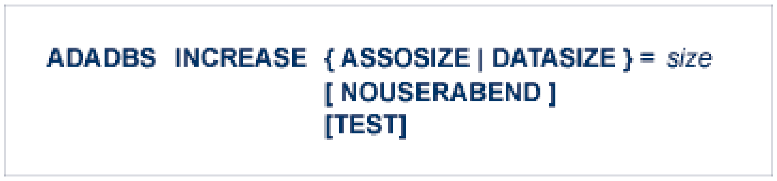
INCREASE: Increase Last Associator or Data Storage Data Set Size

The INCREASE function increases the size of the last data set currently being used for the Associator or Data Storage. This function may be executed any number of times for the Associator. The maximum of 99 Data Storage Space Tables (DSSTs) somewhat limits Data Storage increases before all 99 Data Storage extents must be combined into a single extent with either the REORASSO or REORDB function of the ADAORD utility.

Notes:

1. The Associator and Data Storage data set sizes must be increased separately. It is *not* possible to increase both with a single operation.
2. After an INCREASE operation is completed, the INCREASE function automatically ends the current nucleus session. This allows for the necessary Associator or Data Storage formatting with ADAFRM before a new session is started. An informational message occurs to tell you that the nucleus has been stopped.



```
ADADBS  INCREASE  { ASSOSIZE | DATASIZE } = size
                  [ NOUSERABEND ]
                  [TEST]
```

This chapter covers the following topics:

- Essential Parameter
- Optional Parameters
- Example
- General Procedure
- Operating-System-Specific Procedures

Essential Parameter

ASSOSIZE | DATASIZE: Size to Be Increased

The additional number of blocks or cylinders needed by the Associator or Data Storage data set. To specify blocks, add a "B" after the value; for example, DATASIZE=50B.

Optional Parameters

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

Use the TEST parameter to test 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 about using this parameter.

Example

The Associator is to be increased by 400 cylinders.

```
ADADBS INCREASE ASSOSIZE=400
```

Note:
On BS2000 systems, we strongly recommend using blocks because cylinders do not exist on this platform. PAM page sizes can be more easily calculated from blocks or RABNs (for more information, read *Device and File Considerations*. So this might be, say:

```
ADADBS INCREASE ASSOSIZE=60000B
```

General Procedure

The general procedure for increasing the size of the Associator or Data Storage is as follows:

1. Back up the database using the ADASAV utility. This step is optional but recommended.
2. Execute the ADADBS INCREASE function.
3. Format the additional space being added to the data set with the ADAFRM utility.

Operating-System-Specific Procedures

z/OS Systems

Under z/OS, the same data set may be formatted by specifying the DISP=MOD parameter in the JCL. The SPACE parameter for the data set being increased should be set to

```
SPACE=(CYL,(0,n))
```

where *n* is the amount of space (in cylinders) being added. The ADAFRM control statement should also specify the number of cylinders being added. If the increased part of the data set to be formatted is contained on a new volume, the VOL parameter of the JCL must include references to all volumes containing the data set.

Example 1: OS Single-Volume INCREASE

400 cylinders are to be added to an Associator data set which currently contains 300 cylinders. The control statement for the INCREASE function would be:

```
ADADBS INCREASE ASSOSIZE=400
```

The following JCL example increases the Associator data set using ADAFRM:

```
//DDASSOR1  DD
DSN=....,DISP=MOD,SPACE=(CYL,(0,400))
```

and the actual ADAFRM control statement would be

```
ADAFRM ASSOFRM SIZE=400
```

Example 2: OS Multivolume INCREASE

To provide the increase in example 1 for multiple volumes, specify the volumes in the JCS:

```
//DDASSOR1  DD  DSN=...
//
DISP=(MOD,CATLG),VOL=SER=(V1,V2,...),SPACE=(CYL,(0,400))...
```

Include the following step after the INCREASE step but before the FORMAT step to ensure a correct catalog entry:

```
//UNCATLG  EXEC PGM=IEFBR14
//DDASSOR1  DD  DSN=...,DISP=(SHR,UNCATLG)
```

z/VSE Systems

The following procedures are recommended for increasing Associator or Data storage:

1. Save the current database.
2. End the Adabas session normally with the ADAEND operator command.
3. Update the JCS defining the database to add the new extent on the same volume.

   Before a new Associator or Data extent *on either a different or the same z/VSE volume* can be increased with ADADBS INCREASE and formatted with ADAFRM, that volume’s table of contents (VTOC) must be updated to contain the new extent.

Use a job similar to the following example to update the VTOC for a single volume extent:

```
* $$ JOB JNM=jobname
* $$ LST ...
* $$ PCH ...
// ASSGN SYS001,DISK,VOL=volume,SHR
// DLBL ASSOEXT,’dsname’,99/365,DA
// EXTENT SYS001,volume1,1,0,starttrack1,trackcount1
// EXTENT SYS001,volume1,1,1,starttrack2,trackcount2

// EXEC ASSEMBLY,GO
MODVTOC  CSECT
         BALR 9,0
         BCTR 9,0
         BCTR 9,0
         USING MODVTOC,9
         OPEN  ASSOEXT
         CLOSE ASSOEXT
         EOJ   RC=0
ASSOEXT  DTFPH TYPEFLE=OUTPUT,DEVADDR=SYS001,DEVICE=DISK,MOUNTED=ALL
         END
/*
/&
* $$ EOJ
```

For a two-volume extent, use a job similar to the following example:

```
* $$ JOB JNM=jobname
* $$ LST ...
* $$ PCH ...
// ASSGN SYS001,DISK,VOL=volume1,SHR
// ASSGN SYS002,DISK,VOL=volume2,SHR
// DLBL ASSOEXT,’dsname’,99/365,DA
// EXTENT SYS001,volume1,1,0,starttrack1,trackcount1
// EXTENT SYS002,volume2,1,1,starttrack2,trackcount2
// EXEC ASSEMBLY,GO
MODVTOC  CSECT
         BALR 9,0
         BCTR 9,0
         BCTR 9,0
         USING MODVTOC,9
         OPEN  ASSOEXT
         CLOSE ASSOEXT
         EOJ   RC=0
ASSOEXT  DTFPH TYPEFLE=OUTPUT,DEVADDR=SYS001,DEVICE=DISK,MOUNTED=ALL
         END
/*
/&
* $$ EOJ
```

Note:
This job causes z/VSE error message 4733D to be sent to the console, and the operator is asked for a response. After the JCS has been validated, the operator response should be DELETE.

4. Perform the ADADBS INCREASE operation.

5. Run the new ADAFRM job to format the new extent. The ADAFRM job must specify the FROMRABN parameter, as shown in the following example:

```
ADAFRM ASSOFRM SIZE=size,FROMRABN=rabn-number
```

where *size* is the number of cylinders or blocks by which the data set is to be increased, and *rabn-number* is the first RABN in the new extent.

6. Start the Adabas nucleus.

BS2000 Systems

Use the following procedure to increase the database on BS2000 systems:

1. Execute ADADBS INCREASE as described in section General Procedure.

2. End the Adabas session with ADAEND.

3. Produce a database report by running the ADAREP utility. Use the report to find the first RABN for the new extent in the "Physical Layout of the Database" portion of the report. The RABN range is indicated in the "VOLSER NUMBER" column.

4. Calculate the PAM page space using the information in the *Device and File Considerations* section for the container type as follows:

   ```
   Number-PAM-pages-to-increase = (Number-RABNs-to-increase) * (RABN-STD-block-size-for-this-device)
   ```

   For example, suppose the ADADBS parameter was ADADBS INCREASE ASSOSIZE=60000B. Increasing a 2300 ASSO device by 60000 RABNs needs 120000 PAM pages:

   ```
   /MODIFY-FILE-ATTRIBUTE ADA99.ASSO,SUP=PUB(SPACE=RELATIVE(120000))
   ```

   Note:
   In the old ISP format, this was performed by the FILE command. For example, `/FILE ADA99.ASSO, SPACE=120000`.

5. Format the new space by running the ADAFRM utility. An example for the space added in step 4 is:

   ```
   ADAFRM ASSOFRM   SIZE=400B,FROMRABN=rabn-number
   ```

   where *rabn-number* specifies the first RABN shown on the new extent, as shown in the report.

6. You are entirely responsible for the container expansion above. We therefore strongly recommend that you check carefully that you use the following command to determine whether the highest PAM page calculated from the highest RABN from the ADAREP utility with the highest PAM page is in the container:

   ```
   /SHOW-FILE-ATTRIBUTES container-name,ALL
   ```

The highest PAM page is in the output field HIGH-US-PA.

If the highest PAM page is less than the highest PAM pages from the the ADAREP, accessing the highest RABNs will result in a DMS0922 I/O error.