IMS Warning for z/OS V1 R3 Change in KSDS CI Size Calculation

This is an update to the flash published on July 10, 2002. The location of the INDXCISZ tool mentioned in the last section has changed. This is the only change in this update.

z/OS Version 1 Release 3 DFSMS changes the calculation of KSDS index component minimum CI sizes. This may change the size of CIs for IMS database data sets and RECONs when they are redefined. If installations do not change buffer pool sizes when CI sizes are increased, database opens may fail or database performance may deteriorate. If preventive actions are not taken when RECON CI sizes increase, DBRC performance could suffer. This article explains the change and what an installation should do to prepare for it. The preparation should be done before any of these data sets are redefined under z/OS V1 R3.

The Change

In previous releases, the algorithm which calculates minimum index CI sizes was based on an assumption that key compression would reduce the size of keys in the index component to approximately 5 bytes. z/OS V1 R3 has changed the algorithm. The new algorithm is based on an assumption that key compression will reduce the size of keys to approximately one third of their full size. This means that large key sizes will tend to generate the biggest changes. A specification of a CI size in the AMS DEFINE will be overridden if it is smaller than the calculated size. The minimum size is also the default size. These changes are explained in information APAR II13288.

Why Was This Change Made?

The change in the index component CI size calculation was done to better allocate these data sets. Each entry in the index CI contains the key associated with the data CI to which it points. The key is compressed in the entry. The old algorithm's assumption that keys would be compressed to approximately 5 bytes was often wrong for large keys. The new algorithm's assumption that keys will be compressed to approximately one third of their length will tend to be more accurate. This should avoid situations where the index component CI is severely undersized. When an index CI is not large enough to hold an entry for each data component CI in a CA, some CIs in the CA cannot be used. This may cause a data set to become unnecessarily large with many unused CIs within its CAs. In fact, this could be a reason that an HIDAM index, secondary index, or (S)HISAM database approaches the 4 gigabyte data set size limitation.

Are Changes Required When Migrating to z/OS V1 R3?

Changes are only required if the index component CI size is increased after migrating to z/OS V1 R3. For some combinations of data component CI size and key size, the minimum index component CI size has not changed. No changes are required for any of these data sets. If you have specified the control interval size for an index component and the size is at least as large as the minimum under z/OS V1 R3, you will not need to make any changes for the data set. For other data sets, you will probably need to accommodate the larger index component CI sizes.

Note that the changes in CI sizes are a result of defining data sets under z/OS V1 R3. Existing data sets are not affected until they are redefined. The redefinition is usually a result of executing an IDCAMS DEFINE. It also could occur as a result of a recall of a migrated data set.

Installations may choose to define larger index component CI sizes for affected KSDSs while they are still running under a previous release. This would allow them to make the changes is a more orderly fashion.

IMS Database Data Sets

IMS HIDAM indexes, secondary indexes, (S)HISAM databases, HALDB PHIDAM indexes, and HALDB ILDSs could be affected by this change. When these data sets are defined or redefined under z/OS V1 R3 they may have larger index component CI sizes.   The CIs may no longer use the same database buffer subpools. If they become larger than the largest buffer size defined for the subpool they use, they cannot be opened. Before defining these data sets under z/OS V1 R3 installations should review their database buffer pools. Different numbers of buffers may be required for some subpools and new subpools may need to be defined. IMS VSAM buffer sizes may be 512, 1K, 2K, or multiples of 4K up to 32K bytes. They are defined in the DFSVSAMP data set for batch jobs and utilities. They are defined in the DFSVSMxx member for online systems. Since IMS database data sets are typically redefined when they are reorganized, these data sets will be affected when they are reorganized.

The following are some example data component CI sizes and key sizes for KSDSs. The minimum index component CI sizes are shown for z/OS V1 R3 and previous releases:

Data Component CI Size	Key Size	Index Component Minimum CI Size Previous Releases	Index Component Minimum CI Size z/OS V1 R3
1024	10	4096	4608
1024	20	4096	5632
1024	30	4096	7168
1024	40	4096	10240
2048	10	3072	3072
2048	20	3072	3584
2048	30	3072	4608
2048	40	3072	6144
4096	10	1536	1536
4096	20	1536	2048
4096	30	1536	2560
4096	40	1536	3584
8192	10	1024	1024
8192	20	1024	1024
8192	30	1024	1536
8192	40	1024	2048
12288	10	1024	1024
12288	20	1024	1024
12288	30	1024	1024
12288	40	1024	1536
16384	10	512	512

16384	20	512	512
16384	30	512	1024
16384	40	512	1024

DBRC RECONs

DBRC RECONs could be affected by this change. When these data sets are defined or redefined under z/OS V1 R3 they may have larger index component CI sizes. Users do not specify the size of RECON buffers. Instead, they may use DSPBUFFS to specify the number of data and index buffers. The size of the buffers is determined by the actual size of the CIs. DBRC performance can be seriously degraded if a RECON with a larger CI size is created while an IMS subsystem is executing. If a new RECON is created with a larger CI size while an IMS subsystem is executing, the new RECON could use NSR, not LSR, buffers. This prevents look-aside buffering. Alternatively, the new RECON's index component might use the buffers created for data components. This could adversely affect performance. For these reasons, it is recommended that all RECONs in a set (RECON1, RECON2, and RECON3) have the same CI sizes. Before migrating to z/OS V1 R3, an installation may REPRO these data sets to new data sets with the index component CI size specified to be at least as large as the minimum for z/OS V1 R3. This will prevent a mismatch when a RECON is redefined under z/OS V1 R3.

The following are some typical data component CI sizes used for RECONs and the minimum CI sizes for the index components with z/OS V1 R3 and previous releases. RECONs have a key size of 32 bytes.

Data Component	Index Component	Index Component
CI Size	Minimum CI Size	Minimum CI Size
	Previous Releases	z/OS V1 R3
8192	1024	1536
12288	1024	1024
16384	512	1024
20480	512	1024
24576	512	512

Analyzing Your Existing Data Sets

IBM is supplying a tool that will help with the migration to z/OS V1 R3. This tool may be run to determine the change (if any) to index component CI sizes. The tool reads existing data sets and reports their current index component CI size and the minimum size if the data set is redefined under z/OS V1 R3. The tool is available from the software server (ftp.software.ibm.com) in the s390/mvs/tools directory as INDXCISZ.JCL.TRSD. You will need to download the file in binary format and unterse it on your OS/390 or z/OS system using TRSMAIN.

To use FTP from an MVS system, first allocate a dataset with LRECL=1024 and RECFM=FB. This data set name will be used in the ftp get command.

Enter the TSO 'ftp' command.
At the 'Connect to ?' prompt, enter: ftp.software.ibm.com
At the 'NAME ...' prompt, enter: anonymous
At the 'PASSWORD' prompt, enter you email address.

At the ‘Command:’ prompt, enter: cd s390/mvs/tools
At the ‘Command:’ prompt, enter: binary
At the ‘Command:’ prompt, enter: get INDXCISZ.JCL.TRSD ‘your data set name’

Once the file is placed on your MVS system, it must be untersed using TRSMAIN. Information on using TRSMAIN may be found at http://techsupport.services.ibm.com/390/trsmain.html.

The file contains instructions for link editing and executing the tool. When executing the tool, the SYSIN file must include a control statement to indicate which files are processed by the tool. The following statements would process all files whose names begin with “LEWIS.VSAM.”

```
//S1  EXEC  PGM=INDXCISZ
//STEPLIB DD DSN=your.library.name,DISP=SHR
//SYSOUT  DD SYSOUT=*
//SYSIN   DD   *
LEWISR.VSAM.**
//
```

A sample listing of the output of this tool follows. The heading shows the data set filter used to select the data sets for analysis. The left column shows the data sets whose index CI sizes will be increased when defined under z/OS V1 R3. The two right hand columns show the current index CI size and the minimum (default) size that will be created under z/OS V1 R3. In this sample, 29 KSDSs had data set names beginning with LEWISR.VSAM. Of these, 18 are affected by the change and are listed.

```
                             LISTING FOR FILTER KEY: LEWISR.VSAM.**
 LEWISR.VSAM.L1024.K10                              4096  4608
 LEWISR.VSAM.L1024.K20                              4096  5632
 LEWISR.VSAM.L1024.K30                              4096  7168
 LEWISR.VSAM.L1024.K40                              4096 10240
 LEWISR.VSAM.L12288.K40                             1024  1536
 LEWISR.VSAM.L16384.K30                              512  1024
 LEWISR.VSAM.L16384.K32                              512  1024
 LEWISR.VSAM.L16384.K40                              512  1024
 LEWISR.VSAM.L2048.K20                              3072  3584
 LEWISR.VSAM.L2048.K30                              3072  4608
 LEWISR.VSAM.L2048.K40                              3072  6144
 LEWISR.VSAM.L20480.K32                              512  1024
 LEWISR.VSAM.L4096.K20                              1536  2048
 LEWISR.VSAM.L4096.K30                              1536  2560
 LEWISR.VSAM.L4096.K40                              1536  3584
 LEWISR.VSAM.L8192.K30                              1024  1536
 LEWISR.VSAM.L8192.K32                              1024  1536
 LEWISR.VSAM.L8192.K40                              1024  2048

                             LISTING FOR FILTER KEY: LEWISR.VSAM.**
         THE NUMBER OF ENTRIES PROCESSED WAS:
                   TOTAL CLUSTERS:               29
                   TOTAL AIX:                     0
                   KSDS PROCESSED:               29
                   INDEX CISIZE CHANGE:          18
                   SKIPPED DUE TO ERRORS:         0
                   SKIPPED (OFFLINE):             0
                   SKIPPED DUE TO FIELDS:         0
```