

## **UniData shared memory corruption occurs with smm file system table overflow.**

UniData shared memory corruption occurs if the internal file system table overflows. This corruption can trigger a database crash on a system with the Recoverable File System (RFS) enabled. Non-RFS systems may experience hangs or file corruption.

UniData maintains a table in shared memory that records information about your system's file systems and the available space in each file system. This table is used to quickly determine if a dynamic file has room to expand in its current directory or if a new part file needs to be created in another file system. The table is loaded when the database starts and is refreshed regularly by the smm daemon. The size of this table is determined by the udtconfig parameter N\_FILESYS. The default size is 200 entries.

If udtconfig N\_FILESYS is configured smaller than the actual number of file systems on your system, when smm loads the file systems into shared memory - it can corrupt shared memory. For RFS systems, the specific part of shared memory corrupted is where the CM message queue ID is stored. This can result in RFS being unable to start or trigger a database crash.

If N\_FILESYS is configured too small when the database starts up, the following message is written to smm.errlog:

```
Too many file system, Enlarge the N_FILESYS.
```

If N\_FILESYS is large enough when the database starts, but additional file systems are mounted later by the administrator, the 'Too many file system' message may not be recorded when smm refreshes the file systems in memory. Corruption can still occur, however.

This shared memory corruption can trigger an RFS database crash accompanied by the following example error in \$UDTBIN/smm.errlog:

```
Thu May 4 16:26:07 CM message Q (795373421) has been removed.  
Thu May 4 16:26:07 Exit: SMM error while checking process groups.  
Thu May 4 16:26:07 SMM bringing the system down.
```

**Note:** The number in the error message (795373421) will vary depending on the name of the file system that has corrupted this part of memory.

If you do not have RFS enabled on your system, the shared memory corruption may be limited to an area of memory that is reserved for RFS, but not used for non-RFS systems. In this case, there would not be any problem experienced on your system. If the number of file systems mounted since the last table refresh results in significant overflow beyond the end of the table, other (non-RFS) shared memory structures could be damaged. Possible problems on a non-RFS system could be process or database hangs - or UniData file corruption.

This shared memory corruption has generally only been experienced by customers who temporarily mount unique file systems. Even though the total number of file systems mounted at any point in time may be smaller than the internal table size, you can still experience corruption. Once a file system has been mounted, that entry remains in the table until UniData is restarted - even after the file system has been unmounted.

This problem is recorded as U2 engineering case 8760. It is corrected in UniData releases beginning with 6.1.19 and 7.1.5.

If you are running an earlier release, please check and regularly monitor your file system table with the following command:

```
$UDTBIN/sms -F
```

If your table has filled, you should increase `udtconfig N_FILESYS` and schedule a time to stop and restart UniData. The memory required for each slot in this table is not excessive. For example, changing `N_FILESYS` from 200 to 1000 resulted in expanding the shared memory segment where this table is stored (`smm-ctl`) by only 109 KB. If your table is full or close to full, you should increase `N_FILESYS` to provide a significant number of available slots to be sure you avoid this problem.

Here is an example of the 'sms -F' output. You only need to monitor the first line of the display - making sure the 'Used' number is smaller than the table size.

```
# $UDTBIN/sms -F
```

```
File system table size (N_FILESYS):200, Used: 16
```

Device ID	Available Space (in 512 Bytes)	NFS	File System Name
8388608	1342164	no	/
78118912	0	no	/devices
78381057	0	no	/system/contract
78643200	0	no	/proc
78905345	0	no	/etc/mnttab
79167489	3871648	no	/etc/svc/volatile
79429633	0	no	/system/object
8388611	10253558	no	/usr
81002497	0	no	/dev/fd
8388613	1551478	no	/var
8388614	1813040	no	/tmp
79167490	3871648	no	/var/run
8388612	4607316	no	/opt
8388616	50977908	no	/disk1
8388615	19945936	no	/home
8388617	57725232	no	/disk2