**GOBLET Tools**

GOBLET is a series of command line tools to easily perform spatial operation on GIS data. Each tool performs a specific spatial operation. The main purpose of having a set of command line tools is to facilitate the use of GOBLET in scripts like R or Python or in 3rd party applications like the Feed Resources Framework, Development Domains, etc. GOBLET is open source and cross platform. It can run on Windows, Linux and Mac.

Each GOBLET command line tool can be execute in a local database (created with ***goblet-createdb***) or in a remote server using the –r option. The remote option allows GOBLET to perform spatial operations in high performance MySQL servers or MySQL Cluster servers.

GOBLET does not display data (on screen maps). To display any raster or shape output you can use QGIS (<http://www.qgis.org/>) or any ESRI GIS product.

GOBLET only imports/exports ArcInfo ASCII grids and shapefiles. If you need to use any other raster format like GeoTIFF, you can use GDAL (<http://www.gdal.org/gdal_translate.html>) to convert raster files across different formats.

To use the command line tools, run cmd (Windows) or access a terminal (Linux/Mac) then enter into the GOBLET directory. In windows systems *cd* into the directory “c:\ILRIBIN\GOBLET”.

The examples in this document uses double quotes (“) to enclose strings, in Linux systems uses single quote (‘).

**The tools:**

***goblet-createdb***

Description: Creates a new GOBLET database

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-c <string>, --cellsize <string>

**(required)**  Cell size

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Create a new GOBLET database called “my1kmdb” in the directory “c:\mygobletdbs” (**The directory must exist**) with a cell size of 0.00833333 (~1km cell size).

*goblet-createdb -a C:\mygobletdbs -d my1kmdb -c 0.008333333*

***goblet-importdataset***

Description: Imports a ArcInfo ASCII grid into the database

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-g <string>, --gridfile <string>

**(required)**  Source grid file

-s <string>, --datasetDescription <string>

**(required)** Dataset description

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

**(required)** Database name

-o, --overwrite

Overwrite dataset if exists

-r, --remote

Connect to remote host

Example:

Import the grid called lgp001.asc (assuming it resides in c:/ILRIBIN/GOBLET/tesdata) into a GOBLET grid dataset called “lgp” with description “Length of Growing Period- Kenya, Tanzania and Uganda”.

*goblet-importdataset -a C:\mygobletdbs -d my1kmdb -g ./testdata/lgp001.asc –t lgp -s "Length of Growing Period- Kenya, Tanzania and Uganda"*

***goblet-importshape***

Description: Imports a shapefile into the database

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-s <string>, --datasetDescription <string>

**(required)** Dataset description

-S <string>, --shapefile <string>

**(required)** Shape file

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

**(required)**  Database name

-o, --overwrite

Overwrite the previous dataset if present

-r, --remote

Connect to remote host

Example:

Import the shapefile called kenTanUnga .shp (assuming it resides in c:/ILRIBIN/GOBLET/tesdata) into a GOBLET shape dataset called “kentanuga” with description “Kenya, Tanzania and Uganda”.

*goblet-importshape -a C:\mygobletdbs -d my1kmdb -S ./testdata/kenTanUnga.shp -t kentanuga -s "Kenya, Tanzania and Uganda"*

The import process stores all the shapes and related data into the database. Each shape is stored with a unique id called “shapeid”. To view the contents of a shape dataset use ***goblet-describeshape***.

***goblet-outputraster***

Description: Creates a ArcInfo ASCII grid from a GOBLET grid dataset. Opposite of ***goblet-importdataset***

Parameters:

-c <string>, --combinationtoshow <string>

“CombinationCode,CombinationCode,....”

-S <string>, --constraintbyshapes <string>

Constraint output using shapes: “ShapeDataSet:shapeID,ShapeID,....”

-e <string>, --extent <string>

Extent: “(upperLeft degrees lat,log) (lowerRight degrees lat,log)”

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-o <string>, --outputFile <string>

**(required)**  output ascii grid file name

-i <string>, --inputDatasetType <string>

**(required)** Input Dataset type: (g)rid, (c)lassification, com(b)ination

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

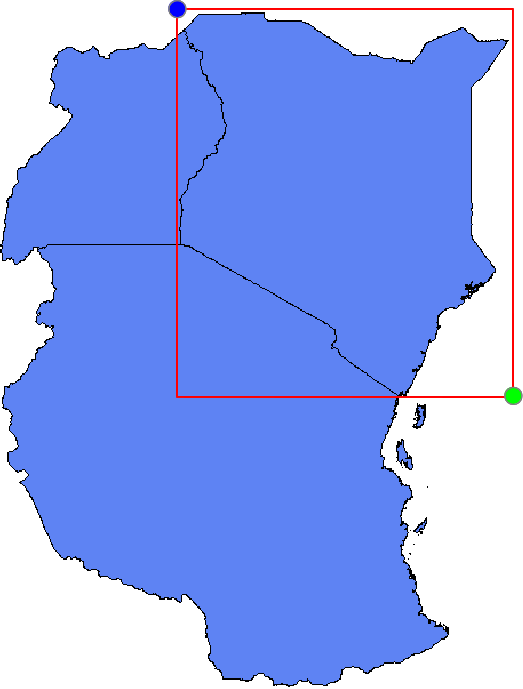
**(required)** Database name

-r, --remote

Connect to remote host

Example:

Output the grid dataset called “lgp” into an ASCII grid called “testout.asc” for an extent covering Kenya. The example assumes that “lgp” exists in the database. The following image shows how to indicate an extent in GOBLET:



An extent represents a box covering from the blue point to the green point in longitude, latitude degrees. For Kenya for example the extent is roughly (**33.76,4.87)** **(42.07,-4.99).**

To output the grid dataset execute:

*goblet-outputraster -a C:\mygobletdbs -d my1kmdb -t lgp -i g -e "(33.76,4.87) (42.07,-4.99)" -o ./testout.asc*

The output using an extend generates a square output. It is also possible to mask the output using a shape dataset. For example, to extract only Kenya use the shape dataset “kentanuga” but only the shapeid = 0.

*goblet-outputraster -a C:\mygobletdbs -d my1kmdb -t lgp -i g -e "(33.76,4.87) (42.07,-4.99)"* ***–S “kentanuga:0”*** *-o ./testout.asc*

To view the contents of a shape dataset use ***goblet-describeshape***.

***goblet-outputshape***

Description: Generates a shapefile from a GOBLET shape dataset. Opposite to ***goblet-importshape***

Parameters:

-b <string>, --combtoshow <string> “classCode,classCode,....”

-c <string>, --classestoshow <string> “classCode,classCode,....”

-e <string>, --extent <string>

Extent: “(upperLeft degrees lat,log) (lowerRight degrees lat,log)”

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-i <string>, --datasetype <string>

**(required)** Dataset type: (s)hapefile, (c)lassification, com(b)ination

-o <string>, --outputfile <string>

**(required)** Output shape file

-t <string>, --dataset <string>

Dataset name

-s <string>, --shapedataset <string>

**(required)** Shape dataset name

-d <string>, --database <string>

**(required)** Database name

-O, --overwrite

Overwrite shape if exists

-r, --remote

Connect to remote host

Example:

Output the shape dataset called “kentanuga” into a shapefile called “testout.shp”.

*goblet-outputshape -a C:\mygobletdbs -d my1kmdb -s kentanuga -i s -o ./testout.shp*

***goblet-aggregatedataset***

Description: Aggregates a grid dataset into a shape dataset.

Parameters:

-f <string>, --combfunction <string>

Aggregation function avg,sum,max,min,and,or,xor,count. Default avg

-S <string>, --constraintbyshapes <string>

Constraint classification using shapes: ShapeDataSet:shapeID,ShapeID,....

-e <string>, --extent <string>

Extent: “(upperLeft degrees lat,log) (lowerRight degrees lat,log)”

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-s <string>, --shapedataset <string>

**(required)** The target shape dataset

-t <string>, --datasets <string>

**(required)** The datasets to aggregate

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Aggregate the grid dataset “lgp” into the provinces in Kenya.

1. Upload the shapefile “./testdata/kenya\_provinces.shp”

*goblet-importshape -a C:\mygobletdbs -d my1kmdb -S ./testdata/kenya\_provinces.shp -t kenprov -s "Kenya Provinces"*

1. Aggregate “lgp” into “kenprov”

*goblet-aggregatedataset -a C:\mygobletdbs -d my1kmdb -t lgp –s kenprov*

The process by default aggregates the average of “lgp” to each shape of “kenprov”. It is also possible to aggregate the maximum, minimum, sum of “lgp” values and number of cells. For binary values it is also possible to aggregate by and, or and xor.

***goblet-classifydataset***

Description: Classifies a grid dataset using a classification code and values from – to.

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-S <string>, --constraintbyshapes <string>

Constraint classification using shapes: ShapeDataSet:shapeID,ShapeID,....

-e <string>, --extent <string>

Extent: “(upperLeft degrees lat,log) (lowerRight degrees lat,log)”

-c <string>, --classdefinition <string>

**(required)** Class definition: “ClassNumber:valueFrom ValueTo,ClassNumber:valueFromValueTo,..”

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Classifies the grid dataset “lgp” with three classes: class 10 with values ranging from 0 to 114, class 20 with values ranging from 114 to 228 and class 30 with values ranging from 228 to 344.

goblet-classifydataset -a C:\mygobletdbs -d my1kmdb -t lgp –c "**10**:0 114,**20**:114 228,**30**:228 344"

***goblet-classifyaggregate***

Description: Classifies an aggregated shape dataset using a classification code and values from – to.

-e <string>, --extent <string>

Extent: “(upperLeft degrees lat,log) (lowerRight degrees lat,log)”

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-c <string>, --classdefinition <string>

**(required)** Class definition: “ClassNumber:valueFrom ValueTo,ClassNumber:valueFrom ValueTo,..”

-s <string>, --shapedataset <string>

**(required)** The aggregated shape dataset

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Classifies the aggregated “lgp” in the shape dataset “kenprov” with three classes: class 10 with values ranging from 0 to 114, class 20 with values ranging from 114 to 228 and class 30 with values ranging from 228 to 344.

goblet-classifyaggregate -a C:\mygobletdbs -d my1kmdb -t lgp -c "10:0 114,20:114 228,30:228 344" -s kenprov***goblet-combinedatasets***

Description: Combines a series of grid datasets into a result grid dataset

Parameters:

-f <string>, --combfunction <string>

Combination function sum or (mul)tiplication Default avg

-S <string>, --constraintbyshapes <string>

Constraint classification using shapes: ShapeDataSet:shapeID,ShapeID,....

-e <string>, --extent <string>

Extent: “(upperLeft degrees lat,log) (lowerRight degrees lat,log)”

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-t <string>, --datasets <string>

**(required)** Datasets to combine. For example: “dataset1,dataset2,dataset3,....”

-d <string>, --database <string>

**(required)**  Database name

-r, --remote

Connect to remote host

Example:

Combine LGP, Population Density and Travel time into a result.

1. Upload and classify Population density

*goblet-importdataset -a C:\mygobletdbs -d my1kmdb -t popdens -s "Population Density" -g ./testdata/pop00den1.asc*

*goblet-classifydataset -a C:\mygobletdbs -d my1kmdb -t popdens -c "100:0 80,200:80 160,300:160 15493"*

1. Upload and classify travel time

*goblet-importdataset -a C:\mygobletdbs -d my1kmdb -t traveltime -s "Travel time in hours" -g ./testdata/time1.asc*

*goblet-classifydataset -a C:\mygobletdbs -d my1kmdb -t traveltime -c "1000:0 4,2000:4 8,3000:8 47"*

1. Combine lgp, popdens and traveltime into one result

*goblet-combinedatasets -a C:\mygobletdbs -d my1kmdb -t "lgp,popdens,traveltime"*

1. Output the combination only for Kenya into a grid file

*goblet-outputraster -a C:\mygobletdbs -d my1kmdb* ***-t NA -i b*** *–e "(33.76,4.87) (42.07,-4.99)" –S "kentanuga:0" -o ./testcomb.asc*

***goblet-combineaggregate***

Description: Combines an aggregated shape dataset into a result shape dataset

Parameters:

-f <string>, --combfunction <string>

Combination function sum,and,or,xor. Default sum

-e <string>, --extent <string>

Extent: “(upperLeft degrees lat,log) (lowerRight degrees lat,log)”

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-s <string>, --shapedataset <string>

**(required)** The aggregated shape dataset

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Combine aggregated LGP, Population Density and Travel time into a result.

1. Aggregate and classify “popdens”

*goblet-aggregatedataset -a C:\mygobletdbs -d my1kmdb -t popdens –s kenprov*

*goblet-classifyaggregate -a C:\mygobletdbs -d my1kmdb -t popdens -c "100:0 80,200:80 160,300:160 2977" -s kenprov*

1. Aggregate and classify “traveltime”

*goblet-aggregatedataset -a C:\mygobletdbs -d my1kmdb -t traveltime –s kenprov*

*goblet-aggregatedataset -a C:\mygobletdbs -d my1kmdb -t traveltime -c "1000:0 4,2000:4 8,3000:8 11" –s kenprov*

1. Combine aggregated “lgp”, “popdens” and “traveltime” into a result

*goblet-combineaggregate -a C:\mygobletdbs -d my1kmdb -s kenprov*

1. Output the result into a shapefile.

*goblet-outputshape -a C:\mygobletdbs -d my1kmdb -s kenprov -i b -o ./combout.shp*

***goblet-gendatasetstats***

Description: Generates a series of statistics from a grid dataset

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Generate basic statistics from the grid dataset “traveltime”

*goblet-gendatasetstats -a C:\mygobletdbs -d my1kmdb -t traveltime*

*The result should be:*

Number of cells: 2069100

Maximum value: 47

Minimum value: 0

Average value: 9.641196655

Standard deviation: 7.456691915

***goblet-genaggregatestats***

Description: Generates a series of statistics from a aggregated shape dataset

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-s <string>, --shapedataset <string>

(required) The aggregated shape dataset

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Generate basic statistics from the aggregated grid dataset “traveltime” in the shape dataset “kenprov”

*goblet-genaggregatestats -a C:\mygobletdbs -d my1kmdb -t traveltime -s kenprov*

*The result should be:*

Number of shapes: 18

Maximum value: 10.90476

Minimum value: 0

Average value: 4.2554

Standard deviation: 2.925105562

***goblet-describeshape***

Description: Display information about a shape dataset. The output can be easy readable for humans or an xml representation for its usage in other computer programs.

Parameters:

-f <string>, --format <string>

Format: (h)uman or (c)omputer. Default human

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Describe the shape kenprov

*goblet-describeshape -a C:\mygobletdbs -d my1kmdb -t kenprov*

*The result should be:*

ID: kenprov

Description: Kenya Provinces

Units:

Shape type: POLYGON

Has metadata: No

No. Colums in raster: 953

No. Rows in raster: 1165

Raster X center: 33.954166412353516

Raster Y center: -4.687500000000000

+---------+--------+-----------+----------+------------+---------------+-----------+

+ shapeid + AREA + PERIMETER + KENPROV\_ + KENPROV\_ID + PROV\_NAME + num\_cells +

+---------+--------+-----------+----------+------------+---------------+-----------+

| 0 | 14.302 | 32.323 | 2.000 | 3.000 | Rift valley | 205980 |

| 1 | 13.103 | 28.620 | 3.000 | 6.000 | Eastern | 188685 |

| 2 | 10.215 | 18.885 | 4.000 | 7.000 | North Eastern | 147007 |

| 3 | 0.000 | 0.067 | 5.000 | 6.000 | Coast | 3 |

| 4 | 0.000 | 0.047 | 6.000 | 3.000 | Coast | 2 |

| 5 | 0.003 | 0.268 | 7.000 | 6.000 | Coast | 42 |

| 6 | 0.665 | 5.497 | 8.000 | 2.000 | Western | 9571 |

| 7 | 1.028 | 10.284 | 9.000 | 1.000 | Nyanzs | 14815 |

| 8 | 1.060 | 6.851 | 10.000 | 4.000 | Central | 15264 |

| 9 | 6.787 | 22.678 | 11.000 | 8.000 | Coast | 97756 |

| 10 | 0.001 | 0.140 | 12.000 | 1.000 | Coast | 8 |

| 11 | 0.000 | 0.080 | 13.000 | 1.000 | Coast | 4 |

| 12 | 0.003 | 0.359 | 14.000 | 1.000 | Coast | 50 |

| 13 | 0.005 | 0.323 | 15.000 | 1.000 | Coast | 75 |

| 14 | 0.055 | 1.529 | 16.000 | 5.000 | Nairobi | 797 |

| 15 | 0.004 | 0.504 | 17.000 | 8.000 | Coast | 69 |

| 16 | 0.002 | 0.273 | 18.000 | 8.000 | Coast | 28 |

| 17 | 0.004 | 0.495 | 19.000 | 8.000 | Coast | 61 |

| 18 | 0.004 | 0.332 | 20.000 | 8.000 | Coast | 58 |

| 19 | 0.009 | 0.605 | 21.000 | 8.000 | Coast | 128 |

| 20 | 0.006 | 0.533 | 22.000 | 8.000 | Coast | 91 |

| 21 | 0.004 | 0.362 | 23.000 | 8.000 | Coast | 51 |

| 22 | 0.001 | 0.168 | 24.000 | 8.000 | Coast | 10 |

| 23 | 0.001 | 0.129 | 25.000 | 8.000 | Coast | 7 |

+---------+--------+-----------+----------+------------+---------------+-----------+

***goblet-resetdataset***

Description: Clears a grid dataset from any previous classification.

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-t <string>, --dataset <string>

**(required)** Dataset name

-d <string>, --database <string>

**(required)** Database name

-r, --remote

Connect to remote host

Example:

Reset the grid dataset “lgp”

*goblet-resetdataset -a C:\mygobletdbs -d my1kmdb -t lgp*

***goblet-resetaggregate***

Description: Clears an aggregated shape dataset from any previous classification.

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-s <string>, --shapedataset <string>

**(required)** The target shape dataset

-d <string>, --database <string>

**(required)** Database name

-c, --clear

Connect to remote host

-r, --remote

Connect to remote host

Example:

Reset the shape dataset “kenyaprov”

*goblet-resetaggregate -a C:\mygobletdbs -d my1kmdb -s kenprov*

***goblet-datasetcalc***

Description: Calculate operations between grid datasets.

Parameters:

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-o <string>, --output <string>

(required) Output type: (f)ile or (d)ataset

-S <string>, --constraintbyshapes <string>

Constraint classification using shapes: ShapeDataSet:shapeID,ShapeID,....

-e <string>, --extent <string>

Extent: '(upperLeft degrees lat,log) (lowerRight degrees lat,log)'

-c <string>, --calculation <string>

(required) Dataset calculation to perform. For example: '(DatasetA +DatasetB) / DatasetC'

-g <string>, --gridfile <string>

Output grid file if output is file

-s <string>, --datasetDescription <string>

Dataset description if output is dataset

-t <string>, --dataset <string>

Dataset name if output is dataset

-d <string>, --database <string>

(required) Database name

-O, --overwrite

Overwrite dataset if exists

-r, --remote

Connect to remote host

Example:

Calculate 2 X traveltime and stored in a result ASCII grid

*goblet-datasetcalc -a C:\mygobletdbs -d my1kmdb –c “traveltime \* 2” –o f –g ./output.asc*

***goblet-calcincombination***

Description: Calculate dataset values aggregated by combination codes (Requires an existing combination)

Parameters:

-s <string>, --descriptions <string>

Descriptions for the calculations separated by coma. Default value is the calculation string.

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-o <string>, --output <string>

(required) Output type: (h)uman readable or (c)omputer readable

-c <string>, --calculation <string>

(required) Calculation to perform. For example: 'sum(DatasetA),sum(DatasetB)'

-d <string>, --database <string>

(required) Database name

-r, --remote

Connect to remote host

Example:

Calculate the sum of travel time for each combination class and show the result as a table.

*golet-calcincombination -a C:\mygobletdbs -d my1kmdb –c “sum(traveltime)” –o h -s “Total travel time”*

*The result should be:*

+------------+-------------------+

+ Class code + Total travel time +

+------------+-------------------+

| 1110 | 625100.000 |

| 1120 | 1394700.000 |

| 1130 | 64800.000 |

| 1210 | 1025100.000 |

| 1220 | 1716200.000 |

| 1230 | 79000.000 |

| 1310 | 4042800.000 |

| 1320 | 10638300.000 |

| 1330 | 125800.000 |

| 2110 | 4400.000 |

| 2120 | 126400.000 |

| 2130 | 21000.000 |

| 2220 | 31100.000 |

| 2230 | 5900.000 |

| 2310 | 3000.000 |

| 2320 | 8300.000 |

| 2330 | 3500.000 |

| 3110 | 400.000 |

| 3120 | 23000.000 |

| 3130 | 4100.000 |

| 3220 | 3200.000 |

| 3230 | 600.000 |

| 3320 | 1900.000 |

+------------+-------------------+

***goblet-*** ***modifydataset***

Description: Modify the description and units of a GOBLET dataset

Parameters:

-U <string>, --units <string>

New unit for the dataset

-s <string>, --description <string>

New description for the dataset

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-t <string>, --dataset <string>

(required) Dataset name

-d <string>, --database <string>

(required) Database name

-r, --remote

Connect to remote host

Example:

Modify the dataset travel to have the description “East Africa travel time”

*goblet-modifydataset -a C:\mygobletdbs -d my1kmdb -t traveltime -s “East Africa travel time”*

***goblet-*** ***listdatasets***

Description: List the dataset in a GOBLET database

Parameters:

-f <string>, --format <string>

Format: (h)uman or (c)omputer

-p <string>, --password <string>

Password. Default no password

-u <string>, --user <string>

User. Default empty

-P <string>, --port <string>

Port number to use. Default 3306

-H <string>, --host <string>

Connect to host. Default localhost

-a <string>, --path <string>

Path to database. Default .

-d <string>, --database <string>

(required) Database name

-r, --remote

Connect to remote host

Example:

List the available dataset

*goblet-* *listdatasets -a C:\mygobletdbs -d my1kmdb*

*The result should be:*

+-------------------+--------------------------+-------+----------+--------------+-------------------+----------------+--------------------+---------------------+-------------------------------+

+ Dataset code + Description + Units + Metadata + Dataset type + Number of columns + Number of rows + X center + Y center + Shape type (if type is shape) +

+-------------------+--------------------------+-------+----------+--------------+-------------------+----------------+--------------------+---------------------+-------------------------------+

| datasetCalcOutput | Nairobi lgp \*2 | class | | grid | 53 | 34 | 36.674991333000008 | -1.441663008999996 | NA |

| kenyaprov | Kenya provinces | | | shape | 953 | 1166 | 33.954165308499995 | -4.687499812500000 | POLYGON |

| lgp | Length of growing period | | | grid | 1510 | 1970 | 29.327170000001001 | -11.745699999998999 | NA |

| popdens | Population density | | | grid | 1510 | 1970 | 29.327170000001001 | -11.745699999998999 | NA |

| traveltime | East Africa travel time | | | grid | 1510 | 1970 | 29.327170000001001 | -11.745699999998999 | NA |

+-------------------+--------------------------+-------+----------+--------------+-------------------+----------------+--------------------+---------------------+-------------------------------+