

# REORDB: Reorder Database

The REORDB function reorders the entire Associator and Data Storage for a database. Files that are not specified are reordered according to their existing definitions.

```

ADAORD REORDB  [DBINDEXCOMPRESSION = { YES | NO } ]
                 [FILE = file-number ]
                 [ACRABN = starting-rabn ]
                 [ALLOCATION = { FORCE | NOFORCE } ]
                 [ASSOPFAC = padding-factor ]
                 [ASSOVOLUME = 'Associator-extent-volume' ]
                 [DATAPFAC = padding-factor ]
                 [DATAVOLUME = 'Data-Storage-extent-volume' ]
                 [DSDEV = device-type ]
                 [DSRABN = starting-rabn ]
                 [DSRELEASE]
                 [DSSIZE = size ]
                 [INDEXCOMPRESSION = { YES | NO } ]
                 [ISNSIZE = { 3 | 4 } ]
                 [MAXISN = highest-isn ]
                 [MAXRECL = record-length ]
                 [NIRABN = starting-rabn ]
                 [NIRELEASE]
                 [NISIZE = size ]
                 [SORTSEQ = { descriptor | ISN | physical-sequence } ]
                 [UIRABN = starting-rabn ]
                 [UIRELEASE]
                 [UISIZE = size ]
                 [LIP = { isn-pool-size | 16384 } ]
                 [LPB = { prefetch-buffer-size | ADARUN-lu } ]
                 [MAXFILES = maximum-number-files ]
                 [NEWDBID = database-identifier ]
                 [NEWDBNAME = database-name ]
                 [NOUSERABEND]
                 [RAID]
                 [TEST]

```

## Note:

If the parameter MAXFILES or NEWDBID is specified, the nucleus automatically terminates at the end of the REORDB function.

This chapter covers the following topics:

- Optional Parameters and Subparameters
- Examples

# Optional Parameters and Subparameters

## **ACRABN: Starting RABN for Address Converter**

The RABN with which the file's address converter extent is to begin. If this parameter is omitted, ADAORD assigns the starting RABN. The space requested must be available in one extent.

## **ALLOCATION: Action to Follow File Extent Allocation Failure**

ALLOCATION specifies the action to be taken if file extent allocations cannot be obtained according to the placement parameters ACRABN, DSRABN, NIRABN, or UIRABN.

By default (that is, ALLOCATION=FORCE), the utility terminates with error if any file extent allocation cannot be met according to RABN placement parameters.

If ALLOCATION=NOFORCE is specified and any allocation with placement parameters fails, the utility retries the allocation without the placement parameter.

## **ASSOPFAC: Associator Padding Factor**

The new Associator block padding factor. The number specified represents the percentage of each Associator block not to be used during the reorder process. A value in the range 1-90 may be specified. The remaining number of bytes after padding must be greater than the largest descriptor value plus 10.

If this parameter is omitted, the current Associator padding factor in effect for the file is used.

## **ASSOVOLUME: Associator Extent Volume**

### **Note:**

The value for the ASSOVOLUME parameter must be enclosed in apostrophes.

ASSOVOLUME identifies the volume on which the file's Associator space (that is, the AC, NI, and UI extents) should be allocated. If the requested number of blocks cannot be found on the specified volume, ADAORD allocates the remaining blocks on other volumes according to its default allocation rules.

If ACRABN, UIRABN, or NIRABN is specified, ADAORD ignores the ASSOVOLUME value when allocating the corresponding extent type.

If ASSOVOLUME is not specified, the file's Associator space is allocated according to ADAORD's default allocation rules.

## **DATAPFAC: Data Storage Padding Factor**

DATAPFAC specifies the new Data Storage padding factor, which is the percentage of each Data Storage block reserved for record expansion when the file is reordered. A value in the range 1-90 may be specified (see the ADALOD LOAD DATAPFAC parameter discussion for more information about setting the padding factor). If this parameter is omitted, the current padding factor for the file is used.

**DATAVOLUME: Data Storage Extent Volume****Note:**

The value for the DATAVOLUME parameter must be enclosed in apostrophes.

DATAVOLUME specifies the volume on which the file's Data Storage space (DS extents) are allocated. If the requested number of blocks requested with DSSIZE cannot be found on the specified volume, ADAORD allocates the remaining blocks on other volumes according to its default allocation rules.

If DSRABN is specified, DATAVOLUME is ignored for the related file.

If DATAVOLUME is not specified, the Data Storage space is allocated according to ADAORD's default allocation rules.

**DBINDEXCOMPRESSION: Compress Database Indexes**

DBINDEXCOMPRESSION indicates whether the indexes of files are rebuilt in compressed or uncompressed form. It applies to all files for which no INDEXCOMPRESSION parameter is specified.

DBINDEXCOMPRESSION can be used to build compressed or uncompressed indexes for all files of the database, making it unnecessary to specify index compression for each file.

**DSDEV: Data Storage Device Type**

DSDEV is the file's Data Storage device type. The specified device type must already be defined to Adabas, normally when the database was created or by the ADADBS utility's ADD function.

If DSDEV is not specified, ADAORD attempts to allocate the file on the device type used before reordering.

**DSRABN: Data Storage Starting RABN**

The beginning RABN for the file's Data Storage extent. If this parameter is omitted, ADAORD assigns the starting RABN.

**DSRELEASE: Release Unused Data Storage Blocks**

Specifying DSRELEASE releases unused Data Storage (DS) blocks belonging to the specified file. If DSRELEASE is not specified, ADAORD allocates *at least* the number of DS blocks that were allocated before the file was reordered.

**Note:**

Adabas calculates the file extent size using any changed padding factor or blocksize values *before* the file is reordered.

**DSSIZE: Data Storage Size**

DSSIZE is the number of blocks or cylinders to be allocated for the file's Data Storage (DS) logical extent. If the value is blocks, it must be followed by "B" (for example, 2000B).

If this parameter is omitted, ADAORD computes the file extent size in proportion to any increase or decrease in the DATAPFAC padding factor.

**FILE: File Number**

The file to which the following parameters apply. Each specified file must be on a separate ADAORD statement following the ADAORD REORDB function statement, and must be immediately followed by the applicable parameters for the file.

For any file whose number is not specified, the current Associator and Data Storage block padding factors and MAXISN value are retained, and all Associator and Data Storage space allocations remain the same.

**INDEXCOMPRESSION: Compress File Index**

INDEXCOMPRESSION indicates whether the index for the file is rebuilt in compressed or uncompressed form. A compressed index usually requires less index space and improves the efficiency of index operations in the Adabas nucleus.

If INDEXCOMPRESSION is *not* specified

- but the DBINDEXCOMPRESSION parameter is specified for the database as a whole, the default is the database value.
- and DBINDEXCOMPRESSION is also *not* specified, the default is the current compression form of the file.

**ISNSIZE: 3- or 4-Byte ISN**

ISNSIZE specifies whether ISNs in the file are to be 3 or 4 bytes long. The default is the value currently used for the file; this value is stored in the file control block (FCB).

**Note:**

It is not possible to change the ISNSIZE of a physically coupled file using ADAORD.

**LIP: ISN Buffer Pool Size**

The LIP parameter can be used to decrease the number of Associator I/O operations when recreating the address converter. For best performance, specify a size that accepts all ISNs of the largest file to be processed.

LIP specifies the size of the ISN pool for containing ISNs and their assigned Data Storage RABNs. The value may be specified in bytes as a numeric value ("2048") or in kilobytes as a value followed by "K" ("2K"). The default for LIP is 16384 bytes (or 16K).

The length of one input record is ISNSIZE + RABNSIZE. Thus the entry length is at least 6 bytes (the ISNSIZE of the file is 3 and the RABNSIZE of the database is 3) and at most 8 bytes (the ISNSIZE is 4 and the RABNSIZE is 4).

**LPB: Prefetch Buffer Size**

LPB specifies the size, in bytes, of the internal prefetch buffer. The maximum value is 32,760 bytes. The default depends on the ADARUN LU parameter. ADAORD may reduce a specified LPB value if the LU value is too small.

**MAXFILES: Maximum Number of Files**

MAXFILES specifies the maximum number of files that can be loaded into the database. The minimum value for this parameter is 3. The highest value permitted is 5000 or one less than the ASSOR1 blocksize, whichever is lower. For example, 2003 is the highest MAXFILES value for a database whose ASSOR1 is stored on a 3380 DASD.

If this parameter is omitted, the current value for MAXFILES is retained.

When MAXFILES is specified, the nucleus terminates after the ADAORD REORDB function is completed.

**MAXISN: Highest ISN Permitted in the File**

The highest ISN that can be allocated for the file. This value must be greater than the current TOPISN value displayed in the ADAREP database report.

ADAORD uses the specified value to calculate the address converter space required. If this parameter is omitted, the current MAXISN value for the file is retained.

**MAXRECL: Maximum Compressed Record Length**

Use the MAXRECL parameter to change the maximum record length, after compression, permitted in the file. Specifying MAXRECL has two effects:

- The file's DATA dataset is allocated only to devices that support the specified length.
- If the file contains Data Storage records that exceed the specified length, ADAORD abends and prints the ERROR-126 message ("Data Storage record too long").

If MAXRECL is not specified, the maximum compressed record length does not change.

**NEWDBID: Database Identifier**

NEWDBID is the ID to be assigned to the database. A value in the range 1-65,535 may be used. For systems using Adabas Online System Security, the value 999 is reserved. If this parameter is omitted, the current database ID is retained.

When NEWDBID is specified, the nucleus terminates after the ADAORD REORDB function is completed.

**NEWDBNAME: Database Name**

NEWDBNAME specifies the name to be assigned to the database. The name can contain up to 16 characters. If the name contains special characters or embedded blanks, it must be enclosed in apostrophes ('...'); for example, 'JAN''S DB'. If this parameter is omitted, the current database name is retained.

**NIRABN: Starting RABN for Normal Index**

NIRABN specifies the RABN with which the file's normal index extent is to begin. If this parameter is omitted, ADAORD assigns the starting RABN.

**NIRELEASE: Release Unused Normal Index Blocks**

Specifying NIRELEASE releases unused normal index (NI) blocks belonging to the specified file. If NIRELEASE is not specified, ADAORD allocates *at least* the number of NI blocks that were allocated before the file was reordered.

**Note:**

Adabas calculates the file extent size using any changed padding factor or blocksize values *before* the file is reordered.

**NISIZE: Normal Index Size**

NISIZE specifies the number of blocks or cylinders to be allocated for the file's normal index. A block count must be followed by "B" (for example, 2000B).

If this parameter is omitted, ADAORD computes the file extent size in proportion to any increase or decrease in the ASSOPFAC padding factor.

**NOUSERABEND: Termination without Abend**

When an error is encountered while the function is running, the utility prints an error message and terminates with user abend 34 (with a dump) or user abend 35 (without a dump).

If NOUSERABEND is specified, the utility will *not* abend after printing the error message. Instead, the message "utility TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

**RAID: Action to Follow Determination That File Does Not Exist**

The RAID parameter instructs ADAORD to ignore any FILE parameters that refer to a file that does not exist in the database.

If RAID is not specified (the default), ADAORD terminates with an error message when it encounters a FILE parameter referring to a file that does not exist in the database.

The RAID parameter is provided for use in recovery jobs built by the Adabas Recovery Aid (ADARAI).

**SORTSEQ: File Processing Sequence**

SORTSEQ determines the sequence in which the file is to be processed. If this parameter is omitted, the records are processed in physical sequence.

If a descriptor is specified, the file is processed in the logical sequence of the descriptor values. *Do not* use a null-suppressed descriptor field, a hyperdescriptor, a phonetic descriptor, a multiple-value field, or a descriptor contained in a periodic group.

**Note:**

Even when the descriptor field is not null suppressed, the record is *not* represented in the inverted list if the descriptor field or a field following it has never been initialized (held a value). Therefore, the record will be dropped when the utility is executed.

If ISN is specified, the file is processed in ascending ISN sequence. For the Adabas checkpoint or security file, only SORTSEQ=ISN is allowed.

**TEST: Text Syntax**

This parameter tests the operation syntax without actually performing the operation. Only the syntax of the specified parameters can be tested; not the validity of values and variables.

**UIRABN: Starting RABN for Upper Index**

UIRABN defines the beginning RABN for the Associator's upper index extent for the file. If this parameter is omitted, ADAORD assigns the starting RABN.

**UIRELEASE: Release Unused Upper Index Blocks**

UIRELEASE releases unused upper index (UI) blocks belonging to the file. If UIRELEASE is not specified, ADAORD allocates *at least* the number of UI blocks that were allocated before the file was reordered.

**Note:**

Adabas calculates the file extent size using any changed padding factor or blocksize values *before* the file is reordered.

**UI SIZE: Upper Index Size**

UI SIZE specifies the number of blocks or cylinders to allocate for the upper index. A block count must be followed by "B" (for example, 2000B). If UI SIZE is omitted, ADAORD allocates space in proportion to an increase or decrease in the ASSOPFAC padding factor.

## Examples

**Example 1:**

```
ADAORD  REORDB
```

The Associator and Data Storage are to be reordered. No changes are to be made to the current database parameters.

**Example 2:**

```
ADAORD  REORDB
ADAORD  MAXFILES=200
ADAORD  NEWDBID=6,NEWDBNAME=DATABASE-6
```

The Associator and Data Storage are to be reordered. A maximum of 200 files are permitted for the database. The database ID and name are to be 6 and DATABASE-6, respectively. If the nucleus is active during the REORDB operation, it will be stopped following the operation (NEWDBID was specified).

### Example 3:

```

ADAORD  REORDB
ADAORD  FILE=1 ,ACRABN=1000 ,NIRABN=2200 ,SORTSEQ=ISN
ADAORD  FILE=2 ,MAXISN=500000
ADAORD
ADAORD  FILE=4 ,ASSOPFAC=5 ,DATAPFAC=20 ,DSSIZE=5 ,DSRABN=1

```

The Associator and Data Storage are to be reordered. The address converter allocation for file 1 is to begin with RABN 1,000. The normal index allocation for file 1 is to begin with RABN 2,200.

The Data Storage portion of file 1 is to be reordered in ascending ISN sequence. The MAXISN for file 2 is to be set to 500,000. The following assignments are made for file 4: the Associator block padding factor is to be changed to 5 percent, the Data Storage block padding factor is set to 20 per cent, and a new DSSIZE of 5 cylinders is assigned starting at RABN 1. All other files are reordered according to their existing definitions.

### Example 4:

```

ADAORD  REORDB
ADAORD  FILE=66
ADAORD  DSRELEASE
ADAORD  NIRELEASE
ADAORD  UIRELEASE
ADAORD  RAID

```

1. ADAORD reorders the entire database.
2. ADAORD releases all unused storage from the Data Storage, normal index, and upper index of file 66.
3. However, if file 66 does not exist in the database, ADAORD does not terminate with an error message; rather, ADAORD ignores this condition and proceeds.

### Example 5:

```

ADAORD  REORDB
ADAORD  DBINDEXCOMPRESSION=YES
ADAORD  FILE=1
ADAORD  FILE=2 ,INDEXCOMPRESSION=NO
ADAORD  FILE=3

```

All files are reordered and rebuilt with compressed indexes, except for file 2, which is rebuilt with an uncompressed index.