

Formatting Database Components

This chapter describes the syntax, parameters, and processing of the ADAFRM utility.

- Formatting Modes
 - Syntax
 - Essential Parameter
 - Optional Parameters
 - Examples
-

Formatting Modes

There are several ADAFRM formatting modes:

1. Format a *new* data set (...FRM functions). Only the data set specified by the function name and the NUMBER parameter is accessed and formatted. The FROMRABN parameter cannot be specified when formatting a new data set.
2. Format *part of an existing* data set (ASSOFRM, DATAFRM, WORKFRM, and TEMPFRM functions). Here, the FROMRABN parameter *must* be specified, except on z/OS platforms. When formatting Work and Data Storage (WORKFRM and DATAFRM functions), the ADAFRM job control must also contain the Associator data sets.

This formatting mode is used in combination with the ADADBS INCREASE function for ASSO and DATA. If a greater WORK is needed, then ADADEF NEWWORK should be used.

3. Reformat *blocks of an existing* data set (...RESET functions). This mode opens all Associator, Data Storage, and Work data sets in the database for access. The FROMRABN parameter is *must* be specified for these functions.
4. If you are using Adabas 8 or later, you can use this utility to clear multiple PLOG headers from the PLOG, without requiring that you reformat the entire PLOG. For more information, read about using the FROMRABN, NUMBER, and SIZE parameters together, as described in the ADAFRM *FROMRABN, NUMBER, and SIZE* parameter documentation.

Syntax

To format the Associator (ASSO..) or Data Storage (DATA..) data set, use this syntax:

```
ADAFRM { ASSOFRM | DATAFRM } SIZE = size
      [DEVICE = device-type ]
      [ { FROMRABN = { starting-rabn | NEXT } |
        NUMBER = { dataset-number | 1 } } ]
      [NOUSERABEND]
```

To format the Work (WORK..) data set, Command log (CLOG..), Protection log (PLOG..), or Sort (SORT..) data set, use this syntax:

Note:

If you are using Adabas 8 or later, you can also use this syntax to clear multiple PLOG headers from the PLOG, without requiring that you reformat the entire PLOG. For more information, read about using the FROMRABN, NUMBER, and SIZE parameters together, as described in the ADAFRM *FROMRABN*, *NUMBER*, and *SIZE* parameter documentation.

```
ADAFRM { WORKFRM | CLOGFRM | PLOGFRM | SORTFRM }
        SIZE = size
        [DEVICE = device-type ]
        [ { FROMRABN = starting-rabn | NUMBER = {dataset-number | 1} } ]
        [NOUSERABEND]
```

To format the Recovery log (RLOG..) data set, use this syntax:

```
ADAFRM RLOGFRM SIZE = size
        [DEVICE = device-type ]
        [NOUSERABEND]
```

To format a Temp (TEMP..) data set, use this syntax:

```
ADAFRM TEMPFRM SIZE = size
        [DEVICE = device-type ]
        [FROMRABN = starting-rabn ]
        [NOUSERABEND]
```

To reformat blocks of an existing Associator, Data Storage, or Work data set, use this syntax:

```
ADAFRM { ASSORESET | DATARESET | WORKRESET }
        SIZE = size
        FROMRABN = start-rabn
        [NOUSERABEND]
```

Essential Parameter

SIZE: Size of Area to be Formatted

SIZE specifies the size of the area to be formatted (or reset). Blocks (a decimal value followed by a "B") or cylinders may be specified. For the RLOGFRM function, the size must be the same as that specified by the RLOGSIZE parameter on the ADARAI utility's PREPARE function. See section Essential Parameter.

If you are using Adabas 8 or later and you want to clear multiple PLOG headers from the PLOG without reformatting the entire PLOG, set the value of the SIZE parameter to "1" and specify values for both the FROMRABN and NUMBER parameters.

Optional Parameters

DEVICE: Device Type

DEVICE is the physical device type on which the area to be formatted is contained. If DEVICE is not specified, the device type specified by the ADARUN DEVICE parameter is used.

FROMRABN: Starting RABN

FROMRABN specifies the RABN at which formatting or resetting is to begin. This parameter may only be used for an existing data set; NUMBER cannot be specified in the same ADAFRM job as FROMRABN.

When FROMRABN is specified with a *xxxxFRM* function, formatting begins at the FROMRABN point and continues up to the *highest complete track* before the RABN computed from FROMRABN + SIZE (assuming a size specified in or converted to blocks). This means that the last track within the specified range (FROMRABN + SIZE) will be formatted *only* if all the track's RABNs are within that range.

If you are using Adabas 8 or later and you want to clear multiple PLOG headers from the PLOG without reformatting the entire PLOG, set the value of the SIZE parameter to "1" and specify values for both the FROMRABN and NUMBER parameters.

When increasing the size of an ASSO or DATA data set, FROMRABN is available as an option only under z/VSE and BS2000. The specified RABN must be one higher than the highest allocated RABN before the logical increase using ADADBS (which must precede the physical increase using ADAFRM). FROMRABN=NEXT instructs ADAFRM to take the first unformatted RABN as the value for FROMRABN. ADAFRM then verifies that the range of blocks determined for formatting by the NEXT value is contained in the free space table (FST). If not, ADAFRM terminates with ERROR-126. On BS2000, it is necessary not to format where a track would be split from one volume to the next (for more information, read *Splitting Data Sets Across Volumes* under *BS2000 Device Types and Block Sizes*).

On z/OS, FROMRABN should only be used to reformat existing blocks as the last record pointer in the VTOC cannot be modified by function FROMRABN. See the examples for ADADBS INCREASE .

This parameter is *required* for the ASSORESET, DATARESET and WORKRESET functions. When specified with the function ASSORESET, the FROMRABN value must be greater than 30.

NOUSERABEND: Termination without Abend

When a parameter error or a functional error occurs while this utility function is running, the utility ordinarily 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.

Note:

When NOUSERABEND is specified, we recommend that it be specified as the first parameter of the utility function (before all other parameters). This is necessary to ensure that its parameter error processing occurs properly.

NUMBER: Data Set Number

NUMBER selects the nonsequential command log, nonsequential protection log, Associator, Data Storage and sort data set to be formatted. The default is 1 (first data set). Values allowed for:

- the Associator (ASSO) or Data Storage (DATA) are 1 through 99;
- protection logs (PLOGs) or command logs (CLOGs) are 1 through 8;
- the recovery log (RLOG) is just 1;
- SORT is either 1 or 2 (1 only on z/VSE systems); and
- WORK or TEMP is either 1 or the default.

ADAFRM ...FRM function statements cannot specify (and will not default to) a NUMBER value if other ADAFRM statements in the same job specify a FROMRABN value.

NUMBER must match the number suffix of the related data definition (DD) statement. See the tables of allowed statements and the examples in section JCL/JCS Requirements and Examples.

If you are using Adabas 8 or later and you want to clear multiple PLOG headers from the PLOG without reformatting the entire PLOG, set the value of the SIZE parameter to "1" and specify values for both the FROMRABN and NUMBER parameters.

Examples

Example 1:

Format 50 cylinders for the Associator, 200 cylinders for Data Storage, 10 cylinders for Work, and 2 cylinders for the recovery log (RLOG).

```
ADAFRM ASSOFRM SIZE=50,DEVICE=3380
ADAFRM DATAFRM SIZE=200,DEVICE=3380
ADAFRM WORKFRM SIZE=10,DEVICE=3380
ADAFRM RLOGFRM SIZE=2
```

Example 2:

One cylinder for nonsequential command log data set 1, and 1 cylinder for nonsequential command log data set 2 are to be formatted.

```
ADAFRM CLOGFRM SIZE=1,DEVICE=3390,NUMBER=1
ADAFRM CLOGFRM SIZE=1,DEVICE=3390,NUMBER=2
```

Example 3:

The first two blocks of an existing Work data set are to be reset to binary zeros.

```
ADAFRM WORKRESET FROMRABN=1,SIZE=2B
```

Example 4:

Assuming the Data Storage data set is on a 3380 disk (9 blocks/track, 15 tracks/cylinder), 100 cylinders-starting at cylinder position 201 relative to the beginning of the data set-will be formatted.

```
ADAFRM DATAFRM SIZE=100,FROMRABN=26992
```

Example 5:

Under z/VSE or BS2000, assuming the Associator of the database has just been increased by 200 cylinders, this job formats the new space in the database. For more detailed examples across all supported platforms, see the ADADBS INCREASE examples in section Operating-System-Specific Procedures.

```
ADAFRM ASSOFRM SIZE=200,FROMRABN=NEXT
```