

Database

ON DEMAND BUSINESS


© 2008 IBM Corporation



Database Enhancements

- Change Accumulation and Prefix Resolution Sort enhancement
- ACBGEN Exploitation of Storage Above the 16M Line
- HALDB Index/ILDS Rebuild enhancement
- Image Copy 2 Fast Replication
- Fuzzy User Image Copy support

Database Utility Enhancements



Change Accum and Prefix Resolution Sort

- Default sort size changed from 200K to MAX
 - ◆ CORE= parameter in JCL for Change Accumulation and Database Prefix Resolution
 - CORE= specifies the storage used by sort
 - MAX allows the installation default value for sort to be used
 - CORE limits the number of sort work data sets used by OEM sort programs
 - UCBs for these data sets are below 16M line for some OEM sort programs
- Benefit
 - ◆ Allows more sort work data sets to be used with OEM sort programs which create UCBs below the 16MB line
 - This could shorten the elapsed execution time of the utilities

4

The default for the CORE= parameter on the Database Change Accumulation and Database Prefix Resolution utilities is changed in IMS V10 from 200K to MAX. When MAX is used, the IMS utility does not limit the amount of space used by sort. It is limited by the default value specified for the sort product. The sort space is particularly important for some non-IBM sort programs which create UCBs for sort work data sets below the 16M line. The sort space may limit the number of these data sets that may be used. By increasing the default space, the number of sort work data sets may be increased. This may shorten the elapsed execution time for these utilities.

A new message has been added to the Database Change Accumulation and Database Prefix Resolution utilities. It is issued when the CORE= parameter is not specified. The message is:

```
DFS3007I SORT CORE SIZE WILL DEFAULT TO CORE=MAX
```

If an installation wants to limit the sort space to less than the system default, it should code the CORE= parameter for the desired value.

ACBGEN Exploitation of Storage Above the 16M Line

- Previous releases were limited in the number of PCBs per PSB
 - ◆ Limitation was due to use of "below the line" storage
 - PSBs with more than approximately 500 PCBs could result in S80A abend


- IMS V10 ACBGEN allocates most of its working storage above the 16M line
 - ◆ Eliminates these out-of-storage abends

- Benefit
 - ◆ Allows up to 2500 PCBs per PSB

5

ACBGEN has been modified to allocate most of its working storage above the 16M line. In previous releases, this working storage was allocated below the line. Very large PSBs could require more storage than was available below the line. Requests for more storage than was available resulted in S80A abends with RC=10 in previous releases. Since V10 allocates storage above the line, it is not limited to the available storage below the line. This should eliminate these abends. Very large PSBs, including those with up to 2500 PCBs, are handled by ACBGEN in V10.

The limitations of 2500 PCBs per PSB and 30,000 SENSEGs per PSB are enforced by the macros used in PSBGENs.



HALDB ILDS Rebuild Enhancement

- New option for HALDB Index/ILDS Rebuild utility (DFSPREC0)
 - ◆ ILDS entries (ILEs) are written to ILDS sequentially in load mode
 - Avoids CI/CA splits during rebuild
 - Creates free space according to data set definition
 - ◆ ILDS entries are sorted in data spaces before they are written
 - ◆ Invoked with RECOVTYP=ILEF or BOTHF
 - Uses 4 data spaces of 2 GB each
 - ◆ Free space creation may result in larger ILDS
- Benefits
 - ◆ Faster execution of DFSPREC0 when rebuilding an ILDS
 - ◆ Free space may improve performance of subsequent reorganizations
 - Reduce CI/CA splits

6

The HALDB Index/ILDS Rebuild utility (DFSPREC0) is used to rebuild either a PHIDAM primary index, an ILDS, or both for a partition. This enhancement applies only to the rebuilding of the ILDS. When an ILDS is rebuilt, the utility reads the partition database data sets looking for segments which are targets of secondary indexes or logical relationships. An indirect list entry (ILE) is written to the ILDS for each target that is found. In previous releases, these writes are done when the target segments are found. This means that the writes are random. They must be done in update mode. The enhancement provides a new way of writing the ILEs. When using the enhancement, ILE entries are not immediately written to the ILDS. Instead, they are stored in data spaces, sorted in ILE key sequence, and then written. The writes are done in load mode. These writes are sequential. Load mode avoids CI/CA splits. These two characteristics can significantly reduce the elapsed time of the writes. Load mode also creates free space as defined in the IDCAM DEFINE for the ILDS.

The new option is invoked with new values for the RECOVTYP parameter. Previously, the valid parameter values were INDEX for rebuilding an PHIDAM index, ILE for rebuilding an ILDS, or BOTH for rebuilding both data sets. The new additional valid values are ILEF for rebuilding the ILDS sequentially or BOTHF for rebuilding both data sets and rebuilding the ILDS sequentially. The "F" in these values signifies that free space will be created.

Some installations limit the number of data spaces that address spaces may create. This is done with an z/OS IEFUSI exit. The default IEFUSI exit for z/OS allows the creation of the 4 data spaces used by DFSPREC0. If your installation has written its own exit, it may not allow 4 data spaces for most address spaces. In this case, you would need a modification to the exit to use the new option.

Since the ILDS that is rebuilt with the new option includes the defined free space, it may create a larger data set than was created in the past with this utility or by the HD Reload utility. The ILDS must be deleted and redefined (DELETE/DEFINE) before the execution of the utility. This is true with the old ILE and BOTH options and with the new ILEF and BOTHF options.

The use of the new option could significantly reduce the elapsed time required for rebuilding ILDSs. The creation of free space in the ILDS could improve the performance of subsequent HD Reload or Online Reorganization executions since the free space should reduce the need for CI/CA splits.



Image Copy 2 Fast Replication

Image Copy 2 Fast Replication

- Fast replication support
 - ◆ Exploits FlashCopy on Enterprise Storage Server (ESS) and DS8000
 - ◆ Exploits SnapShot on RAMAC Virtual Array (RVA)
- Prerequisites for fast replication
 - ◆ z/OS V1R8
 - ◆ Storage Subsystem that supports either
 - Data Set Level FlashCopy V2
 - ESS (Shark), DS8000, ...
 - or
 - SnapShot
 - RAMAC Virtual Array (RVA), ...

8

IMS V10 adds fast replication support. Fast replication is a DFSMSdss function that exploits the FlashCopy capability of the Enterprise Storage Server (ESS or Shark) or TotalStorage DS8000 or the SnapShot capability of the IMS RAMAC Virtual Array (RVA). FlashCopy V2 must be installed on the ESS system. Fast replication also can be used with OEM storage devices which support the DFSMSdss Fast Replication interface.

This support is provided in the Image Copy 2 (DFSUDMT0) utility and the Database Recovery (DFSURDB0) utility. They use data set level Fast Replication functions of DFSMSdss which are only provided by z/OS V1R8.








Image Copy 2 Fast Replication

- Fast replication
 - ◆ Uses DFSMSdss COPY command with FASTREP(REQ) parameter
 - ◆ Output is in same format as database data set
 - ◆ Only one image copy per database data set may be created
 - ◆ Output must be written to same storage system
 - ◆ Copy is done in one phase
 - Time is comparable to the logical copy time for concurrent copy
 - ◆ Issues DFS3141A message when clean image copy completes
 - May be used to indicate that database may be started on online systems
 - ◆ Requires DBRC database registration
 - ◆ Fuzzy image copies of KSDSs
 - Require BWO(TYPEIMS)
 - Require SMS management

9

Fast replication is invoked with the DFSMSdss COPY command with the FASTREP(REQ) parameter. The output or “image copy” is an exact copy of the database data set. It must be written to a volume in the same storage system. Only one output data set may be created for each input data set. Unlike the concurrent copy process, the fast replication copy process is not split into a logical copy phase and a physical copy phase. If you are creating a clean image copy, the database is unavailable for the duration of the copy process; however, the fast replication process is comparable in speed to the logical copy phase of the concurrent copy process.

Message DFS3141A is issued when a clean image copy completes. The DFS3141A message may be used to indicate that the database may be started on online systems. The DFS3141A lists the database, partition, area, or group that was copied. It is followed by DFS3141I messages for each data set that was copied.

Databases must be registered with DBRC. The IDCAMS DEFINE must specify BWO(TYPEIMS) for KSDSs if fuzzy image copies are taken. Fuzzy image copies also require that the database data set be managed by SMS. If a split occurs during the logical copy phase for concurrent copy or the initialization of copy for fast replication, the copy is retried up to nine more times if only one data set is being copied. If multiple data sets are being copied, retries are not attempted. A failure causes DFSMSdss to issue message ADR944E and IMS to issue message DFS3145A. The form of the ADR944E message when a fast replication copy fails due to a CI/CA split during the initialization is:

```
ADR944E (ttt)-mmmmm(yy), DATA SET dsname IN CATALOG catalog_name ON VOLUME
volume_serial_number CANNOT BE PROCESSED BECAUSE IT WAS UPDATED DURING THE FAST
REPLICATION INITIALIZATION
```

The DFS3145A message is:

```
DFS3145A ATTEMPT TO COPY KSDS CONCURRENT WITH UPDATE ACCESS FAILED
```

Image Copy 2 Fast Replication Invocation

- Image Copy 2 DBDS Control Statement
 - ◆ Used to specify a database data set to be copied
 - ◆ Position 62
 - 'F' invokes fast replication
 - ◆ Position 63
 - Blank indicates that DD statement is provided for output data set
 - Output DD name is in positions 22-29
 - DD statement for output must be in JCL
 - 'H' indicates that output data set is dynamically allocated

10

Fast replication copies are invoked by placing an 'F' in position 62 of the DBDS control statement.

If a blank is in position 63 a DD statement must be supplied for the image copy output data set. The data set may either be created prior to the Image Copy 2 step or may be created by the DD statement.

If an 'H' is in position 63 the image copy output data set is created dynamically by DFSMSdss. The attributes of the image copy data set are derived from the database data set. An HLQ control statement is required when an 'H' appears in position 63 of the DBDS control statement. It is shown on the following page.

Other positions on the DBDS control statement have the same meaning that they have without the fast replication option. For example, an 'X' in position 58 invokes a clean image copy and an 'S' in invokes a fuzzy image copy.

Fast replication produces only one image copy data set for a database data set. You cannot specify multiple output data sets. This means that positions 30 through 57 must be blank.

Unlike concurrent copies, fast replication does not have logical copy and physical copy phases. The fast replication process is comparable in speed to the logical copy phase of the concurrent copy process. Since there are not separate logical and physical copy phases, position 59 should be left blank. Position 59 is used for concurrent copies to indicate whether the database becomes available for updates after the logical phase completes or after the physical phase completes.

Position 60 should be blank for fast replication. It is used for the compression option with concurrent copies.

Image Copy 2 Fast Replication Invocation

- Image Copy 2 HLQ Control Statement
 - ◆ Used to specify the data set name of the image copy data set
 - ◆ Position 1
 - 'H' indicates this is an HLQ control statement
 - ◆ Position 3
 - 'Y' time-stamp trailer is appended to output data set name
 - 'N' time-stamp trailer is not appended to output data set name
 - ◆ Positions 4-29
 - High-level qualifier for output data set name
 - ◆ Positions 31-38
 - SMS storage class - optional
 - ◆ Positions 40-47
 - SMS management class - optional

11

An 'H' in position 1 indicates that this is an HLQ (high level qualifier) control statement.

Position 3 indicates whether or not the time-stamp trailer is appended to the output data set name.

Positions 4-29 contain the high-level qualifier used in the output data set name.

Positions 31-38 may be used to specify the SMS storage class that is used by ACS routines.

Positions 40-47 may be used to specify the SMS management class that is used by ACS routines.

IC2 Fast Replication Output Data Set Name


- Output data set name when dynamic allocation is used
 - ◆ If time-stamp is not chosen in HLQ control statement
 - hlq.dbdname.ddname
 - Example for hlq=MY.HLQ, database=DBNAME1, and DD=DDNAME1
MY.HLQ.DBNAME1.DDNAME1
 - ◆ If time-stamp is chosen in HLQ control statement
 - hlq.dbdname.ddname.Dyyddd.Thmmss
 - Example for noon on January 1, 2007
MY.HLQ.DBNAME1.DDNAME1.D07001.T120000

12

This shows two examples of output data set names that are used when dynamic allocation is used for these data sets.

The first example is for a data set when the time-stamp trailer is not selected (position 3 on the HLQ control statement contains an 'N'). The high-level qualifier is MY.HLQ.

The second example is for a data set when the time-stamp trailer is selected (position 3 on the HLQ control statement contains a 'Y'). The high-level qualifier, database name, and DD name are the same as in the first example.



DBRC with Fast Replication

- New GENJCL.IC parameters
 - ◆ SMSONLC - generate IC2 JCL for fuzzy IC with fast replication
 - ◆ SMSOFFLC - generate IC2 JCL for clean IC with fast replication
 - ◆ HLQ(hlq) - specifies high-level qualifier for output data set name
 - ◆ DSNSUF|NODSNSUF - date/time stamp as suffix in output data set name
 - ◆ STORCLAS(name) - specifies Storage Class use by ACS routines for allocating image copy data set
 - ◆ MGMTCLAS(name) - specifies Management Class use by ACS routines for allocating image copy data set

13

SMSONLC and SMSOFFLC are new image copy types recorded in image copy records in the RECONS. Other previously existing image copy types were SMSCIC (IC2 fuzzy IC using concurrent copy), SMSNOCIC (IC2 clean IC using concurrent copy), CIC (concurrent image copy using Image Copy utility), ONLINE (fuzzy image copy using the Online Image Copy utility), and BATCH (clean image copy using the Image Copy utility. High Performance Image Copy (HPIC) V4 also supports fast replication copies, but does not invoke IC2. APAR PK33115 adds the use of SMSONLC and SMSOFFLC by HPIC V4 when it uses fast replication. The HPIC V4 Recovery fund and Create Image Copy function support these IC types as input.




The GENJCL.IC command has new parameters.

The SMSONLC and SMSOFFLC parameters are used to generate fast replication Image Copy 2 utility JCL for either a fuzzy or clean image copy. SMSONLC and SMSOFFLC are also new image copy types recorded in image copy records in the RECONS. Other previously existing image copy types were SMSCIC (IC2 fuzzy IC using concurrent copy), SMSNOCIC (IC2 clean IC using concurrent copy), CIC (concurrent image copy using Image Copy utility), ONLINE (fuzzy image copy using the Online Image Copy utility), and BATCH (clean image copy using the Image Copy utility. High Performance Image Copy (HPIC) V4 supports fast replication copies. APAR PK33115 adds the use of SMSONLC and SMSOFFLC by HPIC V4 when it uses fast replication. The HPIC V4 Recovery fund and Create Image Copy function support these IC types as input.

HLQ(hlq) is used to specify the high-level qualifier for the output data set name when the output data set is created dynamically. 'hlq' may be 1 to 26 characters. The value is placed in positions 31-56 of the DBDS control statement.

DSNSUF specifies that the date/time stamp suffix is to be added to the output data set name for dynamically created output data sets. NODSNSUF specifies that the date/time stamp suffix is not added. NODSNSUF is the default.

STORCLAS(name) and MGMTCLAS(name) are used to specify the storage class name and management class name to be used by the Access Control System (ACS) routines for allocating output data sets which are created dynamically.

Fast Replication with Database Recovery

- Database Recovery Utility reads Fast Replication image copies
 - ◆ No user specification required
 - Utility uses DBRC information to determine that IC is fast replication IC
 - ◆ Uses COPY with FASTREPLICATION(PREFERRED)
 - If Fast Replication copy is on the same storage system with target data set and storage system supports fast replication, fast replication is used to restore the database data set from the image copy
 - Else, traditional copy is used
 - ◆ DBDS DD statement is optional when using Fast Replication IC input
 - 'D' in position 64 of control statement indicates DD statement not used
 - DBRC information is used to allocate the data set
 - ◆ GENJCL.RECOV has added support for Fast Replication
 - DBDSDD|NODBSDD parameters specify if DD statement is used

14

The IMS Database Recovery utility (DFSURDB0) has been updated to read image copies created by using fast replication. In previous releases Database Recovery could read two kinds of image copies. Those in the original Image Copy format have header records followed by copies of the data. Those produced with concurrent copy were in DFSMSdss DUMP format. Fast replication image copies are exact copies of the database data sets, The Database Recovery utility uses DBRC information about the Image Copy to determine its format and how to read it.

Database Recovery invokes fast replication with the “preferred” option. This indicates that fast replication will be used if the image copy and the database data sets are on the same storage system and the storage system supports fast replication. If these conditions are not met, standard copy techniques are used to restore the image copy to the database data set. There are two reason primary reasons that the conditions for fast replication would not be met. First, the user may choose to recover the database data set to a different storage system. Second, the image copy data set may be been moved from its original storage system.

Without fast replication, the database data set must be specified on a DD statement. Dynamic allocation cannot be used. With fast replication this restriction is removed. The database data set may be dynamically allocated. To request dynamic allocation you must place a 'D' in position 64 of the Database Recovery utility control statement.

The DBRC GENJCL.RECOV command has been enhanced to support dynamic allocation of the database data set when using a fast replication image copy as input. This is done with the NODBSDD keyword. The DBDSDD keyword is the default. It indicates that dynamic allocation will not be used and that a DD statement for the database data set will be generated by the GENJCL.RECOV command.



Fast Replication

- Benefits
 - ◆ Exploits FlashCopy and SnapShot
 - Single phase copies
 - Copies produced in seconds
 - ◆ Supports both clean and fuzzy image copies
 - ◆ Full GENJCL support
 - Image Copy 2 and Database Recovery utilities



Fuzzy User Image Copies

Fuzzy User Image Copy Support

- DBRC support for fuzzy user image copies
 - ◆ Fuzzy image copies taken by utility or tool without a DBRC interface
 - Pack dump, DFSMSdss DUMP or COPY not invoked by IC2, etc.
 - ◆ Support:
 - NOTIFY.UIC can specify a fuzzy user image copy
 - CIC and STOPTIME(time) parameters are used
 - GENJCL.RECOV can be used to generate recovery from logs after fuzzy user image copy has been restored
 - USERIC(time) or LASTUIC parameters are used

17

IMS V10 has added support for fuzzy user image copies. User image copies are copies of database data sets that are done without a DBRC interface. Typically, these are done with utilities that are not IMS utilities. In previous releases IMS supported clean user image copies. IMS V10 adds support for image copies taken while the database data set is being updated.

The support has two parts.

First, the NOTIFY.UIC command has been extended to include a specification of a fuzzy image copy, sometimes called a concurrent image copy. The CIC keyword has been added to the NOTIFY.UIC command. It indicates that the user image copy is concurrent or fuzzy. When notifying DBRC of a fuzzy image copy you must specify the stop time of the image copy. This is in addition to the start time which is required for all image copies. The stop time is specified with the STOPTIME(time stamp) parameter.

Second, the GENJCL.RECOV command has been extended to generate the appropriate JCL to include the correct logs for recoveries using fuzzy user image copies. The USERIC(time stamp) parameter specifies the fuzzy user IC that has been restored. LASTUIC may be used instead of USERIC(time). It specifies that the last fuzzy user image copy recorded in the RECONS has been restored. When either of these parameters is used, GENJCL.RECOV selects the proper logs for input to the recovery. These are the logs that were created while the fuzzy user image copy was being taken and any created since that time. USERIC and LASTUIC cause the appropriate S and M control statements to be generated for the Database Recovery utility.

When recovering after restoring a fuzzy user image copy do not specify USEDBDS on the GENJCL.RECOV command. USEDBDS is only for use when a clean user image copy has been restored followed by a NOTIFY.RECOV identifying this clean user image copy.

Fuzzy User Image Copy Support

- DB Recovery utility (DFSURDB0) control statement
 - ◆ 'M' required in column 63 of the S control statement
 - Indicates that a fuzzy user image copy has been restored
 - Indicates that an M control statement follows
 - ◆ M control statement
 - M is column 1
 - Runtime of the user fuzzy image copy in columns 2-32

```

|...+...1...+...2...+...3...+...4...+...5...+...6...
//SYSIN      DD  *
S  DHVNTZ02 VNT0A001                               M
M061531233296-0600

```




- ◆ M control statement is created by GENJCL.RECOV with USERIC(time) or LASTUIC

18

The use of a fuzzy user image copy requires an M in column 63 of the S control statement used by the Database Recovery utility (DFSURDB0). The example shows that the database data set with DDNAME VNT0A001 in database DHVNTZ02 is being recovered. The M in column 63 indicates that a fuzzy user image copy has been restored. An M in column 63 requires that an M control statement also be supplied. The M control statement has an 'M' in column 1 and the runtime of the fuzzy user image copy in columns 2-32. The time is specified in standard time format. The example shows a compressed time of year 2006, day 153, hour 12, minute 33, second 29.6, with an offset of -6 hours. This offset is Central Standard Time in North America.

DBRC verifies that the specified user image copy is recorded in the RECONS.

When GENJCL.RECOV is used with a fuzzy user image copy, the proper M control statement is created. GENJCL.RECOV uses a fuzzy user image copy when either the USERIC(time) or LASTUIC parameter is specified.

Procedures for Clean and Fuzzy User ICs

<ul style="list-style-type: none"> • Clean user IC <ol style="list-style-type: none"> 1. Create clean user IC 2. NOTIFY.UIC ... RUNTIME(time1) • Recovery using clean user IC <ol style="list-style-type: none"> 1. Restore clean user IC 2. NOTIFY.RECOV ... RCVTIME 3. GENJCL.RECOV ... USEDDBS 	<ul style="list-style-type: none"> • Fuzzy user IC <ol style="list-style-type: none"> 1. Create fuzzy user IC 2. NOTIFY.UIC ... CIC RUNTIME(time1) STOPTIME(time2) • Recovery using fuzzy user IC <ol style="list-style-type: none"> 1. Restore fuzzy user IC 2. GENJCL.RECOV ... USERIC(time1) or 2. GENJCL.RECOV ... LASTUIC
--	---

19

This is a comparison of the procedures to use with clean and fuzzy user image copies.

After taking a fuzzy user image copy the NOTIFY.UIC must include both the CIC keyword and the STOPTIME parameter.

Surprisingly, the recovery with a fuzzy user image copy is bit simpler than the recovery with a clean user image copy. You do not have to issue the NOTIFY.RECOV command when using a fuzzy user image copy. The GENJCL.RECOV commands differ. USEDDBS is required after restoring a clean user image copy. The identification of the fuzzy user image copy is required after restoring a fuzzy user image copy. Either a specific copy may be specified with the USERIC(time1) parameter or the last copy may be specified by using LASTUIC. When recovering after restoring a fuzzy user image copy do not specify USEDDBS on the GENJCL.RECOV command. USEDDBS is only for use when a clean user image copy has been restored followed by a NOTIFY.RECOV identifying this clean user image copy.

The recovery processes shown here do not include setting the Recovery Needed flag with the CHANGE.DBDS ... RECOV command. Some installations prefer to set the flag to protect themselves from unintentionally authorizing the database before the recovery is done.



Fuzzy User Image Copies

- Benefits
 - ◆ Expanded supported for fuzzy image copies
 - Non-disruptive to online system
 - ◆ DBRC support for fuzzy user image copies
 - Registration of these copies
 - GENJCL support for selecting the appropriate logs