BaseIndex.txt must be named exactly this for the application to detect and read from it.

It is a csv-like format in that it separates its values like CSV, but doesn’t have consistent headers throughout.

Instead, within each category there is a consistent set of headers.

This document specifies what these categories are, how to format them and ensure they are read by the visualiser. Each line (with exception of context-switching lines) corresponds to some form of **label** within the application that is stored internally and referenced within the application.

The program reads this file by setting a context and interpreting following lines within that context until it detects another context switch. These contexts are:

policygoal,year,scenarios,subscenarios,target,metric,policyouterlabel,country

If a new line is in the text file which contains only the name of the context, the application will switch to that context and interpret the next lines within it.

These contexts may appear in any order with the exception of subscenarios which must come after scenarios.

**The first entry in any line regardless of context is a link to the Base.txt’s field – it links the symbolic name present in Base.txt’s row e.g. T\_E\_FOF to its labelling.**

**policygoal**

Labelling information pertaining to the categorisation of a policy goal and theme information for colouring this segment.

**Example line:**

Envir,<alpha=#00>.<alpha=#FF> Reduced environmental impacts of the EU food-system <alpha=#00>.,"236,236,236","96,142,67","255,255,255","248,254,242","239,254,225","95,110,79"

Fields:

***policy name label*** (matching Base.txt entries for this policy)

***policy description label*** (shown on the concentric arc)

policy description has inserted spacing with some hidden characters. This is a workaround due to the complexities of inserting text along a curve without knowing its length beforehand.

Follow this example, replacing your label with the desired text, and if when you run it it’s not satisfactorily curved, add or remove spaces to the left and right of the text until it renders properly.

***theme.*** A series of RGB(optionally also A) values which determine what colours the policy’s targets get set to when they render. There must be exactly six in the format presented in the example and are interpreted in this order:

targetWedgeColour

policyWheelBackground

policyWheelText

rouletteLight

rouletteDark

rouletteText

**year**

**Example Line:**

2010,

As it is a first entry this is directly maps to the rows in Base.txt with year set to 2010.

**scenarios**

**Example Line:**

REF0,Business-as-usual outlook

Fields:

***scenariolabel,humanreadablename***

Human readable name is the label that is presented on the scenario button and in the scenario summary at the top.

**subscenarios**

**Example Line:**

REF0\_D0,REF0,Consuming a healthy and balanced diet,

Fields:

***subscenariolabel,superscenariolabel,humanreadablename***

subscenarios are visually categorised underneath a scenario(superscenariolabel), and refer to one specific scenario (subscenariolabel).

It must have a superscenario which is the scenariolabel of a valid scenario. The result of this is that when the user selects the superscenario, the subscenario becomes available as a refinement option on the UI.

**target**

**Example Lines:**

T\_E\_FOF,Foodprint of food,Here is some stuff about food, <http://www.google.com/images,200>

T\_E\_CSO,Equity in food consumption

Fields:

**targetlabel,targetdescription,tooltipText(optional),linkURL(optional),tooltipWidth(optional)**

targetDescription is the label that appears on the outer rim of the spidergraph, a short description f the target.

if the ability to hover over this, get a short description and the ability to click on it to go to a webpage is desired, fill in the optional fields like in the first example line. The final parameter, **tooltipWidth**, changes the width of the tooltip from its narrow default. The units can be considered arbitrary, but 200 is a good reference point of width for a short paragraph’s amount of text.

**metric**

**Example line**

M\_E\_FOF,Foodprint of food

**Fields:**

**metricLabel,metricDescription(unused?)**

These link the metric values to Base.txt. I’m unsure if the metric description actually gets used at any point within the program – it may be fine to leave blank.

**policyouterlabel**

**Example Line:**

Equity,<alpha=#00>.<alpha=#FF> People <alpha=#00>.,180,"253,236,206","178,123,13"

**Fields:**

**policyGoalLink,policygoalText,degrees, theme**

These are attached to PolicyGoals via the policyGoalLink. Not every PolicyGoal requires one of these. This adds a ring outside the policy goal **degrees** wide with **policygoalText** with **theme** applied.

The ring starts from the right hand side and sweeps the number of degrees **counterclockwise** around. By arranging the policygoal order in the index file and using varying degrees, this lets us have an outerlabel that spans multiple policygoals. You can see this in the case of the People goal, which spans two policygoals and has degrees set to 180.

Theme is comprised of two colours: background colour and font text colour.

The spacing in **policygoalText** uses the same hack/workaround as **policygoal’s policy description label.** refer to it to see how to change the text effectively.

**country**

**Example Line:**

**EU28**

**fields**

**CountryLabel:** Shorthand code for the country label. This is presented in the UI in a small grid button box so there is not room for more than 4 capital letters representing the country.