
THE WATER RESOURCE INFORMATION SYSTEM

Database Query Tool Primer

Robert Clayton, Environmental Scientist Associate
Environmental Resource Management Division
Watershed Protection and Development Review Department

ABSTRACT

The report provides a technical overview of the development and implementation of the Water Resource Information System (WRIS) Database Query Tool which facilitates citizen-access to the Field Sampling Database through the Austin City Connection Web Site. The reasons why the tool was developed, what technology was chosen for the project, and some idea of the components that are essential to keeping it operational are included.

INTRODUCTION

Any science may be likened to a river. It has its obscure and unpretentious beginning; its quiet stretches as well as its rapids; its periods of drought as well as of fullness. It gathers momentum with the work of many investigators and as it is fed by other streams of thought; it is deepened and broadened by the concepts and generalizations that are gradually evolved.

Carl P. Swanson, *Cytology and Cytogenetics*, Prentice-Hall, 1957

The Database Query Tool is intended to provide easy access to as many scientists and would-be scientists and citizens at large as possible in an efficient manner. Environmental quality data is often labeled *inaccessible* and is criticized for not providing quick and easy answers. This is because it constitutes the *obscure and unpretentious beginning* from which understanding will flow. By giving everyone the opportunity to “browse” this data, it is possible that those who delve freely into this tool will emerge with a better understanding of what water quality monitoring is about. For those who already have an advanced understanding on the topic, they will be provided with rapid access to the wealth of data that we collect on behalf of the citizens of Austin.

As the river of knowledge is deepened and broadened, so too can it become clogged with suspended silt and debris; and as a result the entire nature of its course altered. Consider this document an “owner’s guide” to the Database Query Tool so that all future administrators might keep it functioning properly and prevent it from being consigned too soon to a technological backwater.

DATABASE QUERY TOOL: PREHISTORY

Previous to the advent of this [WRIS: Database Query Tool](#), the [WRIS: Data Request Form](#) was made available through the Austin City Connection web site. This simple web-form enabled

citizens to request information about specific parameters, in specific watersheds around the Austin area; providing them with pull-down lists that enhanced their understanding of the types of data we collect.

When the Data Request Form was initially employed there was a slight surge of interest in our data, concomitant with an article on Polycyclic Aromatic Hydrocarbons in the local newspaper. Since that surge in curiosity, the number of requests for data through the data request form has slackened. [figure 1.1]

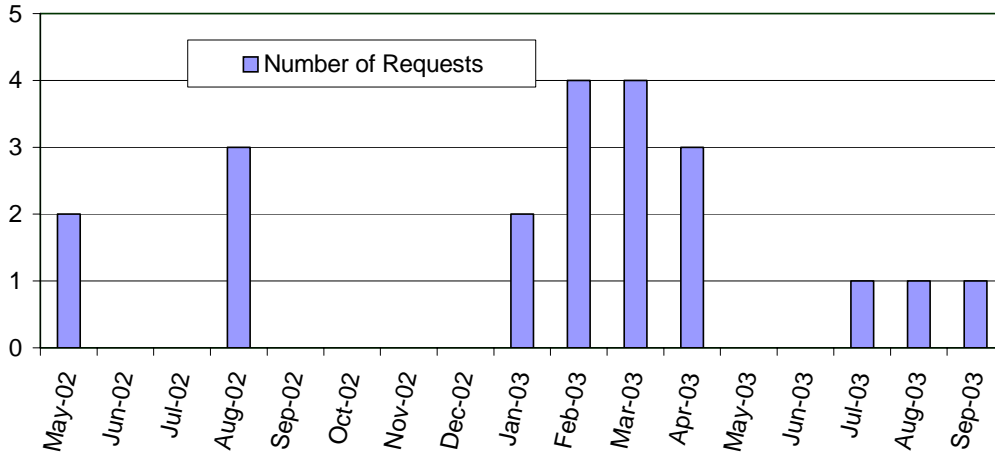


Figure 1.1

The data request form is non-programmatic and simply passes the request to a WRE staff member via email. The staff person then executes the data pull manually and the results are emailed to the requestor within three days. While this is a perfectly useful tool to facilitate the distribution of our publicly-accessible data, we hope that the Database Query Tool will be viewed as a more efficient delivery method, running 24 hours a day, 7 days a week and 365 days a year within constraints of server uptime as determined by the Information Systems Department (ISD). The DQT allows nearly instantaneous access to all Water Resource Evaluation data in the Field Sample Database.

DATABASE QUERY TOOL: INNOVATION THROUGH EXIGENCY ...

A real-time viewer for the data we collect ensures that citizens are equipped to keep themselves informed about the water quality of the creeks and rivers we monitor. With the Data Request Form, there is consideration and interpretation applied to each request; which puts a filter between the data and the citizen. The Database Query Tool was conceived to remove this filter and as a beneficial side effect, improve turnaround time.

The Database Query Tool provides no analysis. The finite purpose of this tool is to generate a comma-delimited file of “raw” environmental quality data to allow end users to do their own analysis. This automates the process of information transparency that is promised to citizens through the Texas Open Records Act.

Other governmental Environmental Quality entities (USGS,LCRA,TCEQ) have similar data distribution web-sites and each have their strengths. It was our goal to meet the minimum

standards established by these existing interfaces. In at least one way we have improved the clarity of the data; by developing an output format that is simple and references units and methods using common, plain-English terminology rather than the more concise, but complicated STORET-codes.

DATABASE QUERY TOOL: TECHNOLOGY OVERVIEW

The challenge in providing database access on the web is to allow anonymous users access to the database without allowing them to damage it, including preventing the application from tying up the database in such a way as to drastically reduce its utility for internal users. In the case of the WRIS database, it is hosted on the same machine as the PIER database (Permitting, Inspection and Environmental Review), which is an application used by several departments. Impact on PIER, and the Field Sample Database (WRIS) itself must be kept to a minimum, and it is also crucial that the Database Query Tool not require much additional staff time for maintenance, since it is designed as a labor saving device.

Query Limits:

In order to minimize disruption to internal users of the database, an arbitrary limit of 15,000 records was established after consultation with ISD. A “statistics” table that is generated by an automated batch job allows us to perform an initial “check-query” that returns a count of records for the criteria specified in the form. If the number exceeds 15,000 then the user is directed to a page that tells them to try again, limiting their criteria. If the count comes back with a value of zero, the user is directed to a page that tells them to try again, broadening their criteria. If the count of records comes back at less than the 15,000-record threshold then the main data query is executed and the output file generated.

There is no limit to the number of queries an individual may run in series.

To see the SQL for this “check-query” see Appendix A.1.

Development Languages:

The Database Access Committee followed an ISD recommendation to develop the application using Cold Fusion, which is an easy-to-develop, database-friendly software package. The guts of the application, the submittal form and the output, are done with Cold Fusion. At the time of this writing, the Cold Fusion product line had transferred from its creator, Allaire Corporation, to Macromedia, the makers of Shockwave™ and Flash™. While Cold Fusion MX is the most current version of Cold Fusion, this application was written to be compliant with Cold Fusion 5.0 tags.

The initial page that contains the HTML form (default.cfm) makes very little use of Cold Fusion tags but primarily relies on HTML and JavaScript. It does, however, use several CF tags. The first is to create a reference to the last 6 digits of the user’s ip address and the second uses an application-level variable that defines the path to the output file. The header and footer includes and the select list includes are also programmed. The action attribute of the form references output.cfm, which has the queries embedded in it as well as the output file creation subroutine and the ftp (file transfer protocol) output functionality and all the error handling. To view the query that is run to pull the output data from the database, see Appendix A.2. To view the Cold Fusion source files with full commenting, please see Appendix B.

JavaScript is used to respond to user-events, validate the form input and to do some pre-processing of hidden form elements to speed up the database access. JavaScript is an event-driven language, whereas Cold-Fusion can only execute code in response to one “event” -- the rendering of the page. JavaScript, on the other hand, cannot access the database. As such, they are a powerful combination. To view the JavaScript source file, please see Appendix C.

For a review of the various JavaScript and Cold Fusion event triggers germane to this application, please see Appendix D.

The appropriate flow of the application is maintained using popup windows for the help and user feedback during the data query and the web pages use cascading style sheets to make the output neat, readable and standardized. To see the stylesheet reference, see Appendix E.

The Database Query tool requires space on three separate servers to function properly, a web server to host the various html (.htm) files and Cold-Fusion (.cfm) templates, a Cold Fusion server to interpret the Cold Fusion templates and perform the database queries for the web server and an FTP server from which web users may download the output file. The Cold Fusion server must have a functioning connection to the Oracle database.

For a complete list of the templates, pages, directory structures and accessory documents required to keep this application functioning, please see Appendix F.

DATABASE QUERY TOOL: BACKGROUND PROCESSES

Since the Database Query Tool accesses the same database that our ERM staff use, relatively few steps are required to ensure that the data citizens can access is the most current. However, due to the construction of the program, a few components are still present that need to be periodically updated.

Select Lists

The four primary Criteria Selection Controls on the default.cfm page are populated using an “Cold Fusion Include” ... which simply inserts the results of one template into the output of another template. Quite often such controls would be populated by a Cold-Fusion template that contains a list generated by a select from a table on the related database. For instance, the *Watershed* Select list might be generated by this query of the watershed table in the WRIS database.

Template: Watershed_Control.cfm

```
<!---- holds the list of values that can be in a select box -- for instance, watershed, site_name ---->

<cfquery datasource="FSDB" name="getWatershedNumber"
cachedwithin="#createTimeSpan(1,0,0,0)#">
  select watershed_no, name
  from watershed
  where watershed_no in (select unique watershed_no
                        from sample_site
                        where sample_site_no in
                        (select unique sample_site_no
                        from field_visit
                        where collecting_entity_no in (1,12)))

  order by name
</cfquery>

<OPTION selected value="0">Choose a Watershed</OPTION>
<cfoutput query="getWatershedNumber">
<OPTION value="#getWatershedNumber.watershed_no#">#getWatershedNumber.name#</OPTION>
</cfoutput>

<!---- generated dynamically -- whoo hoo! ---->
```

Since having these four templates generate the *multiple select* lists every time default.cfm was loaded became rather resource-intensive, it seemed most logical to run these query-templates as stored automated tasks on the Cold Fusion server, wherein the output for each would be dumped to the static templates that can then be used to populate the form control *multiple select* boxes. These tasks should be run near the beginning of a month (so that the sampling year *select box* is updated appropriately within fifteen days of the change of the calendar year).

Statistics Table

Another component that needs to be refreshed is the statistics table that has the row counts for each combination of possible values from the form controls. This table is generated with the following code, a stored procedure on the database server, which is set to run as a batch job once a month:

```
PROCEDURE UPDATE_FSDB_WEIGHTING
IS
BEGIN
DELETE FROM FSDB_WEIGHTING;

INSERT INTO FSDB_WEIGHTING
(
select watershed_no, primary_param_no, medium_no, year, count(row_id) row_ct
from

(select ss.watershed_no, pt.primary_param_no, fs.medium_no, to_char(fv.visit_date,'YYYY') year,
sr.rowid row_id
from sample_site ss, parameter pa, parameter_type pt, field_sample fs, field_visit fv,
sample_result sr
where fs.field_sample_no = sr.field_sample_no and
fv.field_visit_no = fs.field_visit_no and
ss.sample_site_no = fv.sample_site_no and
pa.parameter_no = sr.parameter_no and
pa.param_type_no = pt.param_type_no and fv.collecting_entity_no in (1,12))

group by watershed_no, primary_param_no, medium_no, year
);

COMMIT;

END;
_*****
```

Spooling/Output file Cleanup

The final background component to be managed offline would be the two file areas (the spooling directory on the CF server and the Output directory on the FTP server) that need to have old files deleted from them. Currently staff plans to run scripts on the servers that remove files older than 30 days.

DATABASE QUERY TOOL: HELP SYSTEM

The Help system is fairly simple. The creation of the popup window that houses the help text is run by a JavaScript function called `spawnHelp()` ... which passes the appropriate parameters to `spawnPOP()`, the general window popup function. `SpawnPOP()` requires three variables passed. One is the name of the file that should be displayed in the popup window, second is a name for the window and the third is a string variable that contains the parameters that can be passed to the built-in `.open()` method of the window object. There are two types of values passed. First are the arguments that accept numeric values in Pixels: [Height, Width, Left (location from left screen margin), Top (location from top screen margin)]. The second set of arguments should be assigned a value of "Yes" or "No": [resizable, scrollbars, location, status, toolbar and menubar].

The only other tricky element to the help system is the `parameter_type` and `sampling_media` controls in which the pull-down list changes the contents of the text-area below it. No system has been developed to automate the creation of these pull-down list values; nor the arrays that contain the elements that are fed to the companion text-area controls.

Note that if spaces are present in the control item for the pull-down list, the array must not have spaces in its identifier, but underscores. There is a simple replace function that will convert the spaces to the underscores. IF the name of the array does not equal the option name with underscores instead of spaces the query will fail.

Here's a snippet of the html from the page [help.cfm?help=med_par](#)

```
<select name="m_type"
onChange="swapOptions(window.document.medium_v_p_type.m_type.opti
ons[selectedIndex].text,window.document.medium_v_p_type.m_type,wi
ndow.document.medium_v_p_type.valid_pt);">
<!-- some missing options for clarity --->
<option>Channel Assessment
```

The function that is called on change of this pull-down list will be passed that selected option name ... and replace the space with an underscore. The variable that contains the valid parameter_types for the Channel Assessment method is an array named `channel_assessment`.

When modifying these controls to reflect changes in the Parameter Type and Primary Parameter type lists, please note that the convention must be adhered to for proper operation. Apologies for the inflexibility, however, "we have a system, and it works." (*The Day the Earth Stood Still*, FILM. 1951).

DATABASE QUERY TOOL: FUTURE DEVELOPMENT

In the final stages of testing prior to deployment, the Database Query Tool has reached what we have determined to be the apex of its development cycle. Any future enhancements of this technology would most certainly be developed in a more graphical manner, perhaps through an Arc GIS tool that allowed a user to click, zoom and pan on a map and then trigger a page that gives them context-sensitive options, summary statistics and the ability to pick and choose what data they'd like to view for that site. We can see such a tool being developed as a compliment to ... but not as a replacement for the simple, robust tool that is the Database Query Tool.

Some small avenues for improvement do exist, however. Should XML ever become universally adopted as a standard for data transfer then we could use the native Cold Fusion WDDX extensions to give the user an alternative to the comma-separated output.

The rest of this document is dedicated to the code of the application itself.

Appendix A: Structured Query Language (sql) Queries

Section 1:

dbaLimitChk

```
select sum(row_ct) full_row_ct
from wre_admin.fsdb_weighting fs
where watershed_no #FORM.shed_no# AND
      primary_param_no #FORM.param_no# AND
      #strMediaList#
      year #FORM.date_no#
```

NOTE: the items surrounded by pound signs (#) are either a variable that has been assigned a value with Cold Fusion's <CFSET> tag OR it refers to the value of a FORM text item ... the pound signs give COLD FUSION the understanding that it needs to INTERPRET that signifier and replace it with a VALUE.

Section 2:

getUserDataRequest

```
SELECT
  W.NAME WATERSHED, FS.FIELD_SAMPLE_NO REF_NO,
  To_Char(FV.VISIT_DATE,'MM/DD/YYYY') SAMPLE_DATE,
  DECODE(FS.SAMPLE_TIME,9999,'&nbsp;','TO_CHAR(FS.SAMPLE_TIME,'0999')) SAMPLE_TIME, SS.NAME
  SITE_NAME, ST.DESCRPTION SITE_TYPE,
  SM.DESCRPTION MEDIUM, InitCap(PA.DESCRPTION) PARAMETER,
  nvl(LTRIM(RTRIM(SR.QUALIFIER_CODE)),'&nbsp;:') QUALIFIER,
  SR.PARAM_VALUE RESULT, UM.DESCRPTION UNIT,
  FF.DESCRPTION FILTER, AM.DESCRPTION METHOD

FROM
  SAMPLE_RESULT SR, FIELD_SAMPLE FS,
  FIELD_VISIT FV,          SAMPLE_SITE SS,
  PARAMETER PA,           FILTER_FRACTION FF,
  UNIT_OF_MEASURE UM,     ANALYSIS_METHOD AM,
  SAMPLE_MEDIUM SM,       SITE_TYPE ST,
  WATERSHED W,            PARAMETER_TYPE PT

WHERE
  W.WATERSHED_NO = SS.WATERSHED_NO AND SS.SITE_TYPE_NO = ST.SITE_TYPE_NO AND
  SR.FIELD_SAMPLE_NO = FS.FIELD_SAMPLE_NO AND FS.FIELD_VISIT_NO = FV.FIELD_VISIT_NO AND
  SS.SAMPLE_SITE_NO = FV.SAMPLE_SITE_NO AND SR.PARAMETER_NO = PA.PARAMETER_NO AND
  PA.PARAM_TYPE_NO = PT.PARAM_TYPE_NO AND SR.FILTER_FRACTION_NO = FF.FILTER_FRACTION_NO AND
  SR.U_OF_M_NO = UM.U_OF_M_NO AND SR.METHOD_NO = AM.METHOD_NO AND
  FS.MEDIUM_NO = SM.MEDIUM_NO AND FV.COLLECTING_ENTITY_NO in (1,12) AND
  #strMediaList#
  PT.PRIMARY_PARAM_NO #FORM.param_no# and
  ss.watershed_no #FORM.shed_no# AND
  to_char(fv.visit_date,'YYYY') #FORM.date_no#
```

Appendix B: Cold Fusion Source Code (1 of 8 pages)

application.cfm (pp. 11-12)

db_query_form.cfm (pp. 13-14)

db_query_results.cfm (pp. 14-19)

header.cfm (pp. 19-20)

footer.cfm (pp. 20-21)

help_header.cfm

help_footer.cfm

help.cfm

js_error.cfm

date_static.cfm

media_static.cfm

param_type_static.cfm

watershed_number_static.cfm

application.cfm

```
<!-- application wide font definition -->
<cfset defaultFace = "Verdana, Arial, Helvetica">
<cfset defaultSize = "3">

<cfapplication name= "coaerm_cf_db" sessionmanagement="YES" sessiontimeout="#CreateTimeSpan(0,0,30,0)#">
<cflock timeout="30" THROWONTIMEOUT="Yes" TYPE="Exclusive" SCOPE="Session">
<cfset Session.filecomplete = "false">
</cflock>

<cflock timeout="60" THROWONTIMEOUT="No" TYPE="Exclusive" Scope="Application">
<cfset Application.dbConnector = "orcl_wre">
<cfset Application.ftp_uname = "ftp_wre">
<cfset Application.pass = "Wap1pF33">
<cfset Application.tempdirectory = "E:\INETPUB\WWWROOT\WATERSHED\temp">
<cfset Application.tempDirCleanupIntervalDays = 30>
<cfset Application.outsidedirectory = "ftp://ftp.ci.austin.tx.us/pub/vendors/wre/">
<cfset Application.thisdirectory = "/watershed/">
<!-- thisdirectory is the directory path down from root ... -->
<cfset Application.recordByteWeight = 7.309163>
<!-- this is the approximate number of bytes per record for estimating the file download size
on the db_query_results.cfm page --->
<cfset Application.dbareclimit = 15000>
```

```

<cfset Application.ftpserver = "ftp.ci.austin.tx.us">
<cfset Application.baseHref = "http://isddev/watershed/">
<cfset Application.FormPage = "db_query_form.cfm">
  <!-- note ... when changing value of FormPage ... update the value of var strCfmForm in application.js
  --->
<cfset Application.OutputPage = "db_query_results.cfm">
  <!-- note ... when changing value of OutputPage ... update the value of var strCfmResult in
  application.js --->
<cfset Application.EmailLink = "http://www.ci.austin.tx.us/connect/email_robertclayton.htm">

</cflock>

```

db_query_form.cfm

```

<!-------
Template: db_query_form.cfm (version 6)
Purpose: final.
Use: form page that collects the data from the user and calls db_query_results.cfm to display results.
Author: rob clayton
Date: 9/9/2003

----->

<cfinclude template="header.cfm">
<!---Create the User Form in a table --->
<Table Border="0" width="620">
<TR><TD align=left>
<table border="0"><tr><TD bgcolor="#ADA5EF"></td><TD class="none">Watershed Protection and Development Review<br />Water Resource
Information System<br />Database Query Tool<font size="-2"><a href="default.htm">overview of
this tool</a></font></TD></TR>
<TR><TD colspan="2"></TD></TR>
<tr>
<TD class="inputHead" colspan="2" align="center">Select one or more of each of the Following
Criteria:<br><font size="-2">[hold down the <a href="javascript:shiftHelp();">shift</a> or <a
href="javascript:ctrlHelp();">control [ctrl]</a> key while clicking to select multiple
items]</font></td></tr>

</table>
<table border="1" cellpadding="6">
<tr><td><table align="left" width="100%" cellpadding="1">
<tr>
<TD class="inputHead">Watersheds:<br><font size="-2">required</font></td><td align="right"><a
href="javascript:spawnHelp('help.cfm?help=wsd');"></a></td>
</tr></table></td>
<td><table align="left"><table align="left" width="100%" cellpadding="1"><tr>
<td class="inputHead">Parameter Types:<br><font size="-2">required</font></td><td align="right"><a
href="javascript:spawnHelp('help.cfm?help=par');"></a></td></tr>
</table></td></tr>

<cfoutput>
<form name="DR" action="#Application.thisdirectory##Application.OutputPage#" method="post"
onSubmit="makeSessionID('#RIGHT(CGI.REMOTE_ADDR,6)');return form_check();">

<!-- in this hidden field, store what we're referring to as the "session ID" which is determined by the
date and the last two quartets of the user's ip address --->
<input type="hidden" name="MySessionID" value="null">

<tr><td class="watershed">
<!-- each multiple select control on this page has a javascript associated with the onChange event
called processLists() that is in the application.js file. The values passed to this function are
[Full control Name] (self explanatory? window.document.DR.Watershed)
[Full Name of Text Box for Count Update] (useful as supplemental info?)
[Full Name of Text Box for Display Text Update] (this is a reference for the box where we store the
"Bear Creek, Bee Creek and Shoal Creek" text--with
all it's grammatically correct formatting)
[Full Name of Text Box for Query Language Update] (this a reference for the box where we store the
"watershed no in (1,2,3,4)" SQL terms)

```

```

    [Text for "class" tag in output] (this is the language for the class tag for the td that houses it.
in
the
future revisions you could drop this and use the control name as
same for the class tag. )
--->

<SELECT multiple name="Watershed" size="5"
onChange="processLists(window.document.DR.Watershed,window.document.DR.shed_ct,window.document.DR.shed_nm,
window.document.DR.shed_no, 'watershed');">

<!-- this cfinclude allows for the .cfm file to be generated by a "select name,watershed_No from
watershed" query.
we used static forms to avoid the slowness of the database query for each form. Since the
information is relatively
static ... I suggest that we automate the creation of the watershed_number_static.cfm file as a
scheduled
task within the CF structure. --->

<cfinclude template="select/watershed_number_static.cfm">
</select></td>

    <td class="parameter">

<SELECT multiple name="Parameter_Type" size="5"
onChange="processLists(window.document.DR.Parameter_Type,window.document.DR.param_ct,window.document.DR.pa
ram_nm,window.document.DR.param_no, 'parameter');">

<cfinclude template="select/param_type_static.cfm">
</select></td>
</tr>
<tr><td class="inputHeadOptional"><table align="left" width="100%" cellpadding="1">
<tr>
<td class="inputHeadOptional">Sample Media:<br><font size="-2">note: if unsure, leave as is</font></td><td
class="inputHeadOptional" align="right"><a href="javascript:spawnHelp('help.cfm?help=med');"></a></td>
</tr></table>
</td>
<td><table align="left" width="100%" cellpadding="1"><tr>
<td class="inputHead">Sample Years:<br><font size="-2">required</font></td><td align="right"><a
href="javascript:spawnHelp('help.cfm?help=yea');"></a></td>
</tr></table></td></tr>
<tr>

    <td class="medium">

<SELECT MULTIPLE NAME="Sampling_Media" size="5" onChange="return
processLists(window.document.DR.Sampling_Media,window.document.DR.media_ct,window.document.DR.media_nm,win
dow.document.DR.media_no, 'media');">

    <cfinclude template="select/media_static.cfm">
</select></td>

    <td class="date">

<SELECT MULTIPLE NAME="Sampling_Date" size="5"
onChange="processLists(window.document.DR.Sampling_Date,window.document.DR.date_ct,window.document.DR.date
_nm,window.document.DR.date_no, 'date');">

    <cfinclude template="select/date_static.cfm">
</select></td>

</tr>
<tr><td colspan="2"><table width="100%"><tr><td class="help"><input type="checkbox"
name="OutputCoordinates">Provide Geographic Coordinates for Sample Locations</td><td align="center"><a
href="javascript:spawnHelp('help.cfm?help=coo');"></a></td></tr>
</table>
</td></tr><tr><td colspan="2">
<input type="hidden" name="shed_nm" value="null">
<input type="hidden" name="shed_no" value="null">
<input type="hidden" name="shed_ct" value="null">
<input type="hidden" name="param_nm" value="null">
<input type="hidden" name="param_no" value="null">

```

```



```

db_query_results.cfm

```

<!-------
Template: db_query_results.cfm (version 6)
Purpose: final
Use: display the results of the query formed in db_query_form.cfm
Author: rob clayton
Date: 8/5/2003
----->

<cfinclude template="header.cfm">
<cfif isDefined("URL.cancel")>
<!-- the user requested that the pull get cancelled --->
<cfoutput>
<table border="0" width="620" align="center">
<tr><td align="left">
<H4>Data Request Cancelled</H4>
<P>You requested that we cancel your data pull.</P>
<P>At this time you have three options to proceed. You may:<P>
<table><tr><td><a href="#Application.FormPage#"></a></td><td class="help">Return to the online request application and <a
href="#Application.FormPage#">give it another try with narrowed criteria.</a></td></tr>
<tr><td><a href="http://www.cityofaustin.org/watershed/datarequest.htm"></a></td><td class="help">Submit a query to us using an <a
href="http://www.ci.austin.tx.us/watershed/datarequest.htm">online form</a> and a Watershed Protection
staff person will fulfill your query within three business days.</td></tr>
<tr><td><a href="default.htm"></a></td><td
class="help">Return to the <a href="default.htm">Datarequest Home Page</a></td></tr></table>
</td></tr>
</Table>
</cfoutput>
<cfelseif isDefined("URL.dbDown")>
<!-- the database didn't respond before the timeout ... or some other error on the query --->
<cfoutput>
<table border="0" width="620" align="center">
<tr><td align="left">
<H4>Transient Error: Database Server Down</H4>
<P>The Database Server is temporarily unusable. It is either responding to an unexpectedly high volume of
requests or is down for maintenance.</P>
<P><B>Please wait fifteen minutes and try your request again. If you continue to receive this message,
you may contact our staff by using the email links below.</B></P>
</td></tr>
</Table>
</cfoutput>
<cfelse>
<cfoutput>

```

```

<Table Border="0" width="620">
<TR><TD align=left>
</cfoutput>
<cfset #strOutputFileName# = "null">
<!--If the user has clicked the submit button, run the query to retrieve product items. ---->
<cfif isDefined("FORM.DataRequestSubmit")>
  <cfif #FORM.Sampling_Media# neq 0>
    <cfset strMediaList = "fs.medium_no #FORM.media_no# AND">
  <cfelse>
    <cfset strMediaList = "-- with no media restriction --">
  </cfif>
  <cfset booFoundAs_Seal=false>
  <cfif listFind(#FORM.Sampling_Media#,'17',',') neq 0>
    <cfset booFoundAs_Seal=true>
  </cfif>
  <cfif listFind(#FORM.Sampling_Media#,'18',',') neq 0>
    <cfset booFoundAs_Seal=true>
  </cfif>
  <cfif booFoundAs_Seal>
    <cfif Find("(",#FORM.shed_no#,1) neq 0>
      <cfset #FORM.shed_no# = replace(#FORM.shed_no#,""," ",116)">
    <cfelse>
      <cfset #FORM.shed_no# = replace(#FORM.shed_no#,"=","IN (")>
      <cfset #FORM.shed_no# = #FORM.shed_no# & ",116)">
    </cfif>
  </cfif>
</cfif>
<cftry>
<!-- note ... the alias on the from clause of the following query is a cluge ---->
<cfquery name="dbaLimitChk" datasource="#Application.dbConnector#">
select sum(row_ct) full_row_ct
from wre_admin.fsdb_weighting fs
where watershed_no #FORM.shed_no# AND
      primary_param_no #FORM.param_no# AND
      #strMediaList#
      year #FORM.date_no#
</cfquery>
<cfcatch type="Any">
<cfoutput>
<cflocation url="#Application.outputPage#?dbDown=true&shed_no=#FORM.shed_no#">
</cfoutput>
</cfcatch>
</cftry>
<cfset BOOunderLimit = TRUE>
<cfset BOOzeroRecCt = FALSE>
<cfoutput query="dbaLimitChk">
  <cfset numRecCt = #dbaLimitChk.full_row_ct#>
<!-- 2 ---->
  <cfif #numRecCt# gte #Application.dbareclimit#>
    <cfset BOOunderLimit = FALSE>
  <cfelseif #numRecCt# LT 1>
    <cfset BOOzeroRecCt = TRUE>
    <cfset BOOunderLimit = FALSE>
  <cfelse>
    <cfset BOOunderLimit = TRUE>
  </cfif><!-- 2 ---->
</cfoutput>
<!-- 2 ---->
<cfif BOOunderLimit>
<cfset strOutputFileName = "#FORM.MySessionID#.csv">
<br>
<!-- 3 ---->
<cfif NOT FileExists('#application.tmpdirectory#\#strOutputFileName#')>
<cftry>
  <cfquery name="getUserDataRequest" datasource="#Application.dbConnector#">
<!-- cachedwithin="#CreateTimeSpan(0,0,2,0)#" ---->

SELECT
W.NAME WATERSHED,
FS.FIELD_SAMPLE_NO REF_NO,
To_Char(FV.VISIT_DATE,'MM/DD/YYYY') SAMPLE_DATE,
DECODE(FS.SAMPLE_TIME,9999,'&nbsp;',TO_CHAR(FS.SAMPLE_TIME,'0999')) SAMPLE_TIME,
SS.NAME SITE_NAME,
ST.DESCRPTION SITE_TYPE,
SM.DESCRPTION MEDIUM,
nvl(FS.DEPH,'.1') DEPTH,

```

```

InitCap(PA.DESCRPTION) PARAMETER,
nvl(LTRIM(RTRIM(SR.QUALIFIER_CODE)), '&nbsp;') QUALIFIER,
SR.PARAM_VALUE RESULT,
UM.DESCRPTION UNIT,
FF.DESCRPTION FILTER,
AM.DESCRPTION METHOD
FROM SAMPLE_RESULT SR,
    FIELD_SAMPLE FS,
    FIELD_VISIT FV,
    SAMPLE_SITE SS,
    PARAMETER PA,
    FILTER_FRACTION FF,
    UNIT_OF_MEASURE UM,
    ANALYSIS_METHOD AM,
    SAMPLE_MEDIUM SM,
    SITE_TYPE ST,
    WATERSHED W,
    PARAMETER_TYPE PT
WHERE
W.WATERSHED_NO = SS.WATERSHED_NO AND
SS.SITE_TYPE_NO = ST.SITE_TYPE_NO AND
SR.FIELD_SAMPLE_NO = FS.FIELD_SAMPLE_NO AND
FS.FIELD_VISIT_NO = FV.FIELD_VISIT_NO AND
SS.SAMPLE_SITE_NO = FV.SAMPLE_SITE_NO AND
SR.PARAMETER_NO = PA.PARAMETER_NO AND
PA.PARAM_TYPE_NO = PT.PARAM_TYPE_NO AND
SR.FILTER_FRACTION_NO = FF.FILTER_FRACTION_NO AND
SR.U_OF_M_NO = UM.U_OF_M_NO AND
SR.METHOD_NO = AM.METHOD_NO AND
FS.MEDIUM_NO = SM.MEDIUM_NO AND
FV.COLLECTING_ENTITY_NO in (1,12) AND
#strMediaList#
PT.PRIMARY_PARAM_NO #FORM.param_no# and
ss.watershed_no #FORM.shed_no# AND
to_char(fv.visit_date, 'YYYY') #FORM.date_no#
</cfquery>
    <cfcatch type="Any">
        <cfoutput>
            <cflocation url="#Application.outputPage#?dbDown=true&shed_no=#FORM.shed_no#">
        </cfoutput>
    </cfcatch>
</cftry>
<!-- this next line simply puts the record count into a variable that will reside
through to a half-baked session -->
<cfset Session.numVestigalRecCt = #getUserDataRequest.RecordCount#>
<!-- and make the qualifier *watershed(s)* agree -->
    <!-- 4 --->
    <cfif #FORM.shed_ct# EQ 1>
        <cfset strShedOrSheds = "watershed">
    <cfelse>
        <cfset strShedOrSheds = "watersheds">
    </cfif> <!-- 4 --->
<!-- and make the qualifier *result(s)* agree -->
<!-- 4 --->
<cfif #getUserDataRequest.RecordCount# EQ 1>
    <cfset strResultOrResults = " result:">
<cfelse>
    <cfset strResultOrResults = " results:">
</cfif> <!-- 4 --->
<cfoutput>
<cfset strSummary="Your search for #FORM.param_nm# data in #FORM.media_nm# media for the #FORM.shed_nm#
#strShedOrSheds# and #FORM.date_nm# returned
<B>#getUserDataRequest.RecordCount#</B>#strResultOrResults#</P>">
<P><B>Summary:</B> #strSummary#</P>
<cfscript>
    strDataTag=REReplace(#strSummary#, "(<[^>]*>)|(,)", "", "All");
</cfscript>
<!-- 4 --->
    <cfif #getUserDataRequest.RecordCount# GT 0>
        <cflock name="DataExportLock" Type="Exclusive" Timeout="30">
            <cfwrite ACTION = "write"
                FILE = "#application.tmpdirectory#\#strOutputFileName#"
                OUTPUT = "DATAFILE GENERATED BY COA WATER RESOURCE INFORMATION SYSTEM DATABASE QUERY TOOL"
                ADDNEWLINE = "Yes">
        </cflock>

```

```

<cflock name="TagExportLock" Type="Exclusive" Timeout="30">
  <cffile ACTION = "append"
    FILE = "#application.tempdirectory#\#strOutputFileName#"
    OUTPUT = "#strDataTag#"
    ADDNEWLINE = "Yes">
  </cffile>
</cflock>
</cfif> <!-- 4 -->
</cfoutput>
<!---Display the results of the query and embed the Array ID into the URL ---->
<cfset
strDataDump="Watershed,Ref_no,Sample_Date,Sample_Time,Site_Name,Site_Type,Medium,Depth_(m),Parameter,Quali
fier,Result,Unit,Filter,Method">
  <cfoutput query="getUserDataRequest">
    <cflock name="DataExportWriteLock" type="exclusive" timeout="30">
      <cffile action="append"
        FILE = "#application.tempdirectory#\#strOutputFileName#"
        OUTPUT = "#strDataDump#"
        ADDNEWLINE = "Yes">
      </cffile>
      <cfset strTime = Replace(#getUserDataRequest.sample_time#,"&nbsp;","")>
      <cfset strQual = Replace(stripCR(#getUserDataRequest.qualifier#),"&nbsp;","")>
      <cfset strDataDump = "">
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.Watershed#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.ref_no#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.sample_date#)>
      <cfset strDataDump=ListAppend(strDataDump,#strTime#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.site_name#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.site_type#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.medium#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.depth#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.parameter#)>
      <cfset strDataDump=ListAppend(strDataDump,#strQual#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.result#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.unit#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.filter#)>
      <cfset strDataDump=ListAppend(strDataDump,#getUserDataRequest.method#)>
    </cflock>
  </cfoutput>
</table><H4>Thank you for using the WRIS Database Query Tool!</h4>
<!--- THIS IS THE SECTION WHERE WE DETERMINE THE FILE SIZE ---->
<!-- note that if the amount of data pulled in a record changes, the divisor should change to reflect the
number of Kb per line in output file ---->
<CFSCRIPT>
  i=ceiling(#getUserDataRequest.RecordCount/#Application.recordByteWeight#);
  if(i gt 4){
    while (i/5 neq round(i/5)){
      i=incrementValue(i);
      if(i gt 2500){
        i=5;
      } } }
  numFileSize = i;
</CFSCRIPT>
<cfoutput>
  <P>Downloadable Data File: <a name="bufort" href="javascript:void(0);" title="This file is not yet
ready to be downloaded. Please wait for the page to finish loading.">#strOutputFileName#</a> (~
#numFileSize# Kb)</P>
  <cfif isDefined("FORM.OutputCoordinates")>
    <P>Downloadable Site Location Coordinate File: <a
href="ftp://#Application.ftpserver#/pub/vendors/wre/erm_site_locations.csv">erm_site_locations.csv</a>.</P>
  </cfif>
  <P>Downloadable Column Description File: <a
href="ftp://#Application.ftpserver#/pub/vendors/wre/erm_column_descriptions.txt">erm_column_descriptions.t
xt</a>.</P>
  <cflock name="DataExportWriteLastLine" type="exclusive" timeout="30">
    <cffile action="append"
      FILE = "#application.tempdirectory#\#strOutputFileName#"
      OUTPUT = "#strDataDump#"
      ADDNEWLINE = "Yes">
    </cffile>
  </cflock>
</cfoutput>
<!--- THIS IS THE FTP SECTION ---->

<cfset strFilePath = #Application.tempdirectory#>
<cfset strFileName = #strOutputFileName#>
<cfset strftpServer = #Application.ftpserver#>

```

```

<cflock scope="Session" Type="Exclusive" Timeout="60" Throwontimeout="Yes">
<cfftp action="open"
    server="#strftpServer#"
    port=21
    username="#Application.ftp_username#"
    password="#Application.pass#"
    connection="Session.filePutter"
    agentname="ColdFusion"
    retrycount="1"
    timeout="60"
    stoponerror="No"
    passive="No">
</cflock>
<cfif NOT cffftp.Succeeded>
    Connection aborted!<br />
    <cfoutput>
        error number #cffftp.errorcode#: #cffftp.errortext#
    </cfoutput>
    <cfabort>
</cfif>
<cflock scope="Session" type="ReadOnly" Timeout="60" ThrowonTimeout="Yes">
<cfftp action="PutFile"
    transfermode="AUTO"
    stoponerror="Yes"
    remotefile="#strFileName#"
    localfile="#strFilePath#\#strFileName#"
    connection="Session.filePutter">
</cflock>
<P>
<CFIF NOT CFFFTP.Succeeded>
    <B><I>The Following error occurred while attempting to upload your file:</B><BR />
    <CFOUTPUT>
        <B><I>Error number #CFFFTP.ErrorCode#: #CFFFTP.ErrorText#</I></B>
    </CFOUTPUT>
</CFIF>
<cflock scope="Session" Type="ReadOnly" Timeout="60" Throwontimeout="Yes">
<cfftp action="close"
    connection="Session.filePutter"
    stoponerror="No">
</cflock>
<cfif NOT cffftp.Succeeded>
    Connection aborted!<br />
    <cfoutput>
        error number #cffftp.errorcode#: #cffftp.errortext#
    </cfoutput>
    <cfabort>
</cfif>
<!-- THIS ENDS THE FTP SECTION --->
<cfelse>
<script language="javascript">
    alert('A submission error ocured. The output page cannot be refreshed without submitting a new query.
    Please select your criteria anew and attempt your query again. ');
    controlProcessWindow('processWindow');
<cfoutput>
    window.document.location='#Application.FormPage#';
</cfoutput>
</script>

</cfif>
<!-- removed "HALF-BAKED" from this point --->

<cfelseif BOOzeroRecCt> <!-- 2 --->
<!-- if there are no records in the statistics table --->

    <cfoutput>
        <H4>Database Query Tool Message: <br />Zero Records Returned</H4>
<P>The criteria you specified returned no results.</P>
<P>At this time you have three options to proceed. You may:
<table><tr><td><a href="#Application.thisdirectory##Application.FormPage#"></a></td><td class="help">Return to the online request application and <a
href="#Application.thisdirectory##Application.FormPage#">give it another try with broadened
criteria.</a></td></tr>
<tr><td><a href="http://www.cityofaustin.org/watershed/datarequest.htm"></a></td><td class="help">Submit a query to us using an <a

```

```

href="http://www.ci.austin.tx.us/watershed/datarequest.htm">online form</a> and a Watershed Protection
staff person will fulfill your query within three business days.</td></tr>
<tr><td><a href="default.htm"></a></td><td
class="help">Return to the <a href="default.htm">Datarequest Home Page</a></td></tr></table>
</cfoutput>
<cfelse> <!-- 2 --->
<!-- if the sum of the unique rowids is GTE the threshold --->
  <cfoutput>
    <body onLoad="controlProcessWindow('processWindow')">
      <H4>Database Query Tool Message: <br />Very Large Request</H4>
<p>You selected criteria that specified a data pull of <b>#numRecCt#</b> data records. The current
rejection threshold is
  <b>#Application.dbareclimit#</b> records.</p>
<P>At this time you have three options to proceed. You may:<P>
<table><tr><td><a href="#Application.thisdirectory##Application.FormPage#"></a></td><td class="help">Return to the online request application and <a
href="#Application.thisdirectory##Application.FormPage#">give it another try with narrowed
criteria.</a></td></tr>
<tr><td><a href="http://www.cityofaustin.org/watershed/datarequest.htm"></a></td><td class="help">Submit a query to us using an <a
href="http://www.ci.austin.tx.us/watershed/datarequest.htm">online form</a> and a Watershed Protection
staff person will fulfill your query within three business days.</td></tr>
<tr><td><a href="default.htm"></a></td><td
class="help">Return to the <a href="default.htm">Datarequest Home Page</a></td></tr></table>
</cfoutput>

</cfif> <!-- 2 --->
</cfif> <!-- 1 --->
<cfoutput>
<cfif #strOutputFileName# neq "null">
<script>
changeLink('#Application.outsidedirectory##strOutputFileName#');
</script>
<!-- end of body text --->
</cfif>
</cfoutput>
</cfif>

<cfinclude template="footer.cfm">

```

header.cfm

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<HTML LANG="EN-US">
<head><META HTTP-EQUIV="expires" CONTENT="-1">
<title>Water Resource Information System: Database Query Tool</title>
<cfoutput>
<BASE HREF="#Application.baseHref#">
</cfoutput>
<LINK REL="stylesheet" HREF="http://www.ci.austin.tx.us/accstyles3.css" TYPE="text/css">

<script language="javascript" src="application.js">
</script>
<noscript>
<h1>This web application requires that you view it with a javascript-enabled browser. Please submit this
<a href="http://www.ci.austin.tx.us/watershed/datarequest.htm">data request form</a> to get your data
within 2-3 working days.</h1>
<h2>note: the form below will not work for you because javascript processes the form to assemble the data
query.</h2>
</noscript>
</head>

<!-- BACKGROUND COLOR - if you want a shaded background, replace #ffffff with #ffffff -->
<body onload="menureset();controlProcessWindow('processWindow')" onBeforeUnload="ermResetForm();"
bgcolor="#ffffff">
<center>
<table align="center" bgcolor="#ffffff" width="620" border="0" cellspacing="0" cellpadding="0" vspace="0"
hspace="0">
  <tr valign="middle">
    <td align="center" width="113" rowspan="4" bgcolor="#ffffff">

```

```

<center><a href="http://www.ci.austin.tx.us"></a></center>
</td>
<td rowspan="4" width="10" bgcolor="#ffffff">
<!-- SEARCH FORM -->
<form method="post" action="http://www.ci.austin.tx.us/cgi-bin/htsearch" style="margin-top: 0px; margin-
bottom: 0px;">
    &nbsp;
</td></tr>
<tr><td height="4" align="center" valign="top" bgcolor="#ffffff">
</td></tr>
<tr>
<td align="center" valign="bottom">
<table border="0" cellspacing="1" cellpadding="1" width="100%"><tr><td valign="middle" align="left"
nowrap>
<input type="hidden" name="select" value="htdig">
<input style="FONT-SIZE: 11px" type="text" maxlength="30" name="words" size="10">
<input type="submit" STYLE="FONT-FAMILY: Verdana, Arial, Helvetica; FONT-SIZE: 10px" value="Find!"> <font
face="Verdana, Arial, Helvetica, sans-serif" size="1">
<a href="http://www.ci.austin.tx.us/search/">Options</a></font>
</td></form>
<td align="right" valign="middle" nowrap>

<script language="JavaScript" src="http://www.ci.austin.tx.us/services.js">
</script>

</tr>
<tr>
<td colspan="3" align="center" valign="top" nowrap>

<!--begin navigation links-->
<br>
<FONT FACE="Verdana, sans-serif" size=1 color="#000000">

<a href="http://www.ci.austin.tx.us/help/directory_a.htm">Directory</a>
<font color="#000000">|</font>
<a href="http://www.ci.austin.tx.us/orgchart.htm">Departments</a>
<font color="#000000">|</font>
<a href="http://www.ci.austin.tx.us/faq">FAQ</a>
<font color="#000000">|</font>
<a href="http://www.ci.austin.tx.us/links.htm">Links</a>
<font color="#000000">|</font>
<a href="http://www.ci.austin.tx.us/sitemap.htm">Site Map</a>
<font color="#000000">|</font>
<a href="http://www.ci.austin.tx.us/help.htm">Help</a>
<font color="#000000">|</font>
<a href="http://www.ci.austin.tx.us/connect/response.htm">Contact Us</a></font><br>

</td></tr></table>
<!--end navigation links-->


</td></tr>
</table>
<LINK REL=stylesheet HREF="application.css" TYPE="text/css">

```

Footer.cfm

```

<cfparam name="attributes.thisPath" default="">
<Table Border="0" width="620">
<TR><TD align=left>
<P>All data in the Water Resource Information System should be considered provisional and may or may not
have been reviewed by project staff.</P><P class="disclaim">The Water Resource Information System is
Designed by the Environmental Resource Management Division of the Watershed Protection and Development
Review department in conjunction with the Information Systems Department of the City of Austin.<br /></P>

</TD>
</TR>
<tr>
<cfoutput>

```

```

        <td><font face="Helvetica, sans-serif" size="2" color="##454577">If you need assistance or
experience difficulties with this application, please <a href="#Application.EmailLink#">contact
us</a>.</font></td>
</cfoutput>
</tr>
</Table>

<br clear="all">
<table bgcolor="ffffff" background="http://www.ci.austin.tx.us/images/whiteshim.gif" width="620"
height="70" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="2" align="center">
      <br><br></td>
    </tr>

<tr>
  <td rowspan="5" align="center">
    <br></td>
    <td><font face="Helvetica, sans-serif" size="1" color="##454577">
      <a href="http://www.ci.austin.tx.us/default.htm">Austin City Connection</a><font color="##000000"> -
</font>
    <td colspan="2" align="center">The Official Web site of the City of Austin</td>
  </tr>
<tr>
  <td><font face="Helvetica, sans-serif" size="1" color="##454577">
    <!--this is where you need to add the custom contact email and contact phone number-->
    Contact Us:</font>
    <font face="Helvetica, sans-serif" size="1" color="##454577">
    <a href="http://www.ci.austin.tx.us/connect/email_robertclayton.htm">Send Email</a>&nbsp;or (512) 974-
1865</font></td>
  </tr>
<tr>
  <td>
    <font face="Helvetica, sans-serif" size="1" color="##454577">
    <a href="http://www.ci.austin.tx.us/legal.htm">Legal Notices</a> <font color="##000000">|</font>
    <a href="http://www.ci.austin.tx.us/privacy.htm">Privacy Statement</a></font></td>
  </tr>
<tr>
  <td><font face="Helvetica, sans-serif" size="1" color="##454577">©
    2003 City of Austin, Texas<font color="##000000">.<font color="##454577"> All Rights
    Reserved.</font></font></td>
  </tr>
<tr>
  <td><font face="Helvetica, sans-serif" size="1" color="##454577">P.O.
    Box 1088, Austin, TX 78767 (512) 974-2000</font></td>
  </tr>
</table>
</td></tr>
</center>
</body>
</html>

```

help_header.cfm

```

<html>
<head>
<title>WRIS Help System</title>
<cfoutput>
<BASE HREF="#Application.baseHref#">
</cfoutput>
<LINK REL=stylesheet HREF="application.css" TYPE="text/css">
<script language="javascript" src="application.js">
</script>
</head>
<body bgcolor="#e7e79c" text="##000000">
<TABLE CLASS="help" ALIGN="CENTER">
<cfoutput>
<cfset numCols=37>
<cfif #FindNoCase('MSIE',HTTP_USER_AGENT)# EQ 0>
<cfset numCols=25>

```

```

</cfif>
<TR><TD CLASS="helpnav" align="left"><a href="javascript:void(0);"
onClick="opener.location='#Application.FormPage#';opener.focus();self.focus();">restore main window to
data request page</a> </td><TD CLASS="helpnav" align="right"> <a href="javascript:void(0);"
onClick="self.close();">close this window</a></TD></TR>
<TR><TD CLASS="help" COLSPAN=2><HR COLOR="##AED5EF"></TD></TR>
</cfoutput>

```

help_footer.cfm

```

<cfoutput>
<TR><TD CLASS="help" COLSPAN=2><br /><P>If this help file has not satisfied your questions about what
watershed you'd like to request data for, please <a
href="javascript:opener.location='#Application.EmailLink#';opener.focus();self.focus();">email us</a> and
request clarification and we will do our best to answer your question (and improve the help file based on
your feedback).</P></TD></TR>
</cfoutput>
<TR><TD CLASS="help" COLSPAN=2><HR COLOR="##AED5EF"></TD></TR>
<cfoutput>
<TR><TD CLASS="helpnav" align="left"><a href="javascript:void(0);"
onClick="opener.location='#Application.FormPage#';opener.focus();self.focus();">restore main window to
data request page</a> </td><TD CLASS="helpnav" align="right"> <a href="javascript:void(0);"
onClick="self.close();">close this window</a></TD></TR>
</cfoutput>
</TABLE>
</html>

```

help.cfm

```

<cfinclude template="help_header.cfm">
<cfoutput>
<cfswitch expression="#URL.help#">
<cfcase value="wsd">
<TR><TD CLASS="help" COLSPAN=2><P><B>Watershed</B></P><P>Currently, we have data for
<script>countSheds();</script> watersheds. Some of these are very familiar to those living in the Austin
area and others less well-known.</P><P>You may select a single watershed, or hold down the <a
href="javascript:shiftHelp();">shift</a> or <a href="javascript:ctrlHelp();">control [ctrl]</a> keys while
clicking to select multiple watersheds.</P><P>You may use the <a href="javascript:void(0);"
onClick="javascript:opener.location='http://www.ci.austin.tx.us/watershed/find.htm';opener.focus();self.fo
cus();">Find Your Watershed</a> map to determine which watersheds you are interested in. Note that some of
the watersheds listed on the map are either not within our jurisdiction or we have not had cause to sample
there. So we don't have data for all the watersheds on the map. Conversely, some of the watersheds
listed in this application are not on the map (because they're too tiny). Use the table below to cross-
reference these smaller sub-watersheds to their larger, receiving watershed.<P></P>
</TD></TR>
<TR><TD CLASS="reverse">Sub Watershed</TD><TD CLASS="reverse">Receiving Watershed</TD></TR>
<TR><TD CLASS="help">Chalk Knob Hollow</TD><TD CLASS="help">Barton Creek</TD></TR>
<TR><TD CLASS="help">Danz Creek</TD><TD CLASS="help">Slaughter Creek</TD></TR>
<TR><TD CLASS="help">Davis Spring Branch</TD><TD CLASS="help">Lake Creek</TD></TR>
<TR><TD CLASS="help">Gaines Creek</TD><TD CLASS="help">Barton Creek</TD></TR>
<TR><TD CLASS="help">Grape Creek</TD><TD CLASS="help">Barton Creek</TD></TR>
<TR><TD CLASS="help">Harrison Hollow</TD><TD CLASS="help">Lake Austin</TD></TR>
<TR><TD CLASS="help">Holman Hollow</TD><TD CLASS="help">Barton Creek</TD></TR>
<TR><TD CLASS="help">Long Branch</TD><TD CLASS="help">Barton Creek</TD></TR>
<TR><TD CLASS="help">Long Hollow Creek</TD><TD CLASS="help">Lake Travis</TD></TR>
<TR><TD CLASS="help">Panther Hollow</TD><TD CLASS="help">Lake Austin</TD></TR>
<TR><TD CLASS="help">Rocky Creek</TD><TD CLASS="help">Barton Creek</TD></TR>
<TR><TD CLASS="help">Short Spring Branch</TD><TD CLASS="help">Barton Creek</TD></TR>
<TR><TD CLASS="help">Tar Branch</TD><TD CLASS="help">Walnut Creek</TD></TR>
<TR><TD CLASS="help">White Branch</TD><TD CLASS="help">Barton Creek</TD></TR>
</cfcase>
<cfcase value="par">
<TR><TD CLASS="help" COLSPAN=2><P><B>Parameter Type
</B></P><P>For simplicity's sake, we have organized our data into <script>countParams();</script>
different parameter types.</P><P>Think of this as the type of <I>stuff</I> you are analyzing a sample for.
For each visit we make to a given location, we gather results for many different parameters. By using the
parameter type criteria you can limit the output
to a set of parameters that interest you.</P>
<P>You may select a single parameter type, or hold down the <a href="javascript:shiftHelp();">shift</a> or
<a href="javascript:ctrlHelp();">control [ctrl]</a> keys while clicking to select multiple parameter

```

types.</P><P>Below are some examples of parameters that fall within the classification of each parameter type.</P>

</td></tr>	
<TR><TD CLASS="help" COLSPAN="2">	
<TABLE border="0">	
<TR><TD CLASS="reverse">Parameter Type</TD><TD CLASS="reverse">Description of Data.</TD></TR>	
<TR><TD CLASS="help" VALIGN="top">Biological</td><TD CLASS="help" VALIGN="top">Biological parameters include information on number and types of aquatic algae, bugs, fish and salamanders.</TD></TR>	
<TR><TD CLASS="help" VALIGN="top">Conventional</td><TD CLASS="help" VALIGN="top">Conventional parameters include the traditional water quality parameters like temperature, pH, dissolved oxygen, conductivity and the major ions.</TD></TR>	
<TR><TD CLASS="help" VALIGN="top">Fertilizer</td><TD CLASS="help" VALIGN="top">Fertilizers include data from special studies on the type and composition of fertilizers used in the Austin area.</TD></TR>	
<TR><TD CLASS="help" VALIGN="top">Herbicide</td><TD CLASS="help" VALIGN="top">Herbicides are chemicals used to kill plants and fungi.</TD></TR>	
<TR><TD CLASS="help" VALIGN="top">Inorganic</td><TD CLASS="help" VALIGN="top">Inorganic parameters do not contain carbon and include metals such as mercury and arsenic.</TD></TR>	
<TR><TD CLASS="help" VALIGN="top">Insecticide</td><TD CLASS="help" VALIGN="top">Insecticides are chemicals used to kill bugs.</TD></TR>	
<TR><TD CLASS="help" VALIGN="top">Organic</td><TD CLASS="help" VALIGN="top">Organic analytes typically contain carbon and include a large number of different parameters such as PAHs, PCBs and hydrocarbons.</TD></TR>	
<TR><TD CLASS="help" VALIGN="top">Physical</td><TD CLASS="help" VALIGN="top">Physical analytes include parameters which describe the environmental conditions and habitat of a particular site.</TD></TR>	

<TR><TD CLASS="help" COLSPAN="2"><P>Because the Parameter Type and Sample Media fields overlap (not all parameter types are found in all sample media and vice-versa) you may want to view our documentation about the relationship between these two criteria.</P></TD></TR>

</cfcase><cfcase value="yea"><TR><TD CLASS="help" COLSPAN="2"><P>Sampling Year</P><P>This is the year in which the sample was collected by City of Austin staff. Please note that there was no data collected in 1983; in addition, not all watersheds have been sampled in every year. Consult this documentation to increase your understanding of our sampling frequency.</P><P>If you are receiving the "Very Large Request" error page we advise that you use the Sampling Year criterion to limit your individual queries to the allowable maximum number of records.</P><P>You may select a single year, or hold down the shift or control [ctrl] keys while clicking to select multiple sampling years.</P></cfcase>

<cfcase value="med_par"><TR><TD CLASS="help" COLSPAN="2"><P>Explanation of Logical Sample Media and Parameter Type Relationship</P><P>To make the Database Query Tool more specific and therefore more powerful, you may select both the type of data (Parameter Type) and the medium from which a sample was collected (Sample Medium) in each data query.</P><P>Please note not all parameter types will be associated with each sample media. For example, if you select <i>Inorganics</i> as the parameter type and <i>Benthic Macroinvertebrate Taxa Count</i> as the sample medium, your query will return no results as only <i>Biological</i> and <i>Physical</i> parameter types are associated with that medium.</P><P>The following lists display the valid criteria for each possible parameter type or sample medium.</P></td></tr>

<TR><TD CLASS="help" COLSPAN="2">	
<TABLE width="100%">	
<TR><TD CLASS="reverse">These Parameter Types:</TD></TR>	
<TR><TD CLASS="help"><form name="medium_v_p_type">	
<select name="p_type"	
onChange="swapOptions(window.document.medium_v_p_type.p_type.options[selectedIndex].text,window.document.m	
edium_v_p_type.p_type,window.document.medium_v_p_type.valid_med);">	
<option>Biological	
<option>Conventional	
<option>Fertilizer	
<option>Herbicide	
<option>Inorganic	
<option>Insecticide	
<option>Organic	

```

<option>Physical
</select>
</TD></TR>
<TR>
<TD CLASS="reverse">Are Measured in these Media:</TD></TR>
<TR><TD CLASS="help"><textarea cols=#numCols# rows=12 name="valid_med" readonly>Choose Parameter Type from
list above.</textarea>
</TD></TR>
<TR><TD class="help"><HR></TD></TR>
<TR><TD CLASS="reverse">In this Sample Medium:</TD></TR>
<TR><TD CLASS="help"><select name="m_type"
onChange="swapOptions(window.document.medium_v_p_type.m_type.options[selectedIndex].text,window.document.m
edium_v_p_type.m_type,window.document.medium_v_p_type.valid_pt);" >
<option>Surface Water
<option>Ground Water
<option>Sediment
<option>Soil
<option>Habitat
<option>Benthic Cover
<option>Benthic Macroinvertebrate Taxa Count
<option>Diatom Taxa Count
<option>Fish Salamander Taxa Count
<option>Rain Water
<option>Channel Assessment
<option>Pore Water
<option>Pavement Sealer
<option>Asphalt
</select>
</TD></TR>
<TR>
<TD CLASS="reverse">We look for these Parameter Types:</TD></TR>
<TR><TD CLASS="help"><textarea cols=#numCols# rows=7 name="valid_pt" readonly>Choose Sample Media from
list above.</textarea>
</TD></TR>
</form>
</TD></TR>
</TABLE></TD></TR>
<TR><TD CLASS="help" COLSPAN="2">
<P>If after consulting this help file you are still not sure whether or not you want to limit your query
by sample media,
please set the <B>sample media</B> criterion so that only the "Choose Sample Media [optional]" option is
selected.</td></tr>
</cfcase>
<cfcase value="yea_more">
<TR><TD CLASS="help" COLSPAN=2><P><B>Sampling Year Frequency</B></P><P>
We have had several projects that have intensely studied watersheds for several years; and most watersheds
are only sampled quarterly for one year every three years in a phase-rotation for surface water sampling.
Here's a table showing the number of different years of data we have for each watershed (note, this is
only a rough guide ... the data are of all types ... we may not have 19 years of <b>Benthic</b> data for
Barton Creek, as an example).</P>

<table>
<tr><td class="reverse">Watershed</td><td class="reverse">Years of Data</td></tr>
<tr><td class="help">Barton Creek</td><td class="help">19</td></tr><tr><td class="help">Bull Creek</td><td
class="help">19</td></tr><tr><td class="help">Shoal Creek</td><td class="help">16</td></tr><tr><td
class="help">Town Lake</td><td class="help">14</td></tr><tr><td class="help">Williamson Creek</td><td
class="help">14</td></tr><tr><td class="help">Blunn Creek</td><td class="help">13</td></tr><tr><td
class="help">East Bouldin Creek</td><td class="help">13</td></tr><tr><td class="help">Waller Creek</td><td
class="help">13</td></tr><tr><td class="help">Little Barton Creek</td><td class="help">12</td></tr><tr><td
class="help">Walnut Creek</td><td class="help">12</td></tr><tr><td class="help">Short Spring
Branch</td><td class="help">12</td></tr><tr><td class="help">Eanes</td><td
class="help">11</td></tr><tr><td class="help">Little Walnut Creek</td><td class="help">11</td></tr><tr><td
class="help">West Bull Creek</td><td class="help">11</td></tr><tr><td class="help">Onion Creek</td><td
class="help">11</td></tr><tr><td class="help">Boggy Creek</td><td class="help">10</td></tr><tr><td
class="help">Harper's Branch</td><td class="help">10</td></tr><tr><td class="help">Johnson Creek</td><td
class="help">10</td></tr><tr><td class="help">White Branch</td><td class="help">10</td></tr><tr><td
class="help">Slaughter Creek</td><td class="help">10</td></tr><tr><td class="help">Lake Austin</td><td
class="help">10</td></tr><tr><td class="help">Little Bee Creek</td><td class="help">9</td></tr><tr><td
class="help">Tannehill Branch</td><td class="help">9</td></tr><tr><td class="help">Rocky Creek</td><td
class="help">9</td></tr><tr><td class="help">Rinard Creek</td><td class="help">9</td></tr><tr><td
class="help">Marble Creek</td><td class="help">9</td></tr><tr><td class="help">Taylor Slough
(North)</td><td class="help">9</td></tr><tr><td class="help">West Bouldin Creek</td><td
class="help">9</td></tr><tr><td class="help">Bear Creek</td><td class="help">8</td></tr><tr><td
class="help">Bee Creek</td><td class="help">8</td></tr><tr><td class="help">South Boggy Creek</td><td
class="help">8</td></tr><tr><td class="help">Grape Creek</td><td class="help">8</td></tr><tr><td

```

Fort Branch	7
Taylor Slough (South)	7
Buttermilk Branch	6
Carson Creek	5
Country Club Creek	5
West Country Club Creek	5
Lake Creek	5
Gilleland Creek	5
Brushy Creek	4
Cow Fork Bull Creek	4
Decker Creek	4
Not Applicable	4
Harris Branch	4
Elm Creek	4
Davis Spring Branch (West)	3
Dry Creek	3
South Fork Dry Creek	3
Rattan Creek	3
Panther Hollow	3
North Fork Dry Creek	3
Long Hollow Creek	3
Little Bear Creek	3
Huck's Slough	3
Gaines Creek	3
Dry Creek (North)	3
Colorado River	3
Cottonmouth Creek	3
Wells Branch	3
Turkey Creek	3
Lake Travis	2
Buttercup Creek	1
Danz Creek	1
Harrison Hollow	1
Tar Branch	1
Long Branch	1
Holman Hollow	1
Chalk Knob Hollow	1

Sample Media

This is the medium from which the sample was collected. In other words, this is the substance being analyzed. The individual sampling media for which data is available are described below. If you are unsure about the specific medium in which you are interested, you may leave this field blank, or you may contact us for more information.

You may select a single watershed, or hold down the [shift](#) or [control](#) keys while clicking to select multiple sample media.

[Explanation of the Relationships between Parameter Type and Sample Medium](#)

Sample Medium	Description of Data
Surface Water	Water in streams and lakes.
Ground Water	Underground water sampled from springs and wells.
Sediment	Loose material at the bottom of streams and lakes.
Soil	Loose material not typically submerged in water.
Habitat	Data describing the health and aesthetics of the land areas around streams and lakes.
Benthic Cover	Data describing the composition of the bottom of stream beds.
Benthic Macroinvertebrate Taxa Count	Biological information on the number and types of aquatic bugs and organisms.
Diatom Taxa Count	Biological information on the number and types of diatoms (microscopic algae living in water.)
Rain Water	Information on the quality and amount of rainfall.
Channel Assessment	Data describing the structure and stability of stream banks.
Fish/Salamander Taxa Count	Biological information on the number and types of fish or salamanders in a surface water site.
Fish Tissue	Chemical analyses conducted on the flesh of fish taken from streams and lakes.
Pore Water	Chemical analysis of the water separated from samples of sediment.
Asphalt	Data on the composition of different asphalt pavements.
Pavement Sealer	Data on the chemicals used to cover and protect the surface of asphalt and concrete.

3. Click the checkbox beside **Enable JavaScript**.
 4. Click **OK** button.
-

- <u>Netscape Communicator/Navigator 4.x:</u>

1. From the **Edit** menu, choose **Preferences**.
 2. Click the **Advanced** category.

 3. Click the checkbox beside **Enable JavaScript**.
 4. Click **OK** button.</p>

<p>After you have made these changes, please close and re-open your browser.</p>

<p><u>Other browsers:</u>

To see if your browser supports JavaScript, and for detailed instructions about how to enable this feature, see the online **Help** for your browser.</p></td></tr>

</table>

<cfinclude template="footer.cfm">

date_static.cfm

```
<OPTION selected value="ALL">From 1/1/1982 to present .</OPTION>
<OPTION value="2003">2003</OPTION>
<OPTION value="2002">2002</OPTION>
<OPTION value="2001">2001</OPTION>
<OPTION value="2000">2000</OPTION>
<OPTION value="1999">1999</OPTION>
<OPTION value="1998">1998</OPTION>
<OPTION value="1997">1997</OPTION>
<OPTION value="1996">1996</OPTION>
<OPTION value="1995">1995</OPTION>
<OPTION value="1994">1994</OPTION>
<OPTION value="1993">1993</OPTION>
<OPTION value="1992">1992</OPTION>
<OPTION value="1991">1991</OPTION>
<OPTION value="1990">1990</OPTION>
<OPTION value="1989">1989</OPTION>
<OPTION value="1988">1988</OPTION>
<OPTION value="1987">1987</OPTION>
<OPTION value="1986">1986</OPTION>
<OPTION value="1985">1985</OPTION>
<OPTION value="1984">1984</OPTION>
<OPTION value="1983">1983 (no data)</OPTION>
<OPTION value="1982">1982</OPTION>
```

media_static.cfm

<!---- holds the list of values that can be in a select box -- for instance, watershed, site_name ---->

```
<OPTION selected value="0">Choose Sample Media [optional] .</OPTION>
<OPTION value="1">Surface Water</OPTION>
<OPTION value="9">Ground Water</OPTION>
<OPTION value="2">Sediment</OPTION>
<OPTION value="5">Soil</OPTION>
<OPTION value="7">Habitat</OPTION>
<OPTION value="6">Benthic Cover</OPTION>
<OPTION value="13">Benthic Macroinvert. Taxa Count</OPTION>
<OPTION value="14">Diatom Taxa Count</OPTION>
<OPTION value="8">Rain Water</OPTION>
<OPTION value="12">Channel Assessment</OPTION>
<OPTION value="15">Fish/Salamander Taxa Count</OPTION>
<OPTION value="4">Fish Tissue</OPTION>
<OPTION value="16">Pore Water</OPTION>
<OPTION value="18">Asphalt</OPTION>
<OPTION value="17">Pavement Sealer</OPTION>
```

param_type_static.cfm

```
<OPTION Selected value="0">Choose Parameter Type .</OPTION>
<OPTION value="2" text="(bugs, fish)">Biological</OPTION>
<OPTION value="61">Conventional</OPTION>
<OPTION value="64">Fertilizer</OPTION>
<OPTION value="5">Herbicide</OPTION>
<OPTION value="6">Inorganic</OPTION>
<OPTION value="10">Insecticide</OPTION>
<OPTION value="8">Organic</OPTION>
<OPTION value="11">Physical</OPTION>
```

watershed_number_static.cfm

```
<OPTION selected value="0">Choose a Watershed . . . . .</OPTION>
<OPTION value="1">Barton Creek</OPTION>
<OPTION value="12">Bear Creek</OPTION>
<OPTION value="20">Bear Creek (West)</OPTION>
<OPTION value="11">Bee Creek</OPTION>
<OPTION value="15">Blunn Creek</OPTION>
<OPTION value="17">Boggy Creek</OPTION>
<OPTION value="19">Brushy Creek</OPTION>
<OPTION value="7">Bull Creek</OPTION>
<OPTION value="10">Buttercup Creek</OPTION>
<OPTION value="16">Buttermilk Branch</OPTION>
<OPTION value="23">Carson Creek</OPTION>
<OPTION value="27">Chalk Knob Hollow</OPTION>
<OPTION value="32">Colorado River</OPTION>
<OPTION value="34">Cottonmouth Creek</OPTION>
<OPTION value="31">Country Club Creek</OPTION>
<OPTION value="33">Cow Fork Bull Creek</OPTION>
<OPTION value="39">Danz Creek</OPTION>
<OPTION value="46">Davis Spring Branch</OPTION>
<OPTION value="38">Decker Creek</OPTION>
<OPTION value="40">Dry Branch</OPTION>
<OPTION value="42">Dry Creek (North)</OPTION>
<OPTION value="118">Eanes</OPTION>
<OPTION value="4">East Bouldin Creek</OPTION>
<OPTION value="47">Elm Creek</OPTION>
<OPTION value="9">Fort Branch</OPTION>
<OPTION value="119">Gaines Creek</OPTION>
<OPTION value="48">Gilleland Creek</OPTION>
<OPTION value="51">Grape Creek</OPTION>
<OPTION value="58">Harper's Branch</OPTION>
<OPTION value="59">Harris Branch</OPTION>
<OPTION value="57">Harrison Hollow</OPTION>
<OPTION value="56">Holman Hollow</OPTION>
<OPTION value="60">Huck's Slough</OPTION>
<OPTION value="62">Johnson Creek</OPTION>
<OPTION value="69">Lake Austin</OPTION>
<OPTION value="70">Lake Creek</OPTION>
<OPTION value="71">Lake Travis</OPTION>
<OPTION value="63">Little Barton Creek</OPTION>
<OPTION value="65">Little Bear Creek</OPTION>
<OPTION value="64">Little Bee Creek</OPTION>
<OPTION value="6">Little Walnut Creek</OPTION>
<OPTION value="72">Long Branch</OPTION>
<OPTION value="67">Long Hollow Creek</OPTION>
<OPTION value="76">Marble Creek</OPTION>
<OPTION value="80">North Fork Dry Creek</OPTION>
<OPTION value="81">Onion Creek</OPTION>
<OPTION value="83">Panther Hollow</OPTION>
<OPTION value="87">Rattan Creek</OPTION>
<OPTION value="88">Rinard Creek</OPTION>
<OPTION value="89">Rocky Creek</OPTION>
<OPTION value="2">Shoal Creek</OPTION>
<OPTION value="98">Short Spring Branch</OPTION>
<OPTION value="94">Slaughter Creek</OPTION>
<OPTION value="90">South Boggy Creek</OPTION>
<OPTION value="93">South Fork Dry Creek</OPTION>
<OPTION value="101">Tannehill Branch</OPTION>
<OPTION value="103">Tar Branch</OPTION>
<OPTION value="106">Taylor Slough (North)</OPTION>
```

<OPTION value="107">Taylor Slough (South)</OPTION>
<OPTION value="5">Town Lake</OPTION>
<OPTION value="105">Turkey Creek</OPTION>
<OPTION value="3">Waller Creek</OPTION>
<OPTION value="8">Walnut Creek</OPTION>
<OPTION value="112">Wells Branch</OPTION>
<OPTION value="109">West Bouldin Creek</OPTION>
<OPTION value="108">West Bull Creek</OPTION>
<OPTION value="120">West Country Club Creek</OPTION>
<OPTION value="110">White Branch</OPTION>
<OPTION value="113">Williamson Creek</OPTION>

Appendix C: JavaScript (1 of 8 pages)

```
/* application.js associated with the WRIS database query tool application
   adapted from many sources by robert.clayton@the city of austin's domain.tx.us

   feel free to adapt these scripts to suit your purposes... no warranty expressed or implied
   use at own risk.

   15:03 10/14/2003 */

<!-- declarations -->

var jck = true;
var processWindow = null;
var annoyanceFactor = null;
var strCfmForm = 'db_query_form.cfm';
var strCfmResult = 'db_query_results.cfm';
var annoyanceFactorDos = false;
var Biological = new Array("Benthic Cover"," Benthic Macroinvertebrate Taxa Count"," Diatom Taxa Count",
Fish/Salamander Taxa Count"," Ground Water"," Habitat"," Pore Water"," Sediment"," Surface Water");
var Conventional = new Array("Benthic Cover"," Ground Water"," Habitat"," Pore Water"," Rain Water",
Sediment"," Soil"," Surface Water");
var Fertilizer = new Array("Soil");
var Herbicide = new Array("Ground Water"," Pore Water"," Sediment"," Soil"," Surface Water");
var Inorganic = new Array("Asphalt"," Ground Water"," Pavement Sealer"," Pore Water"," Sediment"," Soil",
Surface Water");
var Insecticide = new Array("Ground Water"," Pore Water"," Sediment"," Soil"," Surface Water");
var Organic = new Array("Asphalt"," Ground Water"," Pavement Sealer"," Pore Water"," Sediment"," Soil",
Surface Water");
var Physical = new Array("Asphalt"," Benthic Cover"," Benthic Macroinvertebrate Taxa Count"," Channel
Assessment"," Diatom Taxa Count"," Fish/Salamander Taxa Count"," Ground Water"," Habitat"," Pavement
Sealer"," Sediment"," Soil"," Surface Water");
var Surface_Water = new Array("Biological"," Conventional"," Herbicide"," Inorganic"," Insecticide",
Organic"," Physical");
var Ground_Water = new Array("Biological"," Conventional"," Herbicide"," Inorganic"," Insecticide",
Organic"," Physical");
var Sediment = new Array("Biological"," Conventional"," Herbicide"," Inorganic"," Insecticide"," Organic",
Physical");
var Soil = new Array("Conventional"," Fertilizer"," Herbicide"," Inorganic"," Insecticide"," Organic",
Physical");
var Habitat = new Array("Biological"," Conventional"," Physical");
var Benthic_Cover = new Array("Biological"," Conventional"," Physical");
var Benthic_Macroinvertebrate_Taxa_Count = new Array("Biological"," Physical");
var Diatom_Taxa_Count = new Array("Biological"," Physical");
var Fish_Salamander_Taxa_Count = new Array("Biological"," Physical");
var Rain_Water = new Array("Conventional");
var Channel_Assessment = new Array("Physical");
var Pore_Water = new Array("Biological"," Conventional"," Herbicide"," Inorganic"," Insecticide",
Organic");
var Pavement_Sealer = new Array("Inorganic"," Organic"," Physical");
var Asphalt = new Array("Inorganic"," Organic"," Physical");

<!-- begin functions -->

function menureset()
{
  for (var i=0; i < document.forms.length; i++)
  {
    for(var j = 0; j < document.forms[i].length; j++)
    {
      if(document.forms[i][j].type.indexOf("select") > -1 &&
        document.forms[i][j].selectedIndex)
        return document.forms[i][j].selectedIndex = 0;
    }
  }
}
// ***** end menureset() *****
```

Appendix C: JavaScript (2 of 8 pages)

```
function changeLink(linkdescription)
{
  for (LinkNo = 0; LinkNo < document.links.length; LinkNo++) {
    if (document.links[LinkNo].href == 'javascript:void(0);') {
      document.links[LinkNo].href = linkdescription;
      document.links[LinkNo].title = "Link to Data file. Different browsers will handle this link
differently. If prompted, choose 'save as' to put file on your hard drive."
    }
  }
}
// ***** end changeLink() *****

function checkFishTaxa(){
  for (var m=0; m < document.DR.Sampling_Media.length; m++)
  {
    if (document.DR.Sampling_Media.options[m].value == 6) {
      var x=m;
    }
    if (document.DR.Sampling_Media.options[m].value == 15) {
      var p=m;
    }
  }

  for (var m=0; m < document.DR.Sampling_Media.length; m++)
  {
    if (document.DR.Sampling_Media.options[m].value == 15) {
      if (document.DR.Sampling_Media.options[m].selected) {
        document.DR.Sampling_Media.options[x].selected=true;
        document.DR.Sampling_Media.options[m].selected=true;
      }
    }
  }
}
// ***** end checkFishTaxa() *****

function checkForNullCriteria()
{
  checkForNull = false;
  if (window.document.DR.shed_no.value == "null"){
    checkForNull = true;
  } else {
    if (window.document.DR.param_no.value == "null") {
      checkForNull = true;
    } else {
      if (window.document.DR.query_changed == "null") {
        checkForNull = true;
      }
    }
  }
}

return checkForNull;
}
// ***** end checkForNullCriteria() *****

function checkMainStatus()
{
  if (checkUrlForFilename(opener.document.location,strCfmResult) == strCfmResult)
  {
    self.close();
  } else {
    null;
  }
}
// ***** end checkMainStatus() *****
```


Appendix C: JavaScript (4 of 8 pages)

```
function countSheds() {

    pickle = checkUrlForFilename(opener.document.location, strCfmForm);

    if (pickle == strCfmForm) {
        document.write(opener.document.DR.Watershed.length-1);
    } else {
        document.write('74');
    }
}
// ***** end countSheds() *****

function countParams() {
    pickle = checkUrlForFilename(opener.document.location, strCfmForm);

    if (pickle == strCfmForm) {
        document.write(opener.document.DR.Parameter_Type.length-1);
    } else {
        document.write('8');
    }
}
// ***** end countParams() *****

function ctrlHelp(){
alert(' Use CTRL to select and deselect individual items. \n\r Press and hold CTRL + click on an item +
click on another item to include it in the selection. \n\r If you hold CTRL and click a selected item it
will deselect it. ');
}
// ***** end ctrlHelp() *****

function ermResetForm(){
if (checkUrlForFilename(window.document.location, strCfmForm) == strCfmForm)
{
    window.document.DR.reset();
}
}
// ***** end ermResetForm() *****

function form_check()
{
if (window.document.DR.Watershed.options[0].selected == true)
{
    alert("please select one or more watersheds and make sure the 'choose a watershed' item is NOT
selected.");
    return false;
} else
{
    if (window.document.DR.Parameter_Type.options[0].selected == true)
    {
        alert("please select one or more types of parameter");
        return false;
    } else
    {
        if (checkForNullCriteria() == false ){

            window.document.DR.validated.value = "true";

spawnPop('uf_processing.htm', 'processWindow', 'width=350,height=250,left=200,top=300,toolbar=no,menubar=no,sc
rollbars=yes,resizable=yes');
            return true;
        } else
        {
            if (annoyanceFactorDos == true) {
                alert('A submission error occured (again).\n Please reselect each query item to ensure your
criteria are registered by the Query tool. If you continue to receive this error please contact
robert.clayton@ci.austin.tx.us and be sure to reference what web browser software you were using at the time
of the error. ');
                window.document.DR.reset();
                return false;
            } else {
```

Appendix C: JavaScript (5 of 8 pages)

[code continues uninterrupted]

```
        alert('A submission error occured. Please reselect each query item to ensure your criteria
are registered by the tool.');
```

```
        annoyanceFactorDos = true;
        window.document.DR.reset();
        return false;
    }
}
}
}
// ***** end form_check() *****

function makeSessionID(input){
var now = new Date();
var mon = now.getMonth();

if (mon < 9) {
    mon = "0" +(mon+1);
} else {
    mon = (mon + 1);
}
var str = ('coaerm' +now.getFullYear() + "-" +mon + "-" +now.getDate() +now.getHours() +now.getMinutes()
+now.getSeconds() +input);
window.document.DR.MySessionID.value = str;
}
// **** end makeSessionID() *****

function notNN4(){
netNarc = true;
if (navigator.appName == 'Netscape') {
if (navigator.appVersion.substr(0,1) == '4'){
netNarc = false;
}
}
return netNarc;
}
// ***** end notNN4() *****

function processLists(input_ctrl,output_ct_ctrl,output_nm_ctrl,output_no_ctrl,cls_desc)
{
var num_CmCt = 0;
var numFlag = 0;
var str_optionNames = "";
var str_valueNums = "";
var str_FirstValue = input_ctrl.options[0].value;

<!-- update the control that tells the form handler that they've modified their query -->

window.document.DR.query_changed.value = "not null";

if (clearSampleMedia(input_ctrl) == true){
<!-- handle dates and numeric/string lists differently. branch using value[0] of control ---->

if (input_ctrl.options[0].value == 0)

{
<!-- we're dealing with a list of string values ---->

for (loop=0; loop < input_ctrl.options.length; loop++)

{ <!-- count through the control to find out the number of selected items and also compile the comma-
delimited list of values for the query to the database ---->

if (input_ctrl.options[loop].value != 0)

{
if (input_ctrl.options[loop].selected == true)

{
num_CmCt = num_CmCt + 1;
if (str_valueNums.length < 1)
```

Appendix C: JavaScript (6 of 8 pages)

[code continues uninterrupted]

```
{ <!-- if the str var is null, don't prepend with comma -->
  str_valueNums = input_ctrl.options[loop].value;
} else { <!-- if the str var has a value, separate with a comma --->
  str_valueNums = str_valueNums + "," + input_ctrl.options[loop].value;
}
}
}
output_ct_ctrl.value=num_CmCt; <!-- update the form field that holds the count --->

if (num_CmCt == 1)
{
  str_valueNums = " = " +str_valueNums;
} else {

  str_valueNums = "in (" +str_valueNums +)";

}

for (loop=0; loop < input_ctrl.options.length; loop++)

{ <!-- no loop through to create the grammatical english version --->

  if (input_ctrl.options[loop].value != 0)

  {
    if (input_ctrl.options[loop].selected == true)
    {
      if (str_optionNames.length < 1)
      {
        if (output_ct_ctrl.value == 1)
        {
          str_optionNames = "<strong class=" +cls_desc+">" +input_ctrl.options[loop].text + "</strong>";
          numFlag++;
        } else {
          str_optionNames = "<strong class=" +cls_desc+">" +input_ctrl.options[loop].text;
          numFlag++;
        }
      } else if (numFlag < (num_CmCt - 1))
      {
        str_optionNames = str_optionNames +", " +input_ctrl.options[loop].text;
        numFlag++;
      } else
      {
        str_optionNames = str_optionNames + "</strong> and <strong class=" +cls_desc+">"
+input_ctrl.options[loop].text + "</strong>";
        numFlag++;
      }
    }
  }
} else {

<!-- we're dealing with date values --->

if (input_ctrl.options[0].selected == true)

{ /* if they select all -- use the appropriate SQL and language */
  str_valueNums = "> 1981";
  str_optionNames = "for all the years available";
  output_ct_ctrl.value=1;
} else {
  /* if they did not select all, run through the array and pull the values
  and put them into a string and keep a running total so as to aid in
  making the language correct (so we'll easily know when we're one away
  from the end of the list later) */
  selct = 0;
  loop=(input_ctrl.options.length-1);
  while (loop > -1)
```

Appendix C: JavaScript (7 of 8 pages)

[code continues uninterrupted]

```
{
  if (input_ctrl.options[loop].selected == true)
  {
    if (str_valueNums.length < 1)
    {
      /* the output string is empty ... put in the first value please */
      str_valueNums = input_ctrl.options[loop].value;
      selct = selct + 1;
    } else {
      str_valueNums = str_valueNums + "," + input_ctrl.options[loop].value;
      selct = selct + 1;
    }
  }
  loop = loop - 1
}
output_ct_ctrl.value=selct;
if (selct == 1)
{
  /* if there's only one ... update the SQL appropriately */
  str_valueNums = "=" + str_valueNums;
} else {
  /* if there's a set of two or more update the SQL appropriately*/
  str_valueNums = "in (" + str_valueNums + ")";
}
loop=(input_ctrl.options.length-1);
while (loop > -1)
{
  if (input_ctrl.options[loop].selected == true)
  {
    if (str_optionNames.length < 1)
    {
      /* this is one date */
      str_optionNames = "<strong class=cls_desc>" + input_ctrl.options[loop].value + "</strong>";
      selct = selct - 1;
    } else {
      if (selct > 1)
      {
        /* this is one of three or more dates */
        str_optionNames = str_optionNames + "</strong>, " + "<strong class=cls_desc>"
+input_ctrl.options[loop].value + "</strong>";
        selct = selct - 1;
      } else {
        /* this is the next to last of a string of two or more dates */
        str_optionNames = str_optionNames + "</strong> and <strong class=cls_desc>"
+input_ctrl.options[loop].value + "</strong>";
      }
    }
  }
  loop = loop - 1
}
/* this makes the years language agree */
if (output_ct_ctrl.value == 1)
{
  str_optionNames = "for the year " + str_optionNames;
} else {
  str_optionNames = "for the years " + str_optionNames;
}
}
output_nm_ctrl.value = str_optionNames;
output_no_ctrl.value = str_valueNums;
}
}
// ***** end function processLists() *****
```

Appendix C: JavaScript (8 of 8 pages)

```
function progBar(theSecs,theDistance)
{
  if (notNN4()){
    if (jck == true) {
      interVal = 1020;
      theDistance = parseInt(theDistance);
      window.document.images[3].width = theDistance;
      window.document.images[2].width = 1;
      thePace = parseInt(theDistance / theSecs);
      jck = false;

    }
    window.document.images[3].width -= thePace;
    if (window.document.images[2].width > (theDistance - thePace)){
      window.document.images[2].width = theDistance;
    } else {
      window.document.images[2].width += thePace;
    }
    if (window.document.images[2].width < theDistance ){
      the_timeout = setTimeout("progBar(" +theSecs +"," +theDistance +");", +interVal);
    } else {
      jck = true;
    }
  } else {
    window.document.images[3].src = "images/shim.gif";
    window.document.images[2].src = "images/shim.gif";
  }
}

function shiftHelp(){
alert(' Use SHIFT to select ALL items between two selected items. \n\r Press and hold SHIFT + click on an
item + click on another item down or up in the list -- all between are selected. ');
}
// ***** end shiftHelp() *****

function spawnHelp(fileName) {
spawnPop(fileName, 'helpWindow', 'height=395,width=390,left=450,top=110,resizable=yes,scrollbars=yes,location=
no,status=no,toolbar=no,menubar=no');
}
// ***** end spawnHelp() *****

function spawnPop(fileName,windowName,optionString)
{
  windowName = window.open(fileName,windowName,optionString);
  windowName.focus();
}
// **** end spawnPop() *****

function swapOptions(the_array_name,src_ctl,aff_ctl)
{
  if (the_array_name.indexOf(" ") != -1) {
    pattern=/ /g;
    the_array_name=the_array_name.replace(pattern, '_');
  }
  var the_array = eval(the_array_name);

  for (loop=0; loop < the_array.length; loop++)
  {
    if (loop == 0){
      aff_ctl.value = " " +the_array[loop];
    }else{
      aff_ctl.value += "\n" +the_array[loop];
    }
  }
}
// ***** end swapOptions *****
```

Appendix D: Database Query Tool Application Event Model

FILE: default.cfm called as a result of following a "Database Query Tool" link on the page: default.htm

Object	Instance	Type	Name or Text	Event	Script
Document	default.cfm	cold fusion template		onLoad	controlProcessWindow('processWindow');
Document	default.cfm	cold fusion template		onBeforeUnload	ermResetForm();
TextLink	0		"shift"	href	javascript:shiftHelp()
TextLink	1		"control [ctrl]"	href	javascript:ctrlHelp()
Img	1		img/help.gif	href	javascript:spawnHelp(x)
Img	2		img/help.gif	href	javascript:spawnHelp(x)
Form	0		DR	action	see cold fusion section
Form	0		DR	onSubmit	makeSessionID(x);return form_check();
Input	0	hidden/text	MySessionID		
Select	1	multiple	Watershed	OnChange	processLists(x,y,z,aa,bb);
Select	2	multiple	Parameter_Type	OnChange	processLists(x,y,z,aa,bb);
Img	3		img/help.gif	href	javascript:spawnHelp(x)
Img	4		img/help.gif	href	javascript:spawnHelp(x)
Select	3	multiple	Sampling_Media	OnChange	processLists(x,y,z,aa,bb);
Select	4	multiple	Sampling_Years	OnChange	processLists(x,y,z,aa,bb);
Img	5		img/help.gif	href	javascript:spawnHelp(x)
Input	1	hidden/text	shed_nm	updated by the onChange script from watershed select.	
Input	2	hidden/text	shed_no	updated by the onChange script from watershed select.	
Input	3	hidden/text	shed_ct	updated by the onChange script from watershed select.	
Input	4	hidden/text	param_nm	updated by the onChange script from parameter type select.	
Input	5	hidden/text	param_no	updated by the onChange script from parameter type select.	
Input	6	hidden/text	param_ct	updated by the onChange script from parameter type select.	
Input	7	hidden/text	media_nm	updated by the onChange script from sampling_media select.	
Input	8	hidden/text	media_no	updated by the onChange script from sampling_media select.	
Input	9	hidden/text	media_ct	updated by the onChange script from sampling_media select.	
Input	10	hidden/text	date_nm	updated by the onChange script from sampling_year select.	
Input	11	hidden/text	date_no	updated by the onChange script from sampling_year select.	
Input	12	hidden/text	date_ct	updated by the onChange script from sampling_year select.	
Input	13	hidden/text	validated	updated by process form_check.	
Input	14	hidden/text	query_changed	updated by process processLists.	
Input	16	reset	DataRequestReset	onClick	javascript:annoyanceFactor=null;

FILE: output.cfm called by cold fusion as a result of form submission on page default.cfm

Object	Instance	Type	Name or Text	Event	Script
Document	output.cfm	cold fusion template		same as above	

inLine Script

changeLink()

APPENDIX E: Cascading Stylesheet Reference (p. 1 of 2)

Application.css

```
TD.app    {
    font-family : geneva, arial, helvetica;
    font-size : 8pt;
    font-weight : Normal;
    border-style : double;
    border-color : #ADA5EF;
}

TD.head   {
    font-family : geneva, arial, helvetica;
    font-size : 10pt;
    font-weight : Bold;
    color : #000000;
    text-align : center;
    border-style : solid;
    border-color : #ADA5EF;
    background-color : #FFFFFF;
}

H4        {
    font-family : geneva, arial, helvetica;
    font-size : 12pt;
    font-weight : Bold;
    color : #000000;
}

P         {
    font-family : geneva, arial, helvetica;
    font-size : 9pt;
    font-weight : Normal;
    color : #000000;
}

P.disclaim {
    font-family : geneva, arial, helvetica;
    font-size : 8pt;
    font-weight : Normal;
    color : #000000;
}

.parameter {
    font-weight : Bold;
    color : #3512B8;
}

.media     {
    font-weight : Bold;
    color : #71D01C;
}

.watershed {
    font-weight : Normal;
    color : #3512B8;
}

.date      {
    font-weight : Bold;
    font-style : Italic;
    color : #3512B8;
}

.inputHead {
    font-family : geneva, arial, helvetica;
    font-weight : Bold;
    font-size : 12pt;
}
```

Appendix E: Cascading Stylesheet Reference (p. 2 of 2)

```
.inputHeadOptional    {
    font-family : geneva, arial, helvetica;
    font-weight : Bold;
    font-size   : 12pt;
    background-color : #ADA5EF;
}

.reverse              {
    font-weight : Bold;
    font-size   : 11pt;
    color       : #FFFFFF;
    border-style : none;
    background-color : #000000;
}

.help                 {
    font-family : geneva, arial, helvetica;
    font-weight : Normal;
    font-size   : 10pt;
    border-style : none;
}

.helpnav              {
    font-family : geneva, arial, helvetica;
    font-weight : Normal;
    font-size   : 7pt;
    border-style : none;
}

.none                 {
    font-weight : Bold;
    font-family : geneva, arial, helvetica;
    font-size   : 12pt;
    color       : #000080;
    border-style : none;
    background-color : #FFFFFF;
}

.blue                 {
    background-color : #ADA5EF;
}
```

Appendix F: Component List

General Resource Files: /

application.css *
application.js *
processing.htm *
request_cancelled.htm *

Introduction Page: /

default.htm *

Cold Fusion Templates: /

application.cfm *
default.cfm *
footer.cfm *
header.cfm *
output.cfm *

Popup Help Files: /

help_coord.htm *
help_med.htm *
help_med_param_type.htm *
help_par.htm *
help_wsd.htm *
help_yea.htm *
help_yea_more.htm *

Routinely Re-generated Files to Populate Select Boxes: select/

date_static.cfm
media_static.cfm
param_type_static.cfm
watershed_number_static.cfm

Various image files: img/

blue_dot.gif
blu_dir.gif
dark_prog.gif
help.gif
help_blu.gif
lite_prog.gif
logo_crop.gif
processing.gif
processingb.gif
shim.gif
wris_logo40.gif