CSI
INTERNATIONAL

White Paper
CSI's VSUM = VSAM Optimization

By: John Mycroft
Office of the CTO
CSI International

CSI
INTERNATIONAL

VSAM - the word strikes dread into the minds of many. How to define a dataset correctly? How to correct the definition of a dataset that needs it? What's the best value for CISZ, freespace and and all those other parameters? How to tell which datasets are causing you problems?
Now I won't say that the LISTCAT report is the most unusable of any utility program but I won't disagree with anyone who does.

So what often happens is that we get hold of someone else's DEFINE job and, with as few changes as possible, use it forever with little thought as to what it means. The result is that we wind up with a load of wasted disk space and files defined with bad performance parameters. We put FREESPACE (10 20) on every dataset definition regardless of what that means. (It means that close to 30% of our dataset is empty space - no big deal for a 1 cylinder dataset but not too good for a dataset occupying four 3390s that never has insertions made to it). As Robert Heinlein didn't say, "There ain't no such thing as "free" space."

The results of poor VSAM parameters include higher hardware costs (you have to pay for disk drives whether they are full or empty), slow performance and repeated reorgs that serve no useful purpose. Given that everyone is trying to contain costs, a software tool such as CSI's VSUM can identify datasets that are causing problems and help decide on better definition parameters to ensure efficient use in the future. VSUM works both online or in batch.

Let's look at a few examples. For each of these, the usual procedure would be to run a LISTCAT of a catalog or several and eyeball them, calculator in hand, for what looks a likely problem area. Gee, that sounds like a job for a computer.

We're running out of disk space but, before we rush out and buy a load more DASD, are there any datasets we could trim back? With VSUM, we can easily identify space that has been allocated but isn't used. We can start with a simple LIST job with a FREECYLINDERS parameter to give us a report of all those datasets with more than some number of empty cylinders. Or run an EMPTY report to show us all the empty datasets we have.

Moving on from there,VSUM can do a detailed analysis of the contents of a KSDS to show up any errors made when the dataset was defined. This can include control intervals in the dataset that cannot be accessed because the index CI size is too small. This function alone can give the information necessary to halve the size of a poorly defined dataset simply by making it all indexable.
How about a report of datasets that haven't been updated since 1999? They are probably candidates for the trash can.

A common procedure with VSAM datasets that experience poor performance is to reorganize them and hope for the best, replicating all the original allocation parameters and often undoing all the good work we paid for with CI and CA splits. But what if those original allocation parameters were less than ideal? VSUM's ANALYZE function can read the whole dataset or part of it to tell us information vital to correct dataset allocation. A good example is the average and maximum record lengths used by VSAM to calculate, among other things, freespace in a CI or CA. Often the subject of guesswork, bad values here can lead to huge amounts of wasted space or totally useless freespace in the dataset. VSUM also does the simple but tedious calculation of the logical to physical IO ratio. It will also go beyond merely reporting on the current state of the file and tell us what values we should use next time the file gets loaded or that we can use IDCAMS to alter right now.

Considering the huge sums of money tied up in VSAM, VSUM provides a rapid way of improving performance, reclaiming wasted space and optimizing application performance. Its functions and reports are of equal value to the VSAM novice and to the experienced DBA.