

v1.0.0

GeoWave Command Line Controls

Commands

Helpful Commands & Flags

GeoWave supports a few extra commands that can be used for informational purposes to debug or explore command usage.

- [Debug](#)
- [Version](#)
- [Help](#)
- [Explain](#)

Debug Flag (--debug)

Use the debug flag to increase the debug logging output by GeoWave on the console to DEBUG. By default, it is set to WARN. This flag must come right after 'geowave' and before any subcommand:

```
geowave --debug <command> <subcommand> <options...>
```

Version Flag (--version)

The version flag will output the build arguments that were used to build GeoWave, as well as the version of the GeoWave tools jar you're using:

```
geowave --version
```

Help Command

The help command will show arguments and their defaults. It can be prepended to any GeoWave command. If you use it while also specifying a sub-command and its arguments, that command's help information will be displayed:

```
geowave help <command> <subcommand>
```

Explain Command

The explain command will show a simplified tabular view of the arguments and their current values. Use this to determine what values are being passed to GeoWave. It also shows hidden parameters and their values, if there are any. An example would be additional Accumulo options:

```
geowave explain store add -t accumulo
```

Config Commands

Commands that affect local configuration only (Required options are designated with an *)

- [AWS](#)
- [GeoServer](#)
- [HDFS](#)
- [List](#)
- [NewCryptoKey](#)
- [Set](#)

geowave config aws

NAME

geowave config aws - Create a local configuration for aws s3

SYNOPSIS

geowave config aws <AWS S3 endpoint URL> (for example s3.amazonaws.com)

DESCRIPTION

This command will create a local configuration for aws s3

OPTIONS

geowave config geoserver

NAME

geowave config geoserver - Create a local configuration for GeoServer

SYNOPSIS

geowave config geoserver [options] <GeoServer URL (for example <http://localhost:8080/geoserver> or <https://localhost:8443/geoserver>), or simply host:port and appropriate assumptions are made>

DESCRIPTION

This command will create a local configuration for connecting to GeoServer.

OPTIONS

- -p, --password

GeoServer Password - Can be specified as 'pass:<password>', 'file:<local file containing the password>', 'propfile:<local properties file containing the password>:<property file key>', 'env:<variable containing the pass>', or stdin

- -u, --username
 - GeoServer User
- -ws, --workspace
 - GeoServer Default Workspace

SSL CONFIGURATION OPTIONS

- --sslKeyManagerAlgorithm
 - Specify the algorithm to use for the keystore.
- --sslKeyManagerProvider
 - Specify the key manager factory provider.
- --sslKeyPassword
 - Specify the password to be used to access the server certificate from the specified keystore file.
 - Can be specified as 'pass:<password>', 'file:<local file containing the password>', 'propfile:<local properties file containing the password>:<property file key>', 'env:<variable containing the pass>', or stdin
- --sslKeyStorePassword
 - Specify the password to use to access the keystore file.
 - Can be specified as 'pass:<password>', 'file:<local file containing the password>', 'propfile:<local properties file containing the password>:<property file key>', 'env:<variable containing the pass>', or stdin
- --sslKeyStorePath
 - Specify the absolute path to where the keystore file is located on system. The keystore contains the server certificate to be loaded.
- --sslKeyStoreProvider
 - Specify the name of the keystore provider to be used for the server certificate.
- --sslKeyStoreType
 - The type of keystore file to be used for the server certificate.
- --sslSecurityProtocol
 - Specify the Transport Layer Security (TLS) protocol to use when connecting to the server. By default, the system will use TLS.
- --sslTrustManagerAlgorithm
 - Specify the algorithm to use for the truststore.
 -

- `--sslTrustManagerProvider`
 - Specify the trust manager factory provider.
- `--sslTrustStorePassword`
 - Specify the password to use to access the truststore file.
 - Can be specified as 'pass:<password>', 'file:<local file containing the password>', 'propfile:<local properties file containing the password>:<property file key>', 'env:<variable containing the pass>', or stdin
- `--sslTrustStorePath`
 - Specify the absolute path to where truststore file is located on system.
 - The truststore file is used to validate client certificates.
- `--sslTrustStoreProvider`
 - Specify the name of the truststore provider to be used for the server certificate.
- `--sslTrustStoreType`
 - Specify the type of key store used for the truststore, i.e. JKS (Java KeyStore).

geowave config hdfs

NAME

geowave config hdfs - Create a local configuration for HDFS

SYNOPSIS

geowave config hdfs [options] <HDFS DefaultFS URL>

DESCRIPTION

This command will create a local configuration for HDFS.

OPTIONS

geowave config list

NAME

geowave config list - List property name within cache

SYNOPSIS

geowave config list [options]

DESCRIPTION

The geowave config list operator will list all properties in local config. -f or --filter will allow you to give a regex to filter the list by (useful regexes may be 'store' or 'index' to isolate properties for one or the other or a particular store/index name to further isolate the list).

OPTIONS

- -f or --filter <arg>
 - Filter list by a regex

geowave config newcryptokey

NAME

geowave config newcryptokey - Generate a new security cryptography key for use with configuration properties

SYNOPSIS

geowave config newcryptokey

DESCRIPTION

This command will generate a new security cryptography key for use with configuration properties. This is primarily used if there is a need to re-encrypt the local configurations based on a new security token, should the old one have been compromised.

OPTIONS

There are currently no options available for this command

geowave config set

NAME

geowave config set - will set a valid property name within the cache

SYNOPSIS

geowave config set [options]

DESCRIPTION

The geowave config set operator will set a valid property name within the cache. This can be useful if you want to update a particular property of a index or store.

OPTIONS

There are currently no options for this command

Store Commands

Commands for managing GeoWave data stores.

- [AddStore](#)
- [ClearStore](#)
- [CopyStore](#)
- [CpStoreCfg](#)
- [RmStore](#)
- [ListTypes](#)
- [RmType](#)
- [Version](#)

geowave store add

NAME

geowave store add - Create a store within Geowave

SYNOPSIS

geowave store add [options] <name>

DESCRIPTION

The geowave store add operator will create a new store in GeoWave.

OPTIONS

- -d, --default
 - Make this the default store in all operations
- *-t, --type <arg>
 - The type of store, such as accumulo, hbase, rocksdb, redis, cassandra, bigtable, dynamodb, kudu, etc.
 - Required!
 - When -t accumulo option is used, additional options are:
 - --gwNamespace
 - The geowave namespace
 - Default is no namespace
 - *-i, --instance

- The Accumulo instance ID
 - Required!
- *-p, --password
 - The password for the user
 - Required!
- *-u, --user
 - A valid Accumulo user ID
 - Required!
- *-z, --zookeeper
 - A comma-separated list of zookeeper servers that an Accumulo instance is using
 - Required!
- When -t, --type hbase option is used, additional options are:
 - *-z, --zookeeper
 - A comma-separated list of zookeeper servers that an Accumulo instance is using
 - Required!
 - --coprocessorJar
 - Path (HDFS URL) to the jar containing coprocessor classes
 - --disableVerifyCoprocessors
 - --scanCacheSize
- When -t, --type redis option is used, additional options are:
 - *-a, --address
 - The address to connect to, such as redis://127.0.0.1:6379
 - Required!
 - --compression
 - Can be "snappy", "lz4", or "none". Defaults to snappy.
- When -t, --type rocksdb option is used, additional options are:
 - --dir
 - The directory to read/write to. Defaults to "rocksdb" in the working directory.
 - --compactOnWrite
 - Whether to compact on every write, if false it will only compact on merge. Defaults to true
 - --batchWriteSize
 - The size (in records) for each batched write. Anything ≤ 1 will use synchronous single

record writes without batching. Defaults to 1000.

- When -t, --type cassandra option is used, additional options are:
 - --contactPoints
 - A single contact point or a comma delimited set of contact points to connect to the Cassandra cluster.
 - Required!
 - --replicas
 - The number of replicas to use when creating a new keyspace.
 - --durableWrites
 - Whether to write to commit log for durability, configured only on creation of new keyspace.
 - --batchWriteSize
 - The number of inserts in a batch write.
- When -t, --type dynamodb option is used, additional options are:
 - --endpoint
 - The endpoint to connect to(specify either endpoint/region not both)
 - --region
 - The AWS region to use(specify either endpoint/region not both)
 - --initialWriteCapacity
 - --initialReadCapacity
 - --maxConnections
 - The maximum number of open http(s) connections active at any given time
 - --protocol
 - The protocol to use. HTTP or HTTPS
 - --cacheResponseMetadata
 - Whether to cache responses from aws(true or false). High performance systems can disable this but debugging will be more difficult.
- When -t, --type kudu option is used, additional options are:
 - --kuduMaster
 - An URL for the Kudu master node
 - Required!
- When -t, --type bigtable option is used, additional options are:
 - --projectId

- --instanceId
- --scanCacheSize

geowave store clear

NAME

geowave store clear - Clear ALL data from a GeoWave store and delete tables

SYNOPSIS

geowave store clear [options] <store name>

DESCRIPTION

The geowave store clear operator will clear ALL data from a GeoWave store and delete tables

OPTIONS

There are currently no options for this command

geowave store copy

NAME

geowave store copy - Copy a data store

SYNOPSIS

geowave store copy [options] <input store name> <output store name>

DESCRIPTION

This command will allow a user to copy a data store

OPTIONS

- * --hdfsHostPort
 - The hdfs host port
- * --jobSubmissionHostPort
 - The job submission tracker
- --maxSplits
 - The max partitions for the input data
- --minSplits
 - The min partitions for the input data

- `--numReducers`
 - Number of threads writing at a time (default: 8)
 - Default: 8

geowave store copycfg

NAME

geowave store copycfg - Copy and modify existing store configuration

SYNOPSIS

geowave store copycfg [options] <name> <new name>

DESCRIPTION

The geowave store copycfg operator will copy and modify an existing GeoWave store. It is possible to override values as you copy, such as `cpstore old new --gwNamespace new_namespace`.

OPTIONS

- `-d, --default`
 - Makes this the default store in all operations

geowave store rm

NAME

geowave store rm - removes an existing store from GeoWave configuration

SYNOPSIS

geowave store rm [options] <store name>

DESCRIPTION

The geowave store rm operator will remove an existing store from GeoWave configuration

OPTIONS

There are currently no options for this command

geowave store listtypes

NAME

geowave store listtypes - display all types in this remote store

SYNOPSIS

geowave store listtypes [options] <store name>

DESCRIPTION

The geowave store listtypes operator will display all types in this remote store.

OPTIONS

There are currently no options for this command

geowave store rmtype

NAME

geowave store rmtype - Remove a type and all associated data from a data store

SYNOPSIS

geowave store rmtype [options] <store name> <type name>

DESCRIPTION

The geowave store rmtype operator will remove a type and all associated data from a data store

OPTIONS

There are currently no options for this command

geowave store version

NAME

geowave store version - Get the version of GeoWave used by a data store

SYNOPSIS

geowave store version [options] <store name>

DESCRIPTION

This command will return the version of GeoWave used by a data store.

OPTIONS

There are currently no options for this command

Index Commands

Commands for managing GeoWave indices.

- [AddIndex](#)
- [CompactIndex](#)
- [ListIndex](#)
- [RmIndex](#)

geowave index add

NAME

geowave index add - Add an index to a data store

SYNOPSIS

geowave index add [options] <store name> <index name>

DESCRIPTION

The geowave index add operator will create an index in a data store if it does not already exist.

OPTIONS

- -c --crs (will only be shown if you have already defined spatial or satial_temporal as your type)
 - The native Coordinate Reference System used within the index. All spatial data will be projected into this CRS for appropriate indexing as needed.
 - Default: EPSG:4326
- -np, --numPartitions
 - The number of partitions. Default partitions will be 1.
 - Default: 1
- -ps, --partitionStrategy
 - The partition strategy to use. Default will be none.
 - Default: NONE
 - Possible Values: [NONE, HASH, ROUND_ROBIN]
- * -t, --type
 - The type of index, such as spatial, or spatial_temporal

geowave index compact

NAME

geowave index compact - Compact all rows for a given index

SYNOPSIS

geowave index compact [options] <store name> <index name>

DESCRIPTION

This command will allow a user to compact all rows for a given index

OPTIONS

There are currently no options for this command

geowave index list

NAME

geowave index list - Display all indices in a data store

SYNOPSIS

geowave index list [options] <store name>

DESCRIPTION

The geowave index list operator will display all indices in a data store

OPTIONS

There are currently no options for this command

geowave index rm

NAME

geowave index rm - Remove an index and all associated data from a data store

SYNOPSIS

geowave index rm [options] <store name> <index name>

DESCRIPTION

The geowave index rm operator will remove an index and all of its data from a data store

OPTIONS

There are currently no options for this command

Statistics Commands

Commands to manage GeoWave statistics.

- [Calcstat](#)
- [ListStats](#)
- [CompactStats](#)
- [ReCalcStats](#)
- [RmStat](#)

geowave stat calc

NAME

geowave stat calc - Calculate a specific statistic in the remote store, given adapter ID and statistic ID

SYNOPSIS

geowave stat calc [options] <store name> <type name> <stat type>

DESCRIPTION

The geowave stat calc operator will calculate a specific statistic in the remote store, given type name and statistic type.

OPTIONS

- --auth
 - The authorizations used for the statistics calculation as a subset of the accumulo user authorization; by default all authorizations are used.
- --json
 - Output in JSON format.
 - Default: false

geowave stat list

NAME

geowave stat list - Print statistics of an existing GeoWave dataset to standard output

SYNOPSIS

geowave stat list [options] <store name> [<type name>]

DESCRIPTION

The geowave stat list operator will print statistics of an existing GeoWave dataset to standard output

OPTIONS

- --auth
 - The authorizations used for the statistics calculation as a subset of the accumulo user authorization; by default all authorizations are used.
- --json
 - Output in JSON format.
 - Default: false

geowave stat compact

NAME

geowave stat compact - Combine all statistics in GeoWave namespace, can be useful for making a datastore more efficient

SYNOPSIS

geowave stat compact [options] <storename>

DESCRIPTION

This command will combine all statistics in a GeoWave namespace

OPTIONS

There are currently no options for this command

geowave stat recalc

NAME

geowave stat recalc - Calculate the statistics of an existing GeoWave dataset

SYNOPSIS

geowave stat recalc [options] <store name> [<type name>]

DESCRIPTION

The geowave stat recalc operator will calculate the statistics of an existing GeoWave dataset

OPTIONS

- `--auth`
 - The authorizations used for the statistics calculation as a subset of the accumulo user authorization; by default all authorizations are used.
- `--json`
 - Output in JSON format.
 - Default: false

geowave stat rm

NAME

geowave stat rm - Remove a statistic from the remote store

SYNOPSIS

geowave stat rm [options] <store name> <type name> <stat type>

DESCRIPTION

The geowave stat rm operator will Remove a statistic from the remote store.

OPTIONS

- `--auth`
 - The authorizations used for the statistics calculation as a subset of the accumulo user authorization; by default all authorizations are used.
 - `--json`
 - Output in JSON format.
 - Default: false
-

Ingest Commands

Commands that ingest data directly into GeoWave or stage data to be ingested into GeoWave (Required options are designated with an *)

- [KafkaToGW](#)
- [ListPlugins](#)
- [LocalToGW](#)
- [LocalToHdfs](#)
- [LocalToKafka](#)
- [LocalToMRGW](#)
- [MRToGW](#)
- [SparkToGW](#)

geowave ingest kafkaToGW

NAME

geowave ingest kafkaToGW - Subscribe to a Kafka topic and ingest into GeoWave

SYNOPSIS

geowave ingest kafkaToGW [options] <store name> <comma delimited index list>

DESCRIPTION

The geowave ingest kafkaToGW operator will ingest supported files that already exist in HDFS

OPTIONS

- --autoOffsetReset
 - What to do when there is no initial offset in ZooKeeper or if an offset is out of range:
 - smallest : automatically reset the offset to the smallest offset
 - largest : automatically reset the offset to the largest offset
 - anything else: throw exception to the consumer
- --avro.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --avro.cql
 - A CQL filter, only data matching this filter will be ingested

- Default: <empty string>
- --avro.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --batchSize
 - The data will automatically flush after this number of entries
 - Default: 10000
- --consumerTimeoutMs
 - By default, this value is -1 and a consumer blocks indefinitely if no new message is available for consumption. By setting the value to a positive integer, a timeout exception is thrown to the consumer if no message is available for consumption after the specified timeout value.
- -x, --extension
 - individual or comma-delimited set of file extensions to accept (optional)
- --fetchMessageMaxBytes
 - The number of bytes of messages to attempt to fetch for each topic-partition in each fetch request. These bytes will be read into memory for each partition, so this helps control the memory used by the consumer. The fetch request size must be at least as large as the maximum message size the server allows or else it is possible for the producer to send messages larger than the consumer can fetch.
- -f, --formats
 - Explicitly set the ingest formats by name (or multiple comma-delimited formats), if not set all available ingest formats will be used
- --gdelt.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --gdelt.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --gdelt.extended
 - A flag to indicate whether extended data format should be used
 - Default: false
- --gdelt.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --geolife.avro

- A flag to indicate whether avro feature serialization should be used
- Default: false
- --geolife.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --geolife.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --geotools-raster.coverage
 - Optional parameter to set the coverage name (default is the file name)
- --geotools-raster.crs
 - A CRS override for the provided raster file
- --geotools-raster.histogram
 - Build a histogram of samples per band on ingest for performing band equalization
 - Default: false
- --geotools-raster.mergeStrategy
 - Optional parameter to choose a tile merge strategy used for mosaic.
 - Default behavior will be **none**.
 - Alternatively 'no-data' will mosaic the most recent tile over previous tiles, except where there are no data values.
- --geotools-raster.nodata
 - Optional parameter to set 'no data' values, if 1 value is giving it is applied for each band, if multiple are given then the first totalNoDataValues/totalBands are applied to the first band and so on, so each band can have multiple differing 'no data' values if needed
 - Default: []
- --geotools-raster.pyramid
 - Build an image pyramid on ingest for quick reduced resolution query
 - Default: false
- --geotools-raster.separateBands
 - Optional parameter to separate each band into its own coverage name. By default the coverage name will have '_Bn' appended to it where **n** is the band's index.
 - Default: false
- --geotools-raster.tileSize
 - Optional parameter to set the tile size stored (default is 256)

- Default: 256
- `--geotools-vector.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--geotools-vector.data`
 - A map of date field names to the date format of the file. Use commas to separate each entry, then the first ':' character will separate the field name from the format. Use '\' to include a comma in the format. For example: "time:MM:dd:YYYY,time2:YYYY/MM/dd hh:mm:ss" configures fields 'time' and 'time2' as dates with different formats
- `--geotools-vector.type`
 - Optional parameter that specifies specific type name(s) from the source file
 - Default: []
- `--gpx.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--gpx.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--gpx.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `--groupId`
 - A string that uniquely identifies the group of consumer processes to which this consumer belongs. By setting the same group id multiple processes indicate that they are all part of the same consumer group.
- `*--kafkaprops`
 - Properties file containing Kafka properties
- `--reconnectOnTimeout`
 - This flag will flush when the consumer timeout occurs (based on kafka property 'consumer.timeout.ms') and immediately reconnect
 - Default: false
- `--tdrive.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false

- `--tdrive.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--tdrive.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `--twitter.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--twitter.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--twitter.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `-v, --visibility`
 - The visibility of the data ingested (optional; default is 'public')
- `--zookeeperConnect`
 - Specifies the ZooKeeper connection string in the form `hostname:port` where host and port are the host and port of a ZooKeeper server. To allow connecting through other ZooKeeper nodes when that ZooKeeper machine is down you can also specify multiple hosts in the form `hostname1:port1,hostname2:port2,hostname3:port3`.

geowave ingest listplugins

NAME

geowave ingest listplugins - List supported data store types, index types, and ingest formats

SYNOPSIS

geowave ingest listplugins [options]

DESCRIPTION

This command will list all the data store types, index types, and ingest formats supported by the version of GeoWave being run.

OPTIONS

There are currently no options for this command

geowave ingest localToGW

NAME

geowave ingest localToGW - Ingest supported files in local file system directly, from S3 or from HDFS

SYNOPSIS

geowave ingest localToGW [options] <file or directory> <storename> <comma delimited index list>

DESCRIPTION

The geowave ingest localToGW operator will run the ingest code (parse to features, load features to geowave) against local file system content.

OPTIONS

- --avro.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --avro.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --avro.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- -x, --extension
 - individual or comma-delimited set of file extensions to accept (optional)
- -f, --formats
 - Explicitly set the ingest formats by name (or multiple comma-delimited formats), if not set all available ingest formats will be used
- --gdelt.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --gdelt.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>

--gdelt.extended

- A flag to indicate whether extended data format should be used
- Default: false

• --gdelt.typename

- A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)

• --geolife.avro

- A flag to indicate whether avro feature serialization should be used
- Default: false

• --geolife.cql

- A CQL filter, only data matching this filter will be ingested
- Default: <empty string>

• --geolife.typename

- A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)

• --geotools-raster.coverage

- Optional parameter to set the coverage name (default is the file name)

• --geotools-raster.crs

- A CRS override for the provided raster file

• --geotools-raster.histogram

- Build a histogram of samples per band on ingest for performing band equalization
- Default: false

• --geotools-raster.mergeStrategy

- Optional parameter to choose a tile merge strategy used for mosaic.
- Default behavior will be **none**. Alternatively 'no-data' will mosaic the most recent tile over previous tiles, except where there are no data values.
- Default: none

• --geotools-raster.nodata

- Optional parameter to set 'no data' values, if 1 value is giving it is applied for each band, if multiple are given then the first totalNoDataValues/totalBands are applied to the first band and so on, so each band can have multiple differing 'no data' values if needed
- Default: []

• --geotools-raster.pyramid

- Build an image pyramid on ingest for quick reduced resolution query

- Default: false
- `--geotools-raster.separateBands`
 - Optional parameter to separate each band into its own coverage name. By default the coverage name will have '_Bn' appended to it where `n` is the band's index.
 - Default: false
- `--geotools-raster.tileSize`
 - Optional parameter to set the tile size stored (default is 256)
 - Default: 256
- `--geotools-vector.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--geotools-vector.data`
 - A map of date field names to the date format of the file. Use commas to separate each entry, then the first ':' character will separate the field name from the format. Use '\' to include a comma in the format. For example: "time:MM:dd:YYYY,time2:YYYY/MM/dd hh:mm:ss" configures fields 'time' and 'time2' as dates with different formats
- `--geotools-vector.type`
 - Optional parameter that specifies specific type name(s) from the source file
 - Default: []
- `--gpx.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--gpx.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--gpx.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `--tdrive.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--tdrive.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>

- `--tdrive.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `-t, --threads`
 - number of threads to use for ingest, default to 1 (optional)
 - Default: 1
- `--twitter.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--twitter.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--twitter.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `-v, --visibility`
 - The visibility of the data ingested (optional; default is 'public')

geowave ingest localToHdfs

NAME

geowave ingest localToHdfs - Stage supported files in local file system to HDFS

SYNOPSIS

geowave ingest localToHdfs [options] <file or directory> <hdfs host:port> <path to base directory to write to>

DESCRIPTION

The geowave ingest localToHdfs operator will stage supported files in the local file system to HDFS

OPTIONS

- `--avro.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--avro.cql`

- A CQL filter, only data matching this filter will be ingested
- Default: <empty string>
- --avro.typeName
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- -x, --extension
 - individual or comma-delimited set of file extensions to accept (optional)
- -f, --formats
 - Explicitly set the ingest formats by name (or multiple comma-delimited formats), if not set all available ingest formats will be used
- --gdelt.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --gdelt.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --gdelt.extended
 - A flag to indicate whether extended data format should be used
 - Default: false
- --gdelt.typeName
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --geolife.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --geolife.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --geolife.typeName
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --geotools-raster.coverage
 - Optional parameter to set the coverage name (default is the file name)

- `--geotools-raster.crs`
 - A CRS override for the provided raster file
- `--geotools-raster.histogram`
 - Build a histogram of samples per band on ingest for performing band equalization
 - Default: false
- `--geotools-raster.mergeStrategy`
 - Optional parameter to choose a tile merge strategy used for mosaic.
 - Default behavior will be `none`. Alternatively 'no-data' will mosaic the most recent tile over previous tiles, except where there are no data values.
 - Default: none
- `--geotools-raster.nodata`
 - Optional parameter to set 'no data' values, if 1 value is giving it is applied for each band, if multiple are given then the first totalNoDataValues/totalBands are applied to the first band and so on, so each band can have multiple differing 'no data' values if needed
 - Default: []
- `--geotools-raster.pyramid`
 - Build an image pyramid on ingest for quick reduced resolution query
 - Default: false
- `--geotools-raster.separateBands`
 - Optional parameter to separate each band into its own coverage name. By default the coverage name will have '_Bn' appended to it where `n` is the band's index.
 - Default: false
- `--geotools-raster.tileSize`
 - Optional parameter to set the tile size stored (default is 256)
 - Default: 256
- `--geotools-vector.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--geotools-vector.data`
 - A map of date field names to the date format of the file. Use commas to separate each entry, then the first ':' character will separate the field name from the format. Use '\' to include a comma in the format. For example: "time:MM:dd:YYYY,time2:YYYY/MM/dd hh:mm:ss" configures fields 'time' and 'time2' as dates with different formats
- `--geotools-vector.type`

- Optional parameter that specifies specific type name(s) from the source file
- Default: []
- --gpx.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --gpx.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --gpx.typeName
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --tdrive.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --tdrive.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --tdrive.typeName
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --twitter.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --twitter.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --twitter.typeName
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)

geowave ingest localToKafka

NAME

geowave ingest localToKafka - Stage supported files in local file system to a Kafka topic

SYNOPSIS

geowave ingest localToKafka [options] <file or directory>

DESCRIPTION

The geowave ingest localToKafka operator will stage supported files in the local file system to a Kafka topic

OPTIONS

- --avro.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --avro.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --avro.typeName
 - A comma-delimited set of typeNames to ingest, feature types matching the specified typeNames will be ingested (optional, by default all types will be ingested)
- -x, --extension
 - individual or comma-delimited set of file extensions to accept (optional)
- -f, --formats
 - Explicitly set the ingest formats by name (or multiple comma-delimited formats), if not set all available ingest formats will be used
- --gdelt.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --gdelt.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --gdelt.extended
 - A flag to indicate whether extended data format should be used
 - Default: false
- --gdelt.typeName
 - A comma-delimited set of typeNames to ingest, feature types matching the specified typeNames will be ingested (optional, by default all types will be ingested)

- `--geolife.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--geolife.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--geolife.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `--geotools-raster.coverage`
 - Optional parameter to set the coverage name (default is the file name)
- `--geotools-raster.crs`
 - A CRS override for the provided raster file
- `--geotools-raster.histogram`
 - Build a histogram of samples per band on ingest for performing band equalization
 - Default: false
- `--geotools-raster.mergeStrategy`
 - Optional parameter to choose a tile merge strategy used for mosaic.
 - Default behavior will be **none**. Alternatively 'no-data' will mosaic the most recent tile over previous tiles, except where there are no data values.
 - Default: none
- `--geotools-raster.nodata`
 - Optional parameter to set 'no data' values, if 1 value is giving it is applied for each band, if multiple are given then the first totalNoDataValues/totalBands are applied to the first band and so on, so each band can have multiple differing 'no data' values if needed
 - Default: []
- `--geotools-raster.pyramid`
 - Build an image pyramid on ingest for quick reduced resolution query
 - Default: false
- `--geotools-raster.separateBands`
 - Optional parameter to separate each band into its own coverage name. By default the coverage name will have '_Bn' appended to it where **n** is the band's index.
 - Default: false
- `--geotools-raster.tileSize`

- Optional parameter to set the tile size stored (default is 256)
- Default: 256
- --geotools-vector.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --geotools-vector.data
 - A map of date field names to the date format of the file. Use commas to separate each entry, then the first ':' character will separate the field name from the format. Use '\' to include a comma in the format. For example: "time:MM:dd:YYYY,time2:YYYY/MM/dd hh:mm:ss" configures fields 'time' and 'time2' as dates with different formats
- --geotools-vector.type
 - Optional parameter that specifies specific type name(s) from the source file
 - Default: []
- --gpx.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --gpx.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --gpx.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- * --kafkaprops
 - Properties file containing Kafka properties
- --metadataBrokerList
 - This is for bootstrapping and the producer will only use it for getting metadata (topics, partitions and replicas). The socket connections for sending the actual data will be established based on the broker information returned in the metadata. The format is host1:port1,host2:port2, and the list can be a subset of brokers or a VIP pointing to a subset of brokers.
- --producerType
 - This parameter specifies whether the messages are sent asynchronously in a background thread. Valid values are (1) async for asynchronous send and (2) sync for synchronous send. By setting the producer to async we allow batching together of requests (which is great for throughput) but open the possibility of a failure of the client machine dropping unsent data.

- `--requestRequiredAcks`
 - This value controls when a produce request is considered completed. Specifically, how many other brokers must have committed the data to their log and acknowledged this to the leader?
- `--retryBackoffMs`
 - The amount of time to wait before attempting to retry a failed produce request to a given topic partition. This avoids repeated sending-and-failing in a tight loop.
- `--serializerClass`
 - The serializer class for messages. The default encoder takes a `byte[]` and returns the same `byte[]`.
- `--tdrive.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: `false`
- `--tdrive.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: `<empty string>`
- `--tdrive.typeName`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `--twitter.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: `false`
- `--twitter.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: `<empty string>`
- `--twitter.typeName`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)

geowave ingest localToMrGW

NAME

geowave ingest localToMrGW - Copy supported files from local file system to HDFS and ingest from HDFS

SYNOPSIS

geowave ingest localToMrGW [options] <file or directory> <hdfs host:port> <path to base directory to write to> <store name> <comma delimited index list>

DESCRIPTION

The geowave ingest localToMrGW operator will Copy supported files from local file system to HDFS and ingest from HDFS

OPTIONS

- --avro.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --avro.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --avro.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- -x, --extension
 - individual or comma-delimited set of file extensions to accept (optional)
- -f, --formats
 - Explicitly set the ingest formats by name (or multiple comma-delimited formats), if not set all available ingest formats will be used
- --gdelt.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --gdelt.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --gdelt.extended
 - A flag to indicate whether extended data format should be used
 - Default: false
- --gdelt.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --geolife.avro

- A flag to indicate whether avro feature serialization should be used
- Default: false
- `--geolife.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--geolife.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `--geotools-raster.coverage`
 - Optional parameter to set the coverage name (default is the file name)
- `--geotools-raster.crs`
 - A CRS override for the provided raster file
- `--geotools-raster.histogram`
 - Build a histogram of samples per band on ingest for performing band equalization
 - Default: false
- `--geotools-raster.mergeStrategy`
 - Optional parameter to choose a tile merge strategy used for mosaic.
 - Default behavior will be `none`. Alternatively 'no-data' will mosaic the most recent tile over previous tiles, except where there are no data values.
 - Default: none
- `--geotools-raster.nodata`
 - Optional parameter to set 'no data' values, if 1 value is giving it is applied for each band, if multiple are given then the first totalNoDataValues/totalBands are applied to the first band and so on, so each band can have multiple differing 'no data' values if needed
 - Default: []
- `--geotools-raster.pyramid`
 - Build an image pyramid on ingest for quick reduced resolution query
 - Default: false
- `--geotools-raster.separateBands`
 - Optional parameter to separate each band into its own coverage name. By default the coverage name will have '_Bn' appended to it where `n` is the band's index.
 - Default: false
- `--geotools-raster.tileSize`
 - Optional parameter to set the tile size stored (default is 256)

- Default: 256
- `--geotools-vector.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--geotools-vector.data`
 - A map of date field names to the date format of the file. Use commas to separate each entry, then the first ':' character will separate the field name from the format. Use '\' to include a comma in the format. For example: "time:MM:dd:YYYY,time2:YYYY/MM/dd hh:mm:ss" configures fields 'time' and 'time2' as dates with different formats
- `--geotools-vector.type`
 - Optional parameter that specifies specific type name(s) from the source file
 - Default: []
- `--gpx.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--gpx.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--gpx.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `--jobtracker`
 - Hadoop job tracker hostname and port in the format hostname:port
- `--resourcemanager`
 - Yarn resource manager hostname and port in the format hostname:port
- `--tdrive.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--tdrive.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--tdrive.typename`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)

- `--twitter.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--twitter.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--twitter.typeName`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `-v, --visibility`
 - The visibility of the data ingested (optional; default is 'public')

geowave ingest mrToGW

NAME

geowave ingest mrToGW - Ingest supported files that already exist in HDFS

SYNOPSIS

geowave ingest mrToGW [options] <hdfs host:port> <path to base directory to write to> <store name>
<comma delimited index list>

DESCRIPTION

The geowave ingest mrToGW operator will ingest supported files that already exist in HDFS

OPTIONS

- `--avro.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--avro.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--avro.typeName`
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- `-x, --extension`

- individual or comma-delimited set of file extensions to accept (optional)
- -f, --formats
 - Explicitly set the ingest formats by name (or multiple comma-delimited formats), if not set all available ingest formats will be used
- --gdelt.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --gdelt.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --gdelt.extended
 - A flag to indicate whether extended data format should be used
 - Default: false
- --gdelt.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --geolife.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --geolife.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --geolife.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --geotools-raster.coverage
 - Optional parameter to set the coverage name (default is the file name)
- --geotools-raster.crs
 - A CRS override for the provided raster file
- --geotools-raster.histogram
 - Build a histogram of samples per band on ingest for performing band equalization
 - Default: false
- --geotools-raster.mergeStrategy

- Optional parameter to choose a tile merge strategy used for mosaic.
- Default behavior will be **none**. Alternatively 'no-data' will mosaic the most recent tile over previous tiles, except where there are no data values.
- Default: none
- `--geotools-raster.nodata`
 - Optional parameter to set 'no data' values, if 1 value is giving it is applied for each band, if multiple are given then the first totalNoDataValues/totalBands are applied to the first band and so on, so each band can have multiple differing 'no data' values if needed
 - Default: []
- `--geotools-raster.pyramid`
 - Build an image pyramid on ingest for quick reduced resolution query
 - Default: false
- `--geotools-raster.separateBands`
 - Optional parameter to separate each band into its own coverage name. By default the coverage name will have '_Bn' appended to it where **n** is the band's index.
 - Default: false
- `--geotools-raster.tileSize`
 - Optional parameter to set the tile size stored (default is 256)
 - Default: 256
- `--geotools-vector.cql`
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- `--geotools-vector.data`
 - A map of date field names to the date format of the file. Use commas to separate each entry, then the first ':' character will separate the field name from the format. Use '\' to include a comma in the format. For example: "time:MM:dd:YYYY,time2:YYYY/MM/dd hh:mm:ss" configures fields 'time' and 'time2' as dates with different formats
- `--geotools-vector.type`
 - Optional parameter that specifies specific type name(s) from the source file
 - Default: []
- `--gpx.avro`
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- `--gpx.cql`

- A CQL filter, only data matching this filter will be ingested
- Default: <empty string>
- --gpx.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --jobtracker
 - Hadoop job tracker hostname and port in the format hostname:port
- --resourcemanager
 - Yarn resource manager hostname and port in the format hostname:port
- --tdrive.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --tdrive.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --tdrive.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- --twitter.avro
 - A flag to indicate whether avro feature serialization should be used
 - Default: false
- --twitter.cql
 - A CQL filter, only data matching this filter will be ingested
 - Default: <empty string>
- --twitter.typename
 - A comma-delimited set of typenames to ingest, feature types matching the specified typenames will be ingested (optional, by default all types will be ingested)
- -v, --visibility
 - The visibility of the data ingested (optional; default is 'public')

geowave ingest sparkToGW

NAME

geowave ingest sparkToGW - Ingest supported files that already exist in HDFS or S3

SYNOPSIS

geowave ingest sparkToGW [options] <input directory> <store name> <comma delimited index list>

DESCRIPTION

Ingest supported files that already exist in HDFS or S3

OPTIONS

- -x, --extension
 - individual or comma-delimited set of file extensions to accept (optional)
 - -f, --formats
 - Explicitly set the ingest formats by name (or multiple comma-delimited formats), if not set all available ingest formats will be used
 - -ho, --hosts
 - The spark driver host Default: localhost
 - -m, --master
 - The spark master designation Default: local
 - -n, --name
 - The spark application name Default: Spark Ingest
 - -c, --numcores
 - Number of cores Default: -1
 - -e, --numexecutors
 - Number of executors Default: -1
 - -v, --visibility
 - The visibility of the data ingested (optional; default is 'public')
-

Analytic Commands

Commands that run mapreduce or spark processing to enhance an existing GeoWave dataset (Required options are designated with an *)

- [DBScan](#)
- [KDE](#)
- [KMeansJump](#)
- [KMeansParallel](#)
- [KMeansSpark](#)
- [NearestNeighbor](#)
- [SQL](#)

The commands below can also be run as a yarn or hadoop api commands (i.e. mapreduce)

NOTE

For instance, if running the analytic using yarn:

```
yarn jar geowave-tools.jar analytic <algorithm> <options> <store>
```

geowave analytic dbscan

NAME

geowave analytic dbscan - Density Based Scanner

SYNOPSIS

geowave analytic dbscan [options] <storename>

DESCRIPTION

The geowave analytic dbscan operator will run a density based scanner analytic on GeoWave data

EXAMPLE

```
yarn jar geowave-tools.jar analytic dbscan -cmi 5 -cms 10 -emn 2 -emx 6 -pmd 1000 -orc 4  
-hdfs localhost:53000 -jobtracker localhost:8032 -hdfsbase /user/rwgdrummer  
--query.adapters gpxpoint my_store
```

Run through 5 max iterations (cmi), with max distance between points as 10 meters (cms), min hdfs input split is 2 (emn), max hdfs input split is 6 (emx), max search distance is 1000 meters (pmd),

reducer count is 4 (orc), the hdfs ipc port is localhost:53000 (hdfs), the yarn job tracker is at localhost:8032 (-jobtracker), the temporary files needed by this job are stored in hdfs://host:port//user/rwgdruammer (hdfsbase), and the data executed against DBSCAN is 'gpxpoint' (query.adapters). The accumulo connection parameters are loaded from my_store.

EXECUTION

DBSCAN uses GeoWaveInputFormat to load data from GeoWave into HDFS. You can use the extract query parameter to limit the records used in the analytic.

It iteratively calls Nearest Neighbor to execute a sequence of concave hulls. The hulls are saved into sequence files written to a temporary HDFS directory, and then read in again for the next DBSCAN iteration.

After completion, the data is written back from HDFS to Accumulo using a job called the "input load runner".

OPTIONS

- * -cmi, --clusteringMaxIterations
 - Maximum number of iterations when finding optimal clusters
- * -cms, --clusteringMinimumSize
 - Minimum Cluster Size
- --cdf, --commonDistanceFunctionClass
 - Distance Function Class implements *org.locationtech.geowave.analytics.distance.DistanceFn*
- * -emx, --extractMaxInputSplit
 - Maximum hdfs input split size
- * -emn, --extractMinInputSplit
 - Minimum hdfs input split size
- -eq, --extractQuery
 - Query
- -b, --globalBatchId
 - Batch ID
- -hdt, --hullDataTypeId
 - Data Type ID for a centroid item
- -hpe, --hullProjectionClass
 - Class to project on to 2D space. Implements *org.locationtech.geowave.analytics.tools.Projection*
- -ifc, --inputFormatClass
 - Input Format Class

- `-conf, --mapReduceConfigFile`
 - MapReduce Configuration
- `* -hdfsbase, --mapReduceHdfsBaseDir`
 - Fully qualified path to the base directory in hdfs
- `* -hdfs, --mapReduceHdfsHostPort`
 - HDFS hostname and port in the format hostname:port
- `-jobtracker, --mapReduceJobtrackerHostPort`
 - [REQUIRED (or resourcemanager)] Hadoop job tracker hostname and port in the format hostname:port
- `-resourcemanager, --mapReduceYarnResourceManager`
 - [REQUIRED (or jobtracker)] Yarn resource manager hostname and port in the format hostname:port
- `-ons, --outputDataNamespaceUri`
 - Output namespace for objects that will be written to GeoWave
- `-odt, --outputDataTypeId`
 - Output Data ID assigned to objects that will be written to GeoWave
- `-oop, --outputHdfsOutputPath`
 - Output HDFS File Path
- `-oid, --outputIndexId`
 - Output Index ID for objects that will be written to GeoWave
- `-ofc, --outputOutputFormat`
 - Output Format Class
- `-orc, --outputReducerCount`
 - Number of Reducers For Output
- `-pdt, --partitionDistanceThresholds`
 - Comma separated list of distance thresholds, per dimension
- `-pdu, --partitionGeometricDistanceUnit`
 - Geometric distance unit (m=meters,km=kilometers, see symbols for javax.units.BaseUnit)
- `* -pmd, --partitionMaxDistance`
 - Maximum Partition Distance
- `-pms, --partitionMaxMemberSelection`
 - Maximum number of members selected from a partition
- `-pdr, --partitionPartitionDecreaseRate`

- Rate of decrease for precision(within (0,1])
- -pp, --partitionPartitionPrecision
 - Partition Precision
- -pc, --partitionPartitionerClass
 - Index Identifier for Centroids
- -psp, --partitionSecondaryPartitionerClass
 - Perform secondary partitioning with the provided class
- * --query.adapters
 - The comma-separated list of data adapters to query; by default all adapters are used.
- --query.auth
 - The comma-separated list of authorizations used during extract; by default all authorizations are used.
- --query.index
 - The specific index to query; by default one is chosen for each adapter.

geowave analytic kde

NAME

geowave analytic kde - Kernel Density Estimate

SYNOPSIS

geowave analytic kde [options] <input storename> <output storename>

DESCRIPTION

The geowave analytic kde operator will run a Kernel Density Estimate analytic on GeoWave data

OPTIONS

- * --coverageName
 - The coverage name
- --cqlFilter
 - An optional CQL filter applied to the input data
- * --featureType
 - The name of the feature type to run a KDE on
- * --hdfsHostPort
 - The hdfs host port

- `--indexId`
 - An optional index ID to filter the input data
- `* --jobSubmissionHostPort`
 - The job submission tracker
- `* --maxLevel`
 - The max level to run a KDE at
- `—maxSplits`
 - The max partitions for the input data
- `* --minLevel`
 - The min level to run a KDE at
- `--minSplits`
 - The min partitions for the input data
- `--tileSize`
 - The tile size
 - Default: 1

geowave analytic kmeansjump

NAME

geowave analytic kmeansjump - KMeans Clustering using Jump Method

SYNOPSIS

geowave analytic kmeansjump [options] <storename>

DESCRIPTION

The geowave analytic kmeansjump operator will execute a KMeans Clustering analytic using a Jump Method

EXAMPLE

```
yarn jar geowave-tools.jar analytic kmeansjump -cmi 15 -zl 1 -emx 4000 -emn 100 -hdfsbase
/usr/rwgdrummer/temp_dir_kmeans -hdfs localhost:53000 -jobtracker localhost:8032
--query.adapters hail -jcp 3 -jrc 4,8 my_store
```

The min clustering iterations is 15 (cmi), the zoomlevel is 1 (zl), the max hdfs input split is 4000 (emx), the min hdfs input split is 100 (emn), the temporary files needed by this job are stored in hdfs:/host:port/user/rwgdrummer/temp_dir_kmeans (hdfsbase), the hdfs ipc port is localhost:53000

(hdfs), the yarn job tracker is at localhost:8032 (-jobtracker), the data executed against is 'hail' (query.adapters), the min k for kmeans parallel sampling is 3 (jcp), the comma separated range of centroids is 4,8 (jrc). The accumulo connection parameters are loaded from my_store.

EXECUTION

KMeansJump uses most of the same parameters from KMeansParallel. It tries every k value given (-jrc) to find the value with least entropy. The other value, jkp, will specify which k values should use kmeans parallel for sampling versus a single sampler (which uses a random sample). For instance, if you specify 4,8 for jrc and 6 for jkp, then k=4,5 will use the kmeansparallel sampler, while 6,7,8 will use the single sampler.

KMeansJump executes by executing several iterations, running the sampler (described above, which also calls the normal k-means algorithm to determine centroids) and then executing a KMeans distortion job, which calculates the entropy of the calculated centroids.

Look at the "EXECUTION" documentation for kmeansparallel operation for discussion of output, tolerance and performance variables.

OPTIONS

- -cce, --centroidExtractorClass
 - Centroid Exractor Class implements
org.locationtech.geowave.analytics.extract.CentroidExtractor
- -cid, --centroidIndexId
 - Index Identifier for Centroids
- -cfc, --centroidWrapperFactoryClass
 - A factory class that implements
org.locationtech.geowave.analytics.tools.AnalyticItemWrapperFactory
- -czl, --centroidZoomLevel
 - Zoom Level Number
- -cct, --clusteringConvergenceTolerance
 - Convergence Tolerance
- *-cmi, --clusteringMaxIterations
 - Maximum number of iterations when finding optimal clusters
- -crc, --clusteringMaxReducerCount
 - Maximum Clustering Reducer Count
- *-zl, --clusteringZoomLevels
 - Number of Zoom Levels to Process
- -dde, --commonDimensionExtractClass

◦ Dimension	Extractor	Class	implements
		<i>org.locationtech.geowave.analytics.extract.DimensionExtractor</i>	

- -cdf, --commonDistanceFunctionClass
 - Distance Function Class implements *org.locationtech.geowave.analytics.distance.DistanceFn*
- -ens, --extractDataNamespaceUri
 - Output Data Namespace URI
- -ede, --extractDimensionExtractClass
 - Class to extract dimensions into a simple feature output
- * -emx, --extractMaxInputSplit
 - Maximum hdfs input split size
- * -emn, --extractMinInputSplit
 - Minimum hdfs input split size
- -eot, --extractOutputDataTypeId
 - Output Data Type ID
- -eq, --extractQuery
 - Query
- -erc, --extractReducerCount
 - Number of Reducers For initial data extraction and de-duplication
- -b, --globalBatchId
 - Batch ID
- -pb, --globalParentBatchId
 - Batch ID
- -hns, --hullDataNamespaceUri
 - Data Type Namespace for a centroid item
- -hdt, --hullDataTypeId
 - Data Type ID for a centroid item
- -hid, --hullIndexId
 - Index Identifier for Centroids
- -hpe, --hullProjectionClass
 - Class to project on to 2D space. Implements *org.locationtech.geowave.analytics.tools.Projection*
- -hrc, --hullReducerCount
 - Centroid Reducer Count
- -hfc, --hullWrapperFactoryClass

- Class to create analytic item to capture hulls. Implements *org.locationtech.geowave.analytics.tools.AnalyticItemWrapperFactory*
- -ifc, --inputFormatClass
 - Input Format Class
- * -jkp, --jumpKplusplusMin
 - The minimum k when K means ++ takes over sampling.
- * -jrc, --jumpRangeOfCentroids
 - Comma-separated range of centroids (e.g. 2,100)
- -conf, --mapReduceConfigFile
 - MapReduce Configuration
- * -hdfsbase, --mapReduceHdfsBaseDir
 - Fully qualified path to the base directory in hdfs
- * -hdfs, --mapReduceHdfsHostPort
 - HDFS hostname and port in the format hostname:port
- -jobtracker, --mapReduceJobtrackerHostPort
 - [REQUIRED (or resourcemanager)] Hadoop job tracker hostname and port in the format hostname:port
- -resourcemanager, --mapReduceYarnResourceManager
 - [REQUIRED (or jobtracker)] Yarn resource manager hostname and port in the format hostname:port
- -ofc, --outputOutputFormat
 - Output Format Class
- -orc, --outputReducerCount
 - Number of Reducers For Output
- * --query.adapters
 - The comma-separated list of data adapters to query; by default all adapters are used.
- --query.auth
 - The comma-separated list of authorizations used during extract; by default all authorizations are used.
- --query.index
 - The specific index to query; by default one is chosen for each adapter.

geowave analytic kmeansparallel

NAME

geowave analytic kmeansparallel - KMeans Parallel Clustering

SYNOPSIS

geowave analytic kmeansparallel [options] <storename>

DESCRIPTION

The geowave analytic kmeansparallel operator will execute a KMeans Parallel Clustering analytic

EXAMPLE

```
yarn jar geowave-tools.jar analytic kmeansparallel -cmi 15 -zl 1 -emx 4000 -emn 100  
-hdfsbase /usr/rwgdrummer/temp_dir_kmeans  
-hdfs localhost:53000 -jobtracker localhost:8032 --query.adapters hail -sms 4 -sxs 8 -ssi  
10 my_store
```

The min clustering iterations is 15 (cmi), the zoomlevel is 1 (zl), the max hdfs input split is 4000 (emx), the min hdfs input split is 100 (emn), the temporary files needed by this job are stored in hdfs://host:port/user/rwgdrummer/temp_dir_kmeans (hdfsbase), the hdfs ipc port is localhost:53000 (hdfs), the yarn job tracker is at localhost:8032 (-jobtracker), the data executed against is 'hail' (query.adapters), the min sample size is 4 (sms, which is kmin), the max sample size is 8 (which is kmax), the minimum number of sampling iterations is 10. The accumulo connection parameters are loaded from my_store.

EXECUTION

KMeansParallel tries to identify the optimal k (sms, smx) for a set of zoom levels (1 → zl). When zoom level is 1, it will perform a normal kmeans and find k clusters. If zoomlevel is 2 or higher, it will take each cluster found, and then try to create sub-clusters (bounded by that cluster), identifying a new optimal k for that sub-cluster. As such, without powerful infrastructure, this approach could take a significant amount of time to complete with zoom levels higher than 1.

KMeansParallel executes by first executing an extraction and de-duplication on data received via GeoWaveInputFormat. The data is copied to HDFS for faster processing. The K-Sampler job is used to pick sample centroid points. These centroids are then assigned a cost, and then weak centroids are stripped before the K-Sampler is executed again. This process iterates several times, before the best centroid locations are found, which are fed into the real K-Means algorithm as initial guesses. K-Means iterates until the tolerance is reached (-cct, which defaults to 0.0001) or the max iterations is met (-cmi).

After execution, KMeansParallel writes the centroids to an output data type (-eot, defaults to centroid), and then creates an informational set of convex hulls which you can plot in GeoServer to visually identify cluster groups (-hdt, defaults to convex_hull).

For tuning performance, you can set the number of reducers used in each step. Extraction/Dedupe reducer count is -crc, Clustering reducer count is -erc, Convex Hull reducer count is -hrc, and Output

reducer count is -orc).

If you would like to run the algorithm multiple times, it may be useful to set the batch id (-b), which can be used to distinguish between multiple batches (runs).

OPTIONS

- -cce, --centroidExtractorClass
 - Centroid Extractor Class implements
org.locationtech.geowave.analytics.extract.CentroidExtractor
- -cid, --centroidIndexId
 - Index Identifier for Centroids
- -cfc, --centroidWrapperFactoryClass
 - A factory class that implements
org.locationtech.geowave.analytics.tools.AnalyticItemWrapperFactory
- -czl, --centroidZoomLevel
 - Zoom Level Number
- -cct, --clusteringConvergenceTolerance
 - Convergence Tolerance
- * -cmi, --clusteringMaxIterations
 - Maximum number of iterations when finding optimal clusters
- -crc, --clusteringMaxReducerCount
 - Maximum Clustering Reducer Count
- * -zl, --clusteringZoomLevels
 - Number of Zoom Levels to Process
- -dde, --commonDimensionExtractClass
 - Dimension Extractor Class implements
org.locationtech.geowave.analytics.extract.DimensionExtractor
- -cdf, --commonDistanceFunctionClass
 - Distance Function Class implements *org.locationtech.geowave.analytics.distance.DistanceFn*
- -ens, --extractDataNamespaceUri
 - Output Data Namespace URI
- -ede, --extractDimensionExtractClass
 - Class to extract dimensions into a simple feature output
- * -emx, --extractMaxInputSplit
 - Maximum hdfs input split size

- * -emn, --extractMinInputSplit
 - Minimum hdfs input split size
- -eot, --extractOutputDataTypeId
 - Output Data Type ID
- -eq, --extractQuery
 - Query
- -erc, --extractReducerCount
 - Number of Reducers For initial data extraction and de-duplication
- -b, --globalBatchId
 - Batch ID
- -pb, --globalParentBatchId
 - Batch ID
- -hns, --hullDataNamespaceUri
 - Data Type Namespace for a centroid item
- -hdt, --hullDataTypeId
 - Data Type ID for a centroid item
- -hid, --hullIndexId
 - Index Identifier for Centroids
- -hpe, --hullProjectionClass
 - Class to project on to 2D space. Implements *org.locationtech.geowave.analytics.tools.Projection*
- -hrc, --hullReducerCount
 - Centroid Reducer Count
- -hfc, --hullWrapperFactoryClass
 - Class to create analytic item to capture hulls. Implements *org.locationtech.geowave.analytics.tools.AnalyticItemWrapperFactory*
- -ifc, --inputFormatClass
 - Input Format Class
- -conf, --mapReduceConfigFile
 - MapReduce Configuration
- * -hdfsbase, --mapReduceHdfsBaseDir
 - Fully qualified path to the base directory in hdfs
- * -hdfs, --mapReduceHdfsHostPort
 - HDFS hostname and port in the format hostname:port

- `-jobtracker, --mapReduceJobtrackerHostPort`
 - [REQUIRED (or resourceman)] Hadoop job tracker hostname and port in the format hostname:port
- `-resourceman, --mapReduceYarnResourceManager`
 - [REQUIRED (or jobtracker)] Yarn resource manager hostname and port in the format hostname:port
- `-ofc, --outputOutputFormat`
 - Output Format Class
- `-orc, --outputReducerCount`
 - Number of Reducers For Output
- `* --query.adapters`
 - The comma-separated list of data adapters to query; by default all adapters are used.
- `--query.auth`
 - The comma-separated list of authorizations used during extract; by default all authorizations are used.
- `--query.index`
 - The specific index to query; by default one is chosen for each adapter.
- `* -sxs, --sampleMaxSampleSize`
 - Max Sample Size
- `* -sms, --sampleMinSampleSize`
 - Minimum Sample Size
- `* -ssi, --sampleSampleIterations`
 - Minimum number of sample iterations

geowave analytic nn

NAME

geowave analytic nn - Nearest Neighbors

SYNOPSIS

geowave analytic nn [options] <storename>

DESCRIPTION

The geowave analytic nn operator will execute a Nearest Neighbors analytic. Analytic 'nn' is similar to DBScan, with less arguments. 'nn' just dumps all near neighbors for every feature id in a list of pairs. Most developers will want to extend the framework to add their own extensions.

EXAMPLE

```
yarn jar geowave-tools.jar analytic nn -emn 2 -emx 6 -pmd 1000 -oop /user/rwgdrummer_out
-orc 4 -hdfs localhost:53000 -jobtracker localhost:8032 -hdfsbase /user/rwgdrummer
--query.adapters gpxpoint my_store
```

The min hdfs input split is 2 (emn), max hdfs input split is 6 (emx), max search distance is 1000 meters (pmd), the sequence file output directory is *hdfs://host:port/user/rwgdrummer_out*, reducer count is 4 (orc), the hdfs ipc port is *localhost:53000* (hdfs), the yarn job tracker is at *localhost:8032* (-jobtracker), the temporary files needed by this job are stored in *hdfs://host:port/user/rwgdrummer* (hdfsbase), and the data executed against is 'gpxpoint' (query.adapters). The accumulo connection parameters are loaded from *my_store*.

EXECUTION

To execute nearest neighbor search in GeoWave, we use the concept of a "partitioner" to partition all data on the hilbert curve into square segments for the purposes of parallelizing the search.

The default partitioner will multiply this value by 2 and use that for the actual partition sizes. Because of this, the terminology is a bit confusing, but the "pmd" option is actually the most important variable here, describing the max distance for a point to be considered a neighbor to another point.

OPTIONS

- -cdf, --commonDistanceFunctionClass
 - Distance Function Class implements *org.locationtech.geowave.analytics.distance.DistanceFn*
- * -emx, --extractMaxInputSplit
 - Maximum hdfs input split size
- * -emn, --extractMinInputSplit
 - Minimum hdfs input split size
- -eq, --extractQuery
 - Query
- -ifc, --inputFormatClass
 - Input Format Class
- -conf, --mapReduceConfigFile
 - MapReduce Configuration
- * -hdfsbase, --mapReduceHdfsBaseDir
 - Fully qualified path to the base directory in hdfs
- * -hdfs, --mapReduceHdfsHostPort

- HDFS hostname and port in the format hostname:port
- -jobtracker, --mapReduceJobtrackerHostPort
 - [REQUIRED (or resourceman)] Hadoop job tracker hostname and port in the format hostname:port
- -resourceman, --mapReduceYarnResourceManager
 - [REQUIRED (or jobtracker)] Yarn resource manager hostname and port in the format hostname:port
- * -oop, --outputHdfsOutputPath
 - Output HDFS File Path
- -ofc, --outputOutputFormat
 - Output Format Class
- -orc, --outputReducerCount
 - Number of Reducers For Output
- -pdt, --partitionDistanceThresholds
 - Comma separated list of distance thresholds, per dimension
- -pdu, --partitionGeometricDistanceUnit
 - Geometric distance unit (m=meters,km=kilometers, see symbols for javax.units.BaseUnit)
- * -pmd, --partitionMaxDistance
 - Maximum Partition Distance
- -pms, --partitionMaxMemberSelection
 - Maximum number of members selected from a partition
- -pp, --partitionPartitionPrecision
 - Partition Precision
- -pc, --partitionPartitionerClass
 - Index Identifier for Centroids
- -psp, --partitionSecondaryPartitionerClass
 - Perform secondary partitioning with the provided class
- * --query.adapters
 - The comma-separated list of data adapters to query; by default all adapters are used.
- --query.auth
 - The comma-separated list of authorizations used during extract; by default all authorizations are used.
- --query.index

- The specific index to query; by default one is chosen for each adapter.

geowave analytic sql

NAME

geowave analytic sql - SparkSQL queries

SYNOPSIS

geowave analytic sql [options] <sql query> - e.g. 'select * from storename[|adaptername] where condition...'

DESCRIPTION

The geowave analytic sql operator will execute a SparkSQL query

OPTIONS

- --csv
 - The output CSV file name
 - --out
 - The output datastore name
 - --outtype
 - The output feature type (adapter) name
 - -s, --show
 - Number of result rows to display
 - Default: 20
-

Accumulo Commands

Utility operations to set accumulo splits and run a test server (Required options are designated with an *)

- [PreSplitPartitionId](#)
- [RunServer](#)
- [SplitEqualInterval](#)
- [SplitNumRecords](#)
- [SplitQuantile](#)

geowave util accumulo dbscan

NAME

geowave util accumulo presplitpartitionid - Pre-split Accumulo table by providing the number of partition IDs

SYNOPSIS

geowave util accumulo presplitpartitionid [options] <storename>

DESCRIPTION

This command will pre-split an accumulo table by providing the number of partition IDs

OPTIONS

- --indexId
 - The geowave index ID (optional; default is all indices)
- --num
 - The number of partitions (or entries)
 - Default: 0

geowave util accumulo runserver

NAME

geowave util accumulo runserver - Runs a standalone mini Accumulo server for test and debug with GeoWave

SYNOPSIS

geowave util accumulo runserver [options]

DESCRIPTION

This command will run a standalone mini single-node accumulo server, which can be used locally for testing and debugging GeoWave, without needing to stand up an entire cluster.

OPTIONS

There are currently no options available for this command

geowave util accumulo splitequalinterval

NAME

geowave util accumulo splitequalinterval - Set Accumulo splits by providing the number of partitions based on an equal interval strategy

SYNOPSIS

geowave util accumulo splitequalinterval [options] <storename>

DESCRIPTION

This command will allow a user to set the accumulated splits through providing the number of partitions based on an equal interval strategy.

OPTIONS

- --indexId
 - The geowave index ID (optional; default is all indices)
- --num
 - The number of partitions (or entries)
 - Default: 0

geowave util accumulo splitnumrecords

NAME

geowave util accumulo splitnumrecords - Set Accumulo splits by providing the number of entries per split

SYNOPSIS

geowave util accumulo splitnumrecords [options] <storename>

DESCRIPTION

This command will set the accumulo datastore splits by providing the number of entries per split.

OPTIONS

- --indexId
 - The geowave index ID (optional; default is all indices)
- --num
 - The number of partitions (or entries)
 - Default: 0

geowave util accumulo splitquantile

NAME

geowave util accumulo splitquantile - Set Accumulo splits by providing the number of partitions based on a quantile distribution strategy

SYNOPSIS

geowave util accumulo splitquantile [options] <storename>

DESCRIPTION

This command will allow a user to set the accumulo datastore splits by providing the number of partitions based on a quantile distribution strategy.

OPTIONS

- --indexId
 - The geowave index ID (optional; default is all indices)
 - --num
 - The number of partitions (or entries)
 - Default: 0
-

GeoServer Commands

Commands that manage geoserver data stores and layers (Required options are designated with an *)

- [AddCS](#)
- [AddCV](#)
- [AddDS](#)
- [AddFL](#)
- [AddLayer](#)
- [AddStyle](#)
- [AddWS](#)
- [GetCS](#)
- [GetCV](#)
- [GetDS](#)
- [GetFL](#)
- [GetSA](#)
- [GetStyle](#)
- [ListCS](#)
- [ListCV](#)
- [ListDS](#)
- [ListFL](#)
- [ListStyles](#)
- [ListWS](#)
- [RmCS](#)
- [RmCV](#)
- [RmDS](#)
- [RmFL](#)
- [RmStyle](#)
- [RmWS](#)
- [SetLS](#)

geowave gs cs add

NAME

geowave gs cs add - Add a GeoServer coverage store

SYNOPSIS

geowave gs cs add [options] <GeoWave store name>

DESCRIPTION

This command will add a GeoServer coverage store

OPTIONS

- -cs, --coverageStore
 - <coverage store name>
- -histo, --equalizeHistogramOverride
 - This parameter will override the behavior to always perform histogram equalization if a histogram exists.
 - Valid values are true and false.
- -interp, --interpolationOverride
 - This will override the default interpolation stored for each layer.
 - Valid values are 0, 1, 2, 3 for NearestNeighbor, Bilinear, Bicubic, and Bicubic (polynomial variant) respectively.
- -scale, --scaleTo8Bit
 - By default, integer values will automatically be scaled to 8-bit and floating point values will not. This can be overridden setting this value to true or false.
- -ws, --workspace
 - <workspace name>

geowave gs cv add

NAME

geowave gs cv add - Add a GeoServer coverage

SYNOPSIS

geowave gs cv add [options] <coverage name>

DESCRIPTION

This command will add a GeoServer coverage

OPTIONS

- * -cs, --cvgstore
 - <coverage store name>
- -ws, --workspace
 - <workspace name>

geowave gs ds add

NAME

geowave gs ds add - Add a GeoServer datastore

SYNOPSIS

geowave gs ds add [options] <GeoWave store name>

DESCRIPTION

This command will add a GeoServer datastore

OPTIONS

- -ds, --datastore
 - <datastore name>
- -ws, --workspace
 - <workspace name>

geowave gs fl add

NAME

geowave gs fl add - Add a GeoServer feature layer

SYNOPSIS

geowave gs fl add [options] <layer name>

DESCRIPTION

This command will add a GeoServer feature layer

OPTIONS

- * -ds, --datastore
 - <datastore name>
- -ws, --workspace

- <workspace name>

geowave gs layer add

NAME

geowave gs layer add - Add a GeoServer layer from the given GeoWave store

SYNOPSIS

geowave gs layer add [options] <GeoWave store name>

DESCRIPTION

This command will add a GeoServer layer from the given GeoWave store

OPTIONS

- -id, --adapterId
 - select just <adapter id> from the store
- -a, --add
 - For multiple layers, add (all | raster | vector)
 - Possible Values: [ALL, RASTER, VECTOR]
- -sld, --setStyle
 - <default style sld>
- -ws, --workspace
 - <workspace name>

geowave gs style add

NAME

geowave gs style add - Add a GeoServer style

SYNOPSIS

geowave gs style add [options] <GeoWave style name>

DESCRIPTION

This command will add a GeoServer style

OPTIONS

- * -sld, --stylesld

- <style sld file>

geowave gs ws add

NAME

geowave gs ws add - Add GeoServer workspace

SYNOPSIS

geowave gs ws add [options] <workspace name>

DESCRIPTION

This command will add a GeoServer workspace

OPTIONS

There are currently no options available for this command

geowave gs cs get

NAME

geowave gs cs get - Get GeoServer CoverageStore info

SYNOPSIS

geowave gs cs get [options] <coverage store name>

DESCRIPTION

This command will return GeoServer CoverageStore info

OPTIONS

- -ws, --workspace
 - <workspace name>

geowave gs cv get

NAME

geowave gs cv get - Get a GeoServer coverage's info

SYNOPSIS

geowave gs cv get [options] <coverage name>

DESCRIPTION

This command will return a GeoServer coverage's info

OPTIONS

- -cs, --coverageStore
 - <coverage store name>
- -ws, --workspace
 - <workspace name>

geowave gs ds get

NAME

geowave gs ds get - Get GeoServer DataStore info

SYNOPSIS

geowave gs ds get [options] <datastore name>

DESCRIPTION

This command will return a GeoServer DataStore info

OPTIONS

- -ws, --workspace
 - <workspace name>

geowave gs fl get

NAME

geowave gs fl get - Get GeoServer feature layer info

SYNOPSIS

geowave gs fl get [options] <layer name>

DESCRIPTION

This command will return GeoServer feature layer info

OPTIONS

There are currently no options available for this command

geowave gs sa get

NAME

geowave gs sa get - Get GeoWave store adapters

SYNOPSIS

geowave gs sa get [options] <store name>

DESCRIPTION

This command will return GeoWave store adapters

OPTIONS

There are currently no options available for this command

geowave gs style get

NAME

geowave gs style get - Get GeoServer Style info

SYNOPSIS

geowave gs style get [options] <style name>

DESCRIPTION

This command will return GeoServer Style info

OPTIONS

There are currently no options available for this command

geowave gs cs list

NAME

geowave gs cs list - List GeoServer coverage stores

SYNOPSIS

geowave gs cs list [options]

DESCRIPTION

This command will list all GeoServer coverage stores

OPTIONS

- -ws, --workspace
 - <workspace name>

geowave gs cv list

NAME

geowave gs cv list - List GeoServer Coverages

SYNOPSIS

geowave gs cv list [options] <coverage store name>

DESCRIPTION

This command will list all GeoServer Coverages

OPTIONS

- -ws, --workspace
 - <workspace name>

geowave gs ds list

NAME

geowave gs ds list - List GeoServer datastores

SYNOPSIS

geowave gs ds list [options]

DESCRIPTION

This command will list all GeoServer datastores

OPTIONS

- -ws, --workspace
 - <workspace name>

geowave gs fl list

NAME

geowave gs fl list - List GeoServer feature layers

SYNOPSIS

geowave gs fl list [options]

DESCRIPTION

This command will list all GeoServer feature layers

OPTIONS

- -ds, --datastore
 - Datastore Name
- -g, --geowaveOnly
 - Show only GeoWave feature layers (default: false)
 - Default: false
- -ws, --workspace
 - Workspace Name

geowave gs style list

NAME

geowave gs style list - List GeoServer styles

SYNOPSIS

geowave gs style list [options]

DESCRIPTION

This command will list all GeoServer styles

OPTIONS

There are currently no options available for this command

geowave gs ws list

NAME

geowave gs ws list - List GeoServer workspaces

SYNOPSIS

geowave gs ws list [options]

DESCRIPTION

This command will list all GeoServer workspaces

OPTIONS

There are currently no options available for this command

geowave gs cs rm

NAME

geowave gs cs rm - Remove GeoServer Coverage Store

SYNOPSIS

geowave gs cs rm [options] <coverage store name>

DESCRIPTION

This command will remove a GeoServer Coverage Store

OPTIONS

- -ws, --workspace
 - Workspace Name

geowave gs cv rm

NAME

geowave gs cv rm - Remove a GeoServer coverage

SYNOPSIS

geowave gs cv rm [options] <coverage name>

DESCRIPTION

This command will remove a GeoServer coverage

OPTIONS

- * -cs, --cvgstore
 - <coverage store name>
- -ws, --workspace
 - <workspace name>

geowave gs ds rm

NAME

geowave gs ds rm - Remove GeoServer DataStore

SYNOPSIS

geowave gs ds rm [options] <datastore name>

DESCRIPTION

This command will remove a GeoServer DataStore

OPTIONS

- -ws, --workspace
 - <workspace name>

geowave gs fl rm

NAME

geowave gs fl rm - Remove GeoServer feature Layer

SYNOPSIS

geowave gs fl rm [options] <layer name>

DESCRIPTION

This command will remove a GeoServer feature Layer

OPTIONS

There are currently no options available for this command

geowave gs style rm

NAME

geowave gs style rm - Remove GeoServer Style

SYNOPSIS

geowave gs style rm [options] <style name>

DESCRIPTION

This command will remove a GeoServer Style

OPTIONS

There are currently no options available for this command

geowave gs ws rm

NAME

geowave gs ws rm - Remove GeoServer workspace

SYNOPSIS

geowave gs ws rm [options] <workspace name>

DESCRIPTION

This command will remove a GeoServer workspace

OPTIONS

There are currently no options available for this command

geowave gs style set

NAME

geowave gs style set - Set GeoServer Layer Style

SYNOPSIS

geowave gs style set [options] <layer name>

DESCRIPTION

This command will set a GeoServer layer style

OPTIONS

- * -sn, --styleName
 - <style name>
-

Landsat8 Commands

Operations to analyze, download, and ingest Landsat 8 imagery publicly available on AWS (Required options are designated with an *)

- [Analyze](#)
- [Download](#)
- [Ingest](#)
- [IngestRaster](#)
- [IngestVector](#)

geowave util landsat analyze

NAME

geowave util landsat analyze - Print out basic aggregate statistics for available Landsat 8 imagery

SYNOPSIS

geowave util landsat analyze [options]

DESCRIPTION

This command will print out basic aggregate statistics that are for available Landsat 8 imagery

OPTIONS

- `--cql`
 - An optional CQL expression to filter the ingested imagery. The feature type for the expression has the following attributes: shape (Geometry), acquisitionDate (Date), cloudCover (double), processingLevel (String), path (int), row (int) and the feature ID is entityId for the scene. Additionally attributes of the individuals band can be used such as band (String), sizeMB (double), and bandDownloadUrl (String)
 - Default: <empty string>
- `--nbestbands`
 - An option to identify and only use a set number of bands with the best cloud cover
 - Default: 0
- `--nbestperspatial`
 - A boolean flag, when applied with `--nbestscenes` or `--nbestbands` will aggregate scenes and/or bands by path/row
 - Default: false

- `--nbestscenes`
 - An option to identify and only use a set number of scenes with the best cloud cover
 - Default: 0
- `--sinceleastrun`
 - An option to check the scenes list from the workspace and if it exists, to only ingest data since the last scene.
 - Default: false
- `--usecachedscenes`
 - An option to run against the existing scenes catalog in the workspace directory if it exists.
 - Default: false
- `-ws, --workspaceDir`
 - A local directory to write temporary files needed for landsat 8 ingest.
 - Default is `<TEMP_DIR>/landsat8`
 - Default: landsat8

geowave util landsat download

NAME

geowave util landsat download - Download Landsat 8 imagery to a local directory

SYNOPSIS

geowave util landsat download [options]

DESCRIPTION

This command will download the Landsat 8 imagery to a local directory

OPTIONS

- `--cql`
 - An optional CQL expression to filter the ingested imagery. The feature type for the expression has the following attributes: shape (Geometry), acquisitionDate (Date), cloudCover (double), processingLevel (String), path (int), row (int) and the feature ID is entityId for the scene. Additionally attributes of the individuals band can be used such as band (String), sizeMB (double), and bandDownloadUrl (String)
 - Default: `<empty string>`
- `--nbestbands`
 - An option to identify and only use a set number of bands with the best cloud cover

- Default: 0
- --nbestperspatial
 - A boolean flag, when applied with --nbestscenes or --nbestbands will aggregate scenes and/or bands by path/row
 - Default: false
- --nbestscenes
 - An option to identify and only use a set number of scenes with the best cloud cover
 - Default: 0
- --sincelastrun
 - An option to check the scenes list from the workspace and if it exists, to only ingest data since the last scene.
 - Default: false
- --usecachedscenes
 - An option to run against the existing scenes catalog in the workspace directory if it exists.
 - Default: false
- -ws, --workspaceDir
 - A local directory to write temporary files needed for landsat 8 ingest.
 - Default is <TEMP_DIR>/landsat8
 - Default: landsat8

geowave util landsat ingest

NAME

geowave util landsat ingest - Ingest routine for locally downloading Landsat 8 imagery and ingesting it into GeoWave's raster store and in parallel ingesting the scene metadata into GeoWave's vector store. These two stores can actually be the same or they can be different.

SYNOPSIS

geowave util landsat ingest [options] <store name> <comma delimited index list>

DESCRIPTION

This command provides a standard ingest routine for downloading Landsat 8 imagery and ingesting it into GeoWave's raster store. In parallel, it allows for ingesting the scene metadata into GeoWave's vector store.

OPTIONS

- `--converter`
 - Prior to ingesting an image, this converter will be used to massage the data. The default is not to convert the data.
- `--coverage`
 - The name to give to each unique coverage. Freemarker templating can be used for variable substitution based on the same attributes used for filtering. The default coverage name is `'${entityId}_${band}'`. If `${band}` is unused in the coverage name, all bands will be merged together into the same coverage.
 - Default: `${entityId}_${band}`
- `--cql`
 - An optional CQL expression to filter the ingested imagery. The feature type for the expression has the following attributes: `shape` (Geometry), `acquisitionDate` (Date), `cloudCover` (double), `processingLevel` (String), `path` (int), `row` (int) and the feature ID is `entityId` for the scene. Additionally attributes of the individuals band can be used such as `band` (String), `sizeMB` (double), and `bandDownloadUrl` (String)
 - Default: `<empty string>`
- `--crop`
 - Use the spatial constraint provided in CQL to crop the image. If no spatial constraint is provided, this will not have an effect.
 - Default: `false`
- `--histogram`
 - An option to store the histogram of the values of the coverage so that histogram equalization will be performed
 - Default: `false`
- `--nbestbands`
 - An option to identify and only use a set number of bands with the best cloud cover
 - Default: `0`
- `--nbestperspatial`
 - A boolean flag, when applied with `--nbestscenes` or `--nbestbands` will aggregate scenes and/or bands by path/row
 - Default: `false`
- `--nbestscenes`
 - An option to identify and only use a set number of scenes with the best cloud cover
 - Default: `0`
- `--overwrite`

- An option to overwrite images that are ingested in the local workspace directory. By default it will keep an existing image rather than downloading it again.
- Default: false
- --pyramid
 - An option to store an image pyramid for the coverage
 - Default: false
- --retainimages
 - An option to keep the images that are ingested in the local workspace directory. By default it will delete the local file after it is ingested successfully.
 - Default: false
- --sincelastrun
 - An option to check the scenes list from the workspace and if it exists, to only ingest data since the last scene.
 - Default: false
- --skipMerge
 - By default the ingest will automerge overlapping tiles as a post-processing optimization step for efficient retrieval, but this will skip the merge process
 - Default: false
- --subsample
 - Subsample the image prior to ingest by the scale factor provided. The scale factor should be an integer value greater than 1.
 - Default: 1
- --tilesize
 - The option to set the pixel size for each tile stored in GeoWave.
 - Default: 512
- --usecachedscenes
 - An option to run against the existing scenes catalog in the workspace directory if it exists.
 - Default: false
- --vectorindex
 - By ingesting as both vectors and rasters you may want each indexed differently. This will override the index used for vector output.
- --vectorstore
 - By ingesting as both vectors and rasters you may want to ingest into different stores. This will override the store for vector output.

- `-ws, --workspaceDir`
 - A local directory to write temporary files needed for landsat 8 ingest. Default is `<TEMP_DIR>/landsat8`
 - Default: `landsat8`

geowave util landsat ingestraster

NAME

`geowave util landsat ingestraster` - Ingest routine for locally downloading Landsat 8 imagery and ingesting it into GeoWave

SYNOPSIS

`geowave util landsat analyze [options]`

DESCRIPTION

This command provides an ingest routine for locally downloading Landsat 8 imagery and ingesting it into GeoWave.

OPTIONS

- `--converter`
 - Prior to ingesting an image, this converter will be used to massage the data. The default is not to convert the data.
- `--coverage`
 - The name to give to each unique coverage. Freemarker templating can be used for variable substitution based on the same attributes used for filtering. The default coverage name is `'${entityId}_${band}'`. If `${band}` is unused in the coverage name, all bands will be merged together into the same coverage.
 - Default: `${entityId}_${band}`
- `--cql`
 - An optional CQL expression to filter the ingested imagery. The feature type for the expression has the following attributes: `shape` (Geometry), `acquisitionDate` (Date), `cloudCover` (double), `processingLevel` (String), `path` (int), `row` (int) and the feature ID is `entityId` for the scene. Additionally attributes of the individuals band can be used such as `band` (String), `sizeMB` (double), and `bandDownloadUrl` (String)
 - Default: `<empty string>`
- `--crop`
 - Use the spatial constraint provided in CQL to crop the image. If no spatial constraint is provided, this will not have an effect.

- Default: false
- --histogram
 - An option to store the histogram of the values of the coverage so that histogram equalization will be performed
 - Default: false
- --nbestbands
 - An option to identify and only use a set number of bands with the best cloud cover
 - Default: 0
- --nbestperspatial
 - A boolean flag, when applied with --nbestscenes or --nbestbands will aggregate scenes and/or bands by path/row
 - Default: false
- --nbestscenes
 - An option to identify and only use a set number of scenes with the best cloud cover
 - Default: 0
- --overwrite
 - An option to overwrite images that are ingested in the local workspace directory. By default it will keep an existing image rather than downloading it again.
 - Default: false
- --pyramid
 - An option to store an image pyramid for the coverage
 - Default: false
- --retainimages
 - An option to keep the images that are ingested in the local workspace directory. By default it will delete the local file after it is ingested successfully.
 - Default: false
- --sincelastrun
 - An option to check the scenes list from the workspace and if it exists, to only ingest data since the last scene.
 - Default: false
- --skipMerge
 - By default the ingest will automerge overlapping tiles as a post-processing optimization step for efficient retrieval, but this will skip the merge process
 - Default: false

- --subsample
 - Subsample the image prior to ingest by the scale factor provided. The scale factor should be an integer value greater than 1.
 - Default: 1
- --tilesize
 - The option to set the pixel size for each tile stored in GeoWave. The default is 512
 - Default: 512
- --usecachedscenes
 - An option to run against the existing scenes catalog in the workspace directory if it exists.
 - Default: false
- -ws, --workspaceDir
 - A local directory to write temporary files needed for landsat 8 ingest. Default is <TEMP_DIR>/landsat8
 - Default: landsat8

geowave util landsat ingestvector

NAME

geowave util landsat ingestvector - Ingest routine for searching landsat scenes that match certain criteria and ingesting the scene and band metadata into GeoWave's vector store.

SYNOPSIS

geowave util landsat ingestvector [options] <store name> <comma delimited index list>

DESCRIPTION

This command provides an ingest routine for searching landsat scenes that match certain criteria, and ingesting the scene and band metadata into GeoWave's vector store.

OPTIONS

- --cql
 - An optional CQL expression to filter the ingested imagery. The feature type for the expression has the following attributes: shape (Geometry), acquisitionDate (Date), cloudCover (double), processingLevel (String), path (int), row (int) and the feature ID is entityId for the scene. Additionally attributes of the individuals band can be used such as band (String), sizeMB (double), and bandDownloadUrl (String)
 - Default: <empty string>
- --nbestbands

- An option to identify and only use a set number of bands with the best cloud cover
 - Default: 0
 - --nbestperspatial
 - A boolean flag, when applied with --nbestscenes or --nbestbands will aggregate scenes and/or bands by path/row
 - Default: false
 - --nbestscenes
 - An option to identify and only use a set number of scenes with the best cloud cover
 - Default: 0
 - --sincelastrun
 - An option to check the scenes list from the workspace and if it exists, to only ingest data since the last scene.
 - Default: false
 - --usecachedscenes
 - An option to run against the existing scenes catalog in the workspace directory if it exists.
 - Default: false
 - -ws, --workspaceDir
 - A local directory to write temporary files needed for landsat 8 ingest. Default is <TEMP_DIR>/landsat8
 - Default: landsat8
-

OSM Commands

Operations to ingest Open Street Map (OSM) nodes, ways and relations to GeoWave (Required options are designated with an *)

- [Ingest](#)
- [Stage](#)

geowave util osm ingest

NAME

geowave util osm ingest - Ingest and convert OSM data from HDFS to GeoWave

SYNOPSIS

geowave util osm ingest [options] <hdfs host:port> <path to base directory to read from> <store name>

DESCRIPTION

This command will ingest and convert OSM data from HDFS to GeoWave.

OPTIONS

- -jn, --jobName
 - Name of mapreduce job
 - Default: Ingest (mcarrier)
- -m, --mappingFile
 - Mapping file, imposm3 form
- --table
 - OSM Table name in GeoWave
 - Default: OSM
- * -t, --type
 - Mapper type - one of node, way, or relation
- -v, --visibility
 - The visibility of the data ingested (optional; default is 'public')

geowave util osm stage

NAME

geowave util osm stage - Stage OSM data to HDFS

SYNOPSIS

`geowave util osm stage [options] <file or directory> <hdfs host:port> <path to base directory to write to>`

DESCRIPTION

This command will stage OSM data from a local directory and write it to HDFS

OPTIONS

- `--extension`
 - PBF File extension
 - Default: `.pbf`
-

Python Commands

Operations to ingest Open Street Map (OSM) nodes, ways and relations to GeoWave (Required options are designated with an *)

- [RunGateway](#)

geowave util python rungateway

NAME

geowave util python rungateway - Runs a Py4J java gateway

SYNOPSIS

geowave util python rungateway [options]

DESCRIPTION

The geowave util python rungateway operator will start the Py4J java gateway required by pygw.

OPTIONS

There are currently no options for this command

Raster Commands

Operations to perform transformations on raster data in GeoWave (Required options are designated with an *)

- [ResizeMR](#)
- [ResizeSpark](#)
- [InstallGDAL](#)

geowave raster resizemr

NAME

geowave raster resizemr - Resize Raster Tiles using MapReduce

SYNOPSIS

geowave raster resizemr [options] <input store name> <output store name>

DESCRIPTION

This command will resize raster tiles that are stored in a GeoWave datastore using MapReduce, and write the resized tiles to a new output store.

OPTIONS

- * --hdfsHostPort
 - The hdfs host port
- --indexName
 - The index that the input raster is stored in
- * --inputCoverageName
 - The name of the input raster coverage
- * --jobSubmissionHostPort
 - The job submission tracker
- --maxSplits
 - The max partitions for the input data
- --minSplits
 - The min partitions for the input data
- * --outputCoverageName
 - The out output raster coverage name

- * --outputTileSize
 - The tile size to output

geowave raster resizespark

NAME

geowave raster resizespark - Resize Raster Tiles using Spark

SYNOPSIS

geowave raster resizespark [options] <input store name> <output store name>

DESCRIPTION

This command will resize raster tiles that are stored in a GeoWave datastore using Spark, and write the resized tiles to a new output store.

OPTIONS

- -ho, --host
 - The spark driver host
- --indexName
 - The index that the input raster is stored in
- * --inputCoverageName
 - The name of the input raster coverage
- -m, --master
 - The spark master designation
- --maxSplits
 - The max partitions for the input data
- --minSplits
 - The min partitions for the input data
- * -n, --name
 - The spark application name
- * --outputCoverageName
 - The out output raster coverage name
- * --outputTileSize
 - The tile size to output

geowave raster installgdal

NAME

geowave raster installgdal - Install GDAL by downloading native libraries

SYNOPSIS

geowave raster installgdal [options]

DESCRIPTION

This command will download GDAL to a directory. Mac OS users should use 'brew install --with-swig -java gdal' instead. The directory should be added to the PATH and Linux users should set LD_LIBRARY_PATH to the directory.

OPTIONS

- --dir
 - The download directory ""

<<<

Vector Commands

Vector data operations (Required options are designated with an *)

- [CQLDelete](#)
- [LocalExport](#)
- [MRExport](#)

geowave vector cqldelete

NAME

geowave vector cqldelete - Delete data that matches a CQL filter

SYNOPSIS

geowave vector cqldelete [options] <storename>

DESCRIPTION

This command will delete all data in a data store that matches a CQL filter.

OPTIONS

- `--adapterId`
 - Optional ability to provide an adapter ID
- `* --cql`
 - CQL Filter for delete
- `--debug`
 - Print out additional info for debug purposes
 - Default: false
- `--indexId`
 - The name of the index (optional)

geowave vector localexport

NAME

geowave vector localexport - Export data directly

SYNOPSIS

geowave vector localexport [options] <store name>

DESCRIPTION

This command will export data from a data store

OPTIONS

- `--adapterIds`
 - Comma separated list of adapter Ids
- `--batchSize`
 - Records to process at a time
 - Default: 10000
- `--cqlFilter`
 - Filter exported data based on CQL filter
- `--indexId`
 - The index to export from
- `* --outputFile`

geowave vector mrexport

NAME

geowave vector mrexport - Export data using map-reduce

SYNOPSIS

geowave vector mrexport [options] <hdfs host:port> <path to base directory to write to> <store name>

DESCRIPTION

This command will perform a data export for data in a data store, and will use MapReduce to support high-volume data stores.

OPTIONS

- --adapterIds
 - Comma separated list of adapter Ids
 - --batchSize
 - Records to process at a time
 - Default: 10000
 - --cqlFilter
 - Filter exported data based on CQL filter
 - --indexId
 - The index to export from
 - --maxSplits
 - The max partitions for the input data
 - --minSplits
 - The min partitions for the input data
 - --resourceManagerHostPort
-

Services Commands

Services operations (Required options are designated with an *)

- [grpc start](#)
- [grpc stop](#)

geowave util grpc start

NAME

geowave util grpc start - Start the GeoWave gRPC server

SYNOPSIS

geowave util grpc start [options]

DESCRIPTION

The geowave util grpc start operator will start the GeoWave gRPC server on a given port number. Remote gRPC clients can interact with GeoWave from this service.

OPTIONS

- --port
 - The port number the server should run on
 - Default: 8980
- --nonBlocking
 - Runs the server in non-blocking mode
 - Default: false

geowave util grpc stop

NAME

geowave util grpc stop - Stop the GeoWave gRPC server

SYNOPSIS

geowave util grpc stop

DESCRIPTION

Shuts down the GeoWave gRPC server

OPTIONS

There are currently no options for this command
