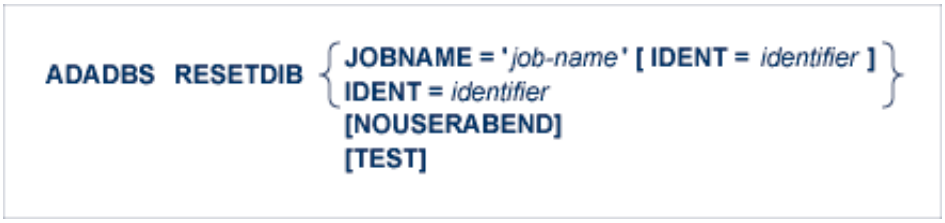
RESETDIB: Reset Entries in Active Utility List

The RESETDIB function resets entries in the active utility list (that is, the data integrity block or DIB).

Adabas maintains a list of the files used by each Adabas utility in the DIB. The DDIB operator command (or Adabas Online System) may be used to display this block to determine which jobs are using which files. A utility removes its entry from the DIB when it terminates normally. If a utility terminates abnormally (for example, the job is canceled by the operator), the files used by that utility remain *in use*. The DBA may release any such files with the RESETDIB function.

Note:
The RESETDIB function can be executed either with or without an active nucleus. To remove a DIB from an abended ADAORD REORDB, REORDATA, REORASSO, ADADBS RESETDIB has to run without an active nucleus.



This chapter covers the following topics:

- Essential Parameters
- Optional Parameters
- Examples

Essential Parameters

JOBNAME: Job Name

This parameter specifies the name of the job whose entry is to be reset. If it is not unique, the IDENT parameter must also be specified.

IDENT: Utility Execution Identifier

A unique number that identifies a utility execution. It may be specified alone or to qualify a job name when the same name has been used for various utility executions. The identifier may be obtained using the operator command DDIB or Adabas Online System.

Optional Parameters

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.

TEST: Test Syntax

The TEST 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. See Syntax Checking with the TEST Parameter for more information on using the TEST parameter in ADADBS functions.

Examples

The entry in the DIB block for job "JOB1" is to be deleted.

```
ADADBS RESETDIB JOBNAME='JOB1'
```

The entry in the DIB block for "JOB2" with IDENT=127 is to be deleted.

```
ADADBS RESETDIB
JOBNAME='JOB2',IDENT=127
```