C Notebook: Hints & Queries

Compiled and Linked by Lee Calcraft

PRINTING A FILE DIRECTLY

I want to print some text from an application which I am writing, and would like to avoid having to drop a save box on to the printer driver. Can you help?
Paul Witheridge.

To achieve your goal you will need to engage in a four-part message dialogue with the printer driver. First you should broadcast a PrintSave message (0x80142). Since this is broadcast, you do not need to know the driver s task handle. If the message is bounced, you should complain that there is no driver present.

The driver will respond with a PrintFile message (0x80140) - or PrintError (0x80144) if the driver is a RISC OS 2 type, and the printer is busy. Your application should ignore the PrintFile message (which has only been retained by Acorn for RISC OS 2 compatibility), and wait for a DataSaveAck message from the driver.

You should then save your file to the name supplied with the DataSaveAck message, and send a DataLoad. The driver will respond with DataLoadAck to indicate that the file is in the print queue.

SIMPLE SCREEN CLEARS
Atle Mjelde Bardholt
The following simple routines can be used to clear the text and graphics screens respectively:

```
#include "kernel.h"
void clear_screen(void)
{
 _kernel_oswrch(12);
}

void clear_graphics(void)
{
 _kernel_oswrch(16);
}
```

RISC_OSLib s DRAG_A_SPRITE
Lee Calcraft
Quite simply - DragASprite is not implemented even on the latest release versions of RISC_OSLib - but watch this space for a possible solution.

CONTRIBUTING TO DESKTOP BOOT FILES
David Pilling
When the user creates a Desktop Boot file from the Acorn icon on the icon bar, the Wimp messaging system is used to ask all applications if they want to contribute. The idea is that applications will add a line to the boot file so that they will be run each time the machine boots up.

You can achieve this by responding to the broadcast message as follows:

```
/* type 17 & 18 messages */

void message(void)
{
 switch (wimpevent.data.msg.hdr.action)
      {
        ......
           break;

      case 10: write_boot_file();
           break;
  etc.
}

void write_boot_file(void)
{
 char string[256];
 int handle;

 handle=wimpevent.data.msg.data.words[0];
 sprintf(string,\
       "Run %s\n",getenv("My_App$Dir"));
 fs_write(handle,string,strlen(string));
}

os_error fs_write(int handle,\
                      void *string,int len)
{
 _kernel_swi_regs r;
 os_error *e;
 r.r[0]=2;
 r.r[1]=handle;
 r.r[2]=(int) string;
 r.r[3]=len;
 e=_kernel_swi(OS_GBPB,&r,&r);
 return e;
}
```

Please send us your C hints - all published hints will be paid for.

caption

caption

caption