Device And File Considerations

Support for new device types that include user-defined block sizes can be implemented in Adabas by modifying one of the table of device-constant entries (TDCEs) reserved for this purpose. A TDCE is X’40’ bytes long and the first free TDCE can be identified by X’0000’ in its first two bytes (TDCDT).

Under BS2000, the address of the first TDCE is at offset ADAIOR+ X‘34’ for all versions of Adabas.

Adabas direct access datasets are always mapped to UPAM files, removing the need to consider physical device characteristics. PAM pages in a dataset are addressed relative to the dataset beginning.

Adabas blocks comprise one or more PAM pages. An Adabas virtual track is made up of a fixed number of blocks, and an Adabas virtual cylinder comprises a fixed number of tracks. The definition of tracks and cylinders are independent of the physical device.

There are a number of predefined virtual devices for BS2000 that should meet most of the storage capacity needs that arise. It should be noted that the virtual memory requirement increases significantly with a larger block size.

Support for new device types, including user-defined block sizes, can be implemented in ADAIOR by modifying one of the TDCEs reserved for this purpose.

- Information to be Zapped into the First Free TDCE
- General Rules for Defining Device Block Sizes
- Device Types and Block Sizes

Information to be Zapped into the First Free TDCE

The information in the following tables must be zapped into the first free TDCE. The rules described in the section *General Rules for Defining Device Block Sizes* must be followed when changing the TDCE.

Label	Offset	Contents
TDCDT	00	Device type in unsigned decimal (X’3385’), must be numeric, and unique among all TDCEs
TDCKSN	02	Constant set number: must be uniquely chosen from the values X’28’ (reserved for BS2000 device type 2006), X’2B’, or X’2E’
TDCF	03	The flag bit must be set—TDCFFBA (X’80’) for FBA/PAM devices or TDCFCKD (X’40’) for CKD devices
TDCDT1	04	Set to zero under BS2000
TDCDT2	05	Set to zero under BS2000
TDCDT3	06	Set to zero under BS2000
TDCDT4	07	Set to zero under BS2000

Label	Offset	Contents
TDCMSBS	08	In BS2000, 32760 for compatibility. Refer to the TDCMSBS default table in *Maximum Sequential Block Size* in the Adabas z/OS installation instructions for more system- and device-related information.
TDCTPC	0A	Number of tracks per cylinder
TDCCIPT	0C	Number of FBA blocks or PAM pages per track (if TDCFFBA is set). For BS2000 less than or equal to 16.
TDCBPCI	0E	Number of bytes per FBA block or PAM page (2048 if TDCFFBA is set)
TDCABPT	10	Number of Associator blocks per track
TDCABS	12	Associator block size
TDCACPB	14	Number of FBA blocks or PAM pages per Associator block (if TDCFFBA is set)
TDCDBPT	16	Number of Data Storage blocks per track
TDCDBS	18	Data Storage block size
TDCDCPB	1A	Number of FBA blocks or PAM pages per Data Storage block (if TDCFFBA is set)
TDCWBPT	1C	Number of Work blocks per track
TDCWBS	1E	Work block size
TDCWCPB	20	Number of FBA blocks or PAM pages per Work block (if TDCFFBA is set)
TDCTSBPT	22	Number of TEMP or SORT blocks per track (if TDCFFBA is set)
TDCTSBS	24	TEMP or SORT block size
TDCTSCPB	26	Number of FBA blocks or PAM pages per TEMP or SORT block (if TDCFFBA is set)
TDCPBPT	28	Number of PLOG blocks per track
TDCPBS	2A	PLOG block size
TDCPCPB	2C	Number of FBA blocks or PAM pages per PLOG block (if TDCFFBA is set)
TDCCBPT	2E	Number of CLOG blocks per track
TDCCBS	30	CLOG block size
TDCCCPB2	32	Number of FBA blocks or PAM pages per CLOG block (if TDCFFBA is set)

In addition, the length of a sequential protection log block may have to be increased. This length is contained in the corresponding PTT entry in CSECT ADAIOI of the load module ADAIOI. The address of the first PTT entry is contained in the fullword at ADAIOI+X‘E4’.

Each PTT entry is X’10’ bytes long and has the structure shown below:

Label	Offset	Contents
PTTPN	00	Program number
PTTFT	01	File type
PTTN	02	DD name characters 2 - 8

PTTF	08	Flags:
		`OUT (X’80’) output`
		`BSAM (X’40’) BSAM`
		`BACK (X’20’) read backwards`
		`JCL (X’10’) BLKSIZE/LRECL/RECFM taken from DATADEF statement or label`
		`UNDEF (X’04’) undefined record format`
		`VAR (X’02’) variable record format`

-	09	Reserved
PTTMBS	0A	Maximum block size
-	0C	Reserved

The PTT entry for the sequential protection log can be identified by X‘12F1’ in its first two bytes.

General Rules for Defining Device Block Sizes

The following general rules must be followed when defining Adabas device block sizes:

- all block sizes must be multiples of 4
- a single block cannot be split between tracks (block size must be less than or equal to the track size)

Block Rules for ASSO/DATA

The following rules apply for Associator and Data Storage blocks:

- Associator block size must be greater than one-fourth the size of the largest FDT, and should be large enough to accept definitions in the various administrative blocks (RABN 1 - 30) and in the FCB
- The block sizes for Associator and Data Storage should be a multiple of 256, less four bytes (for example, 1020) to save Adabas buffer pool space
- The Associator and Data Storage block sizes must be at least 32 less than the sequential block size
- Data Storage block size must be greater than: (maximum compressed record length + 10 + padding bytes)

Block Rule for WORK

The Work block size must be greater than either (maximum compressed record length + 110) or (Associator block size + 110), whichever is greater.

Block Rules for TEMP/SORT

If ADAM direct addressing is used:

```
size > (maximum compressed record length + ADAM record length + 24);
size > 277 (maximum descriptor length + 24)
```

However, TEMP and SORT are generally read and written sequentially; therefore, the larger the TEMP/SORT block size, the better.

Block size for TEMP and SORT must be greater than the block size for Data Storage.

Block Rule for PLOG or SIBA

The following rules apply for PLOG or SIBA blocks:

- The PLOG or SIBA block size must be greater than either (maximum compressed record length + 110) or (Associator block size + 110), whichever is greater.

- It is also recommended that PLOG/SIBA be defined larger than the largest Data Storage block size. This avoids increased I/O caused by splitting Data Storage blocks during online ADASAV operations.

The block size (BLKSIZE) of a sequential file is determined as follows:

```
if PTTF(JCL) then BLKSIZE is taken from file assignment statement or label;
if PTTMBS > 0 then BLKSIZE = PTTMBS;
if PTTMBS = 0 then
if tape then BLKSIZE = 32760;
else BLKSIZE = TDCMSBS;
else if BLKSIZE in file assignment statement or label then use it;
if PTTF(OUT) then
if QBLKSIZE > 0 then BLKSIZE = QBLKSIZE;
if tape then BLKSIZE = 32760;
else BLKSIZE = TDCMSBS;
else error.
```

Note:
QBLKSIZE is an ADARUN parameter.

Using 3480/3490 Tape Cartridge Compression (IDRC)

The use of hardware compression (IDRC) is not recommended for protection log files.

Device Types and Block Sizes

The primary access method for direct access datasets used by Adabas under BS2000 is PAM (primary access method). The device types defined by Adabas establish a logical structure on a PAM dataset in order to process a fixed number of consecutive PAM blocks (e.g., one Adabas block consisting of two PAM blocks or one logical track consisting of four PAM blocks).

These device types are “artificial”; there is no relation to the physical devices being used. A maximum of 16 PAM blocks per track can be combined into one I/O call. For more than 16 PAM blocks per track, parameter chaining is used.

The artificial device types defined by Software AG for BS2000 systems are summarized in the following table. The ASSO, DATA, WORK, PLOG, CLOG, and TEMP/SORT/DSIM block sizes are given in RABNs per track.

Device	Trks/Cyl	PAM Blks/Trk	ASSO	DATA	WORK	PLOG	CLOG	TEMP/SORT/DSIM	Notes
2000	20	4	2048:4	4080:2	4096:2	4096:2	4096:2	4080:2	
2001	19	8	2044:8	4092:4	4096:4	4096:4	8192:2	8192:2	
2002	19	8	4092:4	8188:2	8192:2	8192:2	16384:1	16384:1	see note 1
2003	17	15	2044:15	6140:5	6144:5	6144:5	10240:3	10240:3	
2004	17	15	6140:5	10236:3	10240:3	10240:3	30720:1	30720:1	
2005	11	20	2044:20	4092:10	8192:5	8192:5	10240:4	10240:4	
2006	11	20	4092:10	8188:5	10240:4	10240:4	10240:4	10240:4	
2007	17	15	10236:3	30716:3	30720:3	30720:3	30720:3	30720:3	see note 2
2008	17	16	4092:8	32656:1	32760:1	32760:1	32760:1	32760:1	see note 1
2009	17	16	4092:8	32656:1	32740:1	32740:1	32740:1	32740:1	
2010	15	16	4092:8	8188:4	16380:2	16380:2	16380:2	16380:2	see note 1
2200	15	16	4092:8	8088:4	16380:2	16380:2	16380:2	16380:2	see note 1
2201	15	12	4092:6	12184:2	12288:2	12288:2	12288:2	12288:2	see note 1
2202	15	16	4092:8	16280:2	16380:2	16380:2	16380:2	16380:2	see note 1

Notes:

1. This device can be used with BS2000/NK4 disk types. In these cases, all direct access database files have been defined with a standard block size which is a multiple of 2.
2. Although supported, the 2007 device is not recommended for use with Adabas. Support for the 2007 will be removed in a later Adabas release.

If the current database device is not of a compatible type for NK4 disks and it is necessary to migrate it to those disks, you must use the ADAORD RESTRUCTUREDB utility as described in *Adabas Utilities*.

Splitting Datasets Across Volumes

For private volumes, splitting is possible under every LOGON user ID:

In ISP format:

```
/FILE dataset,DEVICE=D3480,VOLUME=PRIV01,SPACE=60000
/FILE dataset,DEVICE=D3480,VOLUME=PRIV02,SPACE=60000
```

In SDF format:

```
/CREATE-FILE dataset,PRIV-DISK(SPACE=(60000),VOLUME=PRIV01)
/MOD-FILE-ATTR dataset,PROT=(USER-ACC=*ALL)
/MOD-FILE-ATTR dataset,SUP=PRIV-DISK(SPACE=(60000),VOLUME=PRIV02)
```

For public volumes, the splitting is possible under every LOGON user ID if the master catalog entry of the pubset has the attribute:

```
PHYSICAL-ALLOCATION=USER-ALLOWED
```

This attribute is set by issuing the following command under TSOS:

```
/MOD-MASTER-CAT CAT-ID=ABC,PHYSICAL-ALLOCATION=USER-ALLOWED
```

Once this attribute is set, it is possible to split a dataset across two or more public volumes under any LOGON user ID that has the right of space allocation on that particular pubset.

In ISP format:

```
/FILE dataset,VOLUME=ABC.00,DEVICE=D3480,SPACE=60000
/FILE dataset,VOLUME=ABC.01,DEVICE=D3480,SPACE=60000
```

In SDF format:

```
/CREATE-FILE dataset,PUB(SPACE=(60000),VOLUME=ABC.00)
/MOD-FILE-ATTR dataset,PROT=(USER-ACC=*ALL)
/MOD-FILE-ATTR dataset,SUP=PUB(SPACE=(60000),VOLUME=ABC.01)
```

At this point, even a particular physical allocation can be made.

In ISP format:

```
/FILE dataset,VOLUME=ABC.02,DEVICE=D3480,SPACE=(20002,60000,ABS)
```

In SDF format (following the CREATE-FILE and MOD-FILE_ATTR...PROT specifications listed earlier):

```
/MOD-FILE-ATTR dataset,SUP=PUB(SPACE=ABSOLUTE(20002,60000),VOLUME=ABC.02)
```

The example extent covers physical PAM pages 20002 through 80001 on volume ABC.02. The required disk space must, of course, be available. If you are unsure of the available disk space, consult your system administrator.

Saving the Extent List of Datasets

The utility ADAR2E converts the extent list of given datasets into a JOB containing /CREATE-FILE commands. For more information, see the section on the ADAR2E utility in the *Adabas Utilities* documentation.