FlushDataCache Flush the data cache

#include <Memory.h> **Memory Manager**

```
void            FlushDataCache( );
```

The **FlushDataCache** function flushes the current contents of the data cache. Because flushing this cache degrades performance of the CPU, this routine should be called only when absolutely necessary.

Notes: Be sure to check that the trap _HWPriv is implemented before calling this routine. See **Using the Gestalt Manager** for sample code that shows how to determine whether a trap is implemented. If you call this routine and _HWPriv is not implemented, you application will crash.

A cache may contain stale data whenever information in main memory is changed and that information is already cached in the microprocessor's data cache. The MC68030 has a **write-through** cache. This means that any data written to the cache is immediately written out to main memory. You need to flush a write-through cache if you write directly to main memory. This can cause the information in the data cache to become stale.

The MC68040 has a **copy-back** cache. Any data written to the cache is written to main memory only when necessary to make room in the cache for data accessed more recently or when the cache is explicitly flushed. A **copy-back** cache needs to be flushed if you write data that is to be read by non-CPU devices that access RAM directly. A **copy-back** cache can also lead to use of invalid instructions if the stale data in RAM contains executable code. When fetching instructions, the CPU looks in the instruction cache and, if necessary, in main memory, but not in the data cache. See **FlushInstructionCache** for a code example that flushes first the data cache and then the instruction cache of a processor with a **copy-back** cache, such as the MC68040. This operation should be performed whenever self-modifying code is used.