

## **Enlarging a CA IDMS Database**

**Dick Weiland**

**CA Technologies**

**Principal Software Engineer**

It is inevitable that at some time during the career of a DBA they will be faced with the task of expanding the size of one or more areas of a CA IDMS database. The level of criticality of this operation and how it will be accomplished depends on a number of factors such as how suddenly the need to enlarge the database was discovered and the timeframe in which the enlargement must occur. Current physical characteristics of the database and the volume of data to be processed will also have an impact on the procedures that a DBA may decide to implement.

There are only two ways to expand a CA IDMS database area, increase the page size of the area or increase the number of pages that are included within the area. However there are a number of ways to make these changes where each method has results that differ in the amount of time required to perform the procedure and the physical characteristics of the database area at their completion. These physical characteristics will primarily impact the performance of applications running against the expanded database areas.

Within this paper we will examine the various options available to a DBA. Expansions that are generally considered as short-term remedies are the expansion of an area's page size using the EXPAND PAGE utility and an extension of an area's page range using the EXTEND SPACE procedure. These options will be followed by the long-term solution of unloading and reloading the affected areas of the database which may be performed using the UNLOAD/RELOAD, DB-REORG, or REORG utilities. Finally the concepts of Mixed Page Group Support and database segmentation will be introduced.

### **Page Expansion**

The size of a page used within a CA IDMS area is effectively limited by the size of the physical device on which the area is stored which is approximately 32K. As a result any area that is running out of space and has a smaller page size than 32K is a candidate to be enlarged by increasing its page size. The quickest way to accomplish this is to execute the EXPAND PAGE utility against the target area.

EXPAND PAGE requires that new files be allocated for the enlarged database. The utility reads a page from the original physical file with the smaller page size and writes it to the new file with the larger pages. The data from the old page is moved into the top of the new page while the line index space is

moved to the end of the page. This results in additional free space in the middle of the new pages. Diagram 1 illustrates this process.

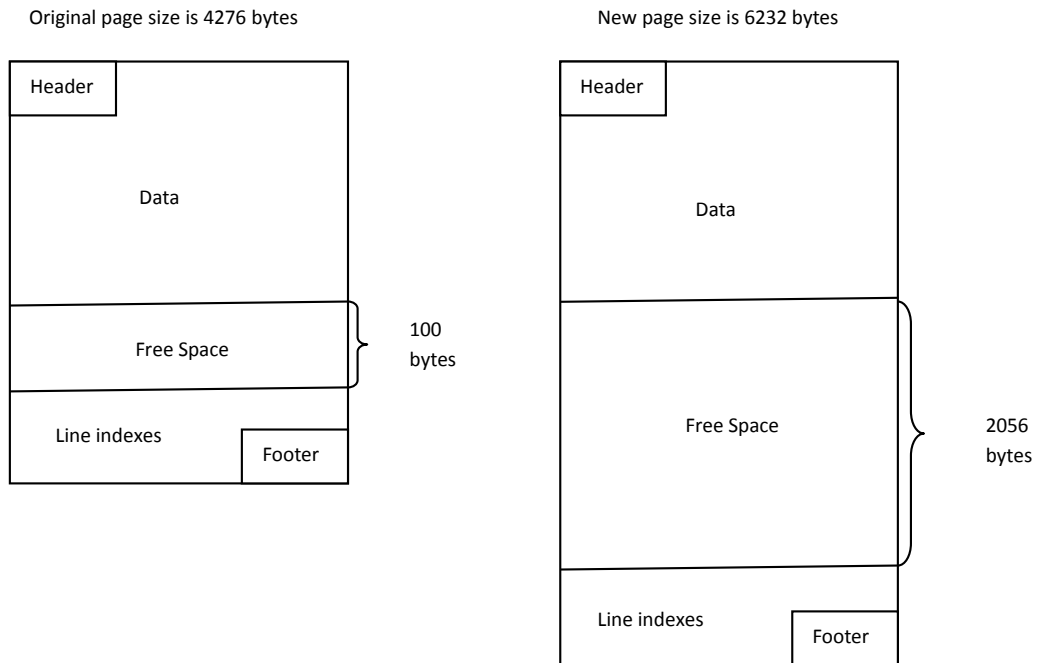


Diagram 1

The EXPAND PAGE utility works on a file basis. That means that if an area maps across multiple files, you must expand each file containing a portion of that area separately. In addition, if an area is mapped to a file that contains multiple areas, all areas in that file will need to have their page size increased or the unaffected areas must be moved to another file whose block size matches the page size of those areas.

Expanding the page size of an area also affects the Space Management Pages (SMP) within the area. The first page of an area is an SMP. After formatting the header and footer areas of the page, the remaining space is divided into halfword fields, each representing a page within an area. The number of pages that can be controlled by an SMP is determined by the following formula.

$$(\text{Page size} - 32) / 2$$

If an area's page size is 4276 the first SMP in the area can control 2,122 pages. The 2,123<sup>rd</sup> page in the area is also formatted as an SMP and can in turn control 2,122 pages. Since the EXPAND PAGE utility is changing the page size of the area it would also change the number of pages controlled by an SMP which could lead to a number of complications if allowed to occur. To prevent these problems it is necessary to add the ORIGINAL PAGE SIZE clause into each area definition within the SEGMENT that is having its page size expanded. That clause must remain within the area's definition with the page size that was specified when the area was first defined until an unload/reload operation is performed against the area.

The major advantage of executing the EXPAND PAGE utility is that it is a rather quick process. The procedure requires modifying the area definitions within the associated segment and regenerating the DMCL prior to allocating the new files and running the EXPAND PAGE utility. The time required to execute the utility is roughly the amount of time needed to read each page in the old area and write each new page in the new area. This results in the EXPAND PAGE utility being a favorite of DBAs when a database unexpectedly runs out of space in the middle of application processing.

However EXPAND PAGE is considered a short-term solution when a database must be enlarged. As an area tends to fill, CA IDMS has had to do more work to find available space for new or expanded record occurrences. This typically results in a reduction in the performance characteristics of that area. EXPAND PAGE simply moves the data from a page in the original area to the corresponding page in the new area. It does nothing to reduce overflow conditions or the existence of fragmented variable length records created due to the earlier space shortages. It is recommended that areas that have been the target of an EXPAND PAGE execution should be scheduled for a full unload and reload when time permits to resolve these negative performance conditions.

Documentation on running the EXPAND PAGE utility can be found in the CA IDMS Utilities Guide.

## **Extend Space**

Extend space is a procedure and not a CA IDMS utility. The procedure allows a user to extend the number of pages within an area's definition by adding additional pages that are contiguously numbered from the last page currently defined within the area. To extend an area's page range does imply that the desired pages to be added to the area are available and not currently assigned to any other area within the database.

The name of the procedure is derived from the clause in the area's physical definition that implements the extension. The following syntax is an example of a typical area's definition within its database SEGMENT.

```
CREATE  
PHYSICAL AREA EMPDEMO.EMP-DEMO-REGION  
PRIMARY SPACE 50 PAGES FROM PAGE 75001
```

```
MAXIMUM SPACE 50 PAGES
PAGE SIZE 4276 CHARACTERS
WITHIN FILE EMPDEMO
FROM 1 FOR ALL BLOCKS;
```

Note that the EMP-DEMO-REGION area is limited by the MAXIMUM SPACE clause to 50 pages. The area will initially contain all 50 pages as defined by the PRIMARY SPACE clause and these pages will be numbered starting from page number 75001. All of the pages will be mapped to a single file with the internal CA IDMS name of EMPDEMO.

The first step would be to define a new file. For this example we will assume that this was done and the new file was given the name of EMPDEMOX. Next you must increase the maximum number of pages that can be contained within the area. This step must be performed prior to the space extension. The following syntax is an example of the maximum number of pages that can be contained in the area being increased to 75.

```
ALTER
PHYSICAL AREA EMPDEMOX.EMP-DEMO-REGION
PRIMARY SPACE 50 PAGES FROM PAGE 75001
MAXIMUM SPACE 75 PAGES;
```

After increasing the maximum number of pages allowed in the area, the area's page range can be extended. The following syntax shows the EMP-DEMO-REGION area being expanded by 25 pages and those pages being mapped to the new file.

```
ALTER
PHYSICAL AREA EMPDEMOX.EMP-DEMO-REGION
PRIMARY SPACE 50 PAGES FROM PAGE 75001
MAXIMUM SPACE 75 PAGES
EXTEND SPACE 25 PAGES
    WITHIN FILE EMPDEMO
        FROM 1 FOR 50 BLOCKS
    WITHIN FILE EMPDEMOX
        FROM 1 FOR 25 BLOCKS;
```

All DMCLs containing this area must be regenerated and new load modules created. Once that is accomplished, one of the newly created DMCLs would be used to format the new file using the FORMAT utility. After the format is performed the area's extension is available for use.

Extending the space for an area is also considered to be a temporary solution. As with an expand page, the resulting database still contains the same amount of overflowed records and possible VLR fragments. The negative effects these items will have during application processing will still be present in the extended database and may grow worse over time as all of the available space will be in the new pages added to the area. This may cause VIA clusters to be extended across additional pages as well as

VLR fragments being created in the new page with little hope that the variable length records will have their existing fragments consolidated. Areas that have been extended should be scheduled for a full unload/reload operation at the earliest possible time.

Extending the space of an area can also have a further complication which must be considered. When an area contains CALC records, the number of pages in the area is one of the factors used in the calculation that determines the page on which the record is to be stored. Since an extend space procedure increases the number of pages in the area, CA IDMS must take precautions to insure being able to locate previously existing CALC record occurrences.

To avoid losing the ability to locate existing CALC record occurrences, CA IDMS will retain the primary number of pages as the number of pages into which a CALC record can be targeted. When a new CALC record is stored and the page to which it is targeted has insufficient space to contain the record, the new record will be overflowed into what is likely to be one of the newly added pages. However, the new record is still associated with its targeted page within the original page range. To access this record CA IDMS will be required to access the original target page and minimally the new page to which the record is overflowed to complete the requested retrieval. Diagram 2 provides a representation of the pages used for CALC records in an extended area. In this example the original page range contained 5 pages and was extended to a 7 page area.

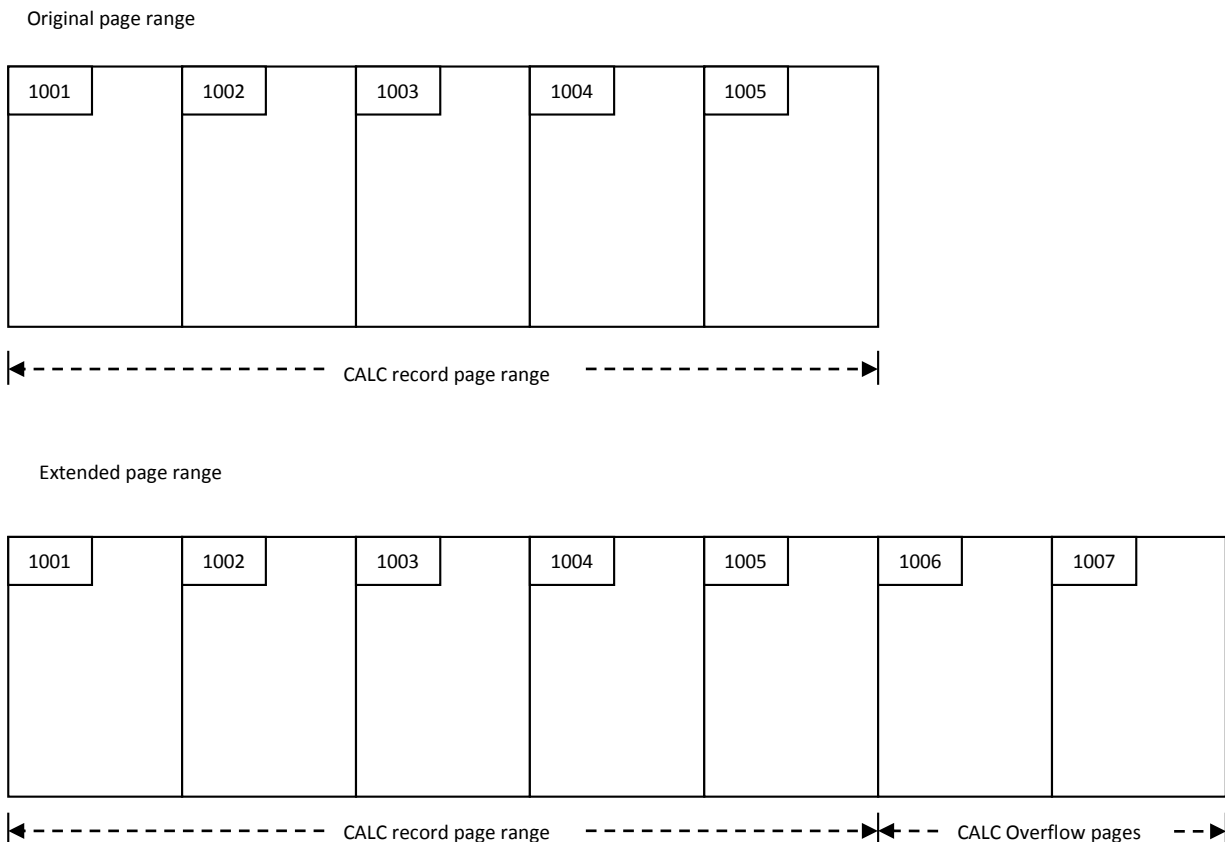


Diagram 2

Because of this difference between the area's entire page range and its CALC page range, it is critical that the original PRIMARY SPACE clause be retained within the PHYSICAL AREA statement until the area has undergone an unload/reload process. Failure to do so will result in minor codes of 26 being returned whenever a CALC retrieval is attempted.

### **Unloading and Reloading**

The only process that should be considered a long-term solution for enlarging a CA IDMS area is a full unload and reload. When performing an unload/reload you have the opportunity to alter the area's page size, its page range, and the number of records that can be stored per page. In addition the following changes can also be made to a non-SQL defined database when performing an unload/reload procedure.

- Change page ranges for records within an area
- Reassign record types to different areas
- Change record location modes
- Store VIA records by means of a different set as long as the record type is already a member of the set when the unload is performed
- Change compression and INDEX BLOCK COUNTS for indexes
- Change area and page range assignments for system-owned indexes

The process of doing a full unload and reload of the database is considered to be a long-term solution since all of an area's data is removed from the database and written to intermediate files. The data is then written back to the new area(s) in a manner that maximizes data clusters and minimizes VLR fragments. Indexes are also rebuilt creating an efficient structure from which future application processing can begin. The drawback of doing an unload/reload process is that it involves a great deal more work than an EXPAND PAGE or an EXTEND SPACE and therefore can run a significantly longer period of time.

CA Technologies provides three sets of utilities that can be used to perform an unload/reload process. The first option is the UNLOAD and RELOAD utilities which work together to perform the operation. These utilities have been available since the inception of CA IDMS and are part of the standard installation of the product. Another utility that has been available for some time is DB-REORG. It is a part of the DB Toolkit for CA IDMS which is typically sold separately from the base CA IDMS product. Finally the REORG utility was made available as a part of the standard CA IDMS product starting with Release 16.0 SP4. REORG was created to address the time it can take to do a full unload/reload of the database by creating an environment where multiple batch jobs can work concurrently to perform the various steps of the operation.

## **UNLOAD/RELOAD**

The UNLOAD and RELOAD utilities have been enlarging CA IDMS databases for over 40 years. As the names imply the UNLOAD utility is used to unload the user data, set, and index contents from the database and write it to various intermediate files. Those files are then used as input to the RELOAD utility which loads the information to the new physical environment. The UNLOAD utility requires that the database be minimally in a retrieval mode to any other copies of CA IDMS and the database must not contain any logically deleted records (LDEL). The PRINT SPACE utility can be used to determine if any LDELs exist and if so the CLEANUP utility can be used to remove those occurrences prior to the execution of UNLOAD.

In general UNLOAD will extract information for any system-owned indexes prior to unloading the data records. However, records stored VIA a system-owned index will be unloaded at that time. UNLOAD then sweeps the specified database areas unloading all CALC and DIRECT records encountered. VIA records are unloaded when their owner is accessed through the area sweep. UNLOAD has the capability to only process specific areas within the database structure to reduce the time required of the operation. It will also extract set information for any sets that cross between a specified area and one that was not selected for processing. This information is passed to the RELOAD utility so these sets can be properly rebuilt in the new version of the database.

The RELOAD utility is actually composed of a number of steps that load the user data into the new version of the database, reconcile set information passed from the UNLOAD utility with new dbkeys for the records resulting from the reload operation, rebuilding any affected system indexes, and updating the prefixes of records with set pointers reflecting the new database structure. These steps may be executed as individual batch job steps along with the sorts required to properly sequence the information. This mode is typically referred to as step-mode. Another option is to execute the RELOAD utility in single-step mode in which the utility will schedule and execute all sorts and reload steps without user intervention.

Details concerning the usage of the UNLOAD and RELOAD utilities can be found in the CA IDMS Utilities Guide.

## **DB REORG**

The DB REORG utility is an add-on product available through the CA IDMS Toolkit. In concept it is similar to the UNLOAD/RELOAD but provides the user with a number of features not available in the standard utilities.

A major difference is that after extracting the data, DB REORG does much of the work of assigning records to various pages within the new database through processing done against intermediate files and minimizes the number of passes required against the database. As a result users are given more control over how records target to database pages by being able to specify a priority to record types. This priority tells DB REORG which record types should be allocated to given pages first, minimizing the amount of overflow occurrences of that type. It also waits to assign overflow records to any database page until all non-overflow records have been assigned to their targeted pages. This has the potential to reduce the overall amount of overflow present within the new database.

Fragmentation of variable length records is also handled in a similar manner. A VLR that cannot fit onto a page in its entirety will not be fragmented until all other VLRs are stored. The result is that initial fragmentation of records is held to a minimum in the new database.

DB REORG does not have a requirement that the old database be free of logically deleted records. If an LDEL is encountered, the utility will resolve the condition and appropriately update any sets in which the LDEL still participated. This eliminates the need for the execution of the CLEANUP utility before starting the unload/reload process.

The DB REORG process is divided into a number of steps. Through the use of parameters, a DBA is allowed to interrupt processing at the completion of each step for the purpose of recovery and restartability in case some error occurs during a subsequent step. However the unload/reload is still a serial process which requires the database to be unavailable for the life of operation.

More details on the use of the DB REORG utility can be found in the IDMS/DB Database Reorg Option User Guide.

## **REORG**

The newest addition to the CA IDMS family of unload/reload utilities is REORG. It was created in response to the problem that very large databases can take days to unload and reload using the previously existing utilities. The REORG utility uses concurrent processing to reduce the overall amount of time it takes to unload and reload a database. REORG roughly divides the source and target databases into equal 'slices'. The number of slices to be used is controlled by the user and controls the number of parallel jobs that will be concurrently executed. To be effective you must have sufficient batch initiators available for all of the batch jobs to concurrently execute.

To control the concurrent jobs REORG creates a control file during its initialization. The control file describes the specifics of the operation and is updated with status information reflecting the progress that has been made during the operation. Each job that performs work on behalf of a REORG operation



updates the control file. Operating system facilities are used to serialize these updates and to coordinate concurrently executing jobs.

The REORG utility has been found to be very effective in reducing the overall amount of time needed to perform an unload and reload of a database. However the nature of the processing does require a bit more front-end work to set up the utility. In particular REORG will use more intermediate datasets than would an UNLOAD/RELOAD or DB-REORG execution. To reduce the effort of allocating these intermediate datasets the CREATE DSMODEL utility was also created. This utility allows a user to generate templates for the work files needed by REORG. These templates will be used during REORG setup processing to optionally dynamically allocate the necessary datasets. When performing a REORG against a database for the first time it is recommended that a user test their procedures against a copy of the database prior to the planned operation to reduce the possibility of encountering out of space conditions on their intermediate files.

The CA IDMS Utilities Guide provides full documentation for using the REORG and CREATE DSMODEL utilities.

## **Very Large Databases**

All of the areas, records, and sets for a CA IDMS database are defined within a schema. In its most basic implementation the total number of pages that a database can utilize is a bit over 16 million. This means that all of the areas defined within the database's schema must fit within 16 million pages. Over the years many databases have grown and their areas have been constantly enlarged using the methods discussed previously but they eventually have exceeded the limitation of the available maximum page range. To overcome this limitation CA IDMS users have the ability to implement page groups or database segmentation individually or in combination.

A page group is an attribute assigned to a database segment when it is defined. All areas defined within that segment are associated with the specified page group. This serves to qualify the page numbers assigned to those areas such that they are unique from page numbers in other segments. The page group for an area is an attribute that is maintained internally within CA IDMS control blocks and is not represented any place within the physical area. Mixed Page Group Support (MPGS) is a feature of CA IDMS that allows a single transaction to access multiple areas where the areas may be defined in different page groups.

When a database has exhausted its primary page range, it is possible that the areas containing that database are separated into multiple page groups. Each page group can utilize up to 16 million pages greatly increasing the amount of space available for the database. Areas within one of these page groups can then be enlarged as needed by the methods described earlier.

However MPGS support does have some limitations as to how the areas can be separated into the various page groups. The details of the limitations to the implementation and the means to work around them can be found in a technical paper entitled Mixed Page Group Implementation.

Another practice used by CA IDMS sites to allow for increases in the size of databases is database segmentation. Database segmentation is a process where the database is divided along certain guidelines making each portion of the database physically or logically independent of the others. There are two types of segmentation used which are called vertical and horizontal segmentation.

Vertical segmentation is the practice of dividing a database into separate functional portions and is implemented at the area level. Areas within the same functional portion of the database can then be grouped together into the same CA IDMS segment and if desired, each CA IDMS segment defined to the database can be assigned to a different page group. Implementation of Mixed Page Group Support typically involves some form of vertical segmentation to allow for the separation of the database's areas into different page groups. It is typically implemented when the database must be viewed in its entirety by the application.

A horizontal segmentation strategy allows the database to be divided along some site specific criteria creating multiple yet functionally equivalent databases. These smaller databases can be processed independently of each other based on a site's requirements. An example of horizontal segmentation would be the division of a company's CUSTOMER-ORDER database in two equivalent databases where one supports the business in North America while the other handles Europe. Each of these databases could have the entire 16 million page range available for its expansion.

Details about the practice of database segmentation can be obtained within a technical paper entitled Database Segmentation.

## **Conclusion**

Enlarging a CA IDMS database is a process that can be performed quickly, as in the cases of expanding pages sizes or extending areas, or can take significant time and effort when the need arises to segment the database. Each enlargement should be evaluated in terms of the time available to implement the change and the resulting performance impact on the database. When an expand page or area extension are implemented, it is strongly recommended that the affected areas undergo an unload/reload process as soon as can be scheduled to eliminate the potential of poor database performance due to overflow and fragmentation conditions.

When you complete an enlargement procedure it is also a good practice to review the possible alternatives still available to a specific database the next time the need for an enlargement occurs. It is a very unpleasant situation when an area is completely filled to realize that you can no longer continue expanding the page size or that you do not have any available pages into which you can extend an area.

Acknowledging these difficulties can provide adequate time to implement Mixed Page Group Support or database segmentation or to schedule time to unload and reload the database before an area reaches a critical level of space utilization.