Sample code ColorDatasetExample.exe

https://github.com/omitindelphi/ColorBand

Illustration of practical task : display arbitrary set of properties in dataset using color code.

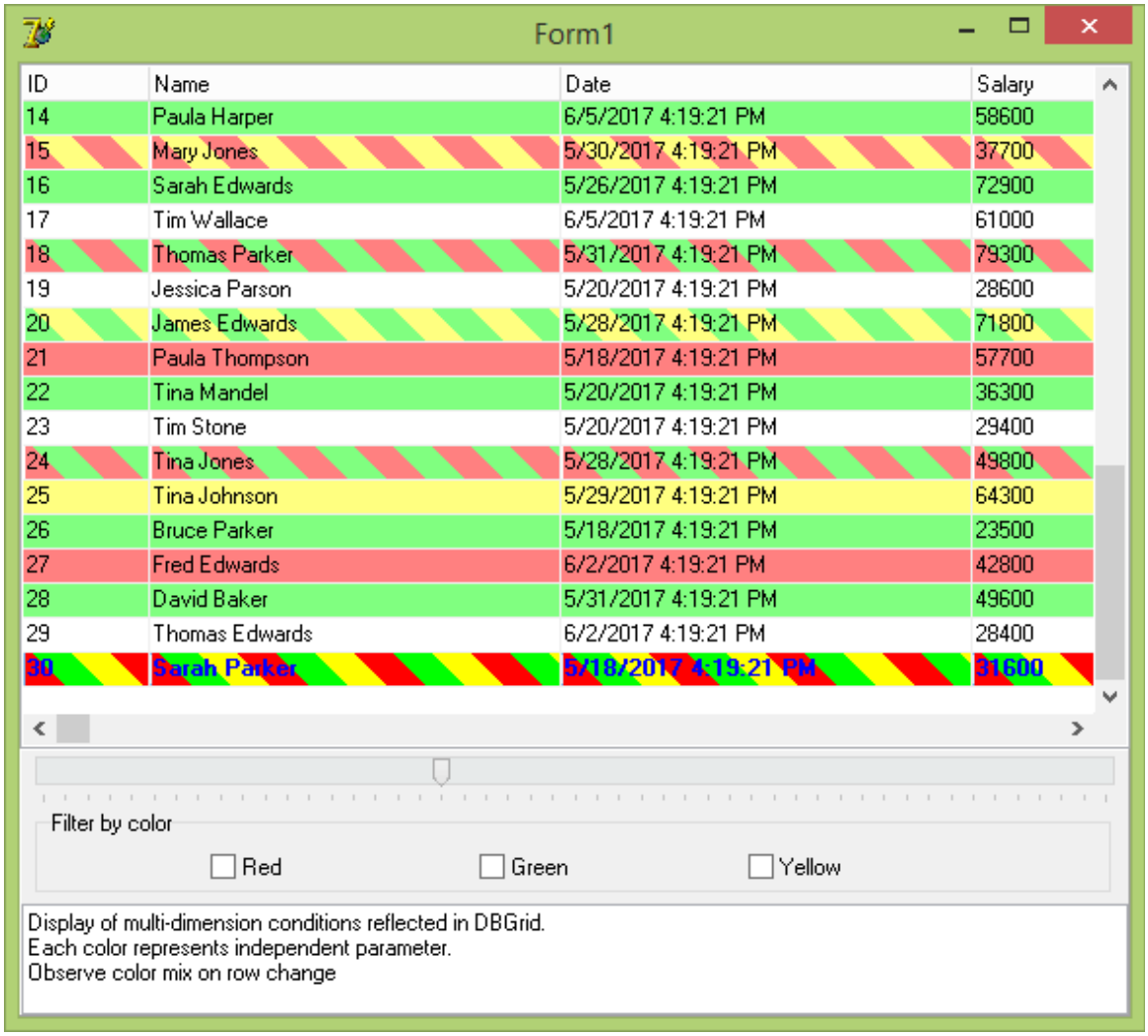
Primary feature: really fast and resource-effective drawing right on canvas without creating/disposing if intermediate bitmaps, making it useful for grid display.

Sometimes (not very often, but sometimes) you need to display collection of rows in dataset, but some rows are special and ought to be highlighted. Obvious solution is to use special color for such marking. Standard solution is to use additional columns in dataset, but they just do not stand out as bright as simple color-coding, and both approaches can be used together - grid can have both status column and row highlighting at the same time

Pretty soon it become clear, that we better have at least two colors for such color-marking, because we need properly display regular row and selected row, too. We can, of course, use font settings for such marking, but it looks just not so great.

Problem becomes much more complicated, if we need display two statuses at once; flat color not necessary would be recognized as combined status; and we will need at least 4 or 8 colors to cover all possible situations – one or other status, combined, selected or not.

Number of situations grows exponentially as number of statuses grows.



Here is my solution of this problem : use simple color-marking, each color represent specific status, dimmed colors represent non-selected row; diagonal strips just looks better than simple horizontal

or vertical ones.

Code can be compiled in D7 or in D10 (win64), trial version; only standard Delphi components used; no extrnal dependencies – deployment package consists only of single executable. Standard in-memory **TClientDataset** used as source dataset, populated by mock data.

How I could judge by our operators' response, and by my own expirience, something in our mind still is having some problems with idea of multiple statuses in single row; to overcome that stumbling point I added filtering by color, and that filtering in some mystic way does simplified recognition.

Now we can instruct operators telling them something like: **in situation #1 double-click on red line**, and **in situation #2 double-click on yellow line**. Green lines are ok, they should disappear soon by itself...

Comments on the code: it is simple proof of concept, used in several cases though. So it is lacking proper abstractions from set of color statuses (too many possibilites for that); and lacking unit testing too (can be added if anybody got interested enough). Probably such grid can be placed on the TFrame, and use grid component with better handling of mouse scroll, but such modification would complicate code and compilation without adding much to idea itself.

File location :
all .dcu files got placed into separate directories, win-64 files separately from win-32 files.
.exe file for D7 got placed ito the same folder that project files; win-64 D10 .exe file got placed to .\Win64 directory.

Originally code was developed under D7, but I verified its compilation and execution by latest D10.2; Some manipulations with .dfm files may be needed to get 100% compatibility (remove properties missing in D7 – I did that), and use of .res file generated by D10 may introduce resource conflicts in D7, so better to remove .res file from version control software and make sure .res file got generated afresh on each build.