Q&A

Q: Is it possible to select the color inside the container of a prop gadget under Release 1.3?

A: The container of a prop gadget is filled with the window's detail pen. However, this pen is also used for titlebar and menu rendering under 1.3. Under 2.0, titlebar rendering is handled separately, but menu rendering and prop gadget rendering still depend on the window detail pen.

Q: Is it possible to have a custom font in a string gadget under 1.3?

A: Not with an Intuition string gadget. You can implement your own if you like, but it's a lot of work. The string gadget's font is the screen's font, which is set when the screen is opened, and cannot be changed after that point.

Q: What exactly will the new ECS Denise do for me? I have Workbench 2.0 on an A2000 with a 1080 color RGB monitor and a Fat Agnus (1 MB chip RAM). What advantage would I get by upgrading to the new Denise?

A: The new Denise will give you a new screen resolution that can be used on any monitor. This mode, called SuperHires, which is nominally 1280 pixels wide and has a maximum of two bitplanes deep. On a multisync monitor, you could also use the ECS Denise's programmable beamsync modes to give a 640 by 480 VGA type screen, but that's not usable on a 1080.

If you have a genlock, you can take advantage of the expanded genlock capabilities of Denise. These are chroma key (any pen number is transparent, not just the background color), bitplane transparency (any bit on a particular bitplane can be transparent), and border transparency toggling.

Q: I'm using GadTools to create some button gadgets. In addition to clicking on the gadget, the user can activate the gadget by hitting its associated key on the keyboard. Each of these gadgets has an underscored alphanumeric character indicating which key activates the gadget. When the user clicks the gadget, the visual image of the button gets pushed. Can I make this happen when the user hits the gadget's key?

A: This is not possible with GadTools, yet.

Q: How do I detect that a disk is ''not a dos disk''?

A: You should be able to detect it like this:

1. Save ((struct Process*) FindTask(NULL))->pr_WindowPtr.
2. Set pr_WindowPtr to -1 (Suppress all requesters).
3. Access the disk. You should get an error without a requester for a non dos disk.
4. Check IoErr().
5. Put thevalue saved in step 1 back into pr_WindowPtr.

Q: The ColorSpec structure defined in <intuition/intuition.h> claims that, for each UWORD of each RGB component, only 6 bits are recognized under V36. Is this true?

A: No it isn't. It should say that only the *bottom* 4 bits are recognized under V36.

Q: What dos.library functions can a Task call?

A:

`CreateProc()`
`CreateNewProc()`
`AllocDosObject()` - All current types that can be allocated via AllocDosObject() may be allocated by a task. Future types may require a process, as may future tags passed to it. If so, the Autodoc will reflect this.
`FreeDosObject()`
`Delay()`
`DateStamp()`
`CompareDates()`
`CheckSignal()`
`StrToLong()`
`SendPkt()`
`DoPkt()`

Q: I have to change the pen color of my rastport fairly often and SetAPen() just takes too long. What does SetAPen() really do, and why does it have so much overhead?

A: SetAPen() (and SetBPen() and SetDrMd()) not only sets the rastport entry, but also recalculates the minterms needed for blits. This recalculation is really just a big lookup table though, so it should not slow your software down too much.
Prior to Release 2, these functions were fairly slow (~700-800 cycles). Under 2.0, they've been reduced to ~200-550, depending on the drawmode used (XOR is fastest, then JAM1, then JAM2). That's not a trivial number of cycles if you have a lot of little drawings to do. You could try sorting your drawings by color so that you do all rendering in color A in one set, change the color value, do all of your color B rendering, change the color value, etc.

If you don't need too many colors, a better solution might be to clone your rastport. By making an exact duplicate of a rastport for each color, you can change the various rendering parameters of one individual rastport (like the APen) without effecting the drawing parameters of other rastport clones.

Q: How do I set the secondary error code of my application?

A:

```
void main(void)
{

    struct Process *proc;

    [...]

    /* The main body of the application goes
here */

    [...]

    proc=FindTask(NULL);
    proc->pr_Result2=105;
    exit(10);
}
```

This sets the secondary error code to 105. Unless there is a primary error to return, the secondary error is meaningless.

Note that exit() may call DOS functions, which could overwrite the result2 you wanted. This depends upon the startup code you use to compile the application. Both the Commodore startup code and the current SAS 5.10 startup will preserve your error code.