

# DEFINE: Defining a Database and Checkpoint File

The database and the checkpoint file must be defined at the same time.

The database parameters include the required ASSOSIZE, DATASIZE, and WORKSIZE parameters and the optional (non-indented) parameters ASSODEV through WORKDEV shown in the syntax diagram.

The FILE=...,CHECKPOINT,... statement is also required for database definition. The checkpoint file parameters (indented under the FILE statement in the syntax diagram) should be specified immediately following the FILE statement. See the examples.

```

ADADEF DEFINE  ASSOSIZE = size-list
                  DATASIZE = size-list
                  WORKSIZE = size
                  FILE = file-number , CHECKPOINT
                    DSSIZE = size
                    MAXISN = maximum-number-of-records-expected
                    [ACRABN = starting-rabn ]
                    [ASSOPFAC = { Associator-padding-factor | 10 } ]
                    [ASSOVOLUME = ' Associator-extent-volume ' ]
                    [DATAPFAC = { Data-Storage-padding-factor | 10 } ]
                    [DATAVOLUME = ' Data-Storage-extent-volume ' ]
                    [DSDEV = device-type ]
                    [DSRABN = starting-rabn ]
                    [DSREUSE = { NO | YES } ]
                    [ISNSIZE = { 3 | 4 } ]
                    [MAXDS = maximum-Data-Storage-secondary-allocation ]
                    [MAXNI = maximum-normal-index-secondary-allocation ]
                    [MAXUI = maximum-upper-index-secondary-allocation ]
                    [NAME = { ' file-name ' | CHECKPOINT } ]
                    [NIRABN = starting-rabn ]
                    [NISIZE = size ]
                    [UIRABN = starting-rabn ]
                    [UISIZE = size ]
                    [ASSODEV = { device-type-list | ADARUN-device } ]
                    [DATODEV = { device-type-list | ADARUN-device } ]
                    [DBIDENT = { database-id | ADARUN-dbid } ]
                    [DBNAME = { database-name | GENERAL-DATABASE } ]
                    [FACODE = { alpha-EBCDIC-key | 37 } ]
                    [FWCODE = { wide-key | 4095 } ]
                    [MAXFILES = { maximum-number-of-files | 255 } ]
                    [NOUSERABEND]
                    [OVERWRITE]
                    [RABNSIZE = { 3 | 4 } ]
                    [REPTOR = YES | NO ]
                    [UACODE = { alpha-ASCII-key | 437 } ]
                    [UES = { YES | NO } ]
                    [UWCODE = wide-key ]
                    [WORKDEV = device-type-list ]

```

This chapter covers the following topics:

- Essential Parameters
- Optional Parameters
- Examples

# Essential Parameters

## ASSOSIZE/ DATASIZE/ WORKSIZE: Database Size

ASSO-/DATA-/WORKSIZE specifies the number of blocks or cylinders to be assigned to the Associator, Data Storage, or Work. A block value must be followed by a "B"; otherwise, the value is assumed to be cylinders.

If the Associator or Data Storage is to be contained on more than one data set, the size of each data set must be specified. If a companion ASSODEV or DATADEV parameter specifies two or more extents, the equivalent ASSOSIZE or DATASIZE parameter must specify the extent sizes as positional operands in the corresponding order (see the examples).

The minimum WORKSIZE allowed is 300 blocks.

### Note:

If ASSOSIZE or DATASIZE is not specified, the ADADEF DEFINE function will not execute. If WORKSIZE is not specified, the function will allocate three (3) cylinders to the Work data set. Because 3 cylinders are usually not enough to start the database, WORKSIZE is considered to be a required parameter.

## DSSIZE: Data Storage Size

DSSIZE specifies the number of blocks or cylinders to be assigned to checkpoint/Data Storage. For blocks, the value specified must be followed by a "B" (for example, DSSIZE=80B).

The size of the checkpoint file specified with the DSSIZE and MAXDS parameters depends on

- the amount of ET data to be stored;
- the number of utility runs for which checkpoint information is to be retained;
- the number of user IDs.

## FILE . . . CHECKPOINT Parameter

The FILE...CHECKPOINT parameter indicates the file number to be used for the checkpoint system file. This parameter is required; the file number must be 5000 or lower.

Adabas uses the checkpoint system file to store checkpoint data and user data provided with the Adabas CL and ET commands.

## MAXISN: Highest ISN to be Used

The highest ISN that may be assigned to the file. The value specified is used to determine the space allocation for the address converter. When determining the MAXISN, consider the importance of ET data and checkpoint data to your site.

Adabas considers ET data to be more important than checkpoint data. As soon as the ET data ISN range in the checkpoint system file is exhausted, the first checkpoint ISN is deleted and given to the ET data. This is an ongoing process. As soon as the MAXISN is reached, a new address converter extent is allocated and given to the checkpoint data. You can delete checkpoint data piece by piece using the Adabas Online System function DELCP.

**Note:**

The way the checkpoint handles data is subject to change in a future release of Adabas.

## Optional Parameters

### ACRABN/ DSRABN/ NIRABN/ UIRABN: Starting RABN

These parameters may be used to cause allocation for their respective areas to begin with the specified RABN:

- ACRABN for the address converter
- DSRABN for Data Storage
- NIRABN for the normal index
- UIRABN for the upper index

### ASSODEV/ DATADEV/ WORKDEV: Device Type

ASSO-/DATA-/WORKDEV specify the device type(s) to be assigned to the Associator, Data Storage, and Work. These parameters are required only if the device type to be used is different from that specified with the ADARUN DEVICE parameter.

WORKDEV, if specified, can only be one device type. If the Associator (ASSODEV) or Data Storage (DATADEV) is to be contained on more than one data set, the device type for each data set must be specified, even if both extents are on the ADARUN DEVICE type.

If multiple extents are used with VSAM data sets, ASSODEV and DATADEV must reflect the dynamic device type; that is, DD/xxxxR1=9999; DD xxxxR2=8888; ... DD/xxxxR5=5555. For example, when defining DDDATAR1 and DDDATAR2, DATADEV=9999,8888.

Space allocation for specified device types must be given in companion ASSOSIZE, DATASIZE, and WORKSIZE parameters on this or another ADADEF statement in the same job. If a ASSODEV or DATADEV parameter specifies more than one extent on the same or different device types (DATADEV=3380,3390, for example), the companion ASSOSIZE or DATASIZE parameter must specify the related extent sizes in corresponding order.

### ASSOPFAC/ DATAPFAC: Padding Factor

ASSOPFAC defines the percentage of space in each Associator RABN block to be reserved for later entries (padding space). This space is used for later descriptor extensions or ISN additions. The percentage value specified, which can range 1-90, should be large enough to avoid the overhead caused when block overflow forces splitting of an existing address block into two new blocks. If ASSOPFAC is not specified, ADADEF assumes a padding factor of 10%.

DATAPFAC defines the percentage of space in each Data Storage RABN block to reserve for later entries (padding space). This space is used when changes to an existing data record cause it to need more space in the block; an updated record that no longer fits in the existing block must be moved to another block. The percentage value specified, which can range 1-90, should be large enough to avoid the overhead caused when block overflow forces splitting of an existing address block into two new blocks. If DATAPFAC is not specified, ADADEF assumes a padding factor of 10%.

**ASSOVOLUME/ DATAVOLUME: Extent Volume****Note:**

Values for ASSOVOLUME and DATAVOLUME must be enclosed in apostrophes.

ASSOVOLUME specifies the volume on which the file's Associator space (that is, the AC, NI, and UI extents) is to be allocated.

DATAVOLUME specifies the volume on which the file's Data Storage space (DS extents) are allocated.

If the requested number of blocks cannot be found on the specified volume, ADADEF retries the allocation while disregarding the ASSOVOLUME or DATAVOLUME parameter value.

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

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

If ASSOVOLUME and/or DATAVOLUME are not specified, the file's Associator and/or Data Storage space, respectively, is allocated according to ADADEF's default allocation rules.

**DBIDENT: Database Identifier**

DBIDENT specifies the identification number to be assigned to the database. A value in the range 1-65535 may be specified. If this parameter is omitted, the value specified with the ADARUN DBID parameter is used.

If multiple databases are to be established, the DBIDENT parameter is required in order to uniquely identify each database.

**DBNAME: Database Name**

DBNAME is the name to be assigned to the database. This name appears in the title of the Database Status Report produced by the ADAREP utility. A maximum of 16 characters may be specified. Enclose the name in single quotation marks if the name includes any special characters other than dashes, or if the name contains embedded blanks.

If this parameter is omitted, a default value of "GENERAL-DATABASE" is assigned.

**DSDEV: Device Type for Data Storage**

DSDEV specifies the device type to be used for the checkpoint file's Data Storage. There is no default value; if DSDEV is not specified, an arbitrary device type is used.

**DSREUSE: Storage Reusage**

DSREUSE indicates whether space which becomes available in the checkpoint file is to be reused. The default is YES.

**FACODE: Encoding for Alphanumeric Fields**

The FACODE parameter specifies the default encoding for alphanumeric fields for all files in the database. The encoding must be derived from EBCDIC encoding; that is, X'40' is the space character. Modal or shift-type double-byte character set (DBCS) encodings are supported; fixed type DBCS (DBCS-only) encodings are not supported. The default encoding key is 37.

The purpose of the database-wide setting is to serve as a default when loading files. Once loaded, the encoding for a file is stored in its FCB.

You can change the default encoding set in this parameter using the ADADEF MODIFY function. Changing the database-wide setting does not affect files already loaded.

### **FWCODE: Encoding for Wide-Character Fields**

The FWCODE parameter specifies the default encoding for wide-character (W) format fields for all files in the database. The default encoding is 4095; that is, Unicode.

The FWCODE parameter can be used to set a wide-character encoding that defines the superset of code points of all user encodings. For example, Unicode encompasses about 50,000 code points as opposed to Host-DBCS and Shift-JIS with about 10,000 code points each.

The purpose of the database-wide setting is to serve as a default when loading files. Once loaded, the encoding for a file is stored in its FCB.

You can change the default encoding set in this parameter using the ADADEF MODIFY function. Changing the database-wide setting does not affect files already loaded.

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

ISNSIZE indicates whether ISNs in the file are 3 or 4 bytes long. The default is 3 bytes.

### **MAXDS/ MAXNI/ MAXUI: Maximum Secondary Allocation**

MAXDS/NI/UI specify the maximum number of blocks per secondary extent for Data Storage, the normal index, and the upper index, respectively. The value specified must be followed by a "B" for blocks (for example, MAXDS=8000B) and cannot be more than 65535B.

### **MAXFILES: Highest File Number**

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 block size, whichever is lower. For example, 2003 is the highest MAXFILES value for a database whose ASSOR1 is stored on a 3380 DASD.

The value specified determines the number of file control blocks and field definition tables to be allocated when the database is being established. Each file control block requires one Associator block and each field definition table requires four Associator blocks.

If this parameter is omitted, a value of 255 is assigned.

Once the database has been established, the value for MAXFILES may be changed only by executing the REORASSO or REORDB functions of the ADAORD utility.

**NAME: Name of the Checkpoint File**

NAME specifies the name for the checkpoint file being defined. This name appears on the Database Status Report produced by the ADAREP utility. The maximum number of characters permitted is 16. The default file name is CHECKPOINT.

**NISIZE: Normal Index Size**

NISIZE specifies the number of blocks or cylinders to be assigned to the normal index. For blocks, the value specified must be followed by "B" (for example, NISIZE=80B).

**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.

**OVERWRITE: Overwrite Existing Database**

Specify OVERWRITE to write over an existing database. OVERWRITE cannot be specified when creating a database with newly formatted data sets.

**RABNSIZE: 3- or 4-Byte RABN**

RABNSIZE specifies the length of RABNs in the database. Specify 3 for 24-bit RABNs or 4 for 31-bit RABNs. The default is 3.

**REPTOR: Set the Event Replicator Server**

The REPTOR parameter is used with ADADEF DEFINE to specify whether a normal Adabas database is being defined, or whether an Event Replicator Server is being defined. Possible values are "YES" or "NO" (the default).

Specify REPTOR=NO for a normal Adabas database. Specify REPTOR=YES for an Event Replicator Server.

**UACODE: User Encoding for Alphanumeric Fields**

The parameter UACODE specifies the default encoding for alphanumeric fields for ASCII users. The encoding must be derived from ASCII encoding; that is, X'20' is the space character. Encodings for multiple-byte character sets are supported. The default encoding is 437.

The UACODE value is not stored in the file being loaded.

You can override the default encoding set in this parameter for a user session using the OP command. You can change it generally using the ADADEF MODIFY function.

### UES: Universal Encoding Support

Setting the parameter UES activates universal encoding support for the database. Any valid xxCODE parameter (FACODE, FWCODE, UACODE, UWCODE) implicitly sets UES=YES.

To deactivate UES, you must explicitly set UES=NO.

You can change the default setting of this parameter generally using the ADADEF MODIFY function.

### UISIZE: Upper Index Size

UISIZE specifies the number of blocks or cylinders to be assigned to the upper index. For blocks, the value specified must be followed by "B" (for example, UISIZE=80B).

### UWCODE: User Encoding for Wide-Character Fields

The UWCODE parameter specifies the user encoding for wide-character (W) format fields. If the parameter is not specified, the default value is the current value of FWCODE.

The purpose of the database-wide setting is to serve as a default when loading files. Once loaded, the encoding for a file is stored in its FCB.

You can override the default encoding set in this parameter for a user session using the OP command. You can change the default setting generally using the ADADEF MODIFY function. Changing the database-wide setting does not affect files already loaded.

## Examples

### Example 1:

```

ADADEF DEFINE
ADADEF ASSOSIZE=200,DATASIZE=600,WORKSIZE=50
ADADEF DBIDENT=1,DBNAME=DATABASE-1
ADADEF MAXFILES=150
ADADEF FILE=1,CHECKPOINT
ADADEF NAME='DB1-CHECKPOINT',MAXISN=5000
ADADEF DSSIZE=2,NISIZE=50B,UISIZE=10B

```

The Associator, Data Storage and Work sizes are equal to 200, 600 and 50 cylinders, respectively. The numeric identifier for the database is 1 and the database name is DATABASE-1. The maximum number of files (and the highest file number) that may be loaded into the database is 150. File 1 is to be reserved for the Adabas checkpoint file. The name of the first system file is to be DB1-CHECKPOINT. The Data Storage size for this file is to be 2 cylinders; the normal index size 50 blocks; the upper index size 10 blocks; and the MAXISN is to be 5000.



**Example 2:**

```
ADADEF DEFINE
ADADEF  ASSODEV=3380,DATADEV=3380,3390,WORKDEV=3380
ADADEF  ASSOSIZE=100,DATASIZE=200,300,WORKSIZE=25
ADADEF  DBIDENT=2,DBNAME='DATABASE_2'
ADADEF  MAXFILES=255
ADADEF  FILE=255,CHECKPOINT,MAXISN=5000
ADADEF  DSSIZE=3,NISIZE=100B,UISIZE=20B
```

The Associator is to be contained on a 3380 device type, and occupies 100 cylinders. Data Storage comprises two data sets: the first data set is 200 cylinders contained on the first DATADEV (3380) device type, and the second data set is 300 cylinders contained on the second DATADEV (3390) device type. The Work space is 25 cylinders on the WORKDEV device (3380).

The numeric identifier for the database is 2, and the database name is DATABASE\_2. A maximum of 255 files may be loaded into the database. An Adabas checkpoint file is loaded during this step.