What's New in Maple 2017

Maple

Geographic Data

Maple 2017 updates the DataSets package to include a built-in geographic information database and a map visualization facility.

▼ GeoNames Dataset

The GeoNames data set is a built-in geographic information database which contains data for over one million geographic locations around the world. The GeoNames data set shares the same set of accessing commands as other built-in data sets.

Obtain a Reference object for the GeoNames data set:

```
with(DataSets) :
with(DataSets:-Builtin) :
ref := Reference("GeoNames")
```

```
        [ Geonames (GeoNamesid)      Name                 Type                 ...  (7 more) ]
        [          68            Boneh-ye Mehdi  "section of populated place"  ...           ]
        [         121              Lab Sefid            "locality"             ...           ]
ref :=  [         166               Gorizi              "locality"             ...           ]
        [          ⋮                   ⋮                    ⋮                   ⋱            ]
        [   (1087362 more)                                                                   ]
```

Each data entry (row) consists of 10 columns where each column denotes certain information about that geographic location. The first column, `GeoNamesid`, is an integer id unique to each row.

View a list of all the column headers (not including `GeoNamesid`):

```
GetHeaders(ref)
```

```
["Name", "Type", "Latitude", "Longitude", "Country", "Province/State/Region", "City/County",
    "Population", "Time Zone"]
```

You can search the data set by indexing into the Reference object. The most interesting and useful indexing method is to index by boolean constraints on column values. For example, find all places in Canada with the name Waterloo and see the actual data:

```
GetData(ref[[Country = "Canada", Name = "Waterloo"]])
[["Waterloo", "populated place", 45.3500799999999984, -72.5158200000000051, "Canada",
   "Quebec", "Monteregie", 4064, "America/Toronto"],
   ["Waterloo", "populated place", 43.4667999999999992, -80.5163900000000012, "Canada",
   "Ontario", undefined, 97475, "America/Toronto"],
   ["Waterloo", "third-order administrative division", 45.3367300000000029,
   -72.5279599999999931, "Canada", "Quebec", "Monteregie", 0, "America/Toronto"]]
```

Find out which states in the US have population greater than 5 million:

```
ref[[Country = "United States", Type = "first-order administrative division", Population
     > 5000000]]
   [ Geonames (GeoNamesid)    Name                   Type                       ...   (8 more) ]
   [       4155751           Florida    "first-order administrative division"  ...            ]
   [       4197000           Georgia    "first-order administrative division"  ...            ]
   [       4361885           Maryland   "first-order administrative division"  ...            ]
   [          ⋮                 ⋮                      ⋮                        ⋱             ]
   [      (18 more)                                                                           ]
```

WorldMap

The WorldMap object under the Builtin subpackage of the DataSets package is new to Maple 2017. It is a versatile map-displaying tool that can cooperate with the GeoNames data set as well as work alone.

The WorldMap object can provide visualization for search results from the GeoNames data set. For example, to generate a map of regions of Austria, first search for the data using the GeoNames data set and then create a WorldMap object using the search results:

$regionsOfAustria := ref[[Country = \text{"Austria"}, Type = \text{"first-order administrative division"}]];$

```
regionsOfAustria :=

  [ Geonames (GeoNamesid)   Name        Type                                   …  (8 more) ]
  [       2761367           Wien        "first-order administrative division"  …           ]
  [       2762300           Vorarlberg  "first-order administrative division"  …           ]
  [       2763586           Tirol       "first-order administrative division"  …           ]
  [          ⋮                ⋮                        ⋮                        ⋱           ]
  [      (6 more)                                                                           ]
```

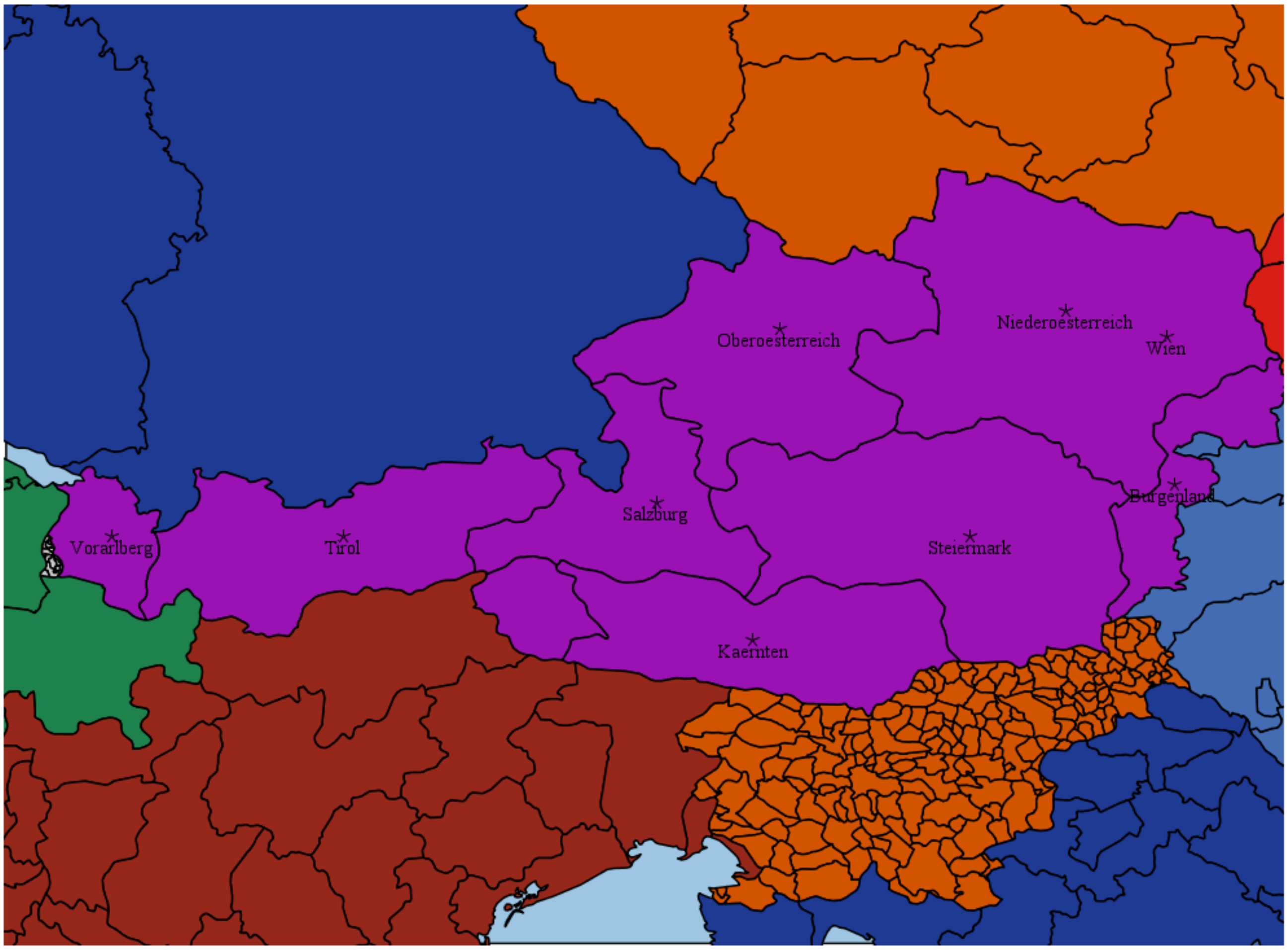
$regionsOfAustriaMap := WorldMap(regionsOfAustria)$

regionsOfAustriaMap :=

A map of the world with 9 points
projection: MillerCylindrical

Then, plot the map by calling the Display command on the WorldMap object:

Display(*regionsOfAustriaMap*, *size* = [1000, 800], *style* = *polygonoutline*)



The WorldMap object can display great circle paths, which are the shortest paths along the earth's surface between two geographic locations.

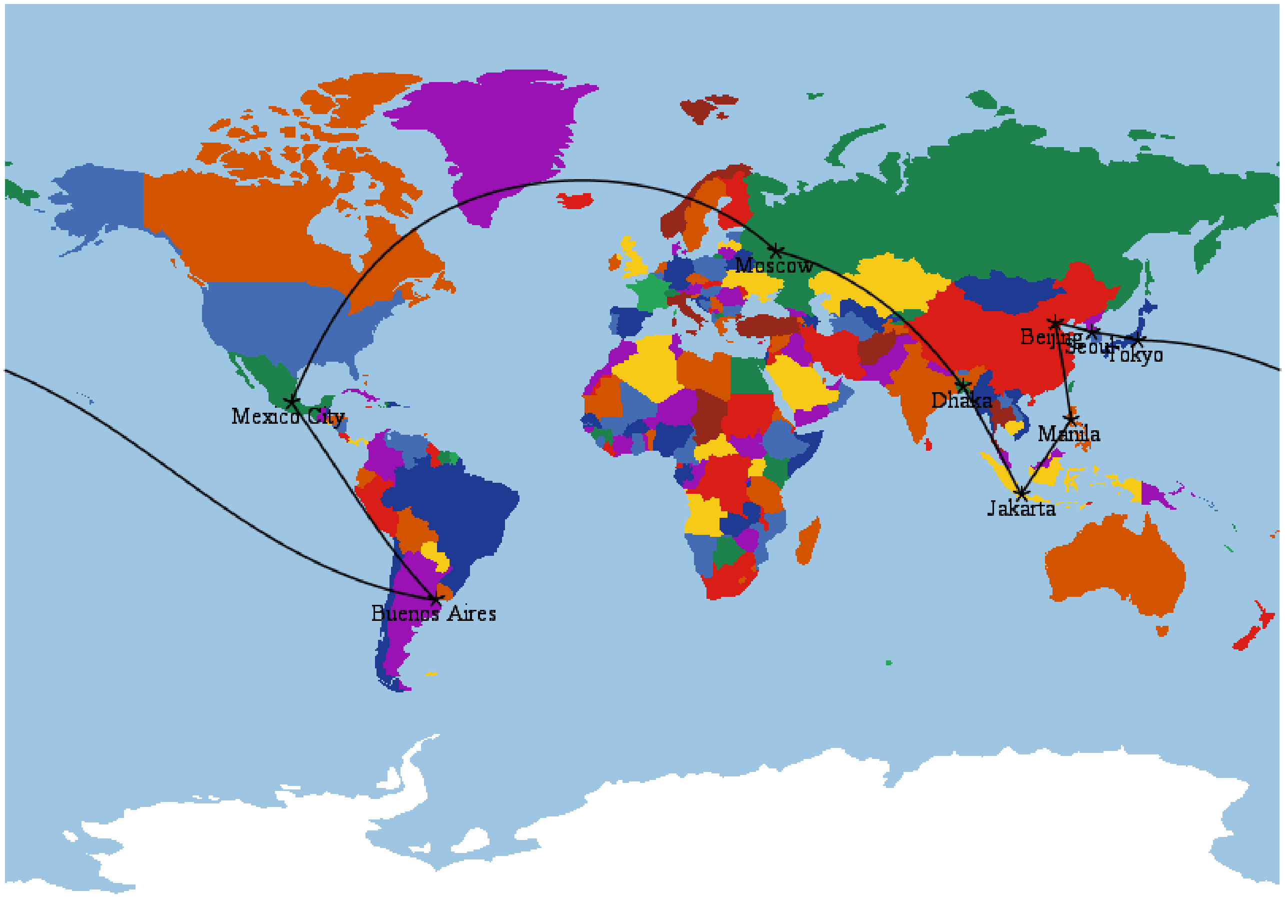
For example, connect all capital cities in the world with population greater than 8 million in a loop:

bigPopulationCapitals := *Reference*("Geonames") [[*Type* = "capital of a political entity",
    Population > 8000000]]

bigPopulationCapitals :=

```
[ Geonames (GeoNamesid)   Name     Type                            ...  (8 more) ]
[       524901            Moscow   "capital of a political entity"  ...           ]
[      1185241            Dhaka    "capital of a political entity"  ...           ]
[      1642911            Jakarta  "capital of a political entity"  ...           ]
[         ⋮                 ⋮                  ⋮                     ⋱            ]
[      (6 more)                                                                   ]
```

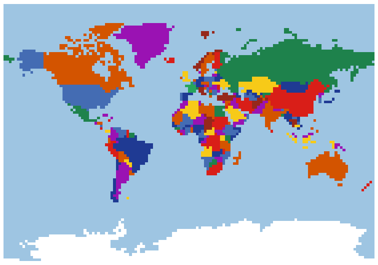
```
bigPopulationCapitalsMap := WorldMap(bigPopulationCapitals) :
seq(AddPath(bigPopulationCapitalsMap, bigPopulationCapitals[i],
    bigPopulationCapitals[modp(i, CountRows(bigPopulationCapitals)) + 1], endpoint=false), i
    = 1 ..CountRows(bigPopulationCapitals)) :
SetCenter(bigPopulationCapitalsMap, [0, 0]) :
ZoomOut(bigPopulationCapitalsMap) :
Display(bigPopulationCapitalsMap, size = [1200, 600])
```



The WorldMap object can display maps under numerous map projections. Some projections are suitable for the type of data visualization presented in previous examples while others make maps themselves interesting and charming to look at.

$m := \mathit{WorldMap}(\;)$

$m :=$



A map of the world

projection: MillerCylindrical

For example, to generate a world map in the Van der Grinten projection, call the Display command on *m* with the projection=VanderGrinten option:

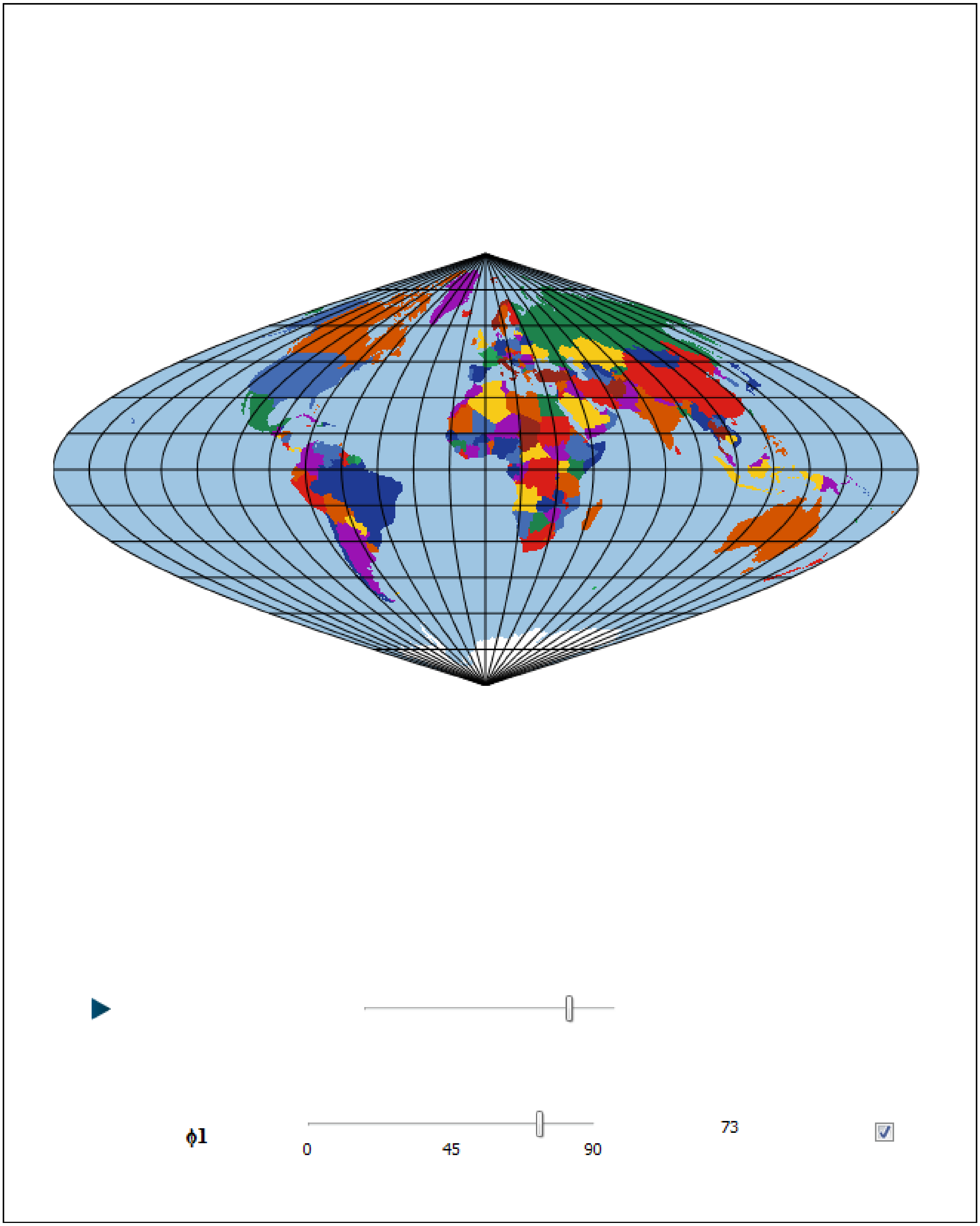
$\mathit{Display}(m, \mathit{projection} = \mathit{VanderGrinten}, \mathit{size} = [600, 600])$

For parameterized map projections, using Display together with the Explore command, you can easily see how the map transforms with respect to changes in the parameters.

For example, the Bottomley projection accepts a standard parallel ($\phi 1$) parameter which makes it vary between the sinusoidal projection ($\phi 1 = 0$) and the Werner projection ($\phi 1 = 90$).

Use Explore to see this transformation:

$$Explore\left(Display\left(m, projection = Bottomley\left(\phi 1\right), grid = true, thickness = 0\right), \phi 1 = 0 ..90, animate = true, size = \left[600, 600\right]\right)$$



▶

ϕ1 73 ☑

The WorldMap object can also highlight the countries of the world based on a set of data using the ChoroplethMap command. The example below shows the countries colored based on their coastline length.

$coastlines := Reference(\text{"Country"})[\,..\,, \text{"Coastline"}]$

$$coastlines := \left[\begin{array}{cc} \textit{Country (Name)} & \textit{Coastline} \\ \textit{Afghanistan} & 0 \\ \textit{Albania} & 362.\ \mathrm{km} \\ \textit{Algeria} & 998.\ \mathrm{km} \\ \vdots & \vdots \\ \textit{(182 more)} & \end{array}\right]$$

Many countries have coastline length 0, so taking the logarithm would yield minus infinity. Instead, we add 1 to each coastline length before taking the logarithm.

$WorldMap\text{:-}ChoroplethMap(m, coastlines, [\,\text{"Khaki"}, \text{"DarkGreen"}\,], transformdata = (x \rightarrow \log(x + 1)))$