Q: *How do you go about attaching a console to an Intuition window on a custom screen?*

A: That depends on what you mean by a console. You can open the *console.device* in the following manner:

```
iorequest->io_Data = yourwindowptr;
iorequest->io_Length = sizeof(struct Window);
OpenDevice("console.device", unit, iorequest,
0);
```

where yourwindowptr = a pointer to the window you opened, iorequest is a ptr to an iorequest you allocated with CreateExtIO(). Using this method, you have to send and receive data from the cosole.device just like any other Exec device (`CMD_READ`, `CMD_WRITE`,...). For more information on this, see the console device section of the RKM:Libraries & Devices Manual

Under 2.0, you can attach a console file handle (CON:) to a window by doing:

```
win = OpenWindow(...)
sprintf( str, "CON://///WINDOW0x%lx", win );
confh = Open( str, MODE_OLDFILE );
```

Using this method, an application can write to the console using standard I/O routines like printf(). The window can be on a custom screen with no problem, just make sure the window is `SIMPLEREFRESH`. When you Close(confh), the window closes too, so only call CloseWindow(win) if the Open() failed.

Q: *How does the program More determine it was launched into the background?*

A: from *<libraries/dosextens.h>*:

```
LONG   cli_Background;  /* Boolean; True if
                           CLI created by RUN  */
```

Q: *Is it OK to put a pointer into a BOOL to test the pointer's validity?*

A: No. BOOL as defined in *<exec/types.h>* is only 16 bits wide, so stuffing a pointer into a BOOL (or returning one as a BOOL) can cause a pointer to look like a `NULL` pointer if the address happens to fall on a 64K boundary.

Q: *How do I determine the ROM version of my A2620/30 card?*

A: Boot with both mouse buttons down. Keep holding. Press ‘‘Shift M’’. Let go of the mouse buttons. Type ‘‘version’’ and hit return. If the board has the -06 ROMs, it will print ‘‘01/15/91’’.

The selection screens look identical to older ROMs and is identical under 1.3 and 2.0. You must hold both mouse buttons down in order to get the screens. One button does not enter the ROMs.

Q: *Can I render directly into a requester's ReqLayer->rp?*

A: Yes, as long as the requester has no `ENDGADGET` gadgets. The reason being that if the user clicks on an `ENDGADGET` gadget, then the requester will go away, and you will be notified through a message (`GADGETUP` or `GADGETDOWN` and `REQCLEAR` if you like). However, like all messages, that information arrives asynchronously. This means that the requester is gone by the time you find out about it. In particular, the requester's ReqLayer and associated RastPort can vanish while you're trying to render into them. The upshot is that it is unsafe to render into the ReqLayer->rp if you have `ENDGADGET`s in that requester.

Q: *On page II-23 of the* Amiga Mail *article ''AmigaDOS Packet Interface Specification'', the article mentions that a handler can disguise an IO request it sends to its underlying device to look like an incoming packet, but it doesn't mention how. How do you do it?*

A:Let i be a pointer to an IOStdReq. Let p be a pointer to a DosPacket. Make the corresponding structures point to each other with the following assignments:

```
i->io_Message.mn_Node.ln_Name = (char *) packet
p->dp_Link = (struct Message *) i
```

Then just SendIO(i). The reply will look just like a packet.

Q: *How do I know what default font to expect when I open a screen or window?*

A: Here is a chart to help...

What you tell OpenScreen	Screen Fonts	Window RPort's Font
A. NewScreen.Font = myfont	myfont	myfont
B. NewScreen.Font = NULL	GfxBase->DefaultFont	GfxBase->DefaultFont
C. {SA_Font, myfont	myfont	myfont
D. {SA_SysFont, 0}	GfxBase->DefaultFont	GfxBase->DefaultFont
E. {SA_SysFont, 1)	Font Prefs Screen text	GfxBase->DefaultFont

Notes:

A and B are the options that existed in 1.3.
C and D are new 2.0-expressions equivalent to A and B respectively.
E is a NEW option for 2.0.
GfxBase->DefaultFont should always be monospace. This is the ''System default text'' from Font Preferences.
Font Preferences ''Screen text'' can be monospace or proportional.
'myfont' can be any font of the programmer's choosing, including a proportional one. This is true under 1.3 and 2.0.