To add a CLAMS Protocol

In CLAMS, protocols are a group of measurements that are collected for a given species. A protocol includes both the measurements and also the measurement order and helps the scientists remember to collect the data they are supposed to collect for various species. Before a survey, scientists will determine which data they want to collect for which species and configure the protocols in the database accordingly. This document will help you define these protocols.

This can be an involved process and this document currently does not go into full detail for every step.

Specimen_protocol

The specimen_protocol table contains the name and description of the protocol. The first step is to add your new protocol to this table. The names must be unique and once data is collected using a specific protocol, you cannot rename or remove the protocol name from this table. So choose wisely.

Measurement_types

The measurement_types table contains all of the measurements defined in CLAMS. You must verify that the measurements you want to include in your protocol exist in this table. To add a measurement type, you must create a unique name, specify the measurement data type, the measurement units, and provide a description.

If you create a new measurement type, you will need to add it to the V_SPECIMEN_MEASUREMENTS view. Instructions on how to do this are outside the scope of this document at this time, except to remember that you will need to add your new measurement in 3 places in the SQL code:

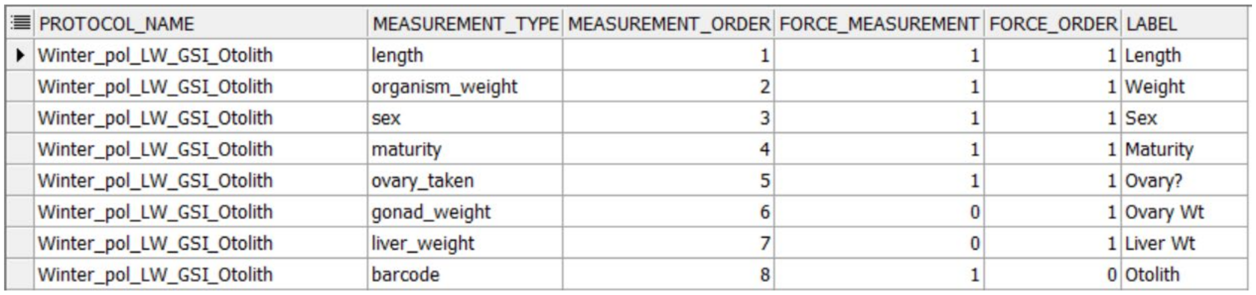
1. In top line 'Create or replace Force Editionable View' clause
2. In Select' clause
3. In 'Join' clause

It is best to copy the existing V_SPECIMEN_MEASUREMENTS code into a text editor, follow the existing format to add your new measurement, and then re-create V_SPECIMEN_MEASUREMENTS using your new SQL. Save a copy of the SQL before modification in case it all goes wrong.

Protocol_definitions

The protocol_definitions table is where you link your measurements with your new protocol. You also define the measurement order, whether the individual measurements are required, if any measurements can be out of order, and the text that will appear on the button for the measurement in CLAMS.

In the image below, you see the definition for the "Winter_pol_LW_GSI_Otolith" protocol.



PROTOCOL_NAME	MEASUREMENT_TYPE	MEASUREMENT_ORDER	FORCE_MEASUREMENT	FORCE_ORDER	LABEL
Winter_pol_LW_GSI_Otolith	length	1	1	1	Length
Winter_pol_LW_GSI_Otolith	organism_weight	2	1	1	Weight
Winter_pol_LW_GSI_Otolith	sex	3	1	1	Sex
Winter_pol_LW_GSI_Otolith	maturity	4	1	1	Maturity
Winter_pol_LW_GSI_Otolith	ovary_taken	5	1	1	Ovary?
Winter_pol_LW_GSI_Otolith	gonad_weight	6	0	1	Ovary Wt
Winter_pol_LW_GSI_Otolith	liver_weight	7	0	1	Liver Wt
Winter_pol_LW_GSI_Otolith	barcode	8	1	0	Otolith

This protocol has 8 measurements: length, organism_weight, sex. maturity, ovary_taken, gonad_weight, liver_weight, and (otolith vial) barcode. The measurements must be defined in the measurement_types table before being added here.

Measurement order starts at 1 and increments by 1 for each measurement. The measurements will be presented on-screen in the order defined here.

Force_measurement should be set to “1” if the measurement must be taken and “0” if it is optional. Measurements that have conditionals attached to them that may disable the measurement must be defined as optional.

Force_order should be set to “1” to force the measurement to be taken in the order defined in the measurement_order column and “0” if the measurement can be taken out of order. Note that measurements that are taken using the same device must be done in order so CLAMS knows which measurement the value sent from the device corresponds to. There may be other reasons to enforce order. For example, you shouldn’t weigh a specimen after cutting it open to check sex and maturity because bits can fall out affecting the weight. In the example above, the barcode measurement is allowed to be out of order because in this protocol, the bar code scanner only provides the otolith vial number so we always know what measurement the value coming from the scanner is associated with. This allows a second biologist who is cleaning the otoliths and placing them in the vials to pick up a vial, scan it, and open it while the first biologist is working through the other measurements.

Label should be set to a string that will be on the measurement button in CLAMS. This is what the user sees when collecting data.

Protocol_map

Protocol_map connects protocols with a species+subcategory. This is where you specify which protocols are available for individual species codes and subcategories. Set the active column to “1” to enable the protocol (meaning it will be available in the specimen module if the user collects a sample of type “measure” in the catch module) or “0” to disable it. We have a number of protocols that are either for different survey areas or survey seasons and we enable/disable protocols depending on when/where we are surveying.

Measurement_setup

Measurement setup connects a workstation, measurement_type, and device (the source of the measurement) for the individual CLAMS modules. This is how CLAMS knows which device will provide the measurement data for a specific measurement. If you added a new measurement for your new protocol, you must create the appropriate entries here, otherwise CLAMS doesn't know how to collect data for that measurement.

In order to add an entry in the measurement_setup table you need to know the workstation ID of the sampling station you are configuring for the measurement, the measurement type you are adding, the device ID of the device providing the measurement, the device's interface, and the CLAMS module the measurement applies to.

Workstation ID, measurement type and device ID should be self explanatory. The device interface informs CLAMS on how to "connect" to the device. There are currently 3 different device interfaces.

1. **Serial** are devices that provide data via a computer's serial port. These devices will be either connected directly to the computer or through the network but the key point is that they will be accessed via a windows "COM" port or Linux TTY. For serial devices, CLAMS will query the device_configuration table and extract various parameters so it can open the correct COM port and parse the incoming data.
2. **SCS** devices are individual "sensors" that are served up by the OMAO SCS system. They can also be served up by the CLAMS SensorLogger application. For SCS devices, CLAMS will connect to the SCS server (IP and port defined in the application_configuration table) and poll the server for the sensor data.
3. **Software** devices are individual dialog boxes that collect data from the user. An example is a dialog that asks if the user is taking a stomach sample or not. It returns "Yes" or "No" as a measurement.

Lastly, the GUI_Module field specifies the CLAMS module that your new measurement setup applies to.

- **TrawlEvent** means the measurement will be used by any of the event modules. Usually the TrawlEvent module only has SCS or Software devices associated with it.
- **Catch** means the entry will be used by the Catch module. The Catch module only has a basket scale configured in measurement setup.
- **Length** means the measurement will be used by the Length module. The Length module only has a length device and a Software sex measurement associated with it.
- **Specimen** means the entry will be used by the specimen module. This is most likely the module you will specify since configurable protocols are only used by the Specimen module.

Validations

The validations table links a validation with a protocol and a measurement. Validations check that specific measurements are sane. If you want your protocol to run validations on specific measurements, you must enter them here.

The validation column contains the name of the Python module that was created to execute the validation. It is case sensitive and does not include the ".py" extension. Both the file name and Python class name must match this entry. The protocol name will be the name of your new protocol. The measurement type will be the measurement that this validation applies. Lastly, the validation order is used when multiple validations apply to a single measurement. If only a single validation is connected to a measurement for a protocol you enter 1. If multiple validations are applied, you set the validation you want to run first as 1, then the next as 2, and so on.

If you add a validation, you must add it to the \validations__init__.py file!

Information on how to write a new validation is currently outside the scope of this document.

Conditionals

The conditionals table links a conditional with a protocol. Conditionals are run after any measurement is taken and can alter the protocol based on the data collected up to that point. If you want your protocol to run conditionals on measurements, you must enter them here.

Just like in the validations table, the conditional column contains the name of the Python module that was created to execute the validation. It is case sensitive and does not include the ".py" extension. Both the file name and Python class name must match this entry. The protocol name will be the name of your new protocol.

If you add a validation, you must add it to the \conditionals__init__.py file!

Information on how to write a new conditional is currently outside the scope of this document.