

---

Subject: 2025.1 (alpha)

Posted by [mirek](#) on Thu, 13 Feb 2025 11:44:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

#### Core:

- In Posix .log files moved to .local/state/u++/log
- Lerp templates and functions for linear interpolation
- FormatIntBase changed to work with int64 instead of int
- DarkTheme function adjusted to better convert low luminosity colors
- New AColor and SColor types allow simplified adjustments for dark mode when runtime theme switch happens.
- Moveable logic changed to improve incomplete type handling

#### CtrlLib, CtrlCore:

- Runtime skin changes
- InfoCtrl::LeftZ/RightZ scaled methods
- Additional Animate functions for animation of widgets or general rectangles
- WhenLeftUp event added to DocEdit/LineEdit/RichEdit
- General chameleon look improvements in Linux/Windows/MacOS
- AllowDarkContent in color selection widgets shows colors converted with DarkTheme if dark mode is active
- Experimental Wayland support (there is now Wayland option in theide setup)

#### RichEdit

- Set last character format tool
- Find previous button in Find dialog
- When find/replace dialog is open Enter key does another find
- PersistentFindReplace now default false
- Ctrl+Click now opens http and https links
- Zooming improvements
- PixelMode - special mode where images at 100% are pixel perfect.
- AllowDarkContent, DarkContent - displays all colors converted with DarkTheme if dark mode is active, effectively allows dark paper in dark mode.

#### Core/SSH

- libssh2 upgraded to version 1.11.1 (#209)

#### ide, umk

- Posix config is now always in the same folder ../theide/.. for all variants of theide and umk (to share the same .var and .bm files)
- RichEdits in theide now support dark theme
- Json/XML viewer
- aux packages now have Remove all menu command
- bmGIT\_BRANCH in build info
- The option in settings to allow access to Downloads folder (default off on MacOS to prevent MacOS asking the permission)
- Search in files new external includes and current assembly options

"Preselect All" option in assembly setup  
Git history of file now shows blame  
Git history of file Copy Hash button  
Multiline cursor (Alt+Shift+Down to start)  
Insert sequence tool  
Automatic rc file creation (with app icon and manifest), CtrlLib: Nicer Win32 dialog icons  
PDB debugger now can use Esc scripts for pretty printing  
xfce terminal support in setup  
experimental glslc support (see upptst/vulkan)

## MacOs

Fixed Assist issues  
Fixed desktop folders (e.g. GetDownloadFolder function)  
Main menu flicker prevention

---

Subject: Re: 2025.1 (alpha)  
Posted by [Tom1](#) on Fri, 21 Feb 2025 22:39:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Hi,

Thanks for the good news! This is much sooner than I expected to see anything about 2025.1.

Will start testing...

Best regards,

Tom

---

Subject: Re: 2025.1 (alpha)  
Posted by [dodobar](#) on Mon, 24 Feb 2025 01:08:02 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Awesome, is there any docs on the animation features?  
is this expandable to colors and layouts? or drawing?

Cheers  
C

---

Subject: Re: 2025.1 (alpha)  
Posted by [Oblivion](#) on Mon, 24 Feb 2025 03:42:51 GMT

Hello dodobar,

Quote:Awesome, is there any docs on the animation features?  
is this expandable to colors and layouts? or drawing?

Cheers  
C

Yes you can animate both values andCtrls.

```
void Animate(Ctrl& ctrl, const Upp::Rect& target, int type = -1)
```

```
void Animate(Ctrl& ctrl, int x, int y, int cx, int cy, int type = -1)
```

Animates a ctrl to transition smoothly to a target rectangle using a specified animation effect. The animation effect type can be `GUIEFFECT_SLIDE`, which moves the control gradually to the target position, or `GUIEFFECT_FADE`, which gradually adjusts the control's transparency from fully transparent to fully opaque. If the type is set to a negative value, the function will use the global UI setting. Any other value will cause the the ctrl to jump directly to the target rectangle without any animation.

```
void Animate(Event<double> update, int duration = 100)
```

Performs GUI animation, repeatedly calling update with increasing numbers from the interval 0..1 for duration milliseconds.

```
void Animate(Vector<Ptr<Ctrl>>& ctrls, const Vector<Rect>& targets, int duration = 100)
```

Animates the transition of multiple ctrls from their current positions to the target positions specified by the targets parameter, over a given duration. The default duration is 100 milliseconds. The number of ctrls must match the number of the target rectangles otherwise the function will silently return without modifying anything.

```
template <class T>
```

```
void Animate(Vector<T>& data, const Vector<T>& targets, Event<> update, int duration = 100)
```

Animates the transition of multiple data values from their current positions to the target positions specified by the targets parameter, over a given duration. The default duration is 100 milliseconds. The number of data values must match the number of the targets values otherwise the function will silently return without modifying anything. After each animation step Animate calls update

Best regards,  
Oblivion

---