

INVERT: Create Descriptors

The INVERT function creates descriptors, subdescriptors, superdescriptors, hyperdescriptors, phonetic descriptors or collation descriptors for existing fields in a file. Several descriptors may be created in a single ADAINV INVERT run, but only for a single file.

```

ADAINV INVERT FILES = file-num
SORTSIZE = size
TEMPSIZE = size
[FIELD = ' field-name [ , option ]... ' ] ...
[COLDE = ' num , name [ , UQ [ , XI ] ] = parent-field ' ]
[HYPDE = ' num , name , length , format [ , option ]... = parent-field , ... ' ]
[PHONDE = ' name ( field-name ) ' ]
[SUBDE = ' name [ , UQ [ , XI ] ] = parent-field ( begin , end ) ' ]
[SUPDE = ' name [ , UQ [ , XI ] ] = { parent-field ( begin , end ) } , ... ' ]
[CODE = cipher-code ]
[LPB = prefetch-buffer-size ]
[LWP = { workpool-size | 1048576 } ]
[NOUSERABEND]
[PASSWORD = ' password ' ]
[SORTDEV = device-type ]
[TEMPDEV = device-type ]
[TEST]

```

This chapter covers the following topics:

- Essential Parameters
- Optional Parameters and Subparameters
- Space Allocation for the INVERT Function
- Examples

Essential Parameters

FILE: File Number

FILE specifies the file in which the descriptor(s) to be created is contained.

SORTSIZE: Sort Size

SORTSIZE specifies the space available for the sort data set or data sets R1/2 (SORTR2 is not supported under z/VSE). The value can be either cylinders (a numeric value only) or blocks (a numeric value followed by a "B"). If blocks are specified, they should be equivalent to a full number of cylinders. The SORTSIZE parameter must be specified. Refer to the *Adabas DBA Reference* documentation for more information on estimating the sort space.

TEMPSIZE: Temporary Storage Size

TEMPSIZE defines the space available for the temp data set. The value may be in cylinders (a numeric value only) or blocks (a numeric value followed by a "B"). This parameter must be specified.

Optional Parameters and Subparameters

CODE: Cipher Code

If the file specified with the FILE parameter is ciphered, an appropriate cipher code must be supplied using the CODE parameter.

FIELD/ COLDE/ HYPDE/ PHONDE/ SUBDE/ SUPDE: Define Descriptor(s)

These parameters may be used to define various types of descriptors. You must specify at least one descriptor definition for the file specified; you may specify more than one descriptor or type of descriptor.

Use the FIELD parameter to define one or more fields as descriptors; use the COLDE parameter for a collation descriptor; HYPDE parameter for a hyperdescriptor; PHONDE for a phonetic descriptor; SUBDE for a subdescriptor; and SUPDE for a superdescriptor.

If provided, a FIELD specification must come before any collation descriptor, hyper-, super-, sub-, or phonetic descriptor specification.

FIELD specifies an existing field (or fields) to be inverted. The field may be an elementary or multiple-value field and may be contained within a periodic group (unless the field is defined with the FI option).

If the descriptor is to be unique, specify "UQ" following the field name. If the uniqueness of the descriptor is to be determined with the index (occurrence number) excluded, specify "XI" as well.

Note:

For Adabas expanded files, ADAINV can only detect unique descriptor violations within the specified component file. If an identical value exists for a unique descriptor in one of the other component files, ADAINV cannot detect it. You must therefore ensure that unique descriptor values remain unique throughout an expanded file.

Although multiple fields can be specified for inversion using the FIELD parameter, only one collation descriptor, hyper-, sub-, super-, or phonetic descriptor is defined per instance of its parameter. Multiple instances of the parameters are allowed per execution of ADAINV.

When inverting a sub- or superfield, the respective SUBDE or SUPDE parameter must specify the same parent fields that were specified when the field was created; otherwise, an error occurs. Begin and end values are taken from the original field definitions.

If a parent field with the NU option is specified, no entries are made in the inverted list for those records containing a null value for the field. For super- and hyperdescriptors, this is true regardless of the presence or absence of values for other descriptor elements.

If a parent field is not initialized and logically falls past the end of the physical record, the inverted list entry for that record is not generated for performance reasons. To generate the inverted list entry in this case, it is necessary to unload short, decompress, and reload the file; or use an application program to initialize the field for each record of the file.

For detailed information about the individual descriptor syntax, subparameter values, and coding, read *Field Definition Statements* in the description of the ADACMP utility.

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; ADAINV may also reduce a specified LPB value if the LU value is too small.

LWP: Work Pool Size

LWP specifies the size of the work pool to be used for descriptor value sorting. The value can be specified in bytes or kilobytes followed by a "K". If no value is specified, the default is 1048576 bytes (or 1024K); however, to shorten ADAINV run time for files with very long descriptors or an unusually large number of descriptors, set LWP to a higher value. To avoid problems with the Sort data set, a smaller LWP value should be specified when defining descriptors for relatively small files.

The minimum work pool size depends on the Sort data set's device type:

Sort Device	Minimum LWP	Minimum LWP
	Bytes	Kilobytes
2000	106496	104K
2314	090112	88K
3375	131072	128K
3380	139264	136K
3390	159744	156K

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.

PASSWORD: File Password

If the file specified with the FILE parameter is security protected, the file's password must be supplied using this parameter.

SORTDEV: Sort Device Type

ADAINV uses the sort data set to sort descriptor values. The SORTDEV parameter indicates the device type to be used for the sort data set. This parameter is required only if the device type to be used is different from that specified with the ADARUN DEVICE parameter. See the z/OS job control information at the end of this section for specific z/OS SORTDEV considerations.

TEMPDEV: Temporary Storage Device Type

ADAINV uses the temp data set to store intermediate data. The TEMPDEV parameter indicates the device type to be used for this data set. This parameter is required only if the device type to be used is different from that specified with the ADARUN DEVICE parameter.

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.

Space Allocation for the INVERT Function

The values for the field being inverted and the ISNs of the records containing the values are written to the inverted list (normal and upper indexes).

If either the normal or upper index logical extent is exhausted during ADAINV execution, ADAINV allocates an additional extent. The size of the extent allocated is equal to 25 percent of the current total size of all the normal index extents currently allocated to the file.

If sufficient space is not available for the new extent or if the maximum number of allocated extents has been reached, ADAINV terminates with an error message.

Examples

Example 1:

```
ADAINV INVERT
FILE=3, FIELD='AR', TEMPSIZE=10, SORTSIZE=5
```

Field AR in file 3 is to be made a descriptor.

Example 2:

```
ADAINV INVERT FILE=5, SUBDE='SA=AA(1,4)'
ADAINV          TEMPSIZE=6, SORTSIZE=3
```

Subdescriptor SA is to be created using field AA (positions 1-4) in file 5 as the parent field.

Example 3:

```
ADAINV INVERT FILE=6,SUPDE='SB=AA(1,4),AB(1,1)'  
ADAINV          TEMPSIZE=5,SORTSIZE=3
```

Superdescriptor SB is to be created using fields AA (positions 1-4) and AB (position 1) in file 6.

Example 4:

```
ADAINV INVERT FILE=1,PHONDE='XX(AA)'  
ADAINV          TEMPSIZE=5,SORTSIZE=3
```

A phonetic descriptor XX is created using field AA as the source field.

Example 5:

```
ADAINV INVERT FILE=6,COLDE='1,Y1=AA'  
ADAINV          TEMPSIZE=5,SORTSIZE=4
```

Collation descriptor CDX=01 named Y1 is created using AA as the source field.