

UNDELFN: Logically Undelete Fields

The UNDELFN function allows you to logically undelete fields in an Adabas database file that were previously logically deleted using the ADADBS DELFN utility function. Undeleting the fields reestablishes the fields in the FDT for the file.

```
ADADBS UNDELFN FILE = file-number
                FIELDLIST = ' field-list '
                [PASSWORD = password ]
                [NOUSERABEND]
                [TEST]
```

This chapter covers the following topics:

- Essential Parameter
 - Optional Parameters
 - Example
-

Essential Parameter

FILE: File Number

FILE specifies the database file number from which the fields should be logically undeleted. The file number may not be the number of a large object (LOB) file; fields in LOB files cannot be deleted, so you should have no reason to undelete them.

FIELDLIST: List of Fields

FIELDLIST specifies a list of one or more fields. At least one field must be specified. If more than one field will be undeleted, separate the field names with commas (.). A maximum of 800 fields may be specified. A field may only be listed once in an ADADBS UNDELFN run.

Descriptor fields cannot be included in the list (they cannot be logically deleted, so there should be no reason to undelete them). Likewise, parent fields of subdescriptor, superdescriptor, hyperdescriptor, phonetic descriptor, and collating descriptor fields cannot be listed.

Optional Parameters

PASSWORD: File Password

PASSWORD specifies the password of the file containing fields to be logically undeleted. This parameter is required if the file is password-protected.

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). When NOUSERABEND is specified, it must be specified as the first parameter (before all other parameters) for the utility function.

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.

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.

Example

In the following example, fields AA, AB, and AC are logically undeleted from file 12.

```
ADADBS UNDELFN FILE=12 FIELDLIST='AA,AB,AC'
```