

DB2 VSAM Cluster Management

| | |
|---|----------|
| DB2 VSAM CLUSTER MANAGEMENT | 1 |
| BASIC VSAM AND SMS INFORMATION | 1 |
| <i>VSAM Extent Consolidation</i> | 2 |
| <i>Space Relief</i> | 2 |
| DB2 USAGE OF VSAM CLUSTER | 3 |
| <i>Primary space allocation</i> | 4 |
| <i>Secondary space allocation (Version 8)</i> | 4 |
| <i>Extending user-managed data sets</i> | 4 |
| <i>Option LARGE on CREATE TABLESPACE statement</i> | 5 |
| <i>Option DSSIZE integer G on CREATE TABLESPACE statement</i> | 5 |
| <i>Creating EA-enabled table spaces and index spaces</i> | 5 |
| SLIDING SECONDARY ALLOCATION SIZE | 6 |
| DB2 LIMITS | 6 |
| 3390 DISK CAPACITY | 7 |
| DB2 CLUSTER FAILURES | 7 |
| <i>IEC161I</i> | 7 |
| <i>DSNP007I</i> | 8 |
| <i>DSNP001I</i> | 8 |

Basic VSAM and SMS Information

A VSAM data set is limited to 4 GB across all volumes unless Extended Addressability is specified in the SMS data class definition. System requirements restrict the number of volumes that can be used for one data set to 59.

A primary space allocation is the initial amount of allocated space. When the primary amount on the first volume is used up, a secondary amount is allocated on that volume. Each time a new record does not fit in the allocated space, the system allocates more space in the secondary space amount. The system repeats allocating this space until the volume is out of space or the volume extent limit of 123 is reached.

A VSAM data set can be expanded to 123 extents per volume. In addition to the limit of 123 extents per volume, these are the other limits on the number of extents for a VSAM data set:

- If non-SMS-managed, then up to 255 extents per component.
- If SMS-managed, then the following are true:
 - If not striped and without the extent constraint removal parameter in the data class, then up to 255 extents per component.
 - If striped and without the extent constraint removal parameter in the data class, then up to 255 extents per stripe.
 - If the extent constraint removal parameter in the data class is set to a value of Y, then the number of extents is limited by the number of volumes for the data set.

VSAM attempts to extend a data set when appropriate. Each attempt to extend the data set might result in up to 5 extents. This is called the „five extents allocation rule“, which may be overridden by the SMS Dataclass „Space Constraint Relief“ attribute. Without, the system tries to satisfy your primary or secondary space request with no more than five extents. If you request a large amount of space or the space is fragmented, the system might need more than five extents.

For nonstriped VSAM data sets, you can specify in the SMS data class parameter whether to use primary or secondary allocation amounts when extending to a new volume. You can expand the space for a nonstriped VSAM component to 255 extents. For SMS-managed VSAM data sets, this extent limit is removed, and the theoretical limit is the maximum number of volumes (59), times 123 extents per volume, or 7257 extents.

You can expand the space for a striped VSAM component to 255 times the number of stripes. The VSAM limit of 255 extents is still enforced for any non-SMS-managed data set. The system reserves the last four extents for extending a component when the system cannot allocate the last extent in one piece (i.e. 251 available extents).

z/OS since 1.7 allows for VSAM LDSs to grow to a theoretical limit of 7257 extents. Your Storage Administrator must set Extent Constraint Removal to YES in order to exceed the previous limit of 255 extents per component.

Many types of data sets are limited to 65535 total tracks allocated on any one volume, and if a greater number of tracks is required, this attempt to create a data set will fail. Data sets that are not limited to 65535 total tracks allocated on any one volume are:

- Large format sequential
- Extended-format sequential
- UNIX files
- PDSE
- VSAM

If a virtual input-output (VIO) data set is to be SMS managed, the VIO maximum size is 2000000 KB (2 GB), as defined in the Storage Group VIO Maxsize parameter.

Space is allocated for non-system-managed data sets or system-managed data sets without the guaranteed space attribute in the storage class as follows. If you allocate a new data set and specify SPACE=(TRK,(2,4)); this initially allocates two tracks for the data set. As each record is written to the data set and these two tracks are used up, the system automatically obtains four more tracks. When these four tracks are used, another four tracks are obtained. The same sequence is followed until the extent limit for the type of data set is reached.

- A sequential data set can have 16 extents on each volume.
- An extended-format sequential data set can have 123 extents per volume.
- A PDS can have 16 extents.
- A direct data set can have 16 extents on each volume.
- A VSAM data set can have up to 255 extents per component.
- A striped VSAM data set can have up to 255 extents per stripe.
- A PDSE can have 123 extents.
- An HFS data set can have 123 extents on each volume.

You can allocate space for a multivolume data set the same as for a single volume data set. DASD space is initially allocated on the first volume only (exceptions are striped extended-format data sets and guaranteed space data sets). When the primary allocation of space is filled, space is allocated in secondary storage amounts (if specified). The extents can be allocated on other volumes. VIO space allocation is handled differently from other data sets.

When a multivolume VSAM data set extends to the next volume, the data class specifies if the initial space allocated on that volume is the primary or secondary amount. The default is the primary amount. After the primary amount of space is used up, space is allocated in secondary amounts. By using a data class, it is possible to indicate whether to take a primary or secondary amount when VSAM extends to a new volume.

When a multivolume non-VSAM, non-extended-format data set extends to the next volume, the initial space allocated on that volume is the secondary amount.

VSAM Extent Consolidation

Starting with z/OS V1.5 the system consolidates adjacent extents for VSAM data sets when extending on the same volume. VSAM extent consolidation is automatic and requires no action on your part. If the extents are adjacent, the new extent is incorporated into the previous extent. VSAM LDSs that are used for DB2 must be SMS managed for extent consolidation to take effect. Extent consolidation is not used for non-SMS managed data sets, in which case the extent allocation works as before.

Space Relief

The SMS Data Class can assign an attribute that would help resolve some space-related problems. During allocation, there might not be enough space on volumes to meet the requested space. SMS volume selection can sometimes solve this problem by trying all candidate volumes before failing the allocation. You can also use the Space Constraint Relief and Reduce Space Up To (%) attributes to request that an allocation be retried with a space request reduced by a certain percent if it fails due to space constraints. SMS retries the allocation by combining any of the following tasks:

- Spreading the requested quantity over multiple volumes. This option is not permitted for data sets allocated with the guaranteed space attribute for DB2 managed data sets.

- Allocating a percentage of the requested quantity. This option is not permitted for data sets allocated with the guaranteed space attribute.
- Using more than 5 extents for primary allocations for a new volume or another volume that is being used for additional extents.

VSAM and non-VSAM multi-striped data sets do not support space constraint relief. However, single-striped VSAM and non-VSAM data sets can use space constraint relief.

Space Constraint Relief generally works on ...

- new datasets (primary allocation)
- extents of datasets on new volumes (which equals new dataset handling)

DB2 Usage of VSAM Cluster

The maximum size is 2 GB for a data set of a simple space, and 1, 2, or 4 GB for a data set containing a partition. Large partitioned table spaces and indexes on large partitioned table spaces have a maximum data set size of 4 GB.

When a data set in a segmented or simple table space reaches its maximum size of 2 GB, DB2 might automatically create a new data set. The primary data set allocation is obtained for each new data set.

When the data set is created, DB2 always allocates a primary allocation space on a volume that has space available and is specified in the DB2 storage group. Any new extension always gets a secondary allocation space. When the extensions reach the end of the volume, DB2 accesses all candidate volumes from the DB2 storage group and issues the access method services command ALTER ADDVOLUMES to add all volumes in the integrated catalog as candidate volumes for the data set. DB2 then makes a request to extend a secondary allocation space on any one of the candidate volumes that has space available. After the extension is successful, DB2 issues the access method services command ALTER REMOVEVOLUMES to remove all candidate volumes from the integrated catalog for the data set.

DB2 extends data sets when:

- The requested space exceeds the remaining space
- 10 percent of the smaller allocation space (but not over 10 allocation units such as tracks or cylinders) exceeds the remaining space

If DB2 fails to extend a data set with a secondary allocation space because there is no secondary allocation space available on any single candidate volume of a DB2 storage group, DB2 tries again to extend with the requested space, if the requested space is smaller than the secondary allocation space. Use IFCID 258 in statistics class 3 to monitor data set extension activity.

Extending nonpartitioned spaces: For a nonpartitioned table space or an index space, DB2 defines the first piece of the page set starting with a primary allocation space, and extends that piece with secondary allocation spaces. When the end of the first piece is reached, DB2 defines a new piece (which is a new data set) and extends to that new piece starting with a primary allocation space.

Extending partitioned spaces: For a partitioned table space or an index space, each partition is a data set; therefore, DB2 defines each partition with the primary allocation space and extends each partition's data set with secondary allocation space, as needed.

When data extension fails: If a data set uses all possible extents, DB2 cannot extend that data set. For a partitioned page set, the extension fails only for the particular partition that DB2 is trying to extend. For nonpartitioned page sets, DB2 cannot extend to a new data set piece, which means the extension for the entire page set fails.

To avoid extension failures, the value of $(PRIQTY + \text{max_extents} \times \text{SECQTY})$ must be at least as large as the data set size (as specified on the DSSIZE clause or the implicit size for that type of page set). For nonpartitioning indexes, that value must reach the value for PIECESIZE (explicitly or implicitly specified). If DB2 reaches the maximum number of extents before reaching the limit, the extension fails.

Table spaces and indexes that span more than one data set require special procedures. Partitioned table spaces can have different partitions allocated to different DB2 storage groups. Nonpartitioned table spaces or indexes only have the additional data sets to rename (those with the lowest level name of A002, A003, and so on).

- A (simple¹ or) segmented tablespace (linear dataset) can reach a maximum number of 32 datasets for a nonpartitioned table space (or 254 data sets for LOB tablespaces).
- For a partitioned tablespace, a partitioning index, or a nonpartitioning index on a partitioned table space, the maximum is 4096 data sets.

Primary space allocation

By default, DB2 uses the following values for primary space allocation of DB2–managed data sets:

- 1 cylinder (720 KB) for non-LOB table spaces
- 10 cylinders for LOB table spaces
- 1 cylinder for indexes

To indicate that you want DB2 to use the default values for primary space allocation of table spaces and indexes, specify a value of 0 for the following parameters on installation panel DSNTIP7.

Secondary space allocation (since DB2 UDB Version 8 for z/OS)

DB2 can calculate the amount of space to allocate to secondary extents by using a sliding scale algorithm. The first 127 extents are allocated in increasing size, and the remaining extents are allocated based on the initial size of the data set:

- For 32 GB and 64 GB data sets, each extent is allocated with a size of 559 cylinders.
- For data sets that range in size from less than 1 GB to 16 GB, each extent is allocated with a size of 127 cylinders.

This approach has several advantages:

- It minimizes the potential for wasted space by increasing the size of secondary extents slowly at first.
- It prevents very large allocations for the remaining extents, which would likely cause fragmentation.
- It does not require users to specify SECQTY values when creating and altering table spaces and index spaces.
- It is theoretically possible to always reach maximum data set size without running out of secondary extents.

In the case of severe DASD fragmentation, it can take up to 5 extents to satisfy a logical extent request. In this situation, the data set does not reach the theoretical data set size

DB2 uses a sliding scale for secondary extent allocations of table spaces and indexes when:

- You do not specify a value for the SECQTY option of a CREATE TABLESPACE or CREATE INDEX statement
- You specify a value of -1 for the SECQTY option of an ALTER TABLESPACE or ALTER INDEX statement.

Otherwise, DB2 always uses a SECQTY value for secondary extent allocations, if one is explicitly specified.

Maximum allocation of secondary extents:

| Maximum data set size (in GB) | Maximum allocation (in cylinders) | Extents required to reach full size |
|----------------------------------|---------------------------------------|--|
| 1 | 127 | 54 |
| 2 | 127 | 75 |
| 4 | 127 | 107 |
| 8 | 127 | 154 |
| 16 | 127 | 246 |
| 32 | 559 | 172 |
| 64 | 559 | 255 |

Extending user-managed data sets

A user-managed data set is allocated by using only volumes defined for that data set in the ICF catalog. Before the current volume runs out of space, you must issue the Access Method Services commands ALTER ADDVOLUMES or ALTER REMOVEVOLUMES for candidate volumes.

¹ depreciated with DB2 Version 9 vor z/OS

Option LARGE on CREATE TABLESPACE statement

Identifies that each partition of a partitioned table space has a maximum partition size of 4 GB, which enables the table space to contain more than 64 GB of data. The preferred method to specify a maximum partition size of 4 GB and larger is the DSSIZE clause. The LARGE clause is for compatibility of releases of DB2 for OS/390 and z/OS prior to Version 6.

Option DSSIZE integer G on CREATE TABLESPACE statement

A value in gigabytes that indicates the maximum size for each partition or, for LOB table spaces, each data set. If you specify DSSIZE, you must also specify Numparts or LOB.

The following values are valid: 1G (1 gigabyte), 2G, 4G, 8G, 16G, 32G and 64G.

To specify a value greater than 4G, the following conditions must be true:

- DB2 is running with DFSMS Version 1 Release 5.
- The data sets for the table space are associated with a DFSMS data class that has been specified with extended format and extended addressability.

For all table spaces except LOB table spaces, if DSSIZE (or LARGE) is omitted, the default for the maximum size of each partition depends on the value of Numparts:

| If Numparts is ... | Maximum partition size is... |
|--------------------|------------------------------|
| 1 to 16 | 4 GB |
| 17 to 32 | 2 GB |
| 33 to 64 | 1 GB |
| 65 to 254 | 4 GB |

The partition size shown is not necessarily the actual number of bytes used or allocated for any one partition; it is the largest number that can be logically addressed. Each partition occupies one data set.

For LOB table spaces, if DSSIZE is not specified, the default for the maximum size of each data set is 4 GB. The maximum number of data sets is 254.

When you define a table space with DSSIZE, you automatically give the same size to all indexes that point to that tablespace.

Creating EA-enabled table spaces and index spaces

DFSMS has an extended-addressability function, which is necessary to create data sets that are larger than 4 GB. Therefore, the term for page sets that are enabled for extended addressability is *EA-enabled*. You must use EA-enabled table spaces or index spaces if you specify a DSSIZE that is larger than 4 GB in the CREATE TABLESPACE statement.

To create EA-enabled page sets, you must:

1. Use SMS to manage the data sets that are associated with the EA-enabled page sets.
2. Associate the data sets with a *data class* (an SMS construct) that specifies the extended format and extended addressability options.
To make this association between data sets and the data class, use an automatic class selection (ACS) routine to assign the DB2 data sets to the relevant SMS data class. The ACS routine does the assignment based on the data set name. No performance penalty occurs for having non-EA-enabled DB2 page sets assigned to this data class, too, if you would rather not have two separate data classes for DB2.

For user-managed data sets, you can use ACS routines or specify the appropriate data class on the DEFINE CLUSTER command when you create the data set.

3. Create the partitioned or LOB table space with a DSSIZE of 8 GB or greater. The partitioning index for the partitioned table space takes on the EA-enabled attribute from its associated table space. See *DB2 SQL Reference* for more information about the correct syntax.
After a page set is created, you cannot use the ALTER TABLESPACE statement to change the DSSIZE. You must drop and re-create the table space.

Also, you cannot change the data sets of the page set to turn off the extended addressability or extended format attributes. If someone modifies the data class to turn off the extended addressability or extended format attributes, DB2 issues an error message the next time that it opens the page set.

Sliding secondary allocation size

DB2 Version 8 adds a new function called *sliding secondary space allocation*, which can be turned on automatically in DB2 V8 and is used for DB2 managed data sets (STOGROUP defined).

The enhancement helps to reduce the number of out-of-space conditions, eliminate need for PRIQTY and SECQTY specification on SQL CREATE, improve user productivity, and avoid the performance penalty associated with small extent sizes.

This patented solution benefits all users of DB2. It delivers autonomic selection of data set extent sizes with a goal of preventing out-of-extent errors before reaching maximum data set size. It particularly benefits users of ERP/CRM vendor applications, which have many small data sets that can grow rapidly.

The V8 new DSNZPARM parameter is MGEXTSZ (with a global scope), option 7 on install panel DSNTIP7. It determines what happens if a DB2 LDS is out of space and DB2 needs to create a new data set (piece). The values are YES and NO. The default value for MGEXTSZ is NO. Sliding secondary is always on, whether you specify YES or NO. The options have the following effects:

- NO
When specifying a value for PRIQTY and SECQTY, the new data set (created after A001) will have the same allocation as A001 for primary and secondary and therefore take on the characteristics of A001's PRIQTY and SECQTY. If PRIQTY and SECQTY were not specified then the allocation works as documented in YES.
- YES
When allocating a new data set for a new piece (after A001), the primary and secondary allocation will be the last size calculated by the sliding secondary of the previous piece. The maximum allocation will not exceed 127 cylinders for objects 16 GB or below, and 559 cylinders for objects 32 GB or 64 GB.
For example, A001 reaches the 2 GB limit and DB2 now needs to create the A002 data set. If A001's last allocation based on sliding scale was 79 cylinders, A002 will be created with a primary and secondary allocation of 79 cylinders.
The maximum allocation will not exceed 127 cylinders for objects 16 GB or below, and 559 cylinders for objects 32 or 64 GB.

DB2 Limits

| | |
|--|---|
| Maximum size for a segmented tablespace (or simple tablespace, but depreciated with Version 9) | 64 gigabytes (for Version 7, Version 8 and Version 9) with a maximum number of ... <ul style="list-style-type: none"> • 32 x 2 gigabytes data sets for a nonpartitioned table space • or 254 data sets for LOB table spaces. |
| Maximum number of partitions in a partitioned table space or partitioned index | <p><u>Version 8:</u></p> <ul style="list-style-type: none"> • 64 for table spaces that are not defined with LARGE or a DSSIZE greater than 2 GB • 254 for table spaces that are defined with LARGE or a DSSIZE greater than 2 GB <p><u>Version 9:</u></p> <ul style="list-style-type: none"> • 64 for table spaces that are not defined with LARGE or a DSSIZE greater than 2 GB. • 4096, depending on what is specified for DSSIZE or LARGE and the page size. |
| Maximum size of a partition (table space or index) | <p><u>Version 8:</u></p> <p>For table spaces that are not defined with LARGE or a DSSIZE greater than 2 GB:</p> <ul style="list-style-type: none"> • 4 gigabytes for 1 to 16 partitions • 2 gigabytes for 17 to 32 partitions • 1 gigabyte for 33 to 64 partitions <p>For table spaces that are defined with LARGE:</p> <ul style="list-style-type: none"> • 4 gigabytes for 1 to 254 partitions |

| | |
|---------------------------|--|
| | <p>For table spaces that are defined with a DSSIZE greater than 2 GB:</p> <ul style="list-style-type: none"> • 64 gigabytes for 1 to 254 partitions <p><u>Version 9:</u> For table spaces that are not defined with LARGE or a DSSIZE greater than 2 GB:</p> <ul style="list-style-type: none"> • 4 GB, for 1 to 16 partitions • 2 GB, for 17 to 32 partitions • 1 GB, for 33 to 64 partitions <p>For table spaces that are defined with LARGE:</p> <ul style="list-style-type: none"> • 4 GB, for 1 to 4096 partitions <p>For table spaces that are defined with a DSSIZE greater than 2 GB:</p> <ul style="list-style-type: none"> • 64 GB, depending on the page size: <ul style="list-style-type: none"> ▪ for 1 to 256 partitions for 4 KB pages ▪ for 1 to 512 partitions for 8 KB pages ▪ for 1 to 1024 partitions for 16 KB pages ▪ and 1 to 2048 partitions for 32 KB pages |
| Maximum LOB data set size | 64 gigabytes 128 TB (for Version 9) |

3390 Disk Capacity

| Model | Cylinders | Tracks | Bytes/volume | Bytes/track |
|---------|-----------|---------|--------------|-------------|
| 3390-1 | 1.113 | 16.695 | 946 GB | 56664 |
| 3390-2 | 2.226 | 33.390 | 1.89 GB | 56664 |
| 3390-3 | 3.339 | 50.085 | 2.83 GB | 56664 |
| 3390-9 | 10.017 | 150.255 | 8.5 GB | 56664 |
| 3390-27 | 32.760 | 491.400 | 27.84 GB | 56664 |
| 3390-54 | 65.520 | 982.800 | 55.68 GB | 56664 |

DB2 Cluster failures

IEC161I

IEC161I rc[(sfi)] - ccc, iii, sss, ddn, ddd, ser, xxx, dsn, cat
where:

rc Is a return code
sfi Is subfunction information (sfi only appears with certain return codes)
ccc Is a function code
iii Is a job name
sss Is a step name
ddn Is a ddname
ddd Is a device number (if the error is related to a specific device)
ser Is a volume serial number (if the error is related to a specific volume)
xxx Is a VSAM cluster name
dsn Is a data set name
cat Is a catalog name.

For information about these codes, see OS/390 MVS System Messages Volume 1.

DSNB204I - OPEN OF DATA SET FAILED. DSNAME = dsn

System action:

- The table space is automatically stopped.
- Programs receive an -904 SQLCODE (SQLSTATE '57011').
- If the problem occurs during restart, the table space is marked for deferred restart, and restart continues. The changes are applied later when the table space is started.

System programmer action: None.

Operator action:

1. Check reason codes and correct.

2. Ensure that drives are available for allocation.
3. Enter the command START DATABASE.

DSNP007I

```
DSNP007I csect - EXTEND FAILED FOR data-set-name. RC=rrrrrrrrr
      CONNECTION-ID=xxxxxxxx, CORRELATION-ID=yyyyyyyyyyyyyy,
      LUW-ID=logical-unit-of-work-id =token
```

Explanation: Extension failed in CSECT 'DSNPmmmm' of the data space manager (DSM). The 'data-set-name' identifies the data set that is being extended. RC is the reason code, which indicates why the extend process failed. (See "Section 4. DB2 Codes" in item SECTION_4_DB2_CO.) The 'connection ID', and 'logical-unit-of-work ID' (LUW-ID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the 'LUW-ID' is not an '*', the value is a 'logical-unit-of-work-id' that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of 'logical-unit-of-work-id' in any DB2 command that accepts 'logical-unit-of-work-id' as input.

Messages that have the same 'connection ID', 'correlation ID', and 'LUW-ID' relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

System Action: The DSM terminates its actions associated with this extend request. Higher level functions might terminate if they cannot continue without a successful extend. In this case, additional console messages and/or SQL codes are produced. In other cases, higher level functions might be able to resolve the space need without the extend, so they continue normally.

System Programmer Response: The following sources of information are available to help determine the cause of the failure:

- See the Programmer Response for message DSNP001I.
- Analyze the RC value in conjunction with the console messages.
- The reason code might apply to the data set identified in the message or to another data set that is needed for the extend process. If the reason code indicates a data set access failure, the console messages indicate the data set name. For example, if the RC indicates 'resource unavailable', console messages identify the resource that is unavailable. Most likely, the message will be DSNT500I.
- Retrieve diagnostic information from the output of the DISPLAY DB...RESTRICT or DISPLAY UTIL(*)... commands.
- If the extend failed on a result table space, refer to "Sort Data Sets" in Part 5 (Volume 2) of DB2 Administration Guide.
- If the extend failed on an SQL insert or update, refer to Part 2 of DB2 Diagnosis Guide and Reference for information on identifying and reporting the problem.
- If you need to enlarge your data set, refer to "Out of DASD Space" in Part 4 (Volume 1) of DB2 Administration Guide.
- To find out how the extend function operates, refer to "Improving DASD Utilization" in Part 5 (Volume 2) of DB2 Administration Guide.

Operator Response: See Operator Response for message DSNP001I for this information.

System Programmer Response: See System Programmer Response for message DSNP001I for this information.

DSNP001I

```
DSNP001I csect - data-set-name IS WITHIN nk BYTES OF AVAILABLE SPACE.
      RC=rrrrrrrrr CONNECTION-ID=xxxxxxxx, CORRELATION-ID=yyyyyyyyyyyyyy,
      LUW-ID=logical-unit-of-work-id=token
```

Explanation: This message warns of an impending space shortage. The data space manager (DSM) extend function CSECT 'DSNPmmmm' detected the space shortage through the use of a 'hedge' quantity 'n'. Reason code 'rrrrrrr' gives some information about why space could not be obtained. (See

"Section 4. DB2 Codes" in item SECTION_4_DB2_CO.)

There might also be some preceding DSM and VSAM messages that can help to determine why space cannot be obtained. The connection, correlation, and logical-unit-of-work IDs identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

System Action: If sufficient space is available for the current request, continue. Otherwise, message DSNP007I follows, and the current SQL request fails.

Operator Response: DSM messages are directed to the database administrator (DBA) or system administrator. However, the DBA or system administrator can request that the operator issue an MVS WRITELOG operator command (for example, 'W A') to obtain MVS SYSLOG output. The SYSLOG sometimes contains media manager messages that might be helpful to the DBA or system administrator in determining why space could not be allocated.

System Programmer Response: Based on the reason code (RC) in the message, as well as information that might be available in access method services or VSAM messages preceding this one, try one of the following:

- If you do not anticipate further extension of this table space or index, ignore the message.
- If the table space or index is backed by storage group-defined data sets and this data set is not at maximum (123) extents, use the ALTER STOGROUP statement to add volumes to the storage group (up to 133 of them).
- If the table space or index is backed by 'user-defined' data sets, and not at maximum (123) extents for the data set, issue an ALTER ADDVOLUMES access method services command to provide additional volumes (up to 133 of them). Or, if the table space is nonpartitioned and the high allocated RBA is greater than or equal to 2GB (2,147,483,648) for the indicated data set, define the data set required for the next sequential piece. (Actually, in this case, there should be an indication in console messages or the reason code that the data set required for the next piece was not found.)
- If the data set is at maximum (123) extents, try reorganizing the table space or index into a data set or data sets having larger primary and secondary values. This process is discussed in Part 2 (Volume 1) of DB2 Administration Guide.
- **Problem Determination:** Examine the console messages, SYS1.LOGREC information, and SYSLOG to determine the reason for the condition. Try to resolve it by the methods described in the Operator Response section of this message. If you feel that this message indicates a problem in an IBM program, you should provide the following sources of information about the problem:
 - Console sheet showing the message and related messages that might precede or follow it.
 - System log printout (obtained via MVS WRITELOG operator command) showing related VSAM/media manager messages that might appear there (usually needed only if you have a message or reason code that indicates a media manager services problem occurred).
 - Integrated catalog facility (ICF) catalog listing showing characteristics of the data set in question and of the ICF catalog itself.
 - VTOC listing of packs associated with the request, showing space available for data sets, catalogs, VTOC index, and the VTOC itself, as well as data set extent information.
 - Any associated dumps.
 - SYS1.LOGREC listing showing entries at or near the time of the problem (unless these appear in an associated dump).
- Depending on the reason code, the results of certain SQL SELECT operations on DB2 catalog tables may be needed. (See "Section 4. DB2 Codes" in item SECTION_4_DB2_CO.)