Subject: Chameleon "engine"
Posted by mirek on Sat, 17 Jun 2006 19:49:48 GMT
View Forum Message <> Reply to Message

So, "chameleon engine" is now part of Draw, however, CtrlLib is yet to be "chameleonized" (as the first step, prompts now load icons from host Win32 OS).

Chameleon is based on "chameleon variables". Like

CH_COLOR(MenuColor, SColorHighlight);

- note that each "variable" has "default value", which is best represented as expression based on other CH variables (should be mandatory). This way it is enough to change just some of CH vars.

Above example line defines function

Color MenuColor();

CH value can be altered using

ChSet("MenuColor", somecolor);

Current CH types are

CH_INT
CH_FONT
CH_COLOR
CH_STRING
CH_ELEMENT

where CH_ELEMENT uses "Value" as its type and is intended to provide "paint description" - there is function

void ChPaint(Draw& w, const Rect& r, const Value& element);
void ChPaint(Draw& w, int x, int y, int cx, int cy, const Value& element);

Chameleon "detection facility" are simply named registered routines that can investigate current platform settings and adjust chameleon variables accordingly. They can also register alternative element painters (Win32 will do this to use XP theming API).

Default painter excepts "Color" and "Image" as values. If element is Color, rectangle is simply filled by color in ChPaint.

Image in default painter is interpreted based on "hotspot" (defined in TheIDE image designer). Hotspot symmetrically determines "fixed size" border of element and its resizable part.

In other words, with chameleon, providing different sking for application is no more complicated than drawing the appearance of all GUI widgets...

Mirek