

## Dynamic SORTWORK Datasets

Historically work datasets used by a SORT program (SORTWRKxx datasets) were needed by SORT in order to reduce the amount of CORE used by SORT to sort large files. It was incumbent on the programmer to code both the correct number and the correct size of the sort work datasets. The space for these datasets had to be allocated in contiguous 'chunks'.

One of the strange quirks of any SORT program (DFSORT, SYNC SORT, CA-Sort etc) was that only the first physical extent of each SORTWRK dataset was ever actually used by sort program. It is because of this fact that you may encounter some SORTWRK datasets with CONTIG coded in the SPACE parameter. Given that SORT will only ever use the first extent of a SORTWRK dataset you may well wonder what happens if you code a secondary space allocation on the DD card. The answer is pretty simple...nothing happens.... SORT literally ignores the secondary allocation. In other words, coding a secondary space allocation has no effect when used on SORTWRK DD cards. If a sort failed due to lack of SORTWRK space then the programmer would either add another SORTWRKxx DD card or increase the space specified on the existing SORTWRKs. This has a couple of potential problems.

The first and most obvious drawback of doing this is that the required space may not be available at the time that the job runs and then the job will fail when it is executed. The job may run perfectly on the rerun as often the disk space availability will vary depending on what other jobs are running at the same time as your sort job runs. The second, and much less obvious, drawback is that increasing the amount of primary space specified on the SORTWRK DD card may, in fact, have absolutely no effect! This is because of the way that MVS actually allocates disk datasets.

The most important point to remember is that SORT only uses the **first** extent of a SORTWRK dataset, however MVS will use up to 16 extents in order to satisfy the primary space requirement. (This is the reason why CONTIG is often used for SORTWRK datasets). In other words, if a disk is so fragmented that a file with a 100-cylinder primary space request can only be allocated with a 60-cylinder first extent and a 40-cylinder second extent to satisfy the primary space request, then SORT will only use the first 60 cylinders!

For some time now there has been a feature of most, if not all, SORT programs that will instruct the SORT to dynamically allocate its own sort work space. Not only that, but if it detects that it requires even more work space it will dynamically allocate more as and when it needs it.

The use of dynamic SORTWRKs will reduce the number of disk space related JCL errors within sort jobs and additionally it may even reduce space related JCL failures in jobs running alongside SORT jobs.

To utilize dynamic SORTWRKs you need your System Programmer to ensure that (for DFSORT) the install option **DYNAUTO=IGNWKDD** has been set then simply remove any SORTWRK DD cards from within the job step executing SORT! It couldn't be easier, and it could save time, money and jobs from failing.

No JCL deck should need to contain SORTWRK's in the SORT steps these days.

On a side note, no software that stops x37 abends is able to make SORT use extra extents.

