z/OS® Operating System

SAM extended format and compression

Extended format

What are extended format data sets?

- Type of <u>SMS-managed</u> physical sequential data set.

- Same characteristics as a physical sequential data set.

- Internally, the data is stored in a different format.

- <u>Sequential data striping</u> is implemented through the use of extended format data sets.

SAM extended format and compression © 2008 IBM Corporation

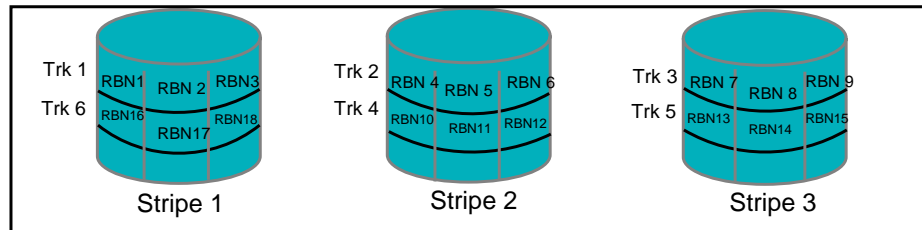Extended format data set can support data striping and compression.

It is a type of SMS-managed physical sequential data set.

Externally, extended format data sets have the same characteristics as sequential data sets. However, records are not necessarily stored in the same format or order as they appear.

Sequential data striping is implemented through the use of extended format data sets. Data is interleaved across multiple volumes for striped data sets.

Extended format
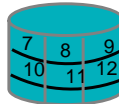
Multi-Striped (Multi-volume)   One logical volume

Trk 1 / Trk 6 — RBN1 RBN 2 RBN3 / RBN16 RBN17 RBN18 — Stripe 1
Trk 2 / Trk 4 — RBN 4 RBN 5 RBN 6 / RBN10 RBN11 RBN12 — Stripe 2
Trk 3 / Trk 5 — RBN 7 RBN 8 RBN 9 / RBN13 RBN14 RBN15 — Stripe 3

Single Stripe, Multi-Volume
Data Class
DSNTYPE=Ext

1 RBN 2 3 / 4 5 6

Storage Class
Sustained Data Rate=nnn

7 8 9 / 10 11 12

*/*where nnn is used to derive the number of stripes*/*

-Data striping distributes data for one data set across multiple volumes.

-A data set can have a maximum of 59 stripes.

-Each stripe must reside on one volume and cannot be extended to another volume.

-I/O can be done in parallel with striped data for better performance.

-To specify extended format, data set type (DSNTYPE) in data class needs to be set to 'Extended', and sustained data rate in storage class needs to be set to the number of stripes needed.

-Sustained data rate can be set to a maximum of 59 stripes. The default is 1, which will give a single striped data set.

Example of blocks in the extended data format

RBN 1

User data		Pad	Suffix

RBN 2

User data		Suffix

RBN 3

User data	Pad	Suffix
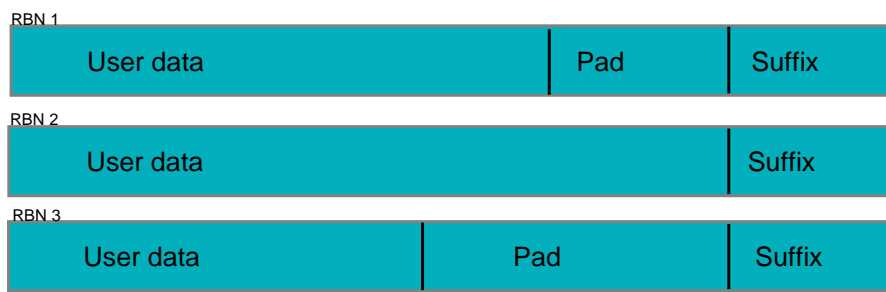
- User data
  - BSAM or QSAM Block
- Padding
  - Format 1 DSCB (DS1BLKL) contains user-perceived maximum block size
  - Above, RBN1 and RBN3 are short user blocks
- Suffix (length = 32 bytes) contains:
  - Length of user data
  - Relative block number
  - 3 bytes to detect control unit padding

SAM extended format and compression

Block in the extended data format:

-User data consists of a BSAM or QSAM block.

-Block can be short block.

-RBN1 and RBN3 are short user blocks and require padding up to the beginning of the suffix.

-Suffix (length of 32 bytes) contains length of user data, relative block number, and 3 bytes to detect control unit padding.

Characteristics of extended format data sets

- Maximum of 123 extents on each volume.

- Maximum of 59 volumes.

- Can occupy any number of tracks.

- Striped data set can contain up to 4GB blocks.

- All physical blocks are the same size.

- Each block has a 32-byte suffix

　　SAM extended format and compression　　© 2008 IBM Corporation

-Extended format sequential data sets have a maximum of 123 extents on each volume.

-Each extended format sequential data set can have a maximum of 59 volumes.

-An extended format data set can occupy any number of tracks.

-An extended format, striped sequential data set can contain up to 4GB blocks.

-All physical blocks in an extended format sequential data set are the same size, but when a program reads a block, the access method returns the length written by the writing program.

-Each block in an extended format data set has 32-byte suffix.

Compressed format

What are compressed format data sets?

- Type of <u>extended format</u> dataset which supports block level compression.

- Same characteristics as extended format sequential data sets.

- Internally, the data within each block is stored in a different format.

- Externally, a compressed format data set:
  - May consist of one or more stripes
  - May be multi-volume (single-stripe only)
  - May not be opened for update

SAM extended format and compression © 2008 IBM Corporation

-Compressed format data sets is a type of extended format data set which supports block level compression.

-An extended format data set can be allocated in the compressed format by specifying COMPACTION=YES in the data class.

-Internally, data within each block is stored in a different format.

-Externally, a compressed format data set may consist of one or more stripes, may be multi-volume, and may not be open for update.

Compressed format (continued)

Types of compression

- **Generic (DBB-based)**

  The system selects a set of Dictionary Building Blocks (DBB) which best reflects the initial data written to the data set. The system later can reconstruct the dictionary by using the information in the dictionary token stored in the catalog.

- **Tailored**

  The system attempts to derive a compression dictionary tailored specifically to the initial data written to the data set. Once derived, the compression dictionary is stored in system blocks which are imbedded within the data set itself.

SAM extended format and compression   © 2008 IBM Corporation

• Two compression techniques are available for compressed format data sets.

• They are DBB-based compression and tailored compression.

• Generic (DBB-based) compression: The system selects a set of dictionary building blocks (DBBs), which best reflects the initial data written to the data set. The system can later reconstruct the dictionary by using the information in the dictionary token stored in the catalog.

• Tailored compression: The system attempts to derive a compression dictionary tailored specifically to the initial data written to the data set. Once derived, the compression dictionary is stored in system blocks which are imbedded within the data set itself.

• The form of compression the system is to use for newly created compressed format data sets can be specified at either or both the data set level and at the installation level.

Compressed format (continued)

User Blocks

System Block - in CI format

| CPFX$_1$ | a |
| CPFX$_2$ | b |

Compress

A

B

C

.

.

.

Data Class

DSNTYPE=Ext

Compaction=Y|G|T

*/*Yes|Generic|Tailored*/*

CPFX$_3$ c$_F$ RDF$_3$

RDF$_2$ RDF$_1$ CIDF

CPFX$_3$ c$_L$

RDF$_3$ CIDF
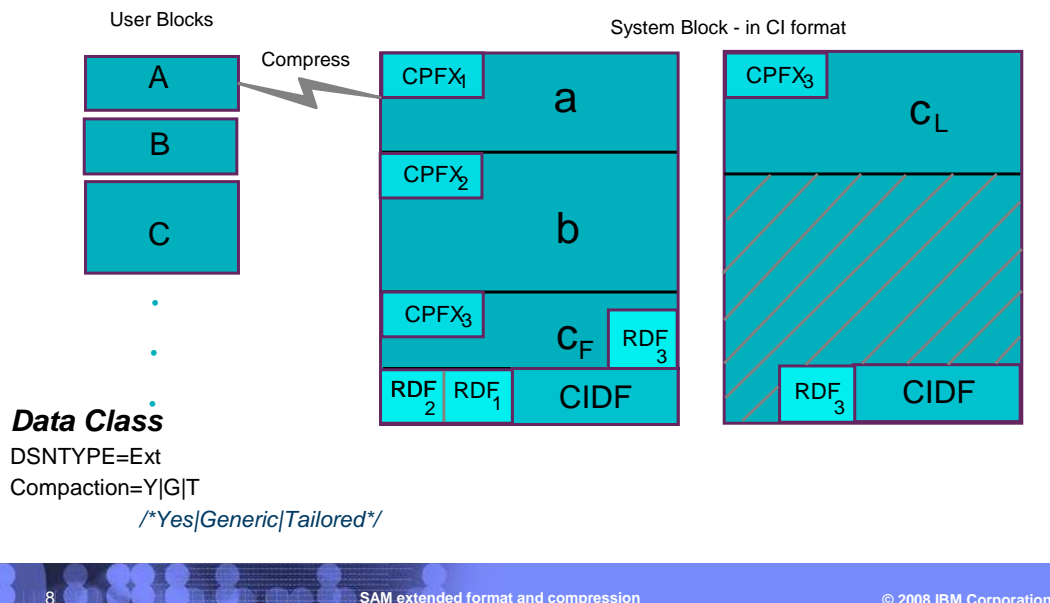
-Each block in the CI has an associated compression record prefix (CPFX) and a record definition field (RDF).

-A compression record prefix (CPFX) contains information about the compressed block, such as whether the record is compressed or not, uncompressed length of record, and some flags.

-A record definition field (RDF) contains information about the compressed record, such as whether it is a spanned record, which segment the record is spanned, and record length.

-A CI has an associated control interval definition field (CIDF).

-A control interval definition field (CIDF) contains information about the CI such as the offset to the free space and length of the free space.

Characteristics of compressed format data sets

- Might or might not contain compressed records.

- Data format consists of physical blocks whose length has no correlation to the logical block size of the data set.

- Compressed format data set cannot be opened for update.

- System attempts to derive a compression token when enough data is written to the data set (between 8K and 64K for generic compression and much more for tailored compression).

•A compressed format data set might or might not contain compressed records.

-The data format for a compressed format data set consists of physical blocks whose length has no correlation to the logical block size of the data set in the DCB, DCBE, and the data set label. The actual physical block size is calculated by the system and is never returned to the user. However, the system maintains the user's block boundaries when the data set is created so that the system can return the original user blocks to the user when the data set is read.

-A compressed format data set cannot be opened for update.

-When the data set is created, the system attempts to derive a compression token when enough data is written to the data set (between 8K and 64K for generic compression and much more for tailored compression). If the system is successful in deriving a compression token, the access method attempts to compress any additional records written to the data set. However, if an efficient compression token could not be derived, the data set is marked as non-compressible and there is no attempt to compress any records written to the data set.

References

- Using Data Sets, SC26-7410

- Macro Instructions for Data Sets, SC26-7408

- SAM Logic, ZW84-4030-03 (IBM confidential)

V1R0-SAM-Extended-Format-100808.ppt

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_V1R0-SAM-Extended-Format-100808.ppt

This module is also available in PDF format at: ../V1R0-SAM-Extended-Format-100808.pdf

11   SAM extended format and compression   © 2008 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.

12   SAM extended format and compression   © 2008 IBM Corporation