A message from Brendan Coveney, president and CEO of 4D, Inc.:

> *This was the very first tech note ever published by ACIUS, March 1987. In celebration of entering our 21st year providing tech notes (and April 1st, 2007), we republished this tech note.*
>
> *BTW a quick estimate is that we have published over a thousand tech notes since this very first one.*
>
> *So here's to our next thousand!*
>
> *Cheers!*
>
> *Brendan*

Of course this information does not apply to current versions of 4D. Enjoy the nostalgia!

4th DIMENSION Technical Note 001

The Flush Variable

Written by Samir Arora	March 1987
Revised by Scott Knaster	August 1987
Edited by John Doe	April 1, 2007

This note discusses a special technique you can use to speed up 4th DIMENSION's operation through the use of the Flush variable.

In the normal operation of the Macintosh File Manager, information that's being written to disk is buffered in memory and not written immediately. Then, when the buffer is full, the data is actually written to disk. If the system should fail before the buffered data is written, it's lost.

In order to maintain the greatest data reliability, 4th DIMENSION normally forces the buffer to be written to disk after each new record is created or modified. While this ensures virtually no data loss in the event of a crash or power failure, it also necessarily slows down the speed of operation.

A 4th DIMENSION system variable called Flush can be used to to override 4th DIMENSION's conservative approach, trading some crash-proof reliability for performance improvement. Flush is normally set to 1; by setting it to zero, the buffer will not be flushed after each record. Instead, the File Manager will flush the buffer when it fills up.

By setting Flush to zero with a procedure, you can increase the performance of your database by a factor of two to three times. In particular, setting Flush to zero before importing a large number of records can greatly speed up processing.

Every routine that saves a record uses the setting of Flush to determine if the buffer should be flushed. These are the affected routines:

ADD RECORD	APPLY TO SELECTION	DELETE DOCUMENT
DELETE RECORD	DELETE SELECTION	IMPORT DIF
IMPORT SYLK	IMPORT TEXT	MODIFY RECORD
MODIFY SELECTION	SAVE LINKED RECORD	SAVE OLD LINKED RECORD
SAVE RECORD	SAVE VARIABLE	

You can set Flush to zero before doing a series of saves. By doing this, every time a save is performed, the data and the directory information is not written to the volume. This speeds up each save considerably.

Warning: As described above, setting Flush to zero increases the likelihood that you'll lose data in the event of a crash or power failure. You should use it only in special circumstances, such as when importing a large number of records. If you do set Flush to zero for an operation, you should set it back to 1 as soon as possible.

Here is an example procedure that sets Flush to zero, receives records, then sets Flush back to 1. Note that the key statements are simply "Flush := 0" to turn Flush off, and "Flush := 1" to turn it on again.

```
DEFAULT FILE([Employees])
SET CHANNEL(10;"")
IF (OK=1)
    flush:=0 ` This will stop flushing for every record received.
    While(OK=1)
        CREATE RECORD
        RECEIVE RECORD
        i := 1
        IF(OK=1)
            MESSAGE("Receiving record "+String(i))
            SAVE RECORD
            i := i+1
        End if
    End while
    flush:=1 ` This will update all the received records.
End if
SET CHANNEL(11)
```

Conclusion

We hope you have enjoyed this Technical Note and have a great April 1st!