ESP

MONTEREY BAY AQUARIUM RESEARCH INSTITUTE

Environmental Sample Processor: Protocol Parameters

5/25/22 Brent Roman   brent@mbari.org

M B A R I

Predefined Protocol Methods

- hab, larv, bac, pufm, kar, da, stx, and standAloneWCR
  - All take two optional parameters, plus
  - an optional code block for non-standard clean up

  - Ruby requires all default parameters to follow explicit ones
    - This means that one must specify the sample volume if options are specified

- Most protocols are configured in the Ruby script file: `mission/protocfg.rb`

Specifying Sample Volumes

- First protocol method parameter, if specified, overrides default sample volumes
  - A single non-negative Float value overrides only sh1 volume
  - Last element of an array of 2 non-negative values overrides WCR volume
  - 1st element overrides sh1 volume as in the single volume case
  - Last element of nil causes WCR puck to be skipped
  - First element of nil skips sh1 sampling (still loads sh1 puck)
  - Sh1 volume must always be numeric. Sh1 cannot be skipped.
  - Examples:
    - hab 123 #123ml sh1 with default WCR
    - larv [123, nil] #123ml sh1 without WCR puck
    - stx [123, 45] #123ml sh1 with 45ml WCR
    - bac 123, 45 # INVALID – 45 is not an option hash
    - bac 123, nil # INVALID – nil is not an option Hash
    - da #default da sample and WCR volumes
    - hab nil #negative lysate test

Default Sample Volumes

- Each protocol has a corresponding *$protocolVol* global variable
- *$protocolVol* is the default value of the protocol's first parameter
  - Examples:
    - $habVol=123 #default 123ml default HAB sh1 with default WCR
    - $habVol=[123, nil] #123ml default HAB sh1 skip WCR pucks
    - $larvVol=[123, 45] #123ml default LARV sh1 with 45ml WCR
    - $larvVol=123, 45 # VALID – same as [123,45] above
    - $habVol=123, nil # VALID – default to skip HAB WCR pucks

- If *$protocolVol* is a single numeric value,
  default WCR volume is determined by protocol's :wcrVol option
    - Discussed in upcoming “Protocol Options” slide
- If *$protocolVol* is a two element array,
  - last element determines the WCR volume,
    overriding :wcrVol option

Protocol Options

- 2nd protocol method parameter, if specified, overrides default *$protocol* options
- Each option is identified by a symbolic name – its hash key
- Ruby hashes merely map arbitrary keys to corresponding values
  - Hash Examples:
    - { :a => 4, :b => 3.14, :foo => “foobar” }
    - {0 => 1, 1 => 0}
  - Note that keys need not be Symbols, but they usually are.
  - Curly braces must be omitted when passing Hash literal to methods
    - Otherwise, it would look like passing in a block of code!!
      - p {:a => 4, :b => 3.14, :foo => “foobar” }   #INVALID

        vs.
      - *p :a => 4, :b => 3.14, :foo => “foobar”*

        alternatively
      - *p a: 4, b: 3.14, foo: “foobar”*
  - Undefined hash keys return nil
    - { :a => 4}[:b] == nil

Default Protocol Options

- Each protocol has a corresponding *$protocol* global variable
- *$protocol* is the default value of the protocol's second parameter
- It will always be a Hash mapping option names to values

- To change a default protocol option for all subsequent invocations:
  $protocol[:option] = newValue

- To override default protocol option(s) for only one invocation:
  protocol [sh1Vol, wcrVol],
  $protocol.with *:option=>newValue*

- First parameter must be specified in order to specify second one.

Basic Protocol Options

- :assay => assay name (:hab, :larv, etc.)
  Identifies the assay in the log and image file names
- :sampler => Sampler object to use (usually *protocol*Sampler)
- :image => array of camera exposures
  as in: [5, 10, 40, :hires,160, :midres, :auto]
- :email => Email topic list specifying who should receive mail
  - eg: HABfans
- :fast => true or false #enables faster cleanup for lab use

Whole Cell Archive Options

- :wcrVol => default WCR sample volume or nil to skip WCR
- :wcrFixes => default list of wcr fixes
  as in: [:fix]*3 #3 fixes using the usual fixative
- :wcrTime => total WCR fix time
  as in: '60:00' #each fix's duration is divided by :wcrFixes.size
- :wcrSampler => sampler to use for WCR water sample

Assay Processing Options

- :hybTemp => sandwich hybridization temperature
- :diluentVol => volume of diluent to add

  if < 0, the specified -volume is added before lysis (pre-diluent)
  if > 0, the specified volume is added after lysis (post-diluent)
- :remix => true or false (defaults to false)
  - enable remix for case where diluentVol < 0
- :lyseHeat => target lysis temperature
- :lyseTime => lysis duration
- :blockDelay => time to delay for sh2 blocking

  as in: `8.minutes #8 minutes by default`

Less Common Protocol Options

- :samplePass => procedure (code) to use for passing sample to PS
- :delay => length of delay to plumb sample for short assays
- :flush => flush procedure

Example: HAB Protocol Options

```
-> $hab
{assay => hab,
 email => HABfans,
 image => [hires, 160, default, 40, midres, auto],
 sampler => HABsampler,
 wcrSampler => HABWCRsampler,
 wcrVol => 50}

-> $hab[:wcrVol] = 1500
1500
-> $hab
{assay => hab,
 email => HABfans,
 image => [hires, 160, default, 40, midres, auto],
 sampler => HABsampler,
 wcrSampler => HABWCRsampler,
 wcrVol => 1500}
```

standAloneWCR Protocol Options

- There is no ~~$standAloneWCR~~ global variable
  - Because WCR is normally incorporated into other assays
- The WCR sample volume must be the 2nd specified.
- Best to define myWCR for your specific needs:

```
def myWCR  vol=$myWCRvol, opts=$myWCRopts, &cleanup
  standAloneWCR([nil, $vol], opts, &cleanup)
end
```

- Setup:

```
$myWCRvol = 100   #default sample volume
$myWCRopts = $bac.with wcrTime: “5:00”
```

- Usage:

```
myWCR           #defaults to 100ml
myWCR  150          #sample 150ml
```

HABDA composite Protocol

- A single protocol combining HAB and DA assays
  - Harmful Algal Bloom
    - Including optional Whole Cell Archive
  - Demonic Acid

- Uses 6 or 7 pucks in all

- Parameters are:
  - HAB sample and WCR volumes
  - DA sample volume
  - Combined HAB and DA options
    - As the last Hash parameter