1. On the fly configuration outside the conf system

Configuration settings are expressed in code and can be

* 1. Provided using the @Configuration decorator
  2. Set programmatically on an instance prior to execution

SearchCommand.prepare

# Enables acting on configuration/option

# settings/metadata prior to execution

SearchCommand.\_execute

# Invokes the command action on each chunk of data

The required\_fields setting enables selected input fields mode. The finished setting signals that the command/splunkd wish to terminate.

1. Metadata is exchanged with each command invocation.

This is called the getinfo exchange.

1. Select metadata is exchanged with each chunk of data that’s processed.

Two specific fields are noteworthy:

finished=True

# Signals that the command/splunkd wish to terminate

partial=True

# Signals that the command is returning a partial result and

# that more results will follow.

inspector=OrderedDict()

# Contains named messages, metrics, and custom metadata

# which are serialized in the order they’re written

1. StreamingCommand

@Configuration(distributed=True) maps to type='streaming'

@Configuration(distributed=False) maps to type=’stateful’

Other options that apply:

distributed={True|False}

generating={True|False}

partial={True|False}

required\_fields=list()

stderr\_dest={‘log’|’none’}

1. GeneratingCommand

In recognition that any command type may be implemented as a generating command we provide quite a few more configuration possibilities.

* GeneratingCommand may produce events

Equivalent to type=’eventing’, generating=True (A new type? yes)

* GeneratingCommand may produce reports

Equivalent to type='reporting', generating=True

* GeneratingCommand that produces events may be distributed Equivalent to type='streaming', generating=True
* Streaming GeneratingCommand may be distributed or centralized

Equivalent to type='streaming', distributed=True, generating=True

Maps to type='streaming', generating=True

* Centralized Streaming GeneratingCommand is

Equivalent to type='streaming', distributed=False, generating=True

Maps to type=’stateful’, generating=True

⇒ Generating command is modal: type=['eventing'|'reporting'|'streaming']

⇒ Streaming GeneratingCommand may be distributed.

## TODO:

* Remove finished from getinfo exchange; exit will be enough.
* Remove stderr\_dest
* Default GeneratingCommand configuration is type=’streaming’, distributed=False, generating=True
* Implement ReportingCommand
* Implement EventingCommand
* Create 'splunklib.searchcommands' logger and use it to log framework issues