

Environmental Sample Processor: Core Protocol and Mission Parameters

5/12/10 Brent Roman brent@mbari.org



Core Protocol Methods

- HAB, LARV, BAC, PUFM, KAR, and even DA
 - 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
 - We'll come back to this.



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 (not puck)
- Sh1 volume must always be numeric. Sh1 cannot be skipped.
- Examples:
 - *hab 123 #123ml sh1* with default WCR
 - *hab [123, nil] #123ml sh1* without WCR puck
 - *hab [123, 45] #123ml sh1* with 45ml WCR
 - *hab 123, 45 # INVALID* – 45 is not an option hash
 - *hab 123, nil # INVALID* – nil is not an option Hash
 - *hab #default hab 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 skipping 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 skipping 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 miscellaneous 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" }`
 - 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`



Protocol Options List

- :assay => assay name (:hab, :larv, etc.)
 - Identifies the assay in the log and image file names
- :sampler => Sampler object to use (usually \$protocol/Sampler)
- :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
- :image => array of camera exposures
 - As in: [5, 10, 40, :hires, 160, :midres, :auto]
- :email => Email object specifying who should receive mail
 - As in: *Email.new ['brent', 'esp001'], :Subject=>"Progress"*
- :samplePass => procedure (code) to use for passing sample to PS
- :blockDelay => time to delay for sh2 blocking
 - As in: '8:00' #8 minutes by default
- :delay => length of delay to plumb sample for short assays
- :fast => true #enables faster cleanup logic for lab use
- :flush => flush procedure



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
- Change a default protocol option for all subsequent invocations:
\$protocol[:option] = newValue
- Override default protocol option(s) for only one invocation:
protocol [sh1Vol, wcrVol], \$protocol.dup.update :option=>newValue
- First parameter must be specified in order to specify second one.
- dup prevents the default \$protocol options from being changed



Sampler Objects and Options

- Sampler objects are created with `Sampler.new`
as in: `defaultSampler = Sampler.new`
- The `Sampler.new` method takes a single optional parameter
 - A Hash of sampler options!
- Sampler objects respond to named methods for each option
as in: `defaultSampler.vacuum #outputs 10..13`
- Options in existing samplers may be changed
as in: `defaultSampler.vacuum = 15..19 #to increase deltaP`
- Some Sampler objects are themselves complex Objects
 - `defaultSampler.exhaleConfig` is Servo configuration for exhalation



Sampler Options and default values

This Hash is returned by `Sampler.default`

```
:bubblePt => nil      #bubble point of filter in psi
:vacuum => 10..13    #deltaP limits in psi while sampling
:maxDelay => 40      #max. seconds to delay to top of stroke
:minRate => 0.2      #min. rate before “clogged” in ml/second
:evacDelay => 15     #delay in seconds during puck evacuation
:numberOfEvacs => 1  #number of cycles of puck evacuation
:maxPumpVolume => 40 #max mls allowed to compress air to bubblePt
:equalized => 1.0    #call deltaP “equal” if abs(top-bottom)<= this in psi
:minPSI => 2.5       #minimum allowed pressure in syringe in psia
:maxResidue => 0.1   #minimum sample stroke volume in ml
:primeVolume => 7    #volume for initial priming sample stroke in ml
:exhaleConfig => nil #default SS servo configuration for exhale stroke
```

- `Sampler.default[:bubblePt]=35` #set default bubble point to 35 psi



Protocol Sampler Objects

- Each core protocol defines its sampler Object as *\$protocol/Sampler*
- (only if no *\$protocol/Sampler* is already defined)

```
$habSampler ||= Sampler.new :bubblePt=>24
```

```
$larvSampler ||= Sampler.new :bubblePt=>24
```

```
$bacSampler ||= Sampler.new :numberOfEvacs=>2, :bubblePt=>70,  
                             :minRate=>25/(3*60.0), :vacuum=>20..24
```

```
$karSampler ||= Sampler.new(:bubblePt=>10, :vacuum=>4..7)
```

- Each of these is defined in the respective *PROTOCOL.rb* file
 - BAC.rb, LARV.rb, HAB.rb, or KAR.rb
- New *:sampler* key in protocol options provides a means of overriding protocol's default sampler for specific invocations



Contextual Sensor Sampling

- Next Week ...

