

## Large Files and Volumes in BS2000/OSD

Issue September 2007

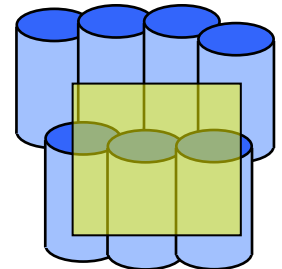
Pages 8

### Files / volumes larger than 32 GB

Observers continue to note an annual percentage increase in the higher two-digit range in the data sector.

This growth not only has an impact on file sizes but also makes it necessary to store the data on volumes / magnetic disk devices.

The physical capacity of magnetic disks is currently about 500 gigabytes (GB) per magnetic disk (Low-cost FC drives in Symmetrix DMX-3 and FibreCAT CX-3). Capacities are likely to double or quadruple in the near future (The SATA II drives in Symmetrix DMX-4 and FibreCAT CX-3 announced for End of 2007 will feature a 750 GB capacity).



BS2000/OSD meets these challenges by raising the limits for files and volumes from the former ceiling of 32 GB (up to BS2000/OSD V4.0).

With the release of BS2000/OSD V5.0, the BS2000 operating system supports files and volumes with capacities up to 4 terabytes (4000 GB).

The supplementary version BS2000/OSD-BC V6.0B is able to support even larger data configurations. The maximum number of files per pubset has been doubled, and the maximum user allocated pubspace within an SM pubset (System Managed Pubset) can exceed the current limit of 4 terabytes.

#### Content

<b>Files / volumes larger than 32 GB</b>	1
<b>Summary</b>	2
<b>Introducing large objects in the datacenter</b>	3
<b>Catalog expansion in BS2000/OSD-BC V6.0B</b>	4
Larger data configurations through catalog expansion	4
Viewing the current catalog situation	4
Automatic catalog enlargement	4
Modifications in the SW products DRV, JV, SCA, SHC-OSD and SPACEOPT	4
<b>Enhanced Support for large SM Pubsets</b>	4
<b>Support for large files in application programs</b>	5
Basics, assembler interfaces	5
High-level programming languages (compilers), LMS, EDT	5
Data storage/management systems	6
Utilities supporting large files	6
<b>Executability of programs in configurations including large files</b>	7
<b>Fujitsu Siemens Computers BS2000/OSD software products for configurations including large files</b>	8
<b>Large volumes</b>	8

## Summary

BS2000/OSD V5.0 permits the definition of pubsets containing volumes and files larger than 32 GB, henceforth referred to as large volumes and large files.

Programs that are to work with large files must be validated and modified if necessary:

In fact, suitably extended DMS user interfaces (assembler macros with 4-byte fields) to enable the use of large files in application programs have been available since OSD V1, but often these are not yet used in the application programs. The semantics of the programs should also now be geared to processing block counters etc. in 4-byte fields. OSD V5.0 provides new language elements at JCL and program level to enable selective access to large files.

For production and execution, programs in high-level programming languages that are to process large files require new versions of C, COBOL, CRTE or Java. In addition, C programs must be suitably modified (use of the 64-bit API).

With the data base management system SESAM/SQL Server as of version 3.1B, using BS2000/OSD V5.0 SESAM/SQL database files may have up to 64 Gigabyte in size. With the data base management system UDS/SQL as of version 2.3B, using BS2000/OSD V5.0 UDS/SQL database files may have up to 128 Gigabyte in size. However, the databases and data storage/management systems ORACLE, LEASY, GOLEM do not need to support large files. Today's capacity limits are far from being exceeded.

In configurations including large files, application programs that do not open large files can usually run without modification. In a small number of cases it may be necessary to adapt the program because of the new format of certain management information. FSC provides a complete ready-to-run software configuration of FSC BS2000 software products for BS2000/OSD-BC as of V5.0 suitable for use in an environment including large files.

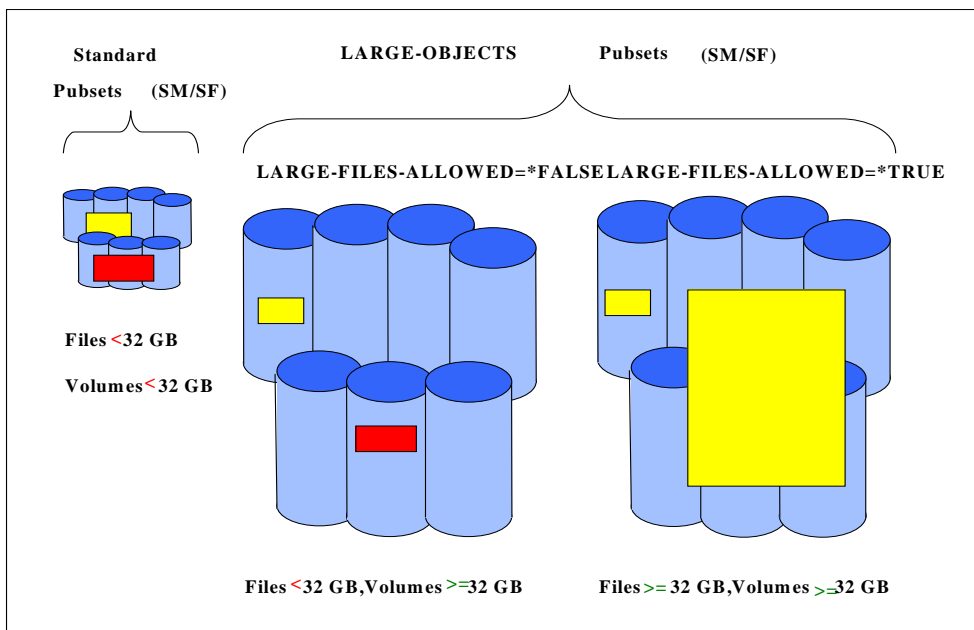
Large volumes can be configured at S and SX servers on the external disk storage subsystems (Symmetrix, FibreCat) connected via Fibre Channel. Large volumes based on BS2000/OSD as of V5.0 can be compatibly introduced without the need for modifications in application programs.

To facilitate migration to large files and large volumes, a manual titled 'Files and volumes larger than 32 GB' has been published, containing a comprehensive description of the technical issues and background.

## Introducing large objects in the datacenter

Large objects are introduced via new pubset types. There are two new classes of pubsets for large objects. The first class permits large volumes, but limits the allowed file size to 32 GB. From the user viewpoint, these pubsets behave almost like conventional pubsets, but they can only be imported under OSD as of V5.0. The second class permits both large volumes and large files. Large objects are supported on SF (Single Feature) and SM (System Managed) pubsets, but no large objects are possible on private disks.

The attributes of a pubset in relation to large volumes and files are specified with SIR-IP at pubset generation time and subsequently cannot be changed to provide less comprehensive support for large objects. Existing pubsets can be upgraded to large pubsets with or without support for large files using the /SET-PUBSET-ATTRIBUTES command.



The new LARGE-VOLUMES operand of the /SET-PUBSET-ATTRIBUTES command specifies whether the pubset may contain volumes > 32 GB. Specifying LARGE-VOLUMES=\*ALLOWED(LARGE-FILES=\*ALLOWED) permits both large volumes and large files.

The following upgrades, and only these, are possible:

- Standard pubset -> Large-object pubset without large files
- Standard pubset -> Large-object pubset with large files
- Large-object pubset without large files-> Large-object pubset with large files

The extended file capacity applies only to non-Pamkey files. The home pubset is excluded from support for files larger than 32 GB.

## Catalog expansion in BS2000/OSD-BC V6.0B

### Larger data configurations through catalog expansion

Every pubset has a catalog in which the file allocations and job variables are stored in catalog entries (TSOSCAT file catalog). A pubset catalog currently (in BS2000 versions V5.0 and V6.0) comprises a maximum of 16,184 4K blocks with large-object pubsets (or 8192 4K blocks with conventional pubsets). This is equivalent to approx. 120,000 - 160,000 files/JVs (or approx. 60,000 - 80,000 files/JVs).

Once this space has been exhausted, no more new IDs can be set up in the pubset. Existing users may also be unable to create new files or increase the size of existing ones.

In BS2000/OSD-BC V6.0B, the catalog is doubled in size. More precisely, the catalog is expanded by 98% (15,808 additional catalog blocks for volsets and SF pubsets).

In the SM pubset, the catalog is split into subcatalogs. Each volume set possesses a catalog for the catalog entries of the files contained in the volume set. In addition, three special catalogs are maintained for the catalog entries of the private disks/tape files, the job variables and the migrated/non-allocated files of the SM pubset.

In BS2000/OSD-BC V6.0B, the SM pubset special catalogs can also be expanded by adding further special catalogs dynamically with a new command (ADD-CATALOG-FILE). Max. 100 subcatalogs are possible per catalog.

Starting with BS2000/OSD-BC V6.0B, the two existing catalog formats "normal" and "large" (implicit, for pubsets containing objects > 32 GB) are joined by a third format: "extra large". The "extra large" catalog format is optional and not compatible with the existing formats, i.e. if it is used for a pubset, it will not be possible to return to an earlier BS2000 version with this pubset. Pubsets with "extra large" catalog format can no more be imported in BS2000/OSD-BC < V6.0B.

Pubsets can be converted into "extra large" pubsets when being imported or when being set up with SIR.

Note: The Large-Objects-Pubset feature for large volumes or large files and the "extra large" catalog format can be introduced independently from each other. That means, a pubset with „extra large“ catalog does not allow large volumes and files by default.

### Viewing the current catalog situation

The new SHOW-PUBSET-CATALOG-ALLOCATION command outputs the catalog type (normal, large, extra large), the occupancy level per catalog (usage/file size), and the expandability per catalog.

### Automatic catalog enlargement

Once a catalog is filled to 90% of its capacity, it is automatically enlarged by the system provided that is possible without changing the catalog format.

For special catalogs in the "extra large" format, a new subcatalog will then be created if none of the existing subcatalogs can be enlarged.

### Modifications in the SW products DRV, JV, SCA, SHC-OSD and SPACEOPT

Support of the "extra large" catalog format is included in the new product versions DRV V3.1 and SHC-OSD V5.0. Support for a 6-digit number of JVs is provided in the JV V14.0C correction version. To support the "extra large" catalog format, correction versions are provided for the SCA and SPACEOPT products: SCA V15.0B and SPACEOPT V3.0B.

## Enhanced Support for large SM Pubsets

The maximum size of a volume set is 4 TB, so the maximum overall size of an SM pubset is 255 \* 4 TB (more than 1.000 TB). In BS2000/OSD-BC V6.0B the maximum user allocated space in an SM pubset can exceed the previous limit of 4 TB. The MODIFY-USER-PUBSET-ATTRIBUTES command is being modified by supporting the value \*UNLIMITED for the TOTAL-SPACE parameter within the PERM-SPACE-LIMITS, TEMP-SPACE-LIMITS and WORK-SPACE-LIMITS operand. The user space quota is displayed with the SHOW-USER-ATTRIBUTES command. The SHOW-FILE-ATTRIBUTES output displays the space in entities of thousands PAM pages, if the previous maximum value of 4 TB is exceeded.

# Support for large files in application programs

## Basics, assembler interfaces

Under BS2000/OSD-BC as of V5.0, support for large files and large volumes is based on the universal use of 4-byte block numbers and 4-byte counters file and disk sizes within the operating system.

Programs that are to process large files must be able to handle these 4-byte fields. Although 4-byte block numbers and block counters have been used at all user interfaces (FSTAT, UPAM) since the release of OSD V1, across-the-board use of these interface versions cannot be taken for granted. Semantic dependencies and assumptions may also exist (example: a 4-byte block number has been used until now to represent special objects in NK-ISAM.)

A hard strategy is therefore being imposed as standard for the introduction of large files. It is assumed that every program that is to access large files must be checked to verify whether its semantics can cope with large files. This is intended to preempt accidental corruption of data. The system prevents uncontrolled overstepping of the 32 GB limit by providing a special function which must be used for opening large files. A parameter enabling the validity of large files to be signaled is being introduced at the OPEN interface as well as via the /ADD-FILE-LINK command interface and the FILE program interface.

## High-level programming languages (compilers), LMS, EDT

For programs written in the high-level programming languages C, COBOL and Java, processing (reading, writing) of files >32 GB is possible under BS2000/OSD V5.0 provided the application has been regenerated using one of the following combinations of compilers and runtime system:

Runtime system:	CRTE as of V2.3C release
C program:	C/C++ C3.0C or higher (only C API (open64, ... ), no C++ IOSTREAM)
COBOL program:	COBOL85 V2.2B or higher, COBOL2000 V1.0 or higher
Java program:	JENV V1.4 or higher

Any programs generated with other compiler and/or runtime system versions or with other compilers of the BS2000 programming system cannot process large files (>32 GB). However, they are still executable despite this restriction, even on system configurations including large volumes or large files.

## Compilation

Neither the source files of the BS2000 compilers nor the output files directly generated by them (object modules, listing files, diagnostic files, etc.) may be large files, but at best library members < 32 GB in a library > 32 GB.

The resource restrictions previously applicable to the individual BS2000 compilers have not changed for BS2000/OSD V5.0 and still hold. Resource usage during compilation of a program is heavily dependent on the individual program structure and cannot be quantified in generally valid limits for file sizes, source or listing lines, etc. In real-world application, resource bottlenecks are encountered by the individual compilers at a very much earlier stage, long before there would be even a remote possibility of reaching the previous maximum file size of 32GB supported by BS2000/OSD. When resource limits are exceeded, appropriate error messages are output (as previously) by the individual compilers.

## LMS/PLAM, EDT

PLAM V3.3A has been extended to enable libraries to be large files. Library members are subject to no additional restrictions in terms of size. LMS and LMSCONV V3.3A and V3.3B have been extended to enable large files to be included as members in libraries and extracted again as files.

Files editable by EDT can be max. 99,999,999 source lines in size (< 26 GB).

## C/C++, NFS, POSIX file system

NFS V3.0, available since May 2001 for BS2000/OSD V3.0 or higher, already supports 64-bit systems with large files. This enables remote file systems containing files larger than 2 GB to be mounted. To allow users to work with files of this size, the corresponding C application interface, which permits access to large files, is supported by C/C++ V3.0C and CRTE V2.2B. Large file systems and files can be set up locally on the BS2000 system as of POSIX A33 under BS2000/OSD V4/V5. A maximum file size of 32 GB -1 is supported locally via ufs under the previous BS2000/OSD version BS2000/OSD-BC V4.0. With POSIX A33 under BS2000/OSD-BC V5.0, the local file system supports large files up to 1024 GB-1. POSIX A33 was released with correction package KP 2/2002 in August 2002.

## Data storage/management systems

The production data held by customers is mainly stored in databases.

The databases and data storage/management systems (SESAM, UDS, ORACLE, LEASY, GOLEM) support large files if necessary and reasonable.

It should be borne in mind that the DB administration activities for files of this size will take a proportionately longer time (long backup runs and recovery times) and may adversely affect 24-hour operation.

This problem can be solved for data residing on Symmetrix storage systems by performing data backup and production in parallel, using HSMS Concurrent Copy and the Symmetrix functions TimeFinder/Mirror (BCV) or TimeFinder/Clone (Clone requires HSMS V8.0B and BS2000/OSD-BC V7.0). For example, the SESAM/SQL online backup with HSMS supports this feature. UDS/SQL and Oracle as well allow online backup using the Symmetrix storage subsystem's TimeFinder feature.

If parallel operation is not possible, it may make sense to split user data over multiple files.

### SESAM/SQL

With BS2000/OSD V5.0, SESAM/SQL server as of V3.1B supports the access to data base files, which can be large up to 64 gigabytes. SESAM/SQL V4.0 supports the partitioning of a table into up to 16 partitions. As a partition can be large up to 64 gigabytes, a table can be large up to 1024 gigabytes. The general release of SESAM/SQL server V4.0 took place in 2006, January.

### UDS/SQL

With BS2000/OSD V5.0 UDS/SQL V2.3B supports the access to data base files, which can be large up to 128 gigabytes. The general release of UDS/SQL server V2.3B took place in July 2003.

### LEASY

For the time being, large files are not supported. Today's capacity limit of 32 GB is considered to be sufficient.

### GOLEM

In GOLEM V9.2 (release 7.2001), the capacity of the largest GOLEM file (Z file) has been increased fourfold, from 8 to 32 GB. It is unlikely that users will come up against this capacity limit in the foreseeable future.

### ORACLE

Support for large files in ORACLE appears not to be an urgent requirement at this stage, because ORACLE distributes the tablespaces across multiple files and consequently individual files come nowhere near the 32 GB limit.

## Utilities supporting large files

The system administrator and the user can call on a number of utility routines, such as HSMS, ARCHIVE, PERCON, SORT, SPACEOPT, etc., to assist in the management and processing of files. New versions of these system support software products are available to support files/volumes larger than 32 GB.

### HSMS/ARCHIVE as of V6.0

As of OSD V5.0, files and volumes  $\geq 32$  GB are supported in BS2000 in specially attributed pubsets. For this, HSMS/ARCHIVE as of V6.0 ensure that

- large files cannot be imported into earlier BS2000 versions,
- large files are only imported into the containers provided for them,
- earlier HSMS versions can reliably handle these new pubsets,
- the attribute "large disk" does not have an impact on the backup resources, so that small files on large disks can be handled in the same way as today (small files on small disks).

### PERCON as of V2.7

PERCON supports large files. This is not visible at the PERCON input interface; all PERCON outputs of record or block counters (in messages or to \*SYSOUT) have been extended from 8 to 16 digits.

### SORT V7.8

SORT supports large files, i.e. input files can be  $>32$  GB, and SORT can create output, scratch and auxiliary files  $> 32$  GB. In addition SORT V7.8B, released with KP 2/2002, supports large POSIX files.

### SPACEOPT as of V2.0

SPACEOPT can reorganize large files as well as small files on large pubsets.

**openFT V9.0**

Support for files > 32 GB is realized with *openFT* V9.0 (released in 2004, February). As of V9.0 it is possible to transfer files with size up to 4 terabytes and administrate them with file management.

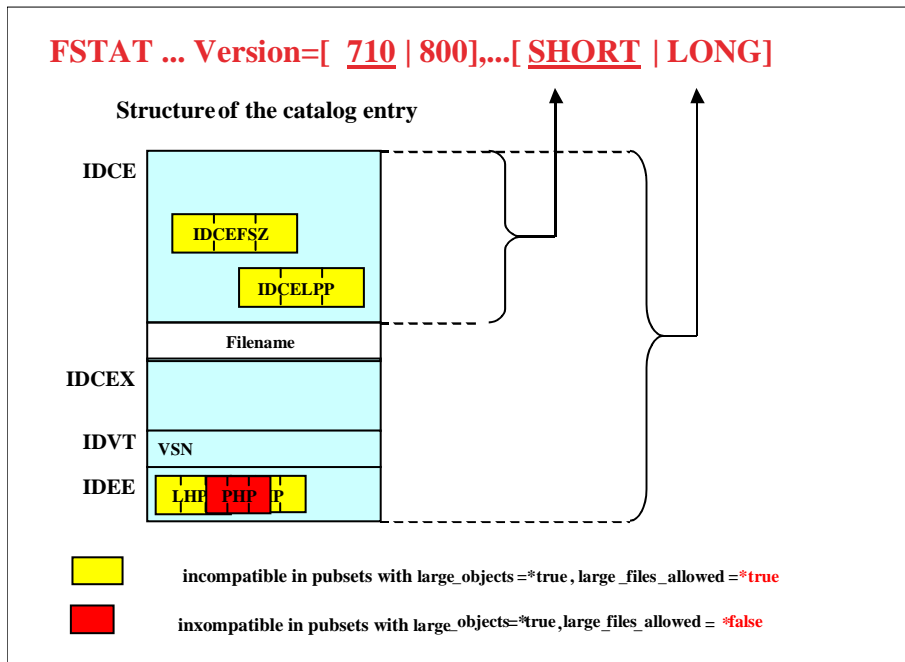
## Executability of programs in configurations including large files

In certain cases the existence of large files under BS2000/OSD-BC as of V5.0 can have repercussions on programs that themselves do not directly process any large files:

If old versions (710, 800) of the FSTAT interface are used, calls including partially qualified file names or wildcards will also lead to errors if the resulting hitlist contains files >= 32 GB.

If the SHORT or LONG operands are specified, the output supplied by the FSTAT 710/800 variants consists of the catalog information in BS2000 V10.0 format. The extent list and the data fields for File-Size and Last-Page-Pointer are represented by 3 bytes only in the output. Calls to these variants that result in an overflow of the 3-byte data fields are rejected (return code DMS0576).

It should be noted in this context that the default value of the FSTAT interface version is VERSION=710 (for compatibility reasons).



In order to cope with the existence of large files, programs of this type must be converted to a higher FSTAT version or implement a specific FSTAT parameterization. In addition to the interface, it may also be necessary to convert data structures in the program.

Even with programs written in high-level programming languages, there may be exceptions/incompatibilities with regard to the executability of existing legacy applications in configurations including large files in BS2000/OSD V5.0, if the programs use system functions whose interfaces have changed relative to OSD V5.0, such as in the case of DMS with the handling of large files. For example, the system function available in C/C++ (systemcmd) combined with an FSTAT command containing a large file in the hitlist runs to error if an old macro version is used in the runtime system when the /FSTAT command is called.

This runtime incompatibility when executing a legacy application can be remedied by re-linking with CRTE V2.3C.

## Fujitsu Siemens Computers BS2000/OSD software products for configurations including large files

To enable customers to create large files, Fujitsu Siemens Computers provides a complete software configuration for BS2000/OSD-BC as of V5.0 which can run in an environment including large files. So that customers can verify whether their specific software configuration will run if large files are present in the system, a classification system is being introduced for the BS2000 products. This classification is published in binding form in the manual titled 'Files and volumes larger than 32 GB', the migration guide for converting to large files and large volumes.

The following classes of products are available:

- Class A: A program can process large files without restriction. This behavior is referred to as LARGE-FILES-capable.
- Class B: A program is not ready for processing large files and/or their metadata, but is able to reject associated accesses, which have to be regarded as errors, in a defined manner, or no accesses to files and their metadata are performed in the program. This behavior is referred to as LARGE-FILES-compatible.
- Class C: A program is not ready for processing large files and is also able to reject associated accesses in a defined manner. This behavior is referred to as LARGE-FILES-incompatible.

LARGE-FILES-compatible or -capable programs are a requirement for configurations that include large files.

Class C (LARGE-FILES-incompatible) is assigned only to versions of Fujitsu Siemens Computers products for which there are successor versions with an A or B classification, as well as to products which have already reached the end-of-maintenance deadline.

## Large volumes

Large volumes can be configured at S and SX servers on the external disk storage subsystems connected via fibre channel (Symmetrix, FibreCAT). When connected via ESCON channel, the maximum volume size is ca. 27 GB with the CKD format and almost 32 GB with the FBA format (Symmetrix DMX).

If while operating larger volumes the volume's load increases corresponding to the larger amount of data, precautions should be taken for their efficient operation:

- In order to increase performance (and data security) external disk subsystems can be operated with the RAID disk technique. For large volume operation, RAID 5 or better RAID 1/0 is recommended. With these RAID levels, the data of a single logical volume are spread over several physical disks (Striping).
- In addition it is recommended to use the PAV BS2000/OSD function (Parallel Access Volume). PAV allows parallelization of the disk input / output operation. For S server/FC, Dynamic PAV is available for BS2000/OSD-BC as of V5.0C since CP 2/06, for SX servers/FC there is an analogous function available via RSC for disk I/Os, for BS2000/OSD-BC as of V6.0B (part of OSD/XC as of V2.0 since CP 2/05).

Both measures in combination: RAID 5 / RAID 1/0 operation and use of PAV / RSC lead to substantial improvements in TP and batch operation (in multi-task batch mode) in terms of I/O times as well as throughput.

BS2000/OSD-BC as of V5.0 permits large volumes to be initialized with VOLIN: VOLIN enables a corresponding attribute (LARGE\_VOLUME) to be stored in the basic record of the SVL.

A soft migration strategy is being adopted for introducing large volumes (without large files):

Large volumes can be compatibly introduced into a system without preparatory measures. Existing data resources can also be stored and processed on large volumes, without conversion overhead or need for validation. Without large files, a volume-size  $\geq 32$  GB only affects the Physical Half Page Numbers (PHPs), which play virtually no role in user interfaces (exception: physical allocation) and in no event can be modified by user programs: With files on large volumes, the FSTAT FORM=LONG tolerates the overflow of physical half pages (PHPs) in the extent list; non-displayable PHPs are assigned the value 'X'FFFFFFF'. It is assumed that PHPs are never or only extremely rarely evaluated in user programs.

Given these general conditions, large volumes can be compatibly introduced, i.e. transparently with regard to application programs.