



The program calls CachePreDMA() and starts an asynchronous DMA write to the second buffer.

The program uses the CPU to write to the last two long words of the first buffer which causes the CPU to load the end of first buffer and the beginning of the second buffer as they are part of the same cache line.

The DMA write writes to the beginning of the second buffer and finishes the transfer.

The program calls CachePostDMA(), which causes the CPU to write the cache line that overlaps the first and second buffer.

Because the overlapping cache line still contains the data that was in the second buffer before the DMA transfer took place, the CPU overwrites the the first two long words of the second buffer.

The current version of *68040.library* (37.4) has the code needed to work around this problem. Note that if any developer used SetFunction() to patch some of the cache control functions, it is important to patch all of these functions with SetFunction() as needed to work around this problem. *68040.library* already does SetFunction() all of the cache functions to deal with copyback issues and this one just happens to be the most complex of them. Version 37.4 of the *68040.library* is available through the CATS closed developer conferences on BIX.

---

*Q: I'm using ExAll() (dos.library 37.44) with ED\_COMMENT. Everything works fine except that the string ead->ed\_Comment points to isn't NULL-terminated.*

*Is this a bug or am I doing something wrong? If it's a bug, is there a more simple workaround than looking at the FileInfoBlock?*

A: This is a bug. It turns out that the V37 FFS does the comment wrong when in ExAll(). It does it as a BSTR rather than a standard C-String. Other filesystems (such as RAM) do it right. Also, filesystems that do not directly support ExAll() but have DOS simulate ExAll() do it right, too.

---

*Q: The 37.4 DOS library Autodoc says that the FPutC() function looks like this:*

```
LONG FPutC(BPTR, UBYTE)
```

*while version 37.4 of the <clib/dos\_protos.h> include file looks like this:*

```
LONG FPutC( BPTR fh, unsigned long ch)
```

*Which one is right?*

A: Would you believe neither? It's really:

```
LONG FPutC(BPTR, LONG ch)
```

There's no particularly good reason for it, other than that's how BCPL did it, and this is a transliterated (directly translated) version of the BCPL. BCPL has only one real type: LONG.

It wouldn't make a difference except that the value returned for success is the longword passed in, and longword comparisons are done against things like '\n', etc. So you must pass in a longword value 0-255.

---

*Q: How do I determine the frequency of the CIA B time of day (TOD) counter?*

A: The CIA B time of day counter counts horizontal sync pulses. The counter rate varies depending on the video mode the Amiga is currently displaying. Instead of using the TOD counter, you want use one of the more conventional CIA timers, and scale the time constants based on whether the system is PAL or NTSC.