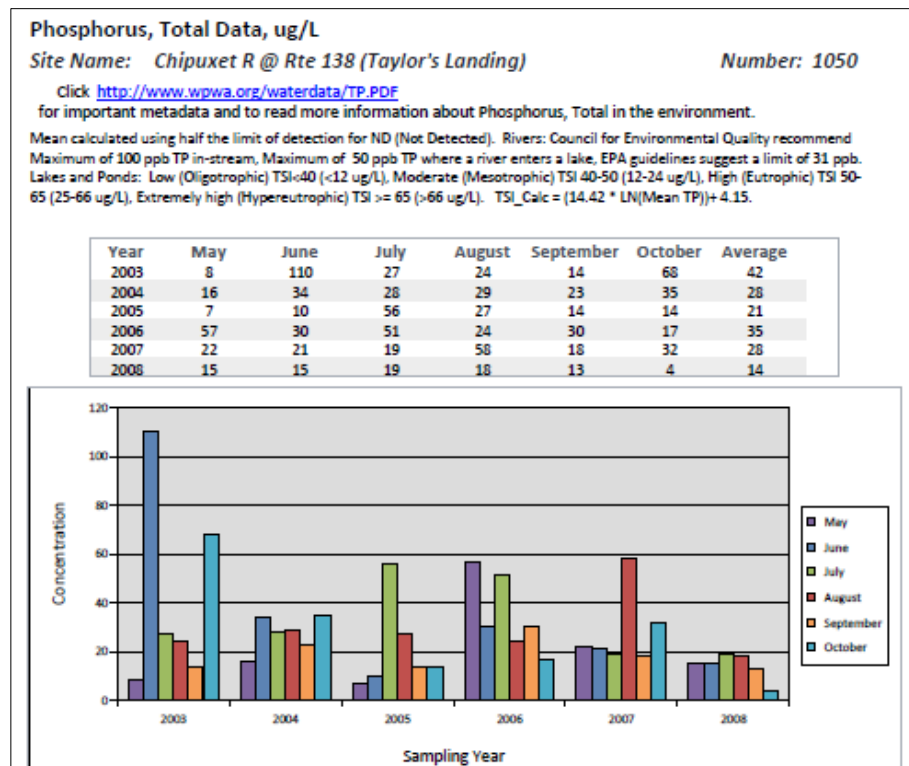


Figure 51. Example of a total phosphorus report WITHOUT trophic status (TSI).

The total phosphorus WITH TSI reports are produced somewhat differently. If "Total Phosphorus, Lakes/Ponds with Trophic Status" is selected from *Frm_Report_Options*, the number "18" is passed into the *ReportIt* macro instead of the *ParamNum*. This is because the total phosphorus *ParamNum* (5) was already used to generate reports for total phosphorus without TSI. The *ReportIt* macro calls the VB function named *TP_TSI_Report* (Figure 52), passing the report sites options number (1, 2, or 3) and the *SiteNum* for which to generate a report if the report sites option is "3" (one site only) into the function as arguments. The *TP_TSI_Report* function first empties the *TBL_One_Param_For_Report* table, and then either adds total phosphorus data for one site (if the report sites option value is "3") or all sites (report sites option of "1" or "2") to the table. The function then calls the *Make_Chemistry_Reports* subroutine using *RPT_TOP_Phosphorus_Trophic* as the report name and either the *QRY_Param_Sites_New_Year_Phosphorus_Trophic* query (report site option "2", Figure 50 above) or the *QRY_For_Rpt_TOP_Phos_Trophic* query (report site options "1" and "3") to create the report generation recordset. Finally, the function empties the *TBL_One_Param_For_Report* table and exits.

```

'-----
' Generate Total Phosphorus with TSI reports by calling Make_Chemistry_Reports
subroutine
'-----
Public Function TP_TSI_Report(WhichSites As Integer, RptSite As Integer)

    ' empty the temporary data table just in case...
    DoCmd.RunSQL "DELETE TBL_One_Param_For_Report.* FROM TBL_One_Param_For_Report;"

    If WhichSites = 3 Then

```



```

DoCmd.RunSQL "INSERT INTO TBL_One_Param_For_Report SELECT TBL_Chemistry_FINAL.*
FROM TBL_Chemistry_FINAL WHERE TBL_Chemistry_FINAL.ParamNum=5 and
TBL_Chemistry_FINAL.SiteNum=" & RptSite & ";", -1
Else
DoCmd.RunSQL "INSERT INTO TBL_One_Param_For_Report SELECT TBL_Chemistry_FINAL.*
FROM TBL_Chemistry_FINAL WHERE TBL_Chemistry_FINAL.ParamNum=5;", -1
End If

If WhichSites = 2 Then
Call Make_Chemistry_Reports("RPT_TOP_Phosphorus_Trophic",
"QRY_Param_Sites_New_Year_Phosphorus_Trophic", "Phosphorus", 0, WhichSites)
Else
Call Make_Chemistry_Reports("RPT_TOP_Phosphorus_Trophic",
"QRY_For_Rpt_TOP_Phos_Trophic", "Phosphorus", 0, WhichSites)
End If

' re-empty the temporary data table just in case...
DoCmd.RunSQL "DELETE TBL_One_Param_For_Report.* FROM TBL_One_Param_For_Report;", -1
End Function

```

Figure 52. The *TP_TSI_Report* function.

Notice that the *QRY_For_Rpt_TOP_Phos_Trophic* query is passed into *Make_Chemistry_Reports* instead of *QRY_Param_Sites_Phosphorus_Trophic* to create the report generation recordset (the list of sites for which to generate reports) for report site options “1” and “3”. Adding the option of creating a report for one site only made total phosphorus with TSI report generation a bit more complex. First, a query named *QRY_Param_Sites_One_Site_Phos_Trophic* was created to generate a simple list of *SiteNums* with data currently in the *TBL_One_Param_For_Report* table (Figure 53). Thanks to the *TP_TSI_Report* function having selectively filled the temporary table, this list would include either ALL *SiteNums* with total phosphorus data or only the single *SiteNum* passed in for report site option “3”.

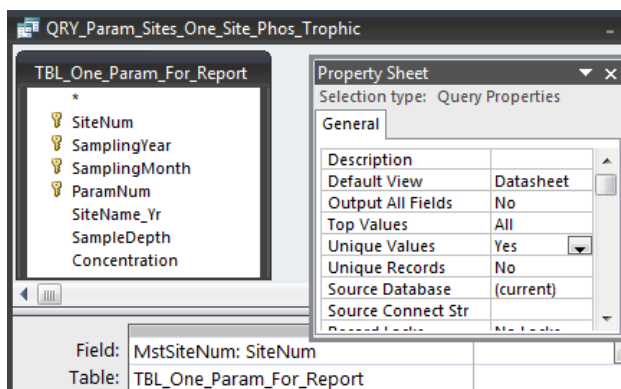


Figure 53. *QRY_Param_Sites_One_Site_Phos_Trophic* query, which generates a list of *SiteNums* with data currently in the *TBL_One_Param_For_Report* table.

Next, this query is linked to *QRY_Param_Sites_Phosphorus_Trophic* within the *QRY_For_Rpt_TOP_Phos_Trophic* query in order to include only the *SiteNum* for which a report is currently being generated for report site option “3”; otherwise, all *SiteNums* with TSI data are included (Figure 54). This is the top-level query used by *RPT_TOP_Phosphorus_Trophic*.

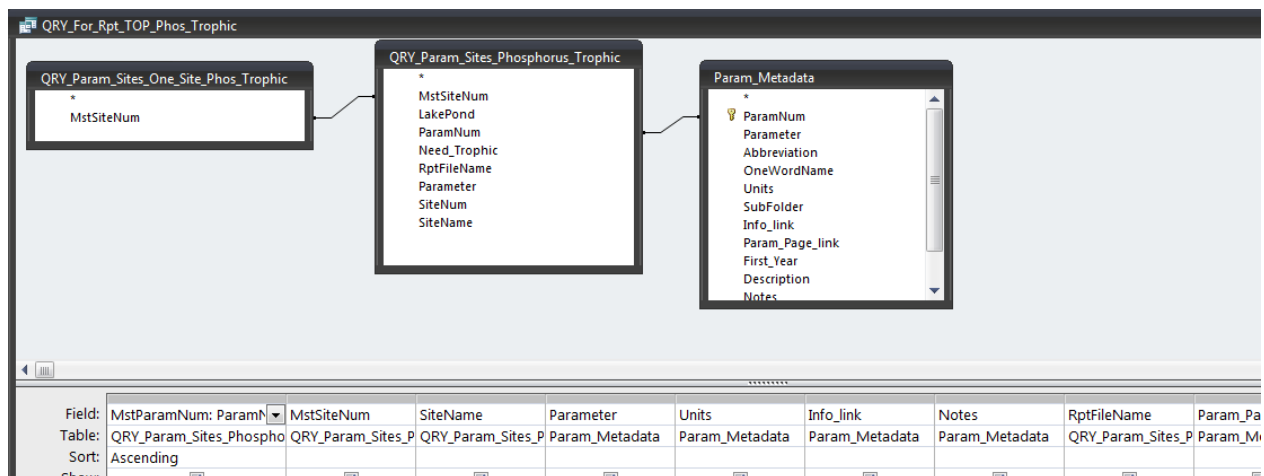


Figure 54. *QRY_For_Rpt_TOP_Phos_Trophic* query.

The subreport containing the data table is named *RPT_Crosstab_Phos_Trophic* and is linked to the top-level report by linking master/child fields as described before. This subreport is created using a query named *QRY_Crosstab_Phosphorus_Trophic* (Figure 55). *QRY_Crosstab_Phosphorus_Trophic* uses the *QRY_Param_Sites_Phosphorus_Trophic* query to keep the subset of the data in the *QRY_Crosstab_TP* crosstab query with TSI data. In addition, *QRY_Crosstab_Phosphorus_Trophic* calls the VB function *Phos_TSI* to calculate trophic status numbers using the average total phosphorus value for each *SiteNum* and *SamplingYear*.

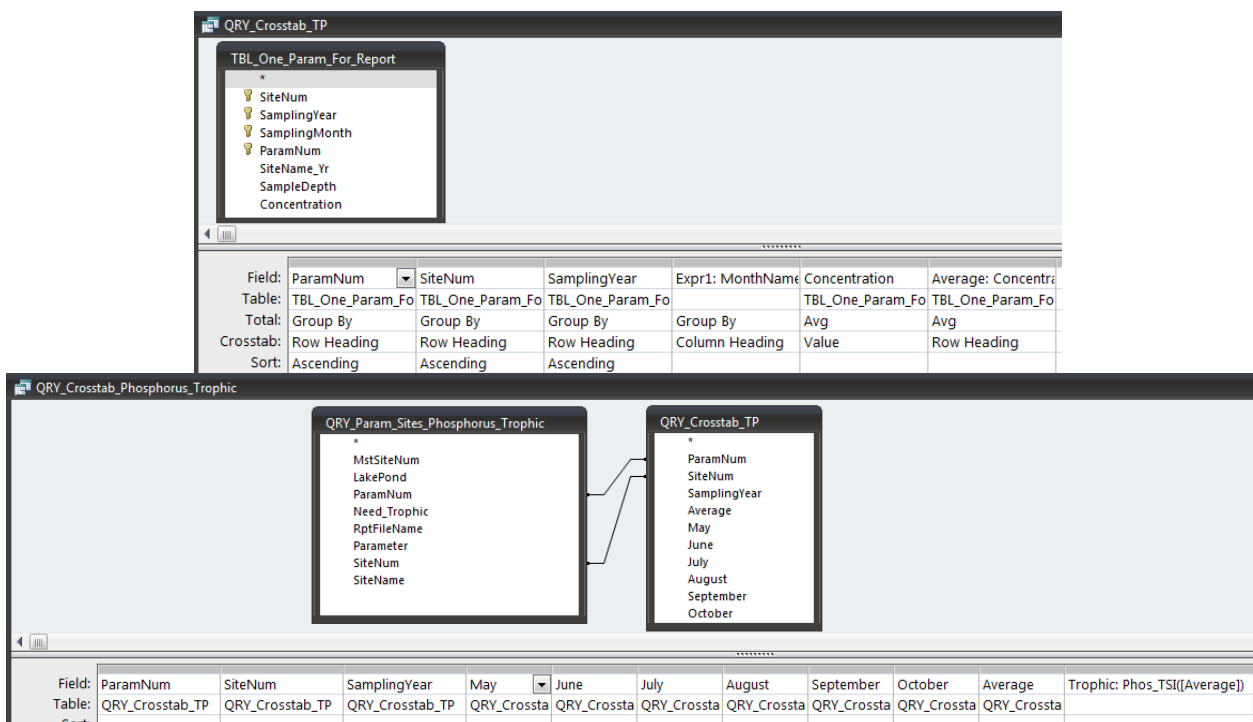


Figure 55. *QRY_Crosstab_TP* and *QRY_Crosstab_Phosphorus_Trophic* queries. Notice the call to the VB function *Phos_TSI* to calculate trophic status numbers.

The subreport containing the data plot is named *FRM_PivotChart_Phos_Trophic* and is also linked to the top-level report by linking master/child fields as described before. This subreport is created based on the *QRY_PivotChart_TP* query (mentioned previously) using a UNION query named *QRY_PivotChart_TP_Threshold* (Figure 56). As with the chlorophyll and Secchi data, the UNION query adds the trophic status threshold data to actual total phosphorus data to allow plotting of informative threshold lines. The threshold data were assigned *SamplingMonth* values of 13 (*MonthText* = “Low”), 14 (*MonthText* = “Moderate”), and 15 (*MonthText* = “High”). The data were plotted and threshold lines reformatted in the same way as with the Chlorophyll and Secchi data. The final report is shown in Figure 57.

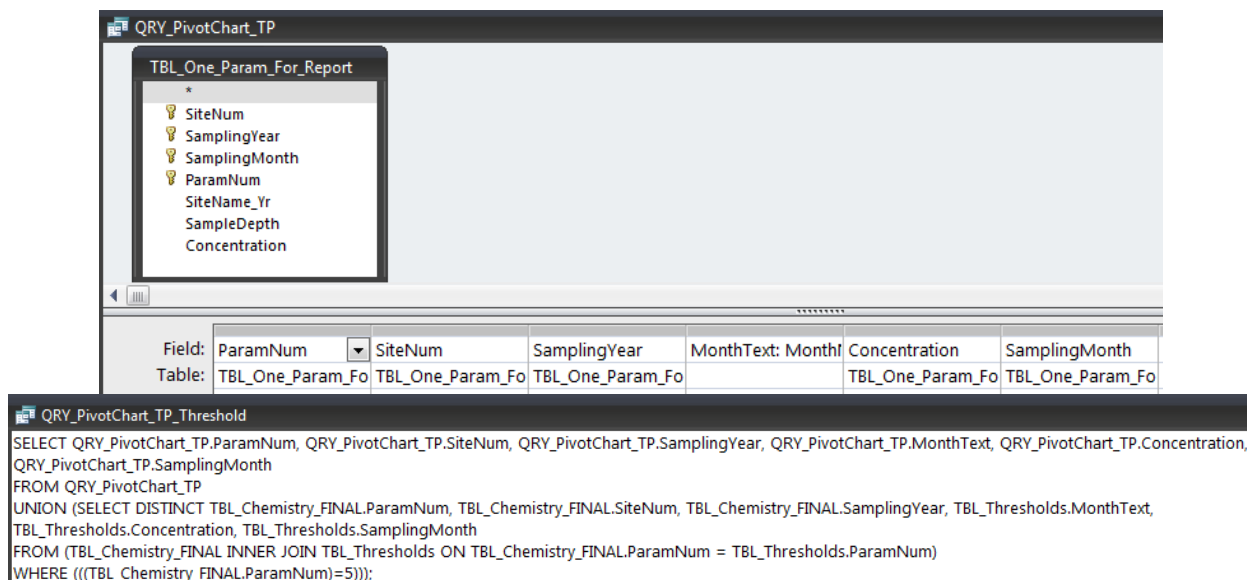


Figure 56. *QRY_PivotChart_TP* and *QRY_PivotChart_TP_Threshold* queries.

Phosphorus, Total Data, ug/L

Site Name: Worden Pond

Number: 1060

Click <http://www.wpwa.org/waterdata/TP.PDF>

for important metadata and to read more information about Phosphorus, Total in the environment.

Mean calculated using half the limit of detection for ND (Not Detected). Rivers: Council for Environmental Quality recommend Maximum of 100 ppb TP in-stream, Maximum of 50 ppb TP where a river enters a lake, EPA guidelines suggest a limit of 31 ppb. Lakes and Ponds: Low (Oligotrophic) TSI<40 (<12 ug/L), Moderate (Mesotrophic) TSI 40-50 (12-24 ug/L), High (Eutrophic) TSI 50-65 (25-66 ug/L), Extremely high (Hypereutrophic) TSI >= 65 (>66 ug/L). $TSI_Calc = (14.42 * \ln(\text{Mean TP})) + 4.15$.

Year	May	June	July	August	September	October	Average	Trophic Status
1988			50			16	33	E
1989	20		39			18	26	E
1990	18		24			9	17	M
1991	23		30			40	31	E
1992	19			36		10	22	M
1993	16						16	M
1995	17		14			12	14	M
1997	14		27			13	18	M
1998	18		17			14	16	M
1999	10		22			23	18	M
2000	23		26			13	21	M
2001	15		22			31	23	M
2003			31				31	E
2004	15		21			17	18	M
2005	15		24				20	M
2006	18		2			21	14	M
2007	14			46		26	29	E
2008	17		18			16	17	M
2009	11			23		31	22	M

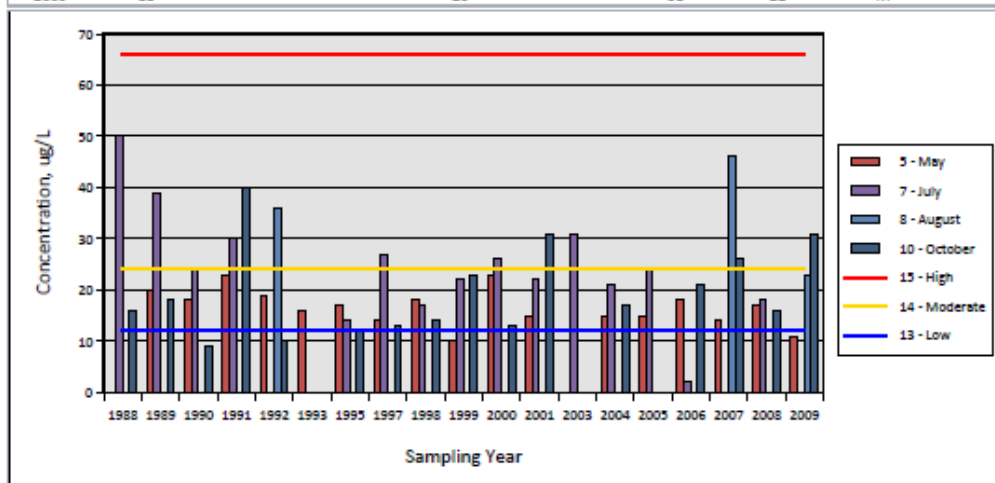


Figure 57. Total phosphorus report with trophic status threshold lines.

Recall that a query is passed into the *Make_Chemistry_Reports* subroutine that generates the recordset used for report generation. If for some reason the query produces an empty recordset, the *Make_Chemistry_Reports* subroutine will output a message to the user to alert them (Figure 58) and attempt to let the user know what the problem may be (for example, asking for total phosphorus with TSI data or alkalinity data for a non-lake/pond site).

```

...
...
If rcdRecQuery.EOF Then
    If WSites = 3 And RPT_TOP_Name = "RPT_TOP_Phosphorus_Trophic" Then
        msgbox ("No data for this site! Be sure you have chosen a lake/pond SURFACE
water site (Site Number ending in 0).")
    Else

```

```

        msgbox ("No data for this site! Please be sure you have chosen a report and
site combination that is likely to exist. For example, there are only Alkalinity data
for lake/pond SURFACE samples.")
    End If
Else
    ' Start looping through the QRY_Param_Sites_Chemistry file to generate a one page
    ' report for each SiteNum. This loop will process every SiteNum and terminate
    ' when the end of the file (EOF) is reached.
...
...

```

Figure 58. Messages output by the *Make_Chemistry_Reports* subroutine if the report generation recordset is empty.

3.5. Generating Bacteria Reports

Bacteria data are stored and reported apart from the rest of the data due to their format. To begin, one of the three bacteria options (“E. coli”, “Enterococci”, or “Fecal Coliform”) is selected from the *FRM_Report_Options* form (Figure 4). For each type of bacteria, there are two possible courses of action from the *DoReport.ReportIt* macro (Figure 59). The Visual Basic function *Make_Bacteria_Reports* is called to generate reports for all three types of bacteria data, but a different query is used to create the site list for report generation depending on the value of the report site option (the first action if the value is “2” (only sites with the latest year of data), the second action if the value is “1” or “3”). Since E. coli bacteria has not been analyzed in several years, choosing E. coli (*ParamNum*=14) along with report site option “2” results in an error message and no other action.

Condition	Action	Arguments
12=[Forms]![Frm_Report_Options]![Rpt_Opt] And 2=[Forms]![Frm_Report_Site_Options]![Rpt_Site]	RunCode	Make_Bacteria_Reports([Forms]![Frm_Report_Options]![Rpt_Opt], "QRY_Param_Sites_New_Year_Bacteria", [Forms]![Frm_Report_Site_Options]![Rpt_Site], Prompt)
...	Close	
...	StopMacro	
12=[Forms]![Frm_Report_Options]![Rpt_Opt]	RunCode	Make_Bacteria_Reports([Forms]![Frm_Report_Options]![Rpt_Opt], "QRY_For_Rpt_TOP_Bacteria", [Forms]![Frm_Report_Site_Options]![Rpt_Site], Prompt)
...	Close	
...	StopMacro	
13=[Forms]![Frm_Report_Options]![Rpt_Opt] And 2=[Forms]![Frm_Report_Site_Options]![Rpt_Site]	RunCode	Make_Bacteria_Reports([Forms]![Frm_Report_Options]![Rpt_Opt], "QRY_Param_Sites_New_Year_Bacteria", [Forms]![Frm_Report_Site_Options]![Rpt_Site], Prompt)
...	Close	
...	StopMacro	
13=[Forms]![Frm_Report_Options]![Rpt_Opt]	RunCode	Make_Bacteria_Reports([Forms]![Frm_Report_Options]![Rpt_Opt], "QRY_For_Rpt_TOP_Bacteria", [Forms]![Frm_Report_Site_Options]![Rpt_Site], Prompt)
...	Close	
...	StopMacro	
14=[Forms]![Frm_Report_Options]![Rpt_Opt] And 2=[Forms]![Frm_Report_Site_Options]![Rpt_Site]	MsgBox	E. coli is no longer analyzed--there are no data from the most recent year of sampling. If you would like to regenerate the report, please select a different bacteria type.
...	Close	
...	StopMacro	
14=[Forms]![Frm_Report_Options]![Rpt_Opt]	RunCode	Make_Bacteria_Reports([Forms]![Frm_Report_Options]![Rpt_Opt], "QRY_For_Rpt_TOP_Bacteria", [Forms]![Frm_Report_Site_Options]![Rpt_Site], Prompt)
...	Close	
...	StopMacro	

Figure 59. Macro actions for bacteria report generation.

The VB function (Figure 60) takes four arguments: *Param* (the parameter number), *QRY_Name* (the name of the query to use as the site list for report generation—either *QRY_Param_Sites_New_Year_Bacteria* (site report option = 2) or *QRY_For_Rpt_TOP_Bacteria* (site report options 1 or 3)), *WhichSites* (1=all sites, 2=sites sampled in the latest year, 3=one site only), and *RptSite* (the *SiteNum* to report if *WhichSites*=3).

```

'-----
' Generate Bacteria reports. This procedure accepts the report name, query providing
' the station list, and subfolder name where the PDF files will be output as
' arguments.
'-----

```

```

Public Function Make_Bacteria_Reports(Param As Integer, QRY_Name As String, WhichSites
As Integer, RptSite As Integer)

' Error handling
On Error GoTo Make_Bacteria_Reports_Err

' Declare the database, recordset, and report filter string variables
Dim dbwpwa As DAO.Database
Dim rcdRecQuery As DAO.Recordset
Dim strFilter, strQuery As String

If WhichSites = 3 And RptSite Like "###1" Then
    MsgBox ("There are no deep water bacteria samples. Please select a SURFACE water
site (Site Number ending in 0).")
    Exit Function
End If

DoCmd.RunSQL "DELETE TBL_One_Bacteria_For_Report.* FROM TBL_One_Bacteria_For_Report;",
-1
DoCmd.OpenQuery ("QRY_Clear_TBL_One_Bacteria_For_Report_Thresholds")

If WhichSites = 3 Then
    SQL_string = "INSERT INTO TBL_One_Bacteria_For_Report
SELECT TBL_Bacteria_Vertical_NUMERIC.* FROM TBL_Bacteria_Vertical_NUMERIC
WHERE TBL_Bacteria_Vertical_NUMERIC.ParamNum=" & Param & " and
TBL_Bacteria_Vertical_NUMERIC.SiteNum=" & RptSite & ";"
Else
    SQL_string = "INSERT INTO TBL_One_Bacteria_For_Report
SELECT TBL_Bacteria_Vertical_NUMERIC.* FROM TBL_Bacteria_Vertical_NUMERIC
WHERE TBL_Bacteria_Vertical_NUMERIC.ParamNum=" & Param & ";"
End If

' insert desired records into temporary data table
DoCmd.RunSQL SQL_string, -1

DoCmd.OpenQuery ("QRY_PivotChart_Bacteria")
DoCmd.OpenQuery ("QRY_PivotChart_Bacteria_Threshold")

Set dbwpwa = CurrentDb
Set rcdRecQuery = dbwpwa.OpenRecordset(QRY_Name)

' Make the mouse an hourglass to indicate a wait
DoCmd.Hourglass True

' Start looping through the QRY_Param_Sites_Bacteria file to generate a one page
' report for each SiteNum. This loop will process every SiteNum and terminate
' when the end of the file (EOF) is reached.

If rcdRecQuery.EOF Then
    MsgBox ("There are no bacteria data for this site!")
Else
    Do Until rcdRecQuery.EOF
        ' Set the filter for the report to the current SiteNum
        strFilter = "[MstSiteNum]=" & rcdRecQuery![MstSiteNum]
        ' Open the report
        DoCmd.OpenReport "RPT_TOP_Bacteria", acViewPreview, "", strFilter, acHidden
        ' Output the file
        DoCmd.OutputTo acOutputReport, "RPT_TOP_Bacteria", "PDFFormat(*.pdf)",
"C:\Users\Elise Torello\documents\WPWA\Field Data\Master Access
Database\WPWASites\Bacteria\" & rcdRecQuery![RptFileName], False, "", 0,
acExportQualityPrint
        ' Close the report
        DoCmd.Close acReport, "RPT_TOP_Bacteria"
    
```

```

        ' Move on to the next record in the query
        rcdRecQuery.MoveNext
        'and loop
    Loop
End If

DoCmd.RunSQL "DELETE TBL_One_Bacteria_For_Report.* FROM TBL_One_Bacteria_For_Report;",
-1

' Turn off the hourglass
DoCmd.Hourglass False

Make_Bacteria_Reports_Exit:
    Exit Function

Make_Bacteria_Reports_Err:
    msgbox Error$
    Resume Make_Bacteria_Reports_Exit
End Function

```

Figure 60. The function Make_Bacteria_Reports to generate reports for all three types of bacteria data.

First, the variables needed by the function are declared. Then, if the site reporting option passed into the function is “3” (report for one site only), the function checks to make sure that the *RptSite* value is for a surface water site. If it is not, an error message is output and the function exits.

The next lines prepare the two temporary tables (*TBL_One_Bacteria_For_Report* and *TBL_One_Bacteria_For_Report_Thresholds*) used during report generation. First, both tables are emptied. Then, *TBL_One_Bacteria_For_Report* is refilled with the appropriate data needed for report generation depending on the value of *WhichSites*. The two “DoCmd.OpenQuery” lines run two APPEND queries (*QRY_PivotChart_Bacteria* and *QRY_PivotChart_Bacteria_Threshold* (Figure 61)), which use the data in *TBL_One_Bacteria_For_Report* and *TBL_Thresholds* to fill *TBL_One_Bacteria_For_Report_Thresholds*. This temporary table is used to create the PivotCharts of bacteria data including threshold lines.

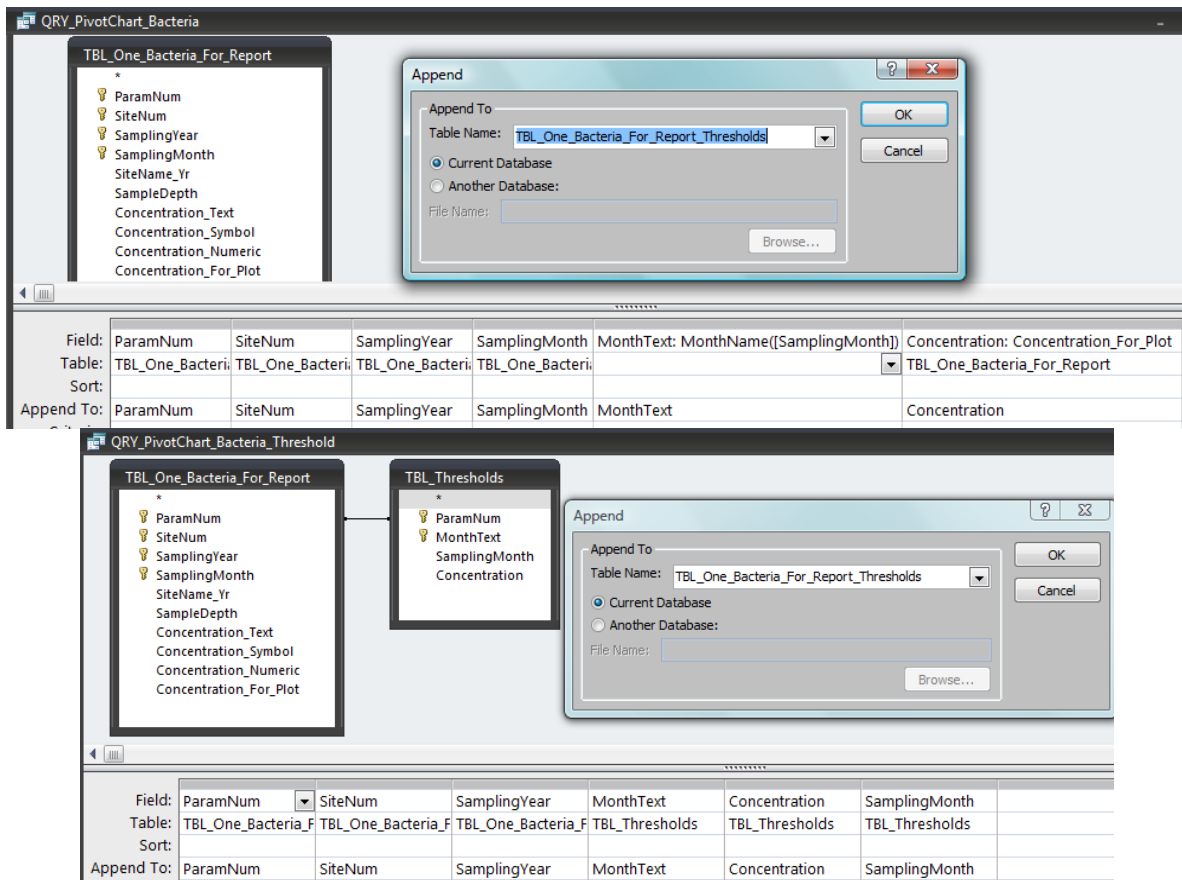
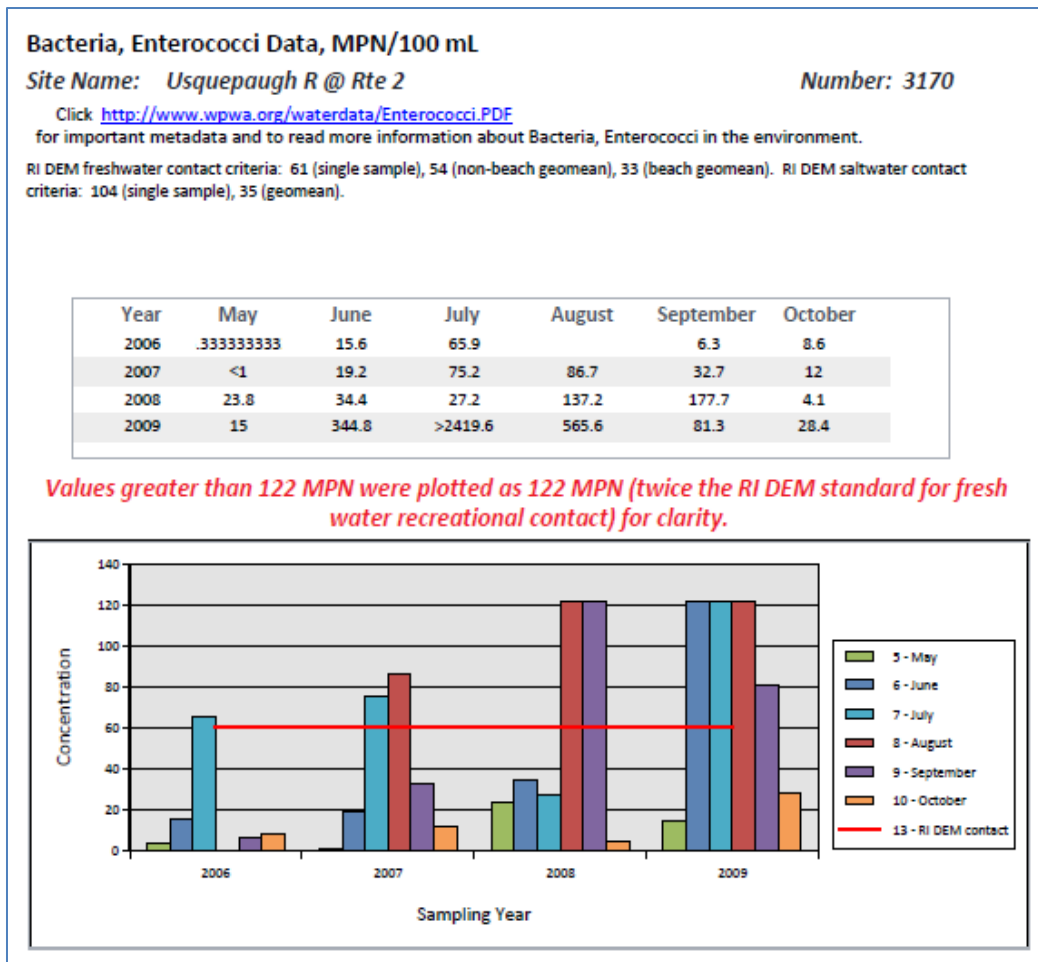


Figure 61. APPEND queries QRY_PivotChart_Bacteria and QRY_PivotChart_Bacteria_Threshold. These queries are executed to fill the temporary table TBL_One_Bacteria_For_Report_Thresholds. Notice the alias given to the Concentration_For_Plot field in QRY_PivotChart_Bacteria, and the "Append To:" fields in both queries.

The VB function then performs very similarly to previous report generating functions. It sets the current database variable, opens a recordset which is generated by the query whose name was passed into the function as an argument, then loops through the recordset generating a one-page PDF report for each SiteNum. When the end of the recordset (*rcdRecQuery.EOF*) is reached, the temporary tables are emptied and the function exits.

The top-level report for bacteria is named *RPT_TOP_Bacteria*. As with previously described reports, the top-level report uses *QRY_Param_Sites_Bacteria*, and then *QRY_For_Rpt_TOP_Bacteria*, queries as data sources. The tabular subreport is named *RPT_Crosstab_Bacteria* and uses *QRY_Crosstab_Bacteria_Vertical* as its data source. The PivotChart subreport is named *FRM_PivotChart_Bacteria_Threshold* and uses *TBL_One_Bacteria_For_Report_Thresholds* as its data source. The threshold data are reformatted as lines in the PivotChart as described previously. Both subreports are linked to the top-level report using master/child fields as in previous reports. The final product is shown in Figure 62.



4. Data Portal (Sampling Site Information) Pages

The next set of procedures generates a sampling site summary, or data portal, report for each site. The entire process is described in Appendix 2.

To generate the reports, it was necessary to generate a list of PDF report file names ONLY for *SiteNum/ParamNum* combinations that exist and therefore have report pages. For example, some sites in the *MasterSiteList* table are not water quality monitoring sites, but were part of special studies, had only macroinvertebrate sampling, etc. Also, some water quality sampling sites did not have every parameter sampled (for example, chlorophyll, Secchi depth, and alkalinity are only sampled in lakes/ponds). The file names generated are included as hyperlinks to the actual PDF data reports from each sampling site data portal page.

First, four queries were designed that list all sites sampled for each chemistry, bacteria, Secchi depth, and chlorophyll *ParamNum*: *QRY_Param_Sites_ALL_Chemistry*, *QRY_Param_Sites_ALL_Bacteria*, *QRY_Param_Sites_ALL_Secchi*, and *QRY_Param_Sites_ALL_Chlorophyll*. Each query creates a recordset containing one record for each site sampled for each parameter in each data table by setting the "Unique Values" property for the query to "Yes". This is shown in for *QRY_Param_Sites_ALL_Chemistry*

(Figure 63)—the other three queries are the same except for the source data. Instead of *TBL_Chemistry_Final*, *QRY_Param_Sites_ALL_Bacteria* uses *TBL_Bacteria_Vertical_NUMERIC*, *QRY_Param_Sites_ALL_Secchi* uses *QRY_Rpt_Secchi*, and *QRY_Param_Sites_ALL_Chlorophyll* uses *QRY_Rpt_Chlorophyll*.

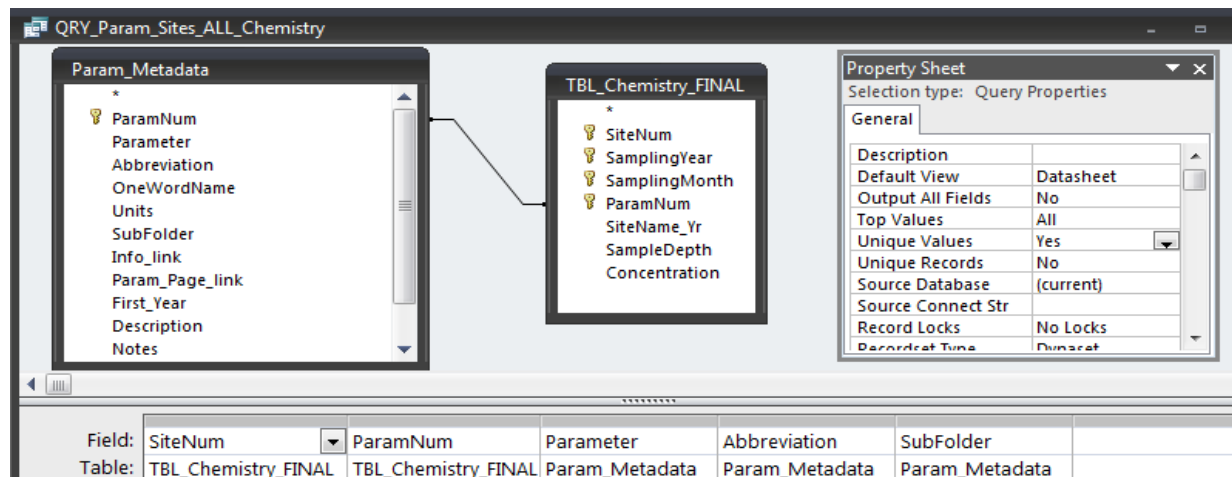


Figure 63. *QRY_Param_Sites_ALL_Chemistry* query. Notice that “Unique Values” is set to “Yes”.

Next, a UNION query named *All_Site_Params_For_Site_Pages* was created from these four queries. The UNION query lists all of the sites with data for each parameter. This query had to be designed in SQL and is very slow to run (Figure 64).

```

All_Site_Params_For_Site_Pages
SELECT QRY_Param_Sites_ALL_Chemistry.*
FROM QRY_Param_Sites_ALL_Chemistry
UNION
SELECT QRY_Param_Sites_ALL_Bacteria.*
FROM QRY_Param_Sites_ALL_Bacteria
UNION
SELECT QRY_Param_Sites_ALL_Secchi.*
FROM QRY_Param_Sites_ALL_Secchi
UNION SELECT QRY_Param_Sites_ALL_Chlorophyll.*
FROM QRY_Param_Sites_ALL_Chlorophyll;

```

Figure 64. *All_Site_Params_For_Site_Pages* UNION query.

Since the query would need to run to generate each report page, the slowness of the query would have been a problem. Therefore, a table (*TBL_All_Data_Files_For_Site_Pages*) was created to contain the dataset generated by the UNION query. This table always exists in the database in an empty state. Each time site pages need to be generated, the *ReportIt* macro first runs a delete query (*All_Site_Params_CLEAR_TABLE*) to empty the table just in case it isn't empty (Figure 65).

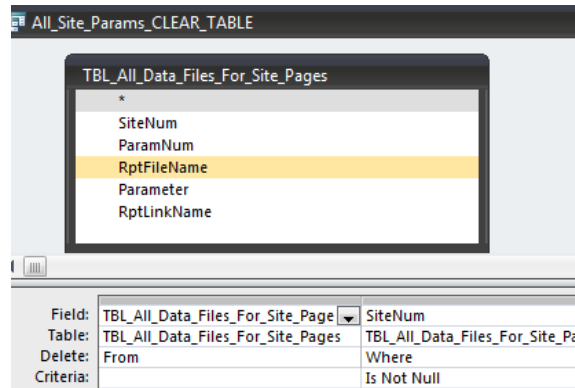


Figure 65. Query used to empty *TBL_All_Data_Files_For_Site_Pages* table.

So, to actually generate the sampling site reports, the first step for the user is to choose the “Sampling Site Pages” option from the *FRM_Report_Options* form (Figure 4). The *DoReport.ReportIt* macro (Figure 66) empties the *TBL_All_Data_Files_For_Site_Pages* table and then calls the VB function *Make_SitePages* (Figure 67) to generate the reports.

20=[Forms]![Frm_Report_Options]![Rpt_Opt]	MsgBox	Please be patient while the reports are generated..., No, Information,
...	Hourglass	Yes
...	OpenQuery	All_Site_Params_CLEAR_TABLE, Datasheet, Edit
...	RunCode	Make_SitePages("RPT_All_Site_Page_Rpts", "All_Sites_For_Site_Page_Rpts",[Forms]![Frm_Report_Site_Options]!
...	OpenQuery	All_Site_Params_CLEAR_TABLE, Datasheet, Edit
...	Close	Form, Frm_Report_Site_Options, Prompt
...	Hourglass	No
...	StopMacro	

Figure 66. Macro actions to produce data portal report pages.

```

'-----
' Generate Sampling Site pages.
'-----
Public Function Make_SitePages(ByVal RPT_TOP_Name As String, ByVal QRY_Name As String,
WhichSites As Integer, RptSite As Integer)

' Error handling
On Error GoTo Make_SitePages_Err

' Declare the database, recordset, and report filter string variables
Dim dbwpwa As DAO.Database
Dim rcdRecQuery As DAO.Recordset
Dim strFilter As String

' Set the current database and recordset variables for use in the do...until loop
Set dbwpwa = CurrentDb

' Make the mouse an hourglass to indicate a wait
DoCmd.Hourglass True

If WhichSites = 1 Then
    DoCmd.OpenQuery ("All_Site_Params_FILL_TABLE")
Else
    If WhichSites = 2 Then
        DoCmd.OpenQuery ("All_Site_Params_New_Year_FILL_TABLE")
    Else
        DoCmd.RunSQL "INSERT INTO TBL_All_Data_Files_For_Site_Pages ( SiteNum,
ParamNum, Parameter, RptLinkName )SELECT All_Site_Params_For_Site_Pages.SiteNum,

```

```

All_Site_Params_For_Site_Pages.ParamNum, All_Site_Params_For_Site_Pages.Parameter,
Make_LinkName([Abbreviation],[SiteNum],[SubFolder]) AS RptLinkName FROM
All_Site_Params_For_Site_Pages WHERE All_Site_Params_For_Site_Pages.SiteNum=" &
RptSite & ";"
    End If
End If

Set rcdRecQuery = dbwpwa.OpenRecordset(QRY_Name)

' Start looping through the QRY_Name file to generate a one page
' report for each Site. This loop will process every SiteNum and terminate
' when the end of the file (EOF) is reached.
Do Until rcdRecQuery.EOF
    ' Set the filter for the report to the current SiteNum and ParamNum
    strFilter = "[MstSiteNum]=" & rcdRecQuery![MstSiteNum]
    ' Open the report in print preview
    DoCmd.OpenReport RPT_TOP_Name, acViewPreview, "", strFilter, acHidden
    ' Output the file
    DoCmd.OutputTo acOutputReport, RPT_TOP_Name, acFormatPDF, "C:\Users\Elise
Torello\documents\WPWA\Field Data\Master Access Database\WPWASites\" &
rcdRecQuery![RptFileName], False, "", 0, acExportQualityPrint
    ' Close the report
    DoCmd.Close acReport, RPT_TOP_Name
    ' Move on to the next record in the query
    rcdRecQuery.MoveNext
    'and loop
Loop

' Turn off the hourglass
DoCmd.Hourglass False

Make_SitePages_Exit:
    Exit Function

Make_SitePages_Err:
    MsgBox Error$
    Resume Make_SitePages_Exit

End Function

```

Figure 67. *Make_SitePages* function to generate data portal pages.

The *Make_SitePages* function takes four arguments: *RPT_TOP_Name* (the name of the top-level sampling site report (*RPT_All_Site_Page_Rpts*)), *QRY_Name* (the name of the query listing all sampling sites (*All_Sites_For_Site_Page_Rpts*)), *WhichSites* (1=all sites, 2=sites sampled in the latest year, 3=one site only), and *RptSite* (the *SiteNum* to report if *WhichSites*=3). Depending on the value of *WhichSites*, one of three queries will be executed to fill the *TBL_All_Data_Files_For_Site_Pages* table (Figure 68). If *WhichSites*=1 (all sites), the *All_Site_Params_FILL_TABLE* query appends the entire contents of the *All_Site_Params_For_Site_Pages* UNION query into the *TBL_All_Data_Files_For_Site_Pages* table. If *WhichSites*=2 (only sites sampled in the latest year), the *All_Site_Params_New_Year_FILL_TABLE* query uses a link with *QRY_Sites_With_Latest_Year_of_Data* to append only the desired sampling sites from the *All_Site_Params_For_Site_Pages* UNION query into the *TBL_All_Data_Files_For_Site_Pages* table. Finally, if *WhichSites*=3 (one site only), a query embedded in the *Make_SitePages* function appends records from the one desired site from the UNION query into *TBL_All_Data_Files_For_Site_Pages* table.

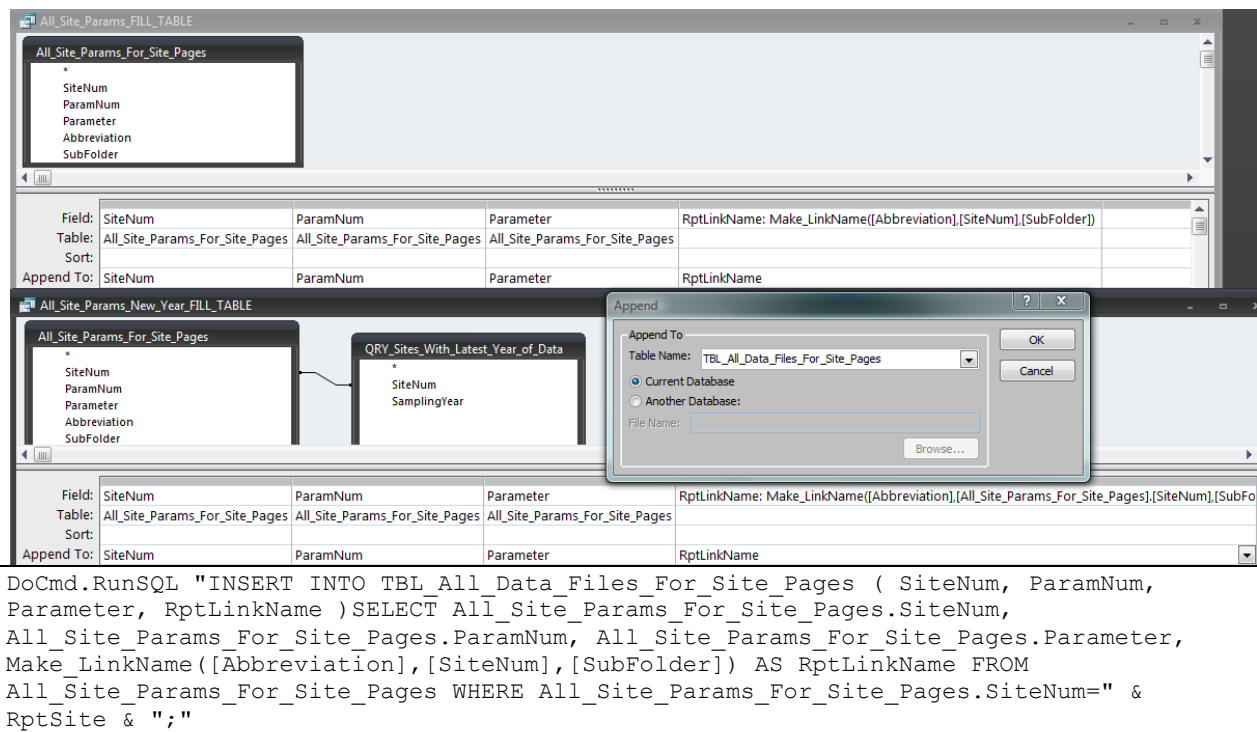


Figure 68. The three possible queries used to fill the *TBL_All_Data_Files_For_Site_Pages* table depending on the value of *WhichSites*.

As with the data reports, the user will see a series of dialog and information boxes requiring input whenever data records are deleted from or added to a table; again, click “OK” or “Yes” to continue with report generation.

In all three queries, a VB function named *Make_LinkName* (Figure 69) is used to create an active URL hyperlink for each data report page. Note the “#” before and after the hyperlink—this is necessary to automatically generate a functioning hyperlink in the PDF reports.

```

'-----
'This function creates the URL for the site or data page file. These are
'the active hyperlinks from Google Maps.
'-----
Public Function Make_LinkName(Abbrev As String, SiteNum As Integer, SubFolder As String)
    Make_LinkName = "#http://www.wpwa.org/waterdata/" & SubFolder & "/" & Abbrev & "_" & SiteNum & ".PDF#"
End Function

```

Figure 69. Visual Basic function for generating the URL name for each data report.

The table *TBL_All_Data_Files_For_Site_Pages* is used in the *All_Sites_Having_Data_Reports* query to create a list of all sites having data, with one record per site, for use in generating top-level site reports (Figure 70). Note the “Total” values for the query—the results are grouped by *SiteNum*, and only the first *ParamNum* for each *SiteNum* is kept in the resulting recordset.

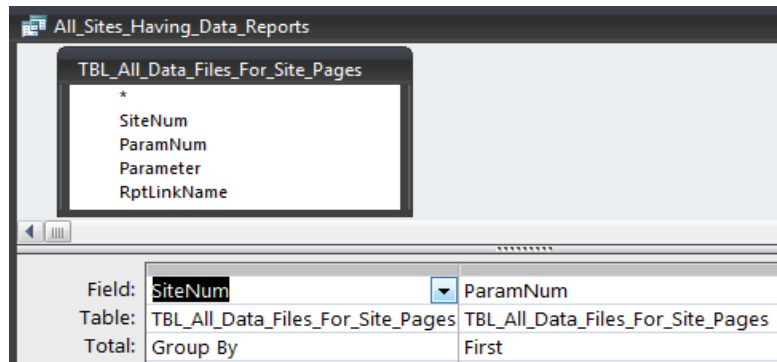


Figure 70. *All_Sites_Having_Data_Reports* query.

The *All_Sites_Having_Data_Reports* query is then used in another query, *All_Sites_For_Site_Page_Rpts*, to include relevant site information from the *MasterSiteList* and *Watershed_Watch_Sites* tables (Figure 71). This query is used by the VB *Make_SitePages* function to loop through all of the sites and generate a one-page report for each site. In addition, *SiteNum* is given the alias *MstSiteNum* to allow linking of master and child fields in the top level site report (*RPT_All_Site_Page_Rpts*, Figure 72).

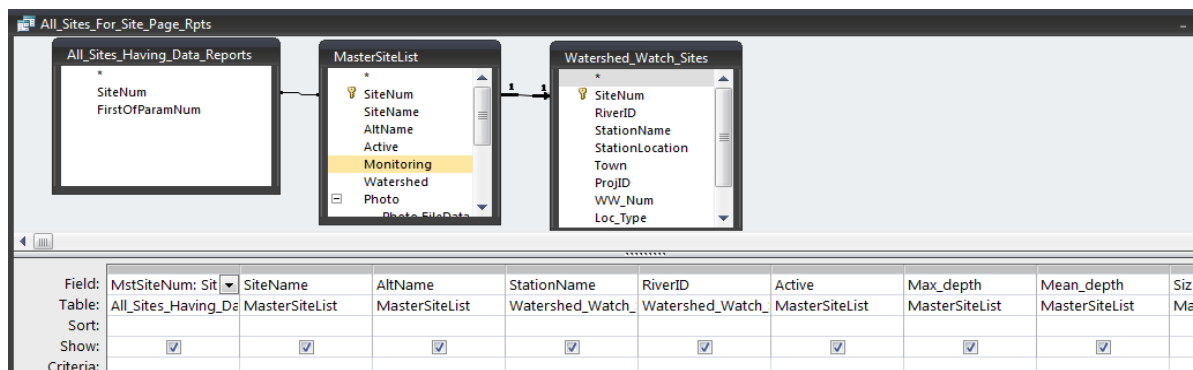


Figure 71. *All_Sites_For_Site_Page_Rpts* query.

RPT_All_Site_Page_Rpts

Page Header

SiteName: SiteNum: MstSiteNum

AltName:

MstSiteNum Header

Town: Town Maximum depth (m): Max_depth

URI WW StationName: StationName Mean depth (m): Mean_depth

URI WW RiverID: RiverID Size (acres): Size_acres

Active sampling site? ☒

Detail

SiteNum Header

For each parameter listed below, click on the file name to access the report.

Parameter	Report File Name (PDF Format)
Parameter	RptLinkName

Comments

Page Footer

Wood-Pawcatuck Watershed Association

University of Rhode Island

URI Watershed Watch COOPERATIVE EXTENSION #URI

Property Sheet

Selection type: Report

Report

Format Data Event Other All

Record Source All_Sites_For_Site_Page_Rpts

Caption RPT_All_Site_Page_Rpts

Pop Up No

Modal No

Display on SharePoint Follow Table Setting

Default View Report View

Allow Report View Yes

Allow Layout View Yes

Picture (none)

Picture Tiling No

Picture Alignment Center

Picture Type Embedded

Picture Size Mode Clip

Width 7.9583"

Auto Center No

Auto Resize Yes

Figure 72. Top-level sampling site report showing the report record source.

The *TBL_All_Data_Files_For_Site_Pages* table is also used to generate a subreport named *RPT_All_Site_Params_FileNames* for inclusion in the top-level site report. This subreport lists all of the hyperlinked data report pages for each site and was incorporated into the Detail section of the top-level report, with Link Master Fields set to *MstSiteNum* and Link Child Fields set to *SiteNum* just like in the data reports. An example of a sampling site data portal page is shown in Figure 73.

Yawgoo Pond

SiteNum: 2010

Town: Exeter, South Kingstown

Maximum depth (m): 10

URI WW StationName: WW68

Mean depth (m): 4

URI WW RiverID: RI0008039L-15

Size (acres): 136

Active sampling site? ☒

For each parameter listed below, click on the file name to access the PDF data report.

Parameter	Report File Name (PDF Format)
Alkalinity	http://www.wpwa.org/waterdata/Other/Alk_2010.PDF
Bacteria, E. coli	http://www.wpwa.org/waterdata/Bacteria/Ecoli_2010.PDF
Bacteria, Enterococci	http://www.wpwa.org/waterdata/Bacteria/Entero_2010.PDF
Bacteria, Fecal coliform	http://www.wpwa.org/waterdata/Bacteria/Fecal_2010.PDF
Calcium	http://www.wpwa.org/waterdata/Dropped/Ca_2010.PDF
Chloride	http://www.wpwa.org/waterdata/Other/Cl_2010.PDF
Chlorophyll	http://www.wpwa.org/waterdata/Chlorophyll/Chl_2010.PDF
Magnesium	http://www.wpwa.org/waterdata/Dropped/Mg_2010.PDF
Nitrogen, Ammonium	http://www.wpwa.org/waterdata/Nitrogen/NH4_2010.PDF
Nitrogen, Nitrate	http://www.wpwa.org/waterdata/Nitrogen/NO3_2010.PDF
Nitrogen, Total	http://www.wpwa.org/waterdata/Nitrogen/TN_2010.PDF
pH	http://www.wpwa.org/waterdata/Other/pH_2010.PDF
Phosphorus, Dissolved	http://www.wpwa.org/waterdata/Phosphorus/DP_2010.PDF
Phosphorus, Total	http://www.wpwa.org/waterdata/Phosphorus/TP_2010.PDF
Secchi	http://www.wpwa.org/waterdata/Secchi/Secchi_2010.PDF
Sodium	http://www.wpwa.org/waterdata/Dropped/Na_2010.PDF



Tuesday, July 06, 2010



Figure 73. Example of a sampling site report showing the hyperlinked data report files.

5. Parameter “More Info” pages

One-page reports were created containing metadata for each sampling parameter. These reports summarize the importance of each parameter, plus display other general and year-specific metadata.

The entire process for creating these reports is described in Appendix 2. To start, choose the "Parameter Metadata Pages" option from the *FRM_Report_Options* form (Figure 4). The *DoReport.ReportIt* macro calls the VB function *Make_ParamPages* to generate the reports (the macro action and VB function are very similar to those described previously).

A query named *Param_MetaDataForTopForm* (Figure 74) was used to create the top-level parameter report (*RPT_Param_More_Info_TOP*, Figure 75) using information from the *Param_Metadata* table. The *ParamNum* field was given the alias *MstParNum* to allow Master/Child field linking with the embedded subreport, *RPT_Param_Meta_Yearly* (note the Source Object for the subreport shown in the Property Sheet in Figure 75). *RPT_Param_Meta_Yearly* was created incorporating year-specific metadata from the table *Param_metadata_yearly*. An example of a parameter information report is shown in Figure 76; hyperlinks to these files are included on the data report pages.

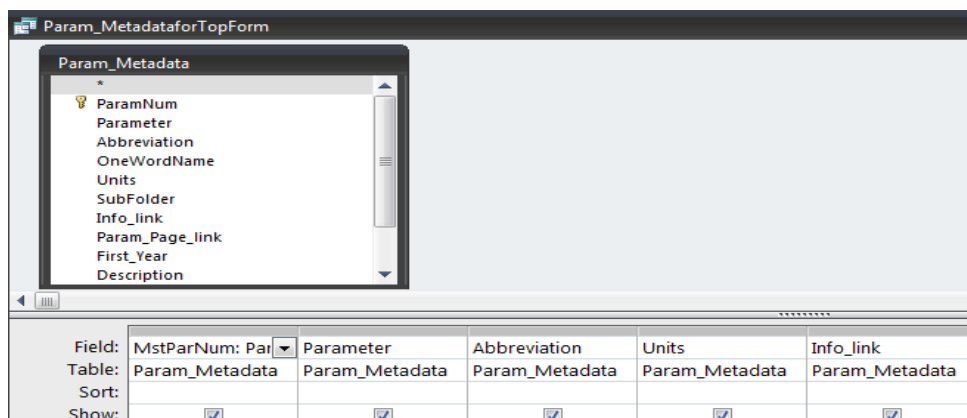


Figure 74. *Param_MetaDataForTopForm* query for creation of parameter metadata reports.

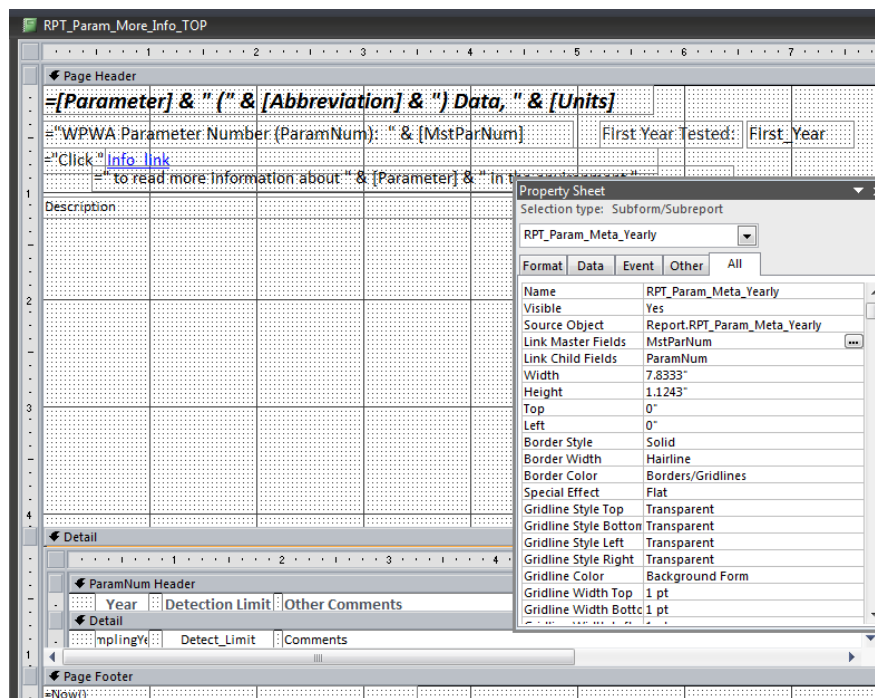


Figure 75. *RPT_Param_More_Info_TOP* report showing linking of Master/Child fields in the subreport.



Figure 76. Example of a parameter information file.

6. Creating a Google Map and Linking the Reports to the Google Maps Placemarks

The process of creating a Google Map, adding placemarks, and linking data reports to the placemarks is actually very simple and is described below.

6.1. Create a Google Account

Go to www.google.com, and on the right-hand side of the page click on “Create an account now”. Enter your email address and choose a password, then follow any other instructions that are on the screen. Once you accept their terms of service, your account will be created. Go ahead and sign in to your account.

6.2. Create a Map Using “My Maps”

Go to www.maps.google.com and click on “My Maps” on the left-hand side, right under where it says “Google maps” (Figure 77).

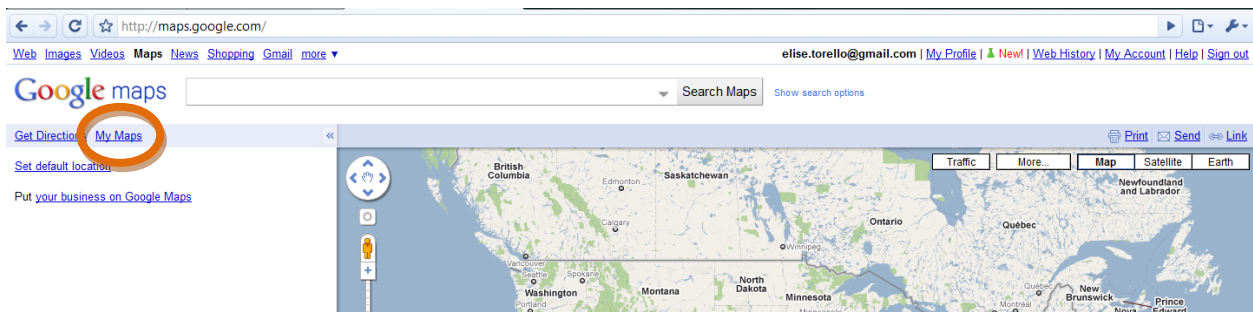


Figure 77. First step in creating a Google Map using My Maps.

Next, click “Create new map” on the left-hand side below “Google maps” and “Get Directions”—you will see the screen in Figure 78.

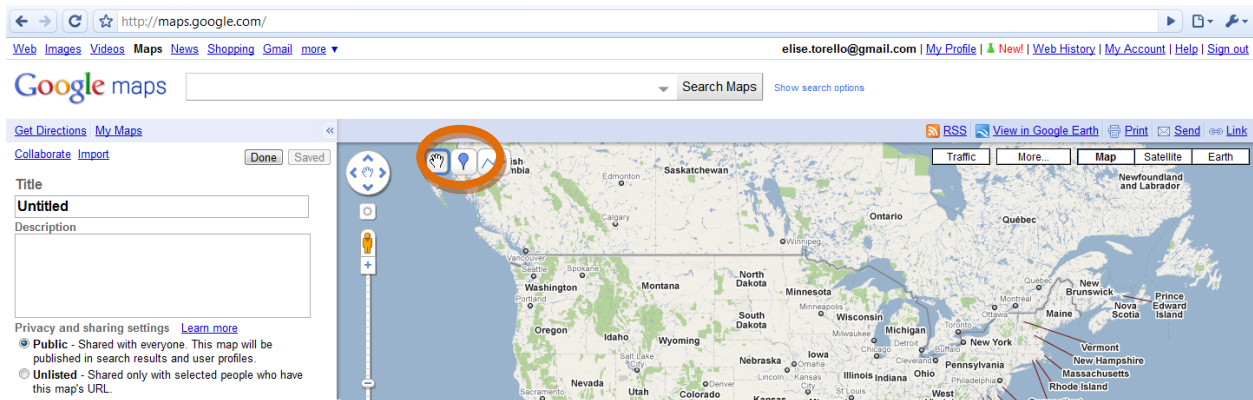


Figure 78. Creating a new Google Map.

Give your map a title and description, and decide if you want it to be public (shared with everyone—published in search results and user profiles) or unlisted (shared only with people who know the map’s URL). Next, zoom in to your region of interest using the slider on the left side of the map add some placemarks. This can be done in two ways—either by clicking on the upside-down teardrop in the map and then clicking on a location or by importing a KML (Keyhole Markup Language) file.

6.2.1. Point and Click Method

Click the upside-down teardrop in the upper left corner of the map (circled in orange in Figure 78), and then click on the location on the map where you want the placemark. Once the placemark is “dropped”, an information balloon will appear where information about the placemark can be entered (Figure 79). Give the placemark a title and enter any other desired information in the description. This can include hyperlinks, photos, videos, text, etc. The icon for the placemark can be changed to one of the styles provided by Google Maps (click on the default blue-teardrop icon to see the other choices available), or a custom placemark can be uploaded.

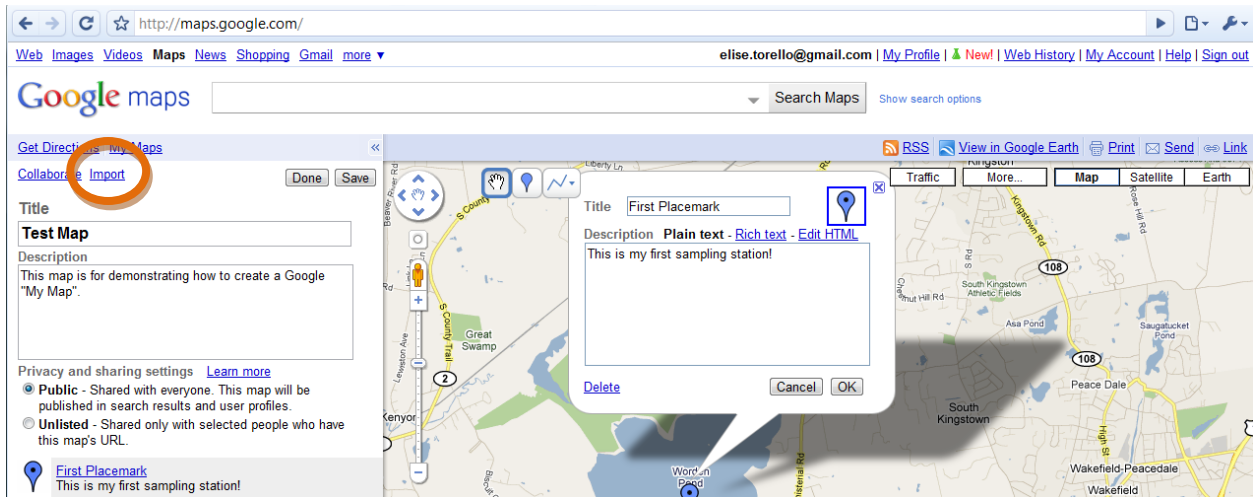


Figure 79. Google Maps placemark information balloon.

6.2.2. Uploading a KML File

KML files can be output by GIS (Geographic Information System) software like ESRI's ArcMap and then imported into Google Maps by clicking "Import" (circled in orange in Figure 79). A window will pop up to browse your computer for the KML file or enter a URL for one (Figure 80). There is an option to replace everything on the current map with the file—be careful not to check this box if you don't mean it! If a KML file created by ArcMap is uploaded, you may need to download the KML file (described below) and edit the style declaration near the top of the file in order to change the placemarks in the Google Map. Once the file is edited, it can be re-imported. When attempting to re-import a KML file to Google Maps, be sure that your file name has no spaces in it or the upload will not work.

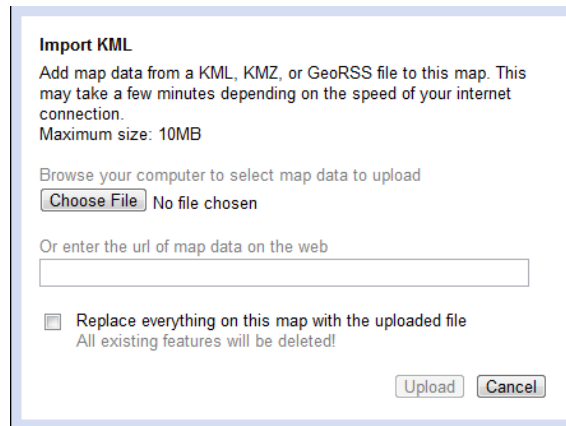


Figure 80. Screen used for importing a KML file into Google Maps.

After the placemarks are entered, save the map (by clicking "Save"—if this button is "grayed out" and says "Saved", Google Maps has already done an auto-save of your map) and click "Done". Your map is created!

Your map can be viewed in several different formats, including "Map", "Satellite", and "Terrain" (Figure 81). The "Map" and "Satellite" views each have their own button in the upper right corner of the map,

but “Terrain” (my personal favorite) is listed under the “More...” button (this option will not be available if you are zoomed in too far—zoom the map out to select this option).

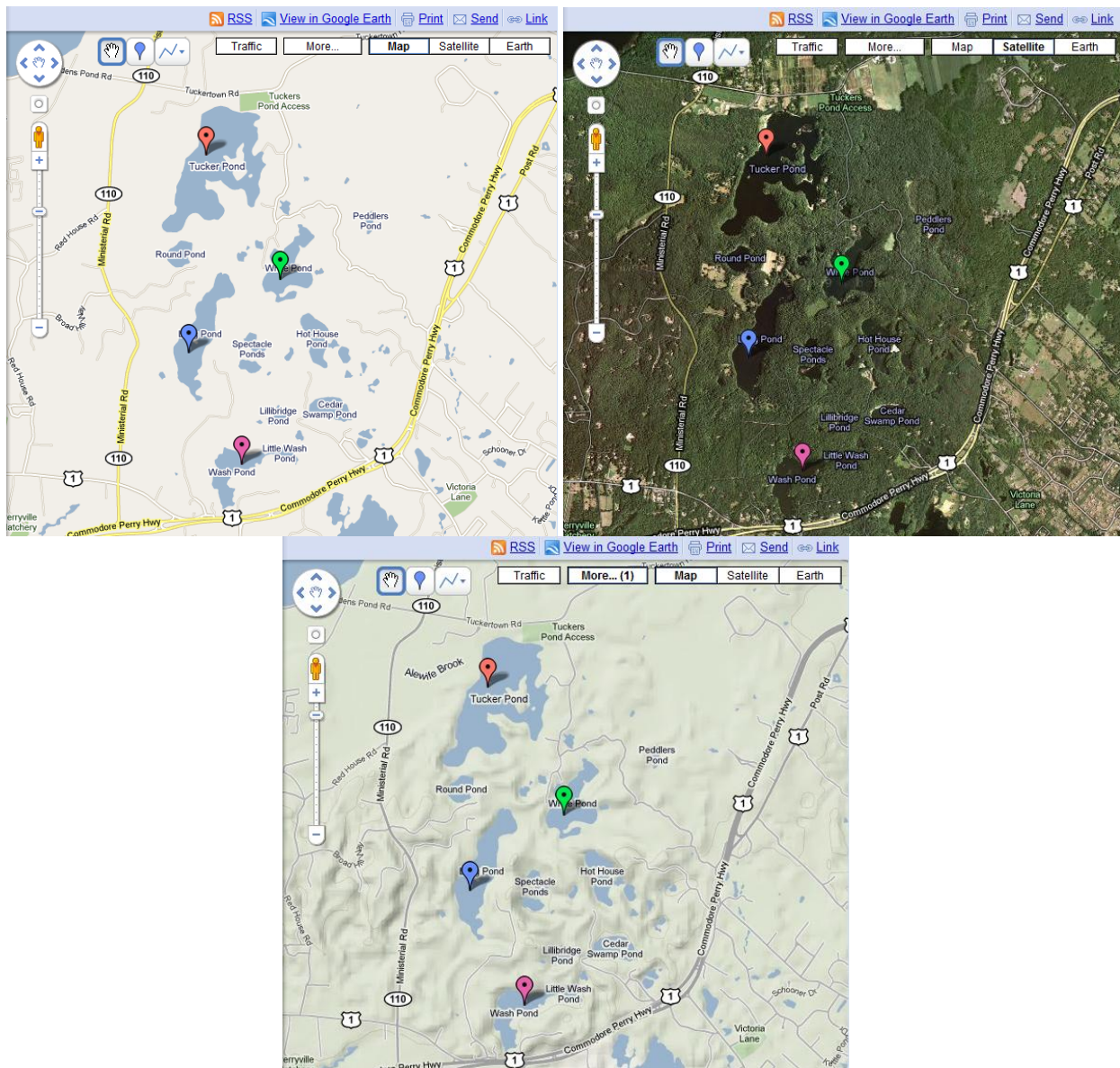


Figure 81. A map in "Map" (upper left), "Satellite" (upper right), and "Terrain" (bottom) views. Notice that in “Terrain” view the “More...” button has change to “More...(1)” since “Terrain” view is selected under that sub-menu.

6.3. Edit the Information in the Pop-up Balloons

Once your placemarks have been entered, you can edit the contents of the information balloons later. This can be done in two ways: within your map, or by exporting the map as a KML file, editing the file, and re-importing it to your map.

6.3.1. Making Changes Within the Map

Log in to your Google account, go to www.maps.google.com, and click on “My Maps”. You should see the map you created listed on the left—click on it to launch the map and then click the “Edit” button. Now you can grab placemarks and move them if needed, or change the content of the information balloons. Click on the placemark to open the balloon to change the contents of the balloon. Don’t forget to save your changes by clicking “Save”, and click “Done” when you are finished.

6.3.2. Exporting and then Re-importing the KML File

This method is useful if there are a lot of changes to the KML file that can be accomplished by cutting/pasting, searching/replacing, etc. From your map, click “Link” above and to the right of the map (circled in orange in Figure 82 below). Right-click over the highlighted text and select “Copy” (or use ctrl-C).

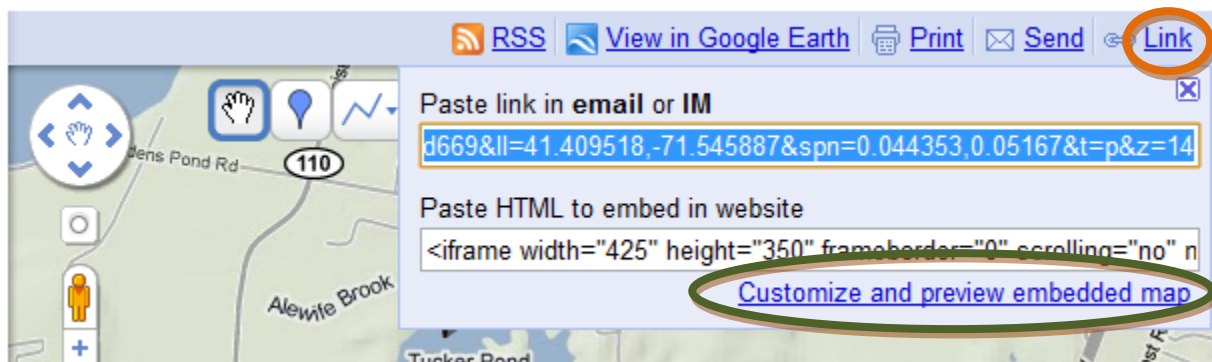



Figure 82. Copying the link address for the Google Map.

Open a new browser window or tab and right-click over the address bar—select “Paste” (or strike ctrl-V). Add `&output=kml` at the end of the URL and hit “enter”. Take care that there are no line breaks or spaces, as there should never be white space or a line break in a URL. This will download the KML file to your computer. The KML file can be edited with a text editor such as Notepad and imported back into Google Maps as described above. *Again, be sure that there are no spaces in the KML file name, or Google Maps will not import the file!* Editing a KML file is not horrendously complicated, but neither is it easy and isn’t for everyone. If you start out with a very simple KML file with just a few placemarks for practice, you will likely begin to make sense of the various tags, etc. in the file. You will see where the placemark names and descriptions live within the file. This is very much worth the effort if you ever anticipate having to make many changes to a map that could best be accomplished using search/replace.

6.4. Embedding a Small, Fully-functional Map in Your Web Page

A nice touch to your web site is to add a small but fully functional version of your Google Map embedded within your page. This is made very easy by Google Maps. As in the previous section, click “Link” above and to the right of the map. Click “Customize and preview embedded map” (circled in

green in Figure 82) to get your map the way you want it to look in your web page (Figure 83). Then, copy and paste the HTML into your web page where you want the map to appear.

 **Google Maps**


1. Customize

Map size

☒ Small
☐ Medium
☐ Large
☐ Custom

Width Height

2. Preview



Map data ©2010 Google - [Terms of Use](#) [Report a problem](#)

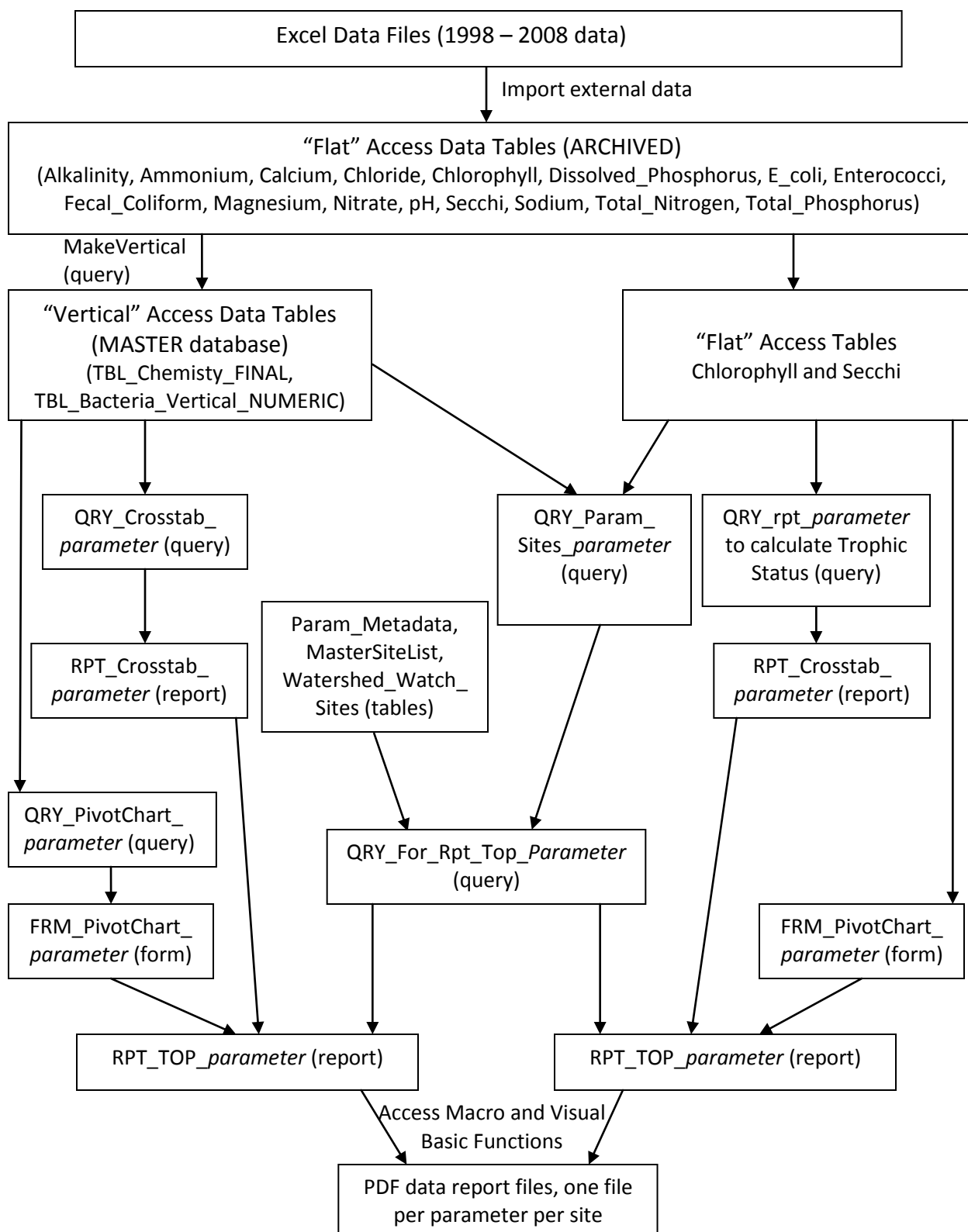
[View Larger Map](#)

3. Copy and paste this HTML to embed in your website

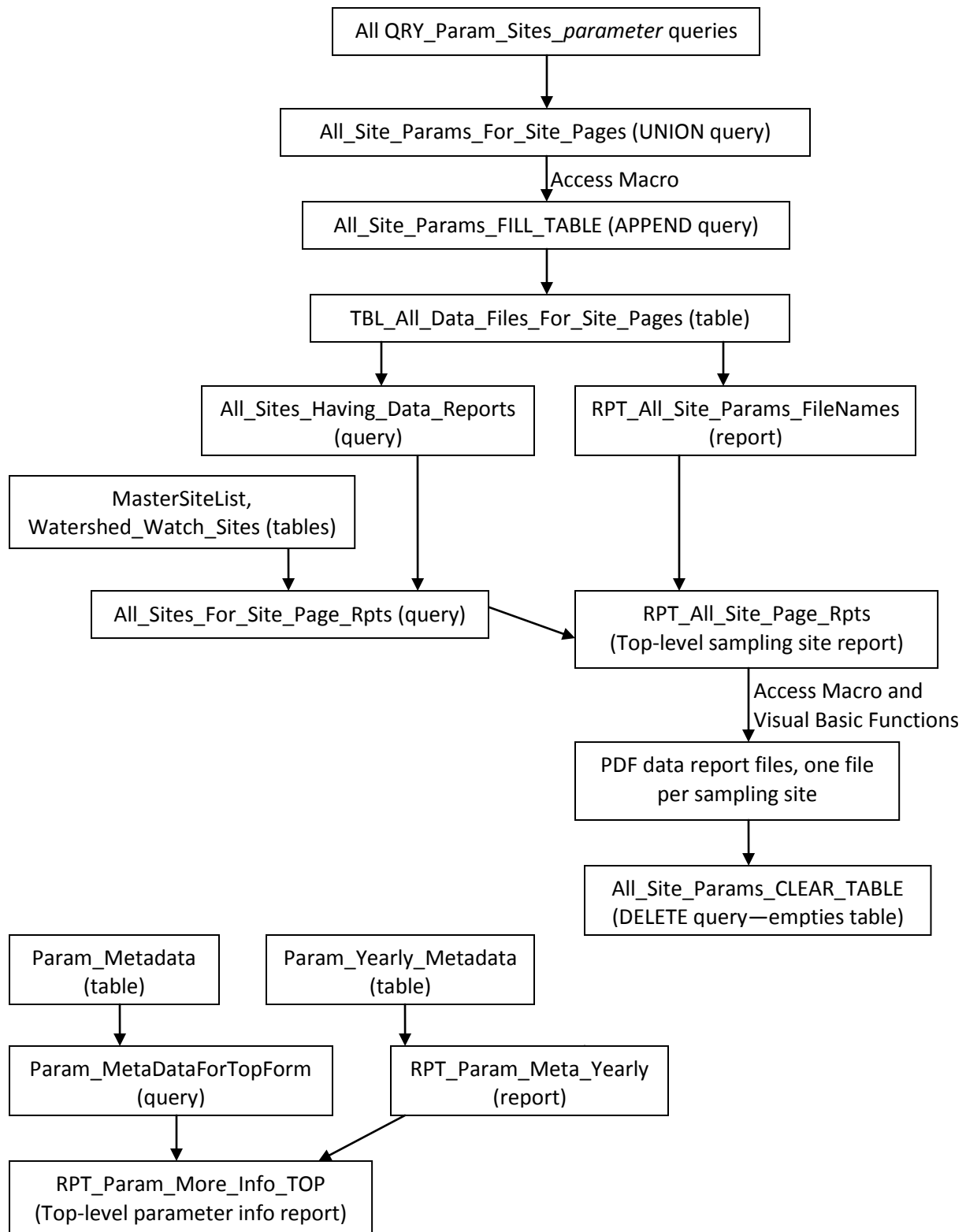
```
<iframe width="300" height="300" frameborder="0"
scrolling="no" marginheight="0" marginwidth="0"
src="http://maps.google.com/maps/ms?
ie=UTF8&hl=en&t=p&msa=0&
msid=117659233277043166885.00048acd78748dbb
7d669&ll=41.413896,-
71.55653&spn=0.038621,0.051498&z=13
&output=embed"></iframe><br /><small>View
```

Figure 83. Embedding a small map in your web page.

Appendix 1: Diagram showing general procedure for creating data report pages in MS Access.



Appendix 2: Diagram showing general procedure for creating sampling site and parameter information pages in MS Access.



Appendix 3: All MS Access Database Objects

Data Tables

MasterSiteList
Watershed_Watch_Sites
Site_LatLon

Param_Metadata
Param_Yearly_Metadata

Chlorophyll
Secchi
TBL_Bacteria_Vertical_NUMERIC
TBL_Chemistry_FINAL
TBL_Thresholds

Data Reports with Temporary Tables, Source Queries, and SubReports

RPT_TOP_Chlorophyll	QRY_For_Rpt_TOP_Chlorophyll
QRY_Param_Sites_Chlorophyll	QRY_Param_Sites_New_Year_Chlorophyll
QRY_rpt_Chlorophyll	RPT_Crosstab_Chlorophyll
FRM_PivotChart_Chlorophyll_Threshold	QRY_PivotChart_Chlorophyll_Threshold (Union)

RPT_TOP_Secchi	QRY_For_Rpt_TOP_Secchi
QRY_Param_Sites_Secchi	QRY_Param_Sites_New_Year_Secchi
QRY_rpt_Secchi	RPT_Crosstab_Secchi
FRM_PivotChart_Secchi_Threshold	QRY_PivotChart_Secchi_Threshold (Union)

RPT_TOP_Bacteria	QRY_For_Rpt_TOP_Bacteria
QRY_Param_Sites_Bacteria	QRY_Param_Sites_New_Year_Bacteria
QRY_Crosstab_Bacteria_Vertical (XTab)	RPT_Crosstab_Bacteria
FRM_PivotChart_Bacteria_Threshold	
TBL_One_Bacteria_For_Report	QRY_Fill_TBL_One_Bacteria_For_Report (Append)
TBL_One_Bacteria_For_Report_Thresholds	QRY_PivotChart_Bacteria (Append)
	QRY_PivotChart_Bacteria_Threshold (Append)
QRY_Clear_TBL_One_Bacteria_For_Report_Threshold (Delete)	

RPT_TOP_Chemistry	QRY_For_Rpt_TOP_Chemistry
QRY_Param_Sites_Chemistry	QRY_Param_Sites_New_Year_Chemistry
QRY_Crosstab_Chemistry (XTab)	RPT_Crosstab_Chemistry
QRY_PivotChart_Chemistry	FRM_PivotChart_Chemistry
TBL_One_Param_For_Report	QRY_Fill_TBL_One_Param_For_Report (Append)

RPT_TOP_Chemistry_Threshold (uses same chemistry crosstab qry/rpt)	FRM_PivotChart_Chemistry_Threshold QRY_PivotChart_Chemistry_Threshold (Union)

RPT_TOP_TP	QRY_For_Rpt_TOP_TP
QRY_Param_Sites_TP	QRY_Param_Sites_New_Year_TP
QRY_Crosstab_TP (XTab)	RPT_Crosstab_TP
QRY_PivotChart_TP	FRM_PivotChart_TP
<hr/>	
RPT_TOP_Phosphorus_Trophic	QRY_For_Rpt_TOP_Phos_Trophic
QRY_Param_Sites_Phosphorus_Trophic	QRY_Param_Sites_New_Year_Phosphorus_Trophic
QRY_Param_Sites_One_Site_Phos_Trophic	
QRY_Crosstab_Phosphorus_Trophic	RPT_Crosstab_Phos_Trophic
QRY_PivotChart_TP_Threshold (Union)	FRM_PivotChart_Phos_Trophic
<hr/>	
RPT_All_Site_Page_Rpts	All_Site_Params_For_Site_Pages (Union)
QRY_Param_Sites_ALL_Bacteria	
QRY_Param_Sites_ALL_Chemistry	
QRY_Param_Sites_ALL_Chlorophyll	
QRY_Param_Sites_ALL_Secchi	
TBL_All_Data_Files_For_Site_Pages	
All_Site_Params_FILL_TABLE (Append)	All_Site_Params_CLEAR_TABLE (Delete)
All_Site_Params_New_Year_FILL_TABLE (Append)	
All_Sites_For_Site_Page_Rpts	RPT_All_Site_Params_FileNames
All_Sites_Having_Data_Reports	
<hr/>	
RPT_Param_More_Info_TOP	Param_MetadataforTopForm
RPT_Param_Meta_Yearly	
<hr/>	
makevertical (Append)	
<hr/>	
QRY_Latest_Year_in_DB	QRY_Sites_With_Latest_Year_of_Data
QRY_Sites_Sampled_Last_Two_Yrs_1	QRY_Sites_Sampled_Last_Two_Yrs_2
QRY_Sites_No_Longer_Active	

Forms for Report Generation

Frm_Report_Options	
Frm_Report_Site_Options	QRY_One_Site_Form

MS Access Macro

DoReport

Visual Basic Module

WPWA Functions