

Class Dependencies

This page outlines the structure and behavior of the FP-Akka quality control workflow as of 19Feb15. Links are provided (in [blue](#)) to the source code for each key class comprising the workflow. Indentation is used to indicated class dependencies (*has-a* and *uses* relationships). For example, **akka.fp.Loader** is dependent on **akka.fp.MongoDBReader**. Maven modules packaging each class are given in [grey](#).

This description is incomplete and includes deep dependencies only for the **akka.fp.NewScientificNameValidator** actor.

As of 2015 Jun 11 some substantive changes have occurred, including reorganization of the package heirarchy, and the production of a DwCaWorkflow that works with input and output files instead of a MongoDB datastore.

The diagram to the right was produced from YesWorkflow markup of the org.filteredpush.akka.workflows.MongoWorkflow class in FP-Akka 1.4.3.

[akka.fp.Loader](#) [FP-Akka module]

- Instantiates a FP-Akka QC Workflow comprising five Akka actors: **akka.fp.MongoDBReader**, **akka.fp.NewScientificNameValidator**, **akka.fp.InternalDateValidator**, **akka.fp.GEORefValidator**, and **akka.fp.MongoSummaryWriter**. (See below for descriptions of each).
- Configures **akka.fp.MongoDBReader** and **akka.fp.MongoSummaryWriter** actors using parsed command-line options.
- Injects **fp.services.COLService** into the *scientificNameService* field of the **akka.fp.NewScientificName** actor.
- Injects **fp.services.InternalDateValidationService** into the *singleDateValidationService* field of the **akka.fp.InternalDateValidator** actor.
- Injects **fp.services.GeoLocate3** into the *geoRefValidationService* field of the **akka.fp.GEORefValidator** actor.
- Executes the workflow.

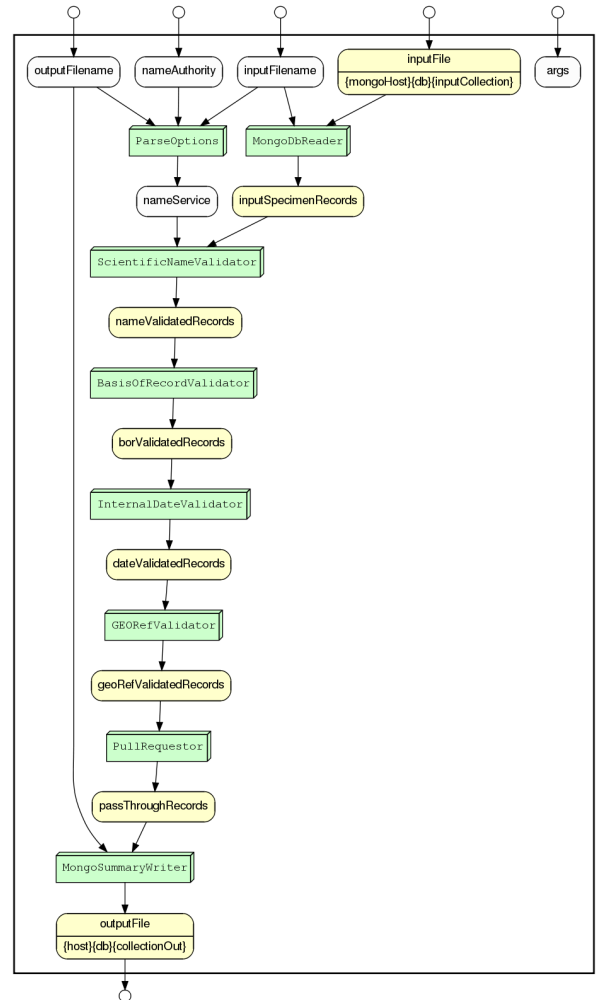
[akka.fp.MongoDBReader](#) [FP-Akka module]

- First actor in the FP-Akka QC workflow.

[akka.fp.NewScientificNameValidator](#) [FP-Akka module]

- Second actor in the FP-Akka QC workflow.
- Akka actor for validating scientific name and authorship fields.
- Uses the class injected into the *scientificNameService* field (**fp.services.COLService** and its parent **fp.services.SciNameServiceParent**) to carry out validation tasks.
- Receives individual SpecimenRecord instances from upstream MongoDBReader.
- For each specimen record:
 - Disassembles the specimen record into fields.
 - Calls the *validateScientificName()* method on the *scientificNameService* (implemented by **fp.services.SciNameServiceParent**) passing the individual field values extracted from the record.
 - Calls getters *getCurationStatus()*, *getCorrectedScientificName()*, *getCorrectedAuthor()*, *getLSID()*, *getComment()*, *getServiceName()* on the *scientificNameService* to extract validation results.
 - If the result returned by *getCurationStatus()* is **CURATED** or **Filled_In**, **replaces** the *scientificName* and *scientificNameAuthorship* fields in the input specimen record with the results from *getCorrectedScientificName()* and *getCorrectedAuthor()* respectively.
 - Adds** to the input specimen record three fields with labels *scinComment*, *scinStatus*, and *scinSource* using the results from *getCurationStatus()*, *getComment()*, and *getServiceName()* on the *scientificNameService*.
 - Forwards the updated specimen record to downstream actors in workflow.
 - Comment by T.M.** The *NewScientificNameValidator* actor overwrites the original *scientificName* and *scientificNameAuthorship* fields in each record that it updates. Downstream actors in the workflow, including *MongoDBWriter* which saves the workflow results, do not have programmatic access to the original values in these fields.

[fp.services.COLService](#) [FP-KurationServices module]



- Injected by **akka.FP.Loader** into the *scientificNameService* field of the **akka.fp.NewScientificNameValidator** actor.
- Derived from **fp.services.SciNameServiceParent** (to which it defers most method calls from the actor).
- Overrides *nameSearchAgainstServices()* to look up scientific name and author in the Catalog of Life using the web service at <http://www.catalogueoflife.org/col/webservice>.
- By the FP-Akka 1.4.0, release this is one of several different Service classes that can be injected.

[fp.services.SciNameServiceParent](#) [FP-KurationServices module]

- Parent class of **fp.Services.COLService**. Implements most of the methods called by **akka.fp.NewScientificNameValidator** actor on the *scientificNameService*.
- Most of name validation logic is defined in *validateScientificName()* method which calls methods on other objects and services.
- Begins by checking internal consistency of scientific name fields *scientificName*, *genus*, *subgenus*, *specificEpithet*, *verbatimTaxonRank*, *taxonRank*, *infraspecificEpithet* using the static *checkConsistencyToAtomicField()* method defined in **fp.util.SciNameServiceUtil**.
- Under some conditions calls **GNISupportingService.resolveDataSourcesNameInLexicalGroupFromGNI()**, **SciNameServiceUtil.checklistBankNameSearch()**, and *nameSearchAgainstServices()*.
- *Comment by T.M.* The internal logic of *fp.services.SciNameServiceParent* is unclear to me.
- *Comment by T.M.* A comment on line 150 states that the call to *nameSearchAgainstServices()* uses the GNI search. The comment value returned to **akka.FP.NewScientificNameValidator** via *getComment()* appears to state the same (see line 161). However, as implemented in *fp.Services.COLService*, it is the Catalog of Life web service that is used.

[fp.services.GNISupportingService](#) [FP-KurationServices module]

- Implements *resolveDataSourcesNameInLexicalGroupFromGNI()* called by *validateScientificNameAgainstServices()* method in **fp.services.SciNameServiceParent**.

[fp.util.SciNameServiceUtil](#) [FP-KurationServices module]

- Implements the *checkConsistencyToAtomicField()* method invoked by **fp.Services.SciNameServiceParent**.
 - *Comment by T.M.* *checkConsistencyToAtomicField()* appears only to construct a scientific name from atomic fields for comparison with provided *scientificName* only if there is content in the *genus*, *specificEpithet*, and *infraspecificEpithet* fields. Otherwise it reports UNABLE_DETERMINE_VALIDITY, "can't construct sciName from atomic fields." Is this the correct behavior?
- Implements the *checkMisspelling()* method invoked by **fp.Services.SciNameServiceParent**.
- Implements the *checklistBankNameSearch()* method invoked by **fp.Services.SciNameServiceParent**.

[edu.harvard.mcz.nametools.NameUsage](#) [FP-KurationServices module]

- Used by **fp.util.SciNameServiceUtil** during calls to *checklistBankNameSearch()*.
[fp.services.InternalDateValidationService](#)

[org.gbif.nameparser.NameParser](#)

- Used by **fp.util.SciNameServiceUtil** during calls to *checkConsistencyToAtomicField()*.

[org.gbif.api.model.checklistbank.ParsedName](#)

- Used by **fp.util.SciNameServiceUtil** during calls to *checkConsistencyToAtomicField()*.

[akka.fp.InternalDateValidator](#) [FP-Akka module]

- Third actor in the FP-Akka QC workflow.
- Akka actor for validating the specimen collector and collection date fields.
- Uses the class injected into the *singleDateValidationService* field (**fp.services.InternalDateValidationService**) to carry out validation tasks.
- Receives individual SpecimenRecord instances from upstream MongoDBReader.
- For each specimen record:
 - Disassembles the specimen record into fields.
 - Calls the *validateDate()* method on the *singleDateValidationService* (implemented by **fp.services.InternalDateValidationService**) passing the individual field values extracted from the record.
 - Calls getters *getCurationStatus()*, *getCorrectedDate()*, *getComment()*, *getServiceName()* on the *singleDateValidationService* to extract validation results.

- If the result returned by *getCurationStatus()* is **CURATED** or **Filled_In**, **replaces** the *eventDate* field in the input specimen record with the results from *getCorrectedDate()*.
- **Adds** to the input specimen record three fields with labels *dateComment*, *dateStatus*, and *dateSource* using the results from *getCurationStatus()*, *getComment()*, and *getServiceName()* on the *singleDateValidationService*.
- Forwards the updated specimen record to downstream actors in workflow.

[fp.services.InternalDateValidationService](#) [FP-KurationServices module]

- Implements date validation methods called by **akka.fp.InternalDateValidator**.
- Most of name validation logic is defined in *validateDate()* method which calls methods on other objects and services.
- *validateDate()* calls the private *parseDate()* method to check internal consistency of fields.
- Then uses the *checkWithAuthorSolr()* method to validate the collector and collection date with the Filteredpush entomologists list (in a Solr server) and sets *curationStatus* to UNABLED_CURATED if collection date is not within the life span of the collector.
- ~~Although there is a *checkWithAuthorHarvard()* method that validates collector and collection date against the Harvard List of Botanists, the code is hardwired only to use the *checkWithAuthorSolr()*.~~ As of FP-Akka 1.4.0 this has been fixed, with both the SCAN entomologists list and the HUH botanists list being used as sources.
- The **validation of collector** with respect to **collection date** is meaningfully performed **only for entomologists** ~~not for botanists~~ entomologists and botanists, not for other collectors.

[akka.fp.GEORefValidator](#) [FP-Akka module]

- Fourth actor in the FP-Akka QC workflow.
- Akka actor for validating georeference fields.

[fp.services.Geolocate3](#) [FP-KurationServices module]

- GeoLocate3 checks decimalLatitude and decimalLongitude for sanity, compares them with the value of country using a shapefile of country boundaries, and checks them against the georeferences returned for the country /state/county/locality by the Tulane GeoLocate service (confusingly using the *geolocate2* service call). A check is also made if the locality is on land, **all marine localities will fail georeference validation**.

[akka.fp.MongoSummaryWriter](#) [FP-Akka module]

- Fifth and final actor in the FP-Akka QC workflow.
- Writes out JSON containing the modified record, summary provenance, and a block of provenance provided by each actor.