Tethys Web Client Manual

Tethys, Antioch mosaic, 3rd century from Baltimore Museum of Art

Table of Contents

1 Introduction

This document provides examples of common tasks using the web interface to Tethys. The Tethys Web Client is a web application that provides a more visual interface to a subset of the functionality of the MATLAB client. It provides maps, plots, query tools and advanced query tools. Time and ~~Spatial~~ spatial filters can be built into the queries. It also provides a set of tools for saving queries for later modification, refining simple queries, and exporting query results in MATLAB, XML, and R data formats.

2 The Simple Queries View/Tab

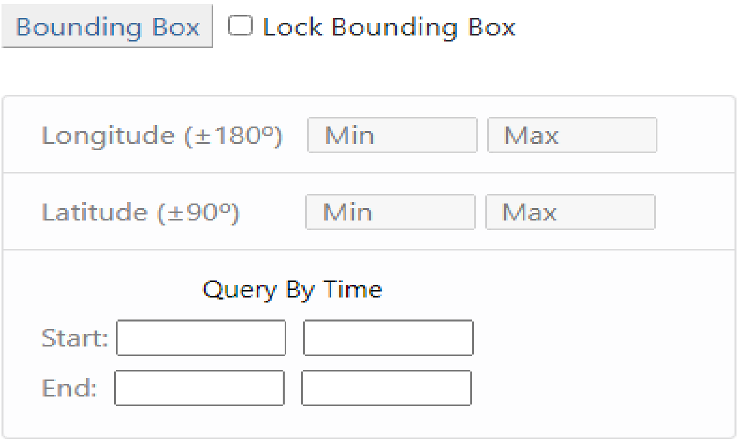
2.1 Orienting yourself to the controls

The web interface will automatically open to the 'Simple Queries' tab, these tab controls are toward the top of the web page under the Tethys banner. There are named from left to right as follows: Simple Queries, Advanced Queries, and Plots.

In the Simple Queries pane, all of the controls are aligned vertically alongside the left-hand side of the pane. These controls help to build and view the results of a simple query. Additionally there are controls for refining the query, saving the query results in different formats and loading queries that you have saved.

2.2 Bounding Box Control and Time Filter

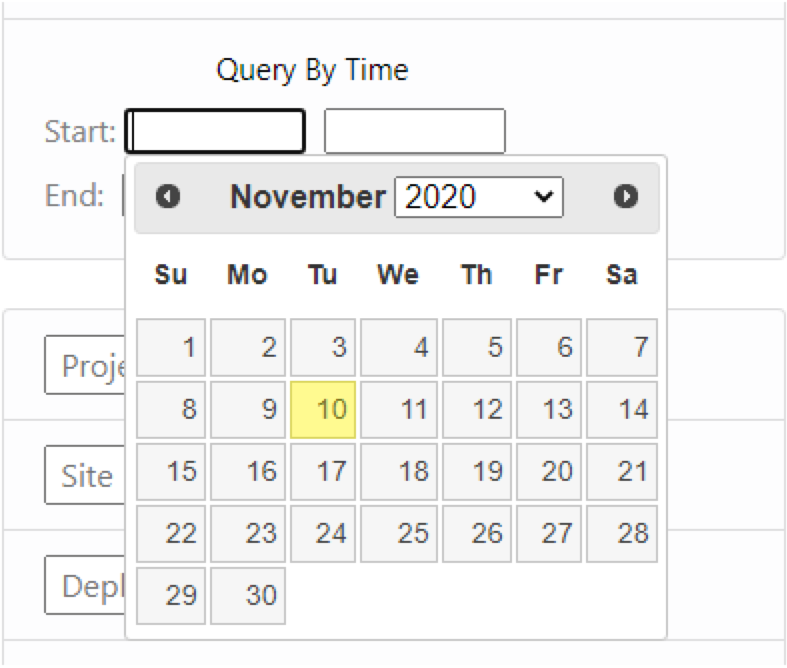
The first section contains an optional set of bounding box controls that will allow you to refine your query spatially. All queries can be filtered on a graphic bounding box on the map that you can control.



To activate the controls click the ‘Bounding Box’ button at the top. You will notice a transparent red-outlined square appear on the map. You can grab the box on the map and drag the entire box. You can also grab the corners or the sides to resize and shape the bounding box area. You will notice when you drag and reshape the box that the Longitude and Latitude coordinates will automatically update. By selecting the ‘Lock Bounding Box’ checkbox the bbox will lock in place while you scroll or zoom the map. You do not have to provide any Bounding Box information for a query, it is optional. To turn off the bounding box coordinate and remove them from you query you can simply click the ‘Bounding Box’ button and the coordinates will disappear.

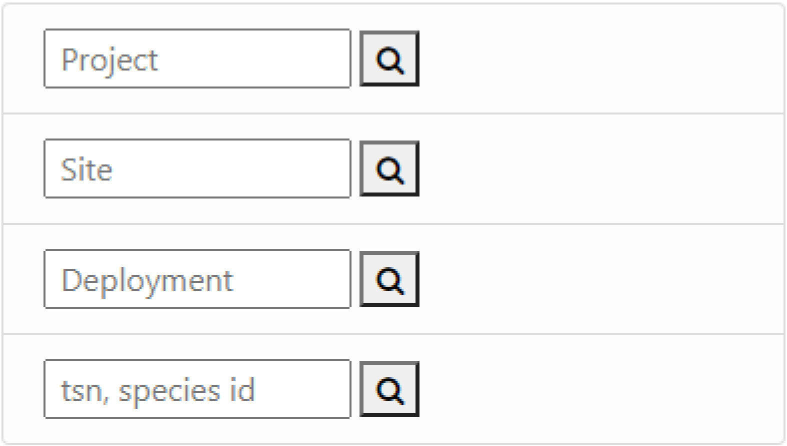
2.2.1 Time Controls

In addition to spatial filters each of the above four queries can be narrowed to include a time range for the data. The first row of the “Query By Time” section is the start time. By clicking on the box, a calendar date can be chosen for the Start date in the left hand column (shown below). A time component for the Start date can be chosen for the hour, minute and second in the right hand column. The bottom row contains the same controls but for the end date/time.

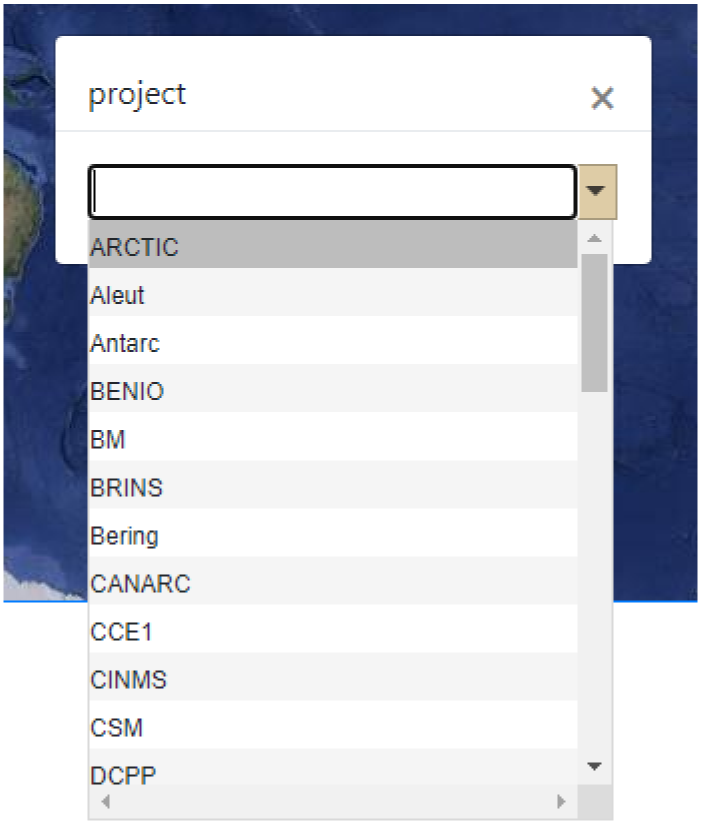


2.3 Species, Site, Project, Deployment Controls

A set of four controls appear directly below the Bounding Box and Time control.

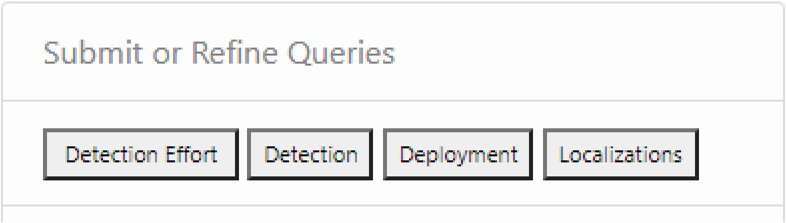


The first control allows you to enter a Project name. The second and third controls allow you to choose a Site name and Deployment ID. The last control allows you to enter a species Latin name, or species abbreviation. You will notice a small magnifying glass button to the right of each of the four controls, clicking on that button will present you with a list of possible values for that control. Each of the possible values is filtered depending on the other choices you have made in the other controls. A species library must be chosen when using the species parameter. By default, one will be chosen for you when you load the page. The project possible values is shown below.



2.4 Submit or Refine Queries

The next group of controls entitled ‘Submit or Refine Queries’ allows you to choose the type of simple query you wish to perform and optionally refine it in the advanced viewer. Other options include export/save results, save queries and load queries. To begin we will split up this into two sections easier to understand. Below is how the first section of those controls appear.



Four buttons submit specific pre-built queries, based on the parameters entered in section 2.3 and the spatial (described above) and/or temporal filters (described below). The queries are "Detection Effort", "Detection", "Deployment", and "Localizations"

The Detection Effort query, returns deployment information about where efforts were looking for a combination of parameters. For example, you can return the Site IDs within a Project where Risso's Dolphin (Grampus griseus) were searched. Alternatively, you could return all of the species searched for at a specific site or across an entire Project.

The Detection query, returns detection information and deployment information for actual detections given a set of parameters. For example, you can return detections for a specific Species, Site, Project, and Deployment combination. You may, for example, want to know all of the Blue Whale Detections for a Project, Site, and Deployment combination. Or you may simply want all the detections at a specific Site.

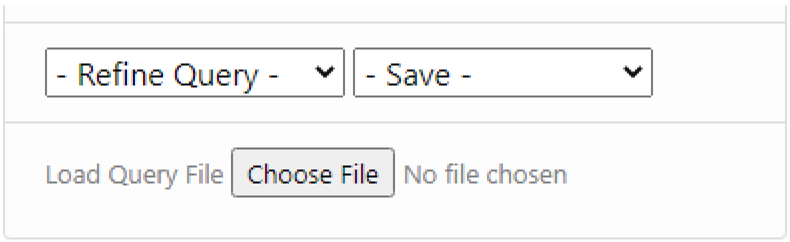
The Deployment query returns deployment information for a combination of the same parameters.

The Localization query returns information about tracks along a localization study. It returns the timestamps and lat/longs for specific waypoints along a track. It also returns the bounding box for the entire track.

2.5 Refine queries in advanced viewer, Save Results, Loading Queries

The next section of controls (see below) starts with a drop-down box for refining the Detection Effort, Detection, and Deployment pre-built queries in the advanced viewer. By selecting one of the three options, a new advanced view opens populated with your parameters; from here you can refine the query by adding or subtracting new parameters. This will be described in detail in section 3. where we discuss the advanced viewer.

For now, let us focus on saving the results of a query. The 'Save' drop-down has three options that can be used after you have gotten the results of the query. The first option, 'Save Query' saves the parameters of the query you performed, so that you can re-run your query in the future. This is particularly useful with the more complex queries built in section 3. When you save a query onto your file system you can get back to it, by opening it using the bottom button entitled 'Load Query File', this will reload your query. The next option in the drop-down is 'Save Results as Matlab File', choosing this option will export your results as a MATLAB data structure in the form of a .mat file that can be used directly in MATLAB. The next option in the drop-down is 'Save Result as XML'; this option allows you to export and save your results as XML. The last option 'Save Results as R' saves your results as a data-tree R object that can be dragged and dropped into a R console. Exploring the results in this R format will be discussed in section 2.6



2.6 Viewing Results in an R session.

To view your results in the R data.tree format your Tethys server that the web client is using will have to be configured to produce R results. Details about how to configure your Tethys server to do this are in the Tethys Manual.

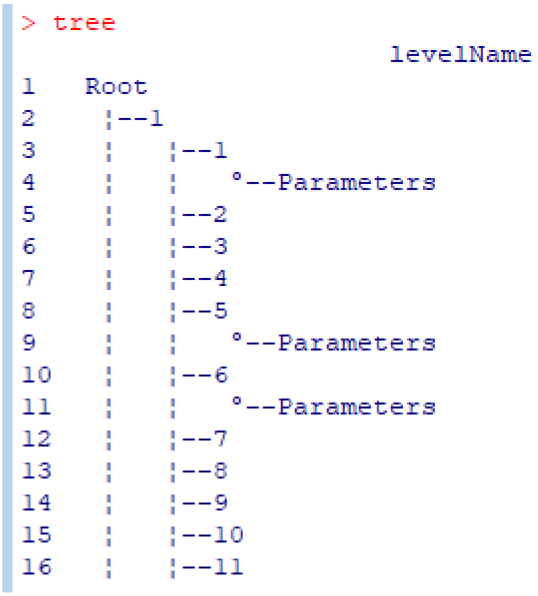
If you have a server that produces R results choosing the R save option in the web client from the above drop down will download an R result. You will need R 4.x for your desktop/laptop installed to view the results. The following link has directions and the download for installing R

[Download R-4.0.3 for Windows. The R-project for statistical computing. (r-project.org)]

When you have installed R you will need to install an R package and load it. From the R command line type: install.packages('data.tree'). Note you will only have to install the data.tree library once.

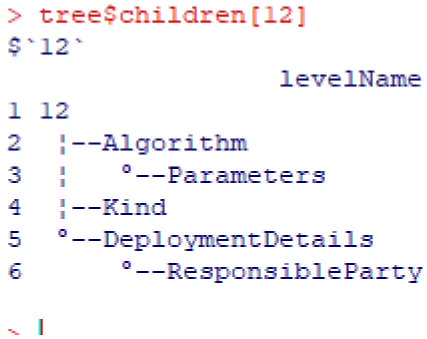
Before you can load the results into your R session, type: library('data.tree') at the R command line.

You are now ready to just drag and drop the R results file into your R session. Dragging and dropping the file into R produces a root node call 'tree'. By typing the name 'tree' into your R command prompt you will see a graphical representation of the tree structure. Like the following:



```
> tree
                         levelName
1   Root
2    ¦--1
3    ¦   ¦--1
4    ¦   ¦   °--Parameters
5    ¦   ¦--2
6    ¦   ¦--3
7    ¦   ¦--4
8    ¦   ¦--5
9    ¦   ¦   °--Parameters
10   ¦   ¦--6
11   ¦   ¦   °--Parameters
12   ¦   ¦--7
13   ¦   ¦--8
14   ¦   ¦--9
15   ¦   ¦--10
16   ¦   ¦--11
```

There are different methods for navigating and pulling data from the tree. Just to name a few you can pull an indexed record from the data by typing: tree$children[12]. This will pull the 12th record and in the case of DetectionEffort would produce the following graphic.



```
> tree$children[12]
$`12`
                   levelName
1 12
2  ¦--Algorithm
3  ¦   °--Parameters
4  ¦--Kind
5  °--DeploymentDetails
6      °--ResponsibleParty

> |
```

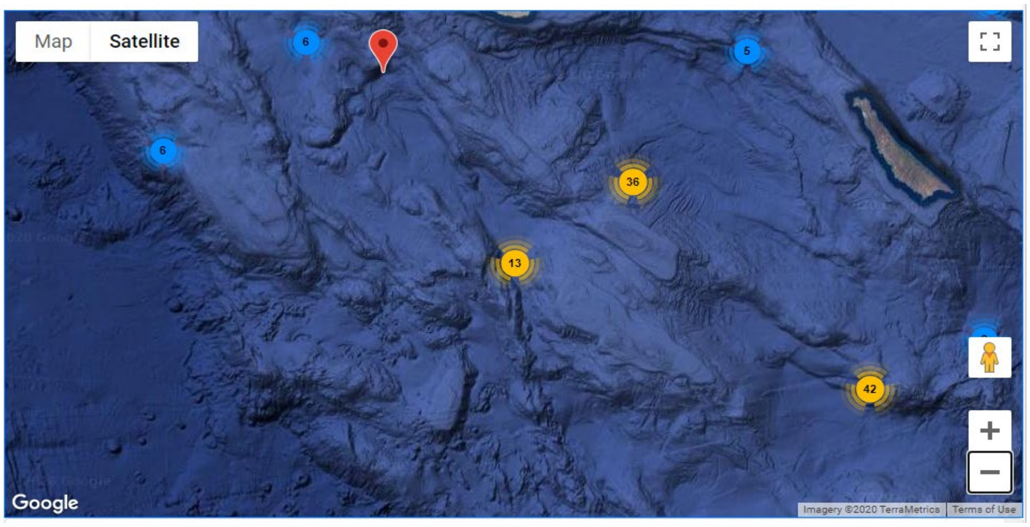
A parallel method exists, where you can also simply type: tree$'12'. Building on that to get the Algorithm from the 12th record, type: tree$'12'$Algorithm. These are just the basics for a complete description of converting, reading, editing the tree please see the official link below:

Introduction to data.tree (r-project.org)

2.7 Viewing the Results of Simple Queries.

To the right of the inputs for building the query there is a map, and below it an empty pane, these are the areas where your results will appear. Once you have set your parameters and optionally set filters, clicking the query button will submit the query and a progress indicator will appear to let you know when the query is finished. When the query is finished, the results will be sent to the results sections of the page (map and XML section). If the pre-built query returns latitude and latitudes (every query other than Deployment), the results will be mapped. For all queries, results under 200 MB in size results will also be returned as XML in the pane below the map. If your results are larger than the limit they can be downloaded through the 'Save' drop-down.

In the mapped results, individual data points are represented as red balloons. Hovering over the red markers will reveal information about that data point. Yellow or blue circular markers represent clusters of data points with a number that denotes how many data points are clustered there. Clicking on the circular markers will expand to show the individual data points in the cluster.



3 Advanced Queries

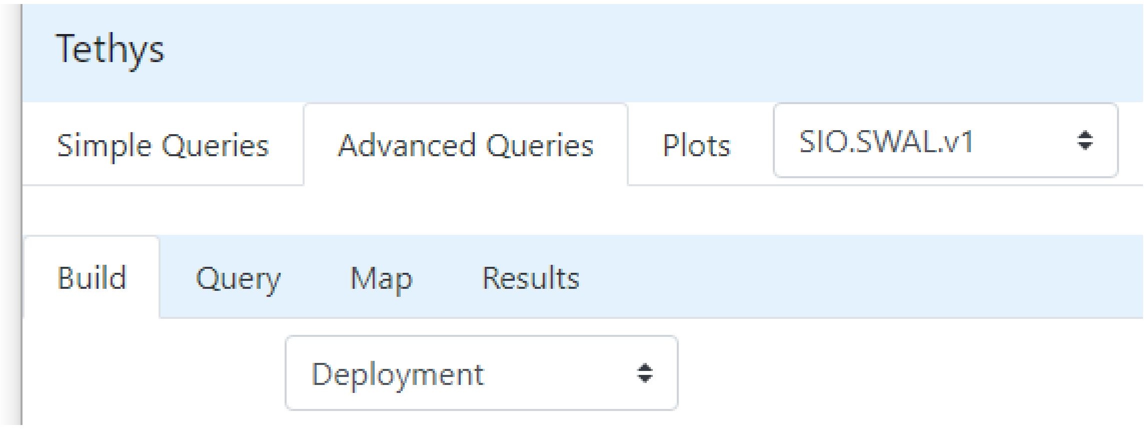
There are two ways to activate an advanced query. You can go there directly by using the 'Advanced Queries' tab at the top of the page or you can refine a simple query by using the 'Refine Query' drop-down. Opening an advanced query from its tab will open a new blank query while using the refine option will populate the advanced query with the parameters of a simple query.

3.1 Controls for the Advanced Queries View

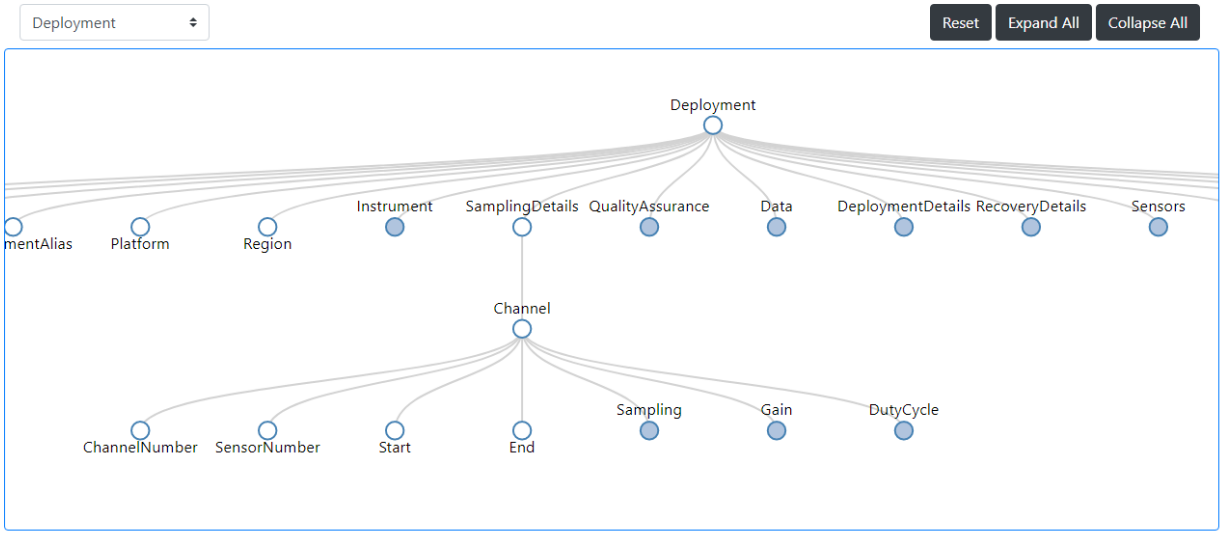
3.1.1 The Build Tab

Shown below is the top left of the page that opens when the Advanced Queries tab is selected either by the user directly or through the refinement tool. Below the main tabs is a second tier of tabs, titled 'Build','Query','Map','Results'. These tabs give access to different views within the Advanced View itself. The image shown below shows the Advanced View opened in the Build

Tab, this is the default tab when the Advanced View opens. The Build tab is where you will build the actual query.

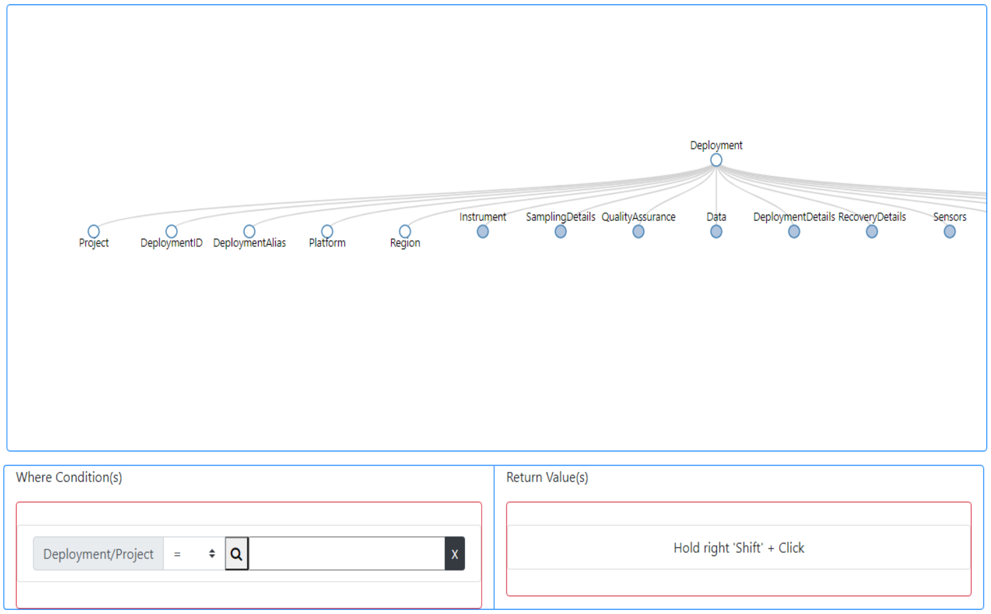


The Build tab contains a drop-down at the top that allows you to choose the collection from which to choose parameters. For example, you may want Deployment Details from the Deployment collection, and you may want to select a species ID from the OnEffort/Detection section of the Detections collection. Choosing a collection from the drop-down will give you a tree that depicts the collection. Shown below is the tree view of the Deployment collection with the Sampling Details node expanded.



Notice that the tree spreads off the page to the left and the right; this is because this view has been zoomed to show you the nodes of the tree. Navigating the tree is as simple as just dragging it and using your mouse wheel to zoom in or out. The blue nodes in the tree are expandable. By clicking a blue node, it will automatically expand to reveal more nodes. Each node is a variable recorded in the data for all of the deployment documents in a collection.

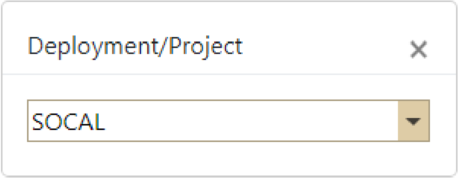
Below the tree view there is a single row, containing two columns. The left column are Where Conditions for your query and the right column are the Return Values you would like returned in the query results. To populate these columns, hold the left Shift key down and click on a node that you would like to include in the Where clause. Let us say that you want to query on the Project name. The Project node is on the far left of the tree, hold the left Shift key and click on the Project node, a Where condition will appear in the row below the tree as shown below.



Deployment/Project =

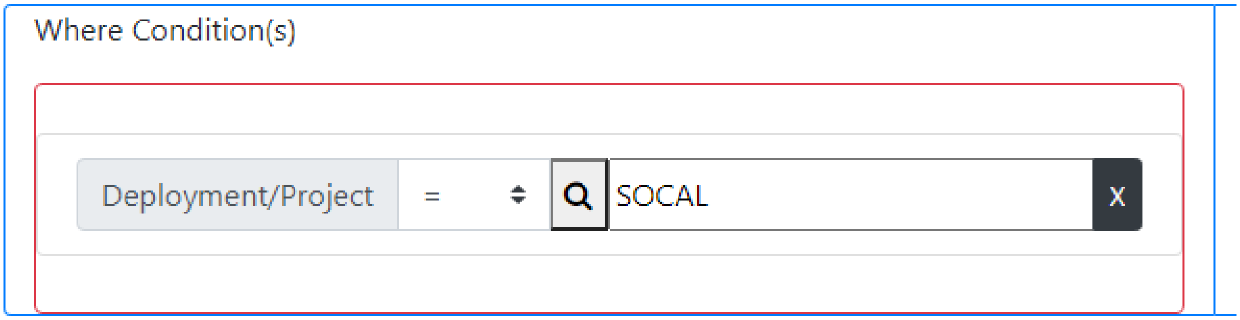
Hold right 'Shift' + Click

Notice that the node name will appear on the left followed by an operator drop-down. Click on the drop-down arrows and other operators will appear from which you can choose. The default is the equals' operator. The next button is a magnifying glass; by clicking on it you will get a drop-down with all of the allowable values for this variable. Below are images of the values drop-down, and the Where Condition populated with the value chosen using the magnifying glass drop-down. By clicking on other nodes, other conditions can be added as new rows.



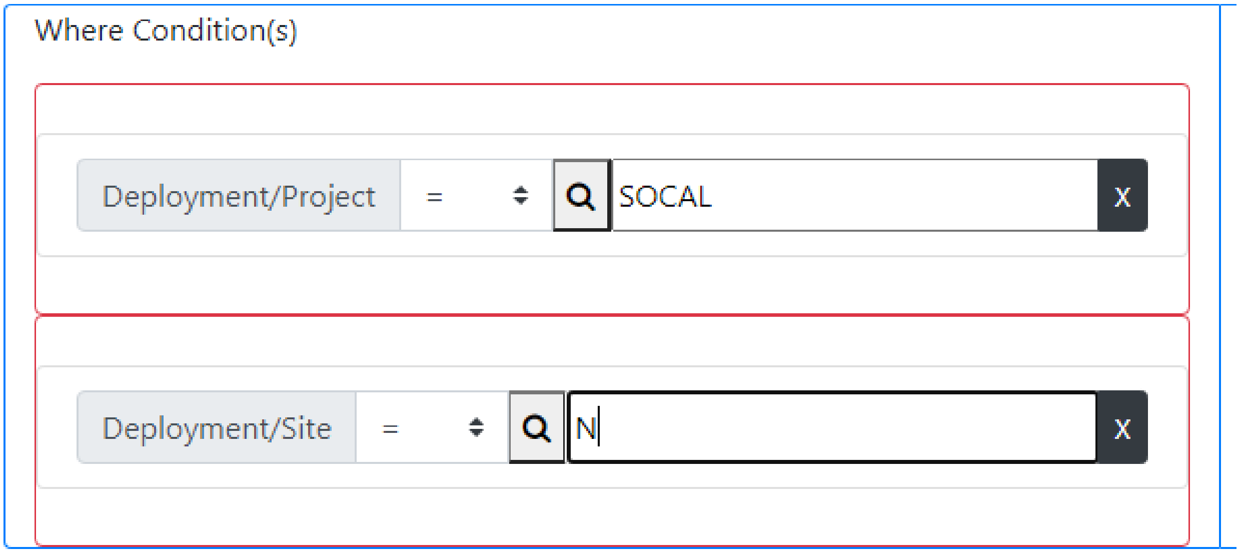
Deployment/Project

SOCAL


Where Condition(s)

Deployment/Project = SOCAL

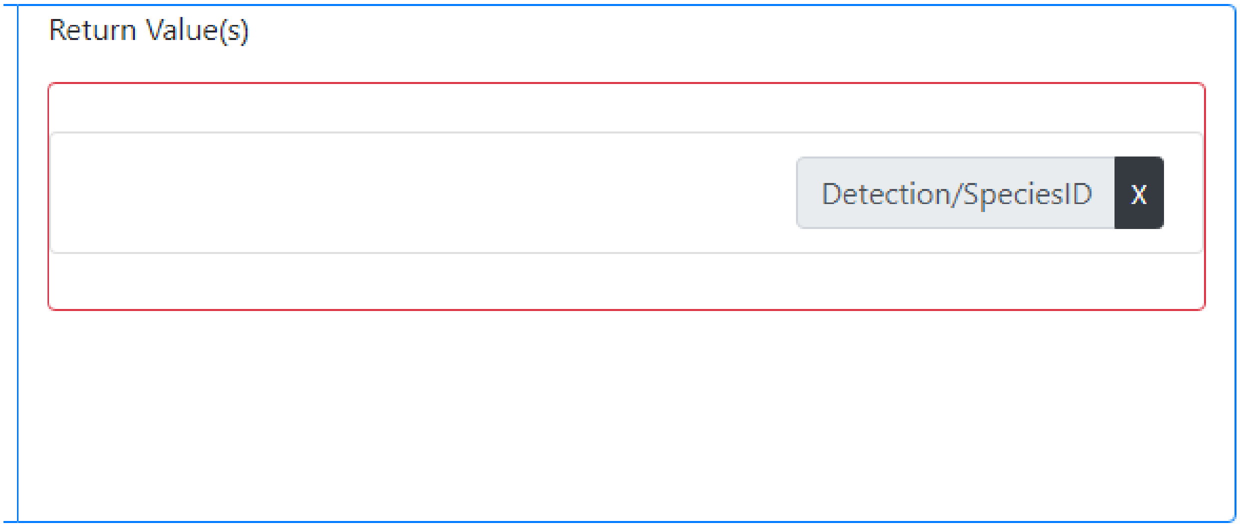
Shown below is a new row with an additional condition added.


Where Condition(s)

Deployment/Project = SOCAL

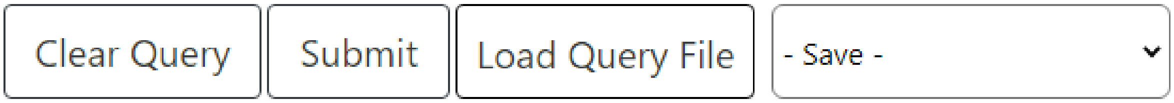
Deployment/Site = N

Adding Return Values follows the same procedure but when adding Return Values hold the right 'Shift' key while clicking on a node. Let us say that we would like to return some information from the Detections collection, below is what that would look like.



Notice that we have decided to return the Species ID from the OnEffort/Detection node for the conditions defined in the Where Conditions column. So we return all species detected at site N in the SOCAL project. Again adding more return values by clicking on nodes will add new rows to the 'Return Value(s)' column. Also, notice that there are no values to be set on Return Values.

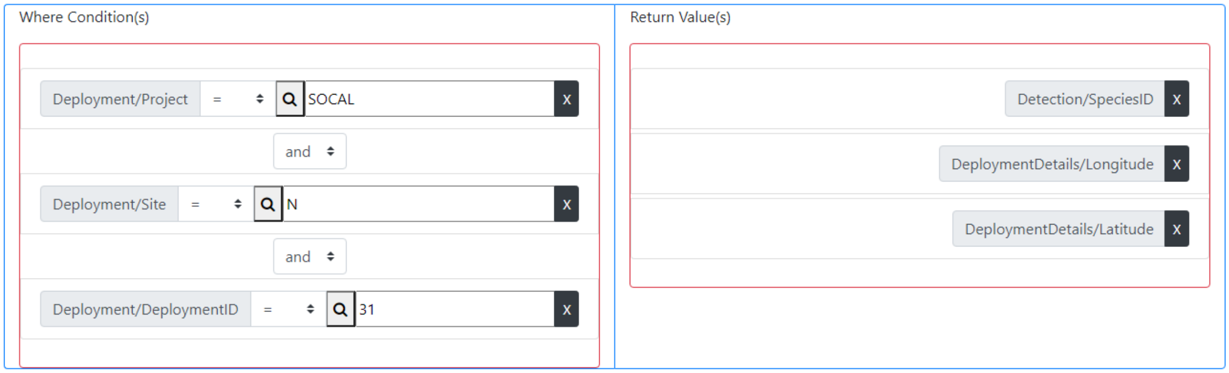
At the bottom of the page, you will see the following buttons.



Once you have finished building your query, you can click on 'Submit' to retrieve your results. Notice that you can clear a query, submit a query, or build on an older query you have saved, and load those parameters into a new query by clicking on 'Load Query File'. You can also use the 'Save' drop-down to save your results in the same formats as the 'Save' drop-down described for the Simple Queries.

3.1.2 Viewing Your Results

Let say that we have the following query.



Where Condition(s)

Deployment/Project = SOCAL X

and

Deployment/Site = N X

and

Deployment/DeploymentID = 31 X

Return Value(s)

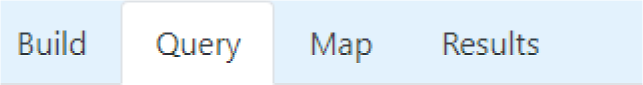
Detection/SpeciesID X

DeploymentDetails/Longitude X

DeploymentDetails/Latitude X

It is essentially the same query as above, but with an extra Where condition, for the Deployment ID and two extra Return values, the latitude and longitude for the detections. Upon clicking the ‘Submit’ button, a progress indicator will appear while the query is being processed. When the query finishes the results are sent back to the web client for viewing, and downloading/exporting.

Next to the Build tab there are three more tabs named, Query, Map, and Results.

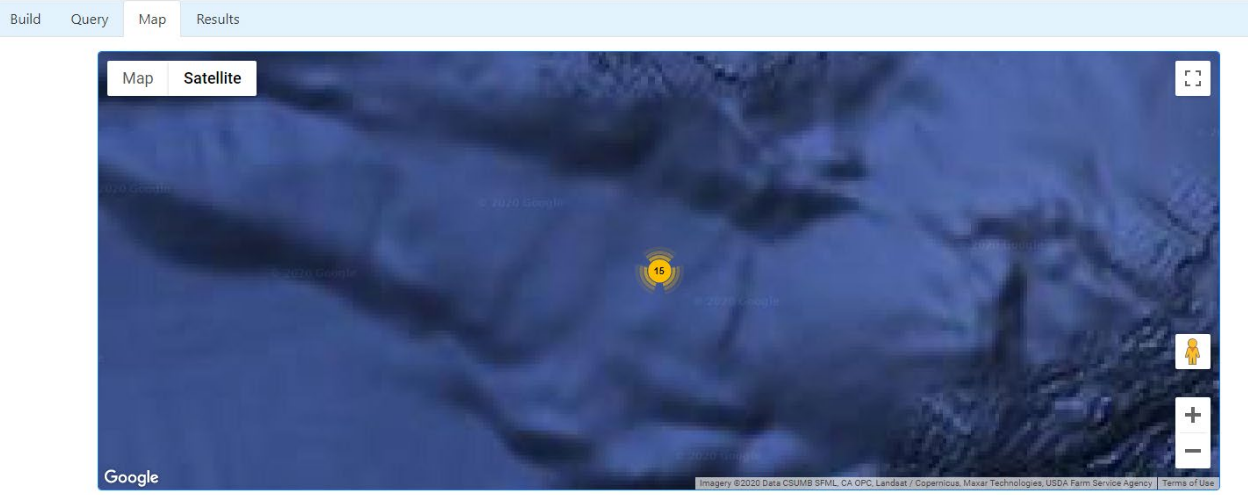


The Query tab shows the structure of the query as dbXML. The database uses this query language. Viewing the query can sometimes give you a sanity check when you receive results that you did not expect, or as a last minute check before submitting your query. The following is the dbXML query for the query parameters shown above.

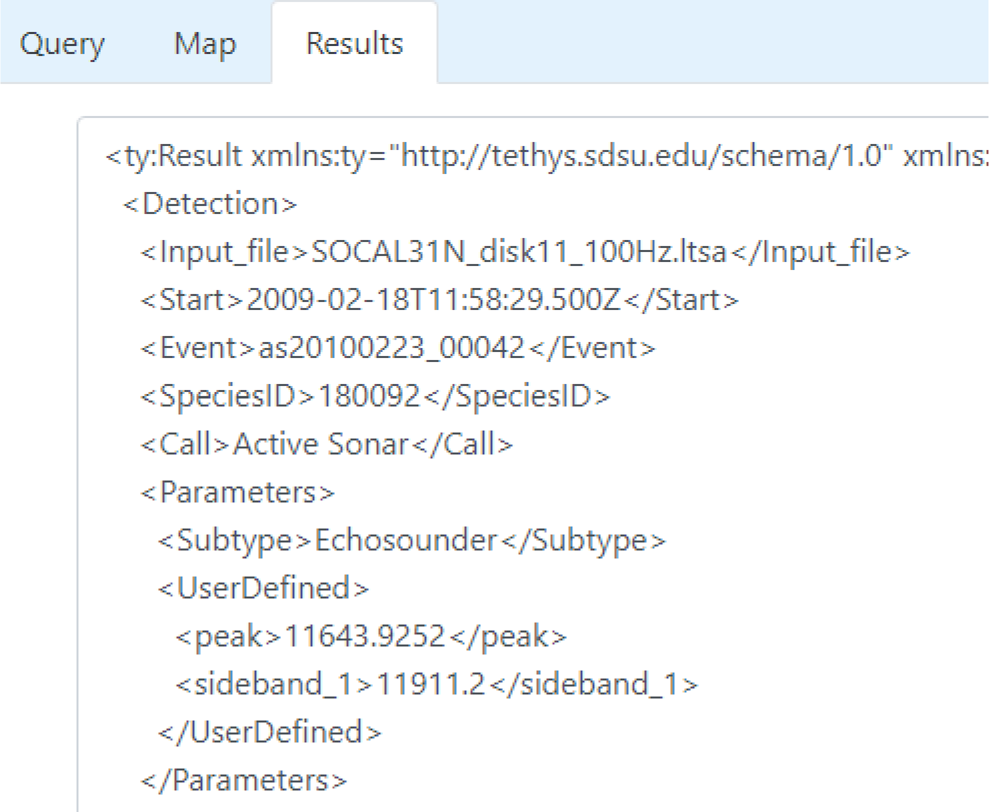
```
import schema namespace ty="http://tethys.sdsu.edu/schema/1.0" at "tethys.xsd";
import module namespace lib="http://tethys.sdsu.edu/XQueryFns" at "Tethys.xq";
<ty:Result XMLns:xsi="http://www.w3.org/2001/XMLSchema-instance">{
   for $Detections1 in collection('Detections')/ty:Detections  ,
   $Deployment1 in collection('Deployments')/ty:Deployment
   where $Deployment1/Project = "SOCAL" and
   $Deployment1/Site = "N" and
   $Deployment1/DeploymentID = 31  and
   $Deployment1/Project = $Detections1/DataSource/Project and
   $Deployment1/Site = $Detections1/DataSource/Site and
   number($Deployment1/DeploymentID) = number($Detections1/DataSource/Deployment)
   return <out>{
      for $Detections2 in $Detections1/OnEffort/Detection
      return lib:tsn2abbrev( $Detections2/SpeciesID , "SIO.SWAL.v1")
      , $Deployment1/DeploymentDetails/Longitude,
      $Deployment1/DeploymentDetails/Latitude
   }</out>
}
```

</ty:Result>

The next tab is ‘Map’. If you include latitudes and longitudes in your Return Values, you will be able to view the coordinates of your results in the map as shown below.


Build
Query
Map
Results
Map
Satellite
15
Google
Imagery ©2020 Data CSUMB SFML, CA OPC, Landsat / Copernicus, Maxar Technologies, USDA Farm Service Agency
Terms of Use

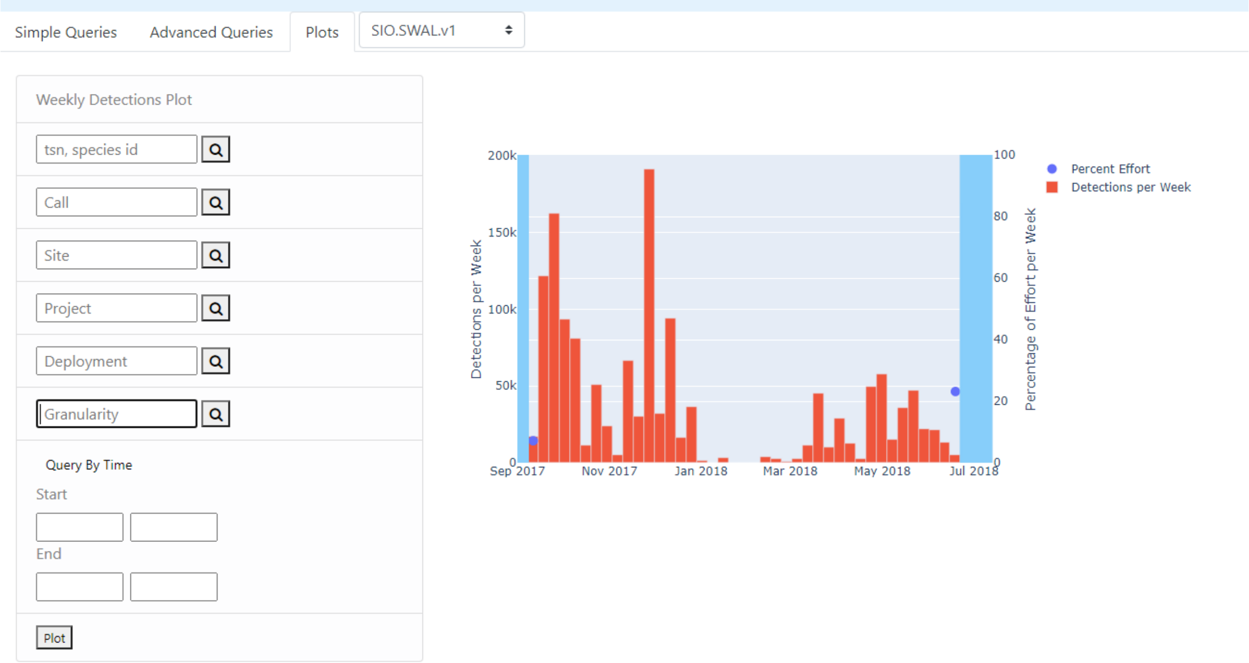
The last tab is the Results tab, your results will be shown here as XML if your result set is smaller than 200 MB. If it is larger you can easily download the results as XML from the ‘Save’ drop-down.



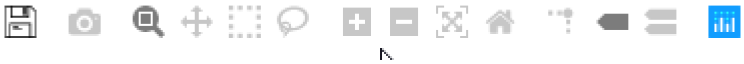
```
<ty:Result xmlns:ty="http://tethys.sdsu.edu/schema/1.0" xmlns:
 <Detection>
  <Input_file>SOCAL31N_disk11_100Hz.ltsa</Input_file>
  <Start>2009-02-18T11:58:29.500Z</Start>
  <Event>as20100223_00042</Event>
  <SpeciesID>180092</SpeciesID>
  <Call>Active Sonar</Call>
  <Parameters>
   <Subtype>Echosounder</Subtype>
   <UserDefined>
    <peak>11643.9252</peak>
    <sideband_1>11911.2</sideband_1>
   </UserDefined>
  </Parameters>
```

4 Plots Tab

The last tab at the top level is ‘Plots’. The Plots tab contains a utility for generating weekly effort plots. These plots graph the weekly percent effort and detections per week for a deployment. As with all parameters in the web client, a magnifying glass search tool will allow you to search for available values. The query can also be filtered on time just like all of the queries in the client. When you fill out your values, the ‘Plot’ button at the bottom will send your plot request to the database. When the plot is generated, it will appear in the pane to the right of the controls.



When you use your mouse to hover above and to the right of the plot, you will see a see a set of controls for the plot as shown below.



Hovering over each tool will show you what it does. There are tools for saving the plot as both SVG and a PNG image. The other tools will help to highlight data values and navigate the plot.

5 Conclusion

We hope that this overview has helped with the major tools available in the Tethys Web Client. For further help or comments on features, you can contact the developers directly at jeff.cavner@gmail.com.