DBVERIFY - Database file Verification Utility (文档 ID 35512.1)

转到底部

Introduction

~~~~~~~~~~~~

  This article describes the basic details of the DBVERIFY (or DBV)

  utility which can be used to check Oracle datafiles for signs of

  corruption. The article gives summary details of how to use

  DBV and gives an indication of what output to expect, along with

  notes on how to interpret the output. There is also an example at the

  end of the article.

Availability

~~~~~~~~~~~~

  The DBV utility is supplied with Oracle7 release 7.3.2 onwards and

  with all Oracle8 / 8i releases.

  DBV can be used against data files from earlier Oracle releases

  but it must be executed from the ORACLE\_HOME environment in which it

  is installed - you CANNOT just copy the executable about.

  Eg: DBVERIFY 7.3 can check Oracle 7.1 data files provided it is run

  from the 7.3 $ORACLE\_HOME

Purpose

~~~~~~~

  DBV checks Oracle datafiles to ensure that:

  - The datafile has a valid header

  - Each datablock in the file has a special "wrapper" which identifies

  the block - this "wrapper" is checked for correctness

  - DATA (TABLE) and INDEX blocks are internally consistent

  - From 8.1.6 onwards: That various other block types are internally

  consistent (such as rollback segment blocks)

  The tool can be used to give some degree of confidence that a

  datafile is free from corruption. It opens files in a read only mode

  and so cannot change the contents of the file being checked.

Usage

~~~~~

  DBV can be run against datafiles which are currently opened by a

  database instance - there is no need to shutdown the database.

  Datafiles are opened read-only by DBV so it cannot corrupt

  the contents. There was a bug on 8.0.4 where DBV could not be used

  on opened datafiles on Windows NT but that bug was fixed on 8.1.6.

  Bug:727547

  Unix:

  Any release: dbv FILE=filename [options]

  Windows NT:

  7.3: DBVERF73 FILE=filename [options]

  8.0: DBVERF80 FILE=filename [options]

  8.1: DBV FILE=filename [options]

  VMS:

  In versions less than 9.2.0 DBV cannot be used on VMS systems against files which are currently

  opened by an instance.

  Any release: DBV FILE=filename [options]

  MVS:

  7.3: Does not exist

  8.0/8.1: DBV FILE=/DSN/filename

  Options:

  Keyword Description Meaning

  --------- ------------------ -----------------

  FILE File to Verify This is the name of the file to verify.

  See "Limitations" below if your datafile

  name has no suffix.

  START Start Block This is the first datablock to check in

  the file. This defaults to the first

  block in the file and need only be

  specified if you want to check just

  a portion of a given file.

  END End Block This is the last datablock to check in the

  file. This defaults to the last block of

  the file but may need specifying for RAW

  devices (See "Limitations" below)

  BLOCKSIZE Logical Block Size This is the database block size of the

  datafile you wish to scan. The value

  defaults to "2048".

  This parameter must be set to the

  DB\_BLOCK\_SIZE of the datafile to be

  scanned.

  LOGFILE Output Log This is the name of file to output the

  results to. The default is "NONE" and

  output is sent to terminal.

  FEEDBACK Display Progress If set to a value above 0 (the default)

  then DBV outputs a "." for every N pages

  of the datafile checked. This is useful

  to see that DBV is working through the

  file.

  PARFILE Parameter file Parameters can be specified in a

  parameter file and PARFILE used to cause

  the file contents to be used as input

  parameters. The PARFILE can contain any

  of the above options.

  HIGH\_SCN Scn Highest Block SCN To Verify

  (scn\_wrap.scn\_base OR scn)

  Find the blocks exceeding the SCN.

  Available in version 9.2.0.6 and above.

  USERID Username/Password If the file you are verifying is an

  Automatic Storage Management (ASM) file,

  you must supply a USERID. This is because

  DBVERIFY needs to connect to an Oracle

  instance to access ASM files.

  SEGMENT\_ID TS#.FILE#.BLOCK# Specifies the segment that you want to verify.

  For more info, review Note:139962.1

  For help on command line parameters in a given version type

  "dbv help=y" at the command line.

Limitations and Special Notes

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  - As DBV performs checks at a block level it cannot detect problems

  such as INDEX versus TABLE mismatches which can be detected by the

  'ANALYZE TABLE .. VALIDATE STRUCTURE CASCADE' command.

  - This utility can ONLY be used against DATA files.

  It CANNOT be used to verify redo log files or control files.

  - You can use DBV to verify an Automatic Storage Management (ASM) file.

  However, the database must be opened and the option USERID has to be used

  Example : dbv file=+DG1/ORCL/datafile/system01.dbf userid=system/sys

  DBV checks the userid/password for ASM managed files, which is not possible when database is not open.

  - On most releases on Unix DBV expects a filename extension.

  This means that DBV cannot be used against datafiles with no

  filename suffix, or against RAW devices.

  The workaround is to create a symbolic link to the raw device where

  the link name MUST have an extension.

  Eg: ln -s /dev/rdsk/mydevice /tmp/mydevice.dbf

  Now use DBV against /tmp/mydevice.dbf

  - For RAW devices you should use the END parameter to avoid running

  off the end of the Oracle file space.

  eg: "dbv FILE=/dev/rdsk/r1.dbf END=<last\_block\_number>"

  If you get the END value too high DBV can report the last page/s of the

  file as corrupt as these are beyond the end of the Oracle portion of

  the raw device.

  You can find value for END from the V$DATAFILE view by dividing the

  BYTES value by the database block size.

  Eg: To find out the END value to use for file#=5:

  SVRMGRL> show parameter db\_block\_size

  NAME TYPE VALUE

  ----------------------------------- ------- ------

  db\_block\_size integer 2048

  SVRMGRL> select BYTES/2048 from v$datafile where FILE#=5;

  BYTES/2048

  ----------

  5120

  So the command would be:

  dbv file=/dev/rdsk/r1.dbf blocksize=2048 END=5120

  - DBV may not be able to scan datafiles larger than 2Gb and

  may report "DBV-100". This is reported in Bug:710888 for Unix and

  Bug:1372172 for 8.1.6 on NT. This problem is platform and release

  specific so if you get DBV-100 errors check the filesize first.

  - DBV from 8.1.6 onwards may report spurious errors for rollback segment

  blocks if the database has been migrated from Oracle7. See Bug:1359160

  and Note:118008.1.

  - DBV only checks a block in isolation - it does not know if the block

  is part of an existing object or not.

  - DBV is broken on SCO Unix - see Bug:814249

  - DBV of a lower version should not be used against a higher DB version.

  - Running DBV on Standby database in MOUNT mode will fail with:

  DBV-00111: OCI failure (4159) (ORA-00604: error occurred at recursive SQL level 1

  ORA-01219: database not open: queries allowed on fixed tables/views only

  ORA-06512: at "SYS.X$DBMS\_DBVERIFY", line 22

  As DBV needs the database to be open.

Known Bugs

~~~~~~~~~~

You can restrict the list below to issues likely to affect one of the following versions by clicking the relevant button:

NB

Prob

Bug

Fixed

Description

II

13350245

11.2.0.3.BP06, 11.2.0.4, 12.1.0.1

DBV reports false SCN corruption code 6054 / 6056 - "csc higher than block scn" for V7 block

II

12778063

12.1.0.1

DBV reports false SCN corruption code 6054 / 6056 - "csc higher than block scn" for V7 block -superceded

II

9393307

11.2.0.2, 12.1.0.1

DBVerify / RMAN / ANALYZE not detecting logical corruption in index with avsp < 0

E

II

8837919

11.2.0.2, 12.1.0.1

DBV / RMAN enhanced to detect ASSM blocks with ktbfbseg but not ktbfexthd flag set as in Bug 8803762

E

III

8720802

10.2.0.5, 11.2.0.1.BP07, 11.2.0.2, 12.1.0.1

Add check for row piece pointing to itself (db\_block\_checking,dbv,rman,analyze)

E

III

7517208

10.2.0.5, 11.2.0.1

DBV enhanced to identify Logical SCN Block corruptions

II

6820317

11.2.0.1

DBVERIFY fails with DBV-600 [22] if no write permissions on file

-

5472917

10.2.0.4, 11.1.0.6

DBVERIFY may error if ASM disk disconnected

E

-

5031712

10.2.0.4, 11.1.0.6

DBV enhanced to report NOLOGGING corrupt blocks with DBV-201 instead of DBV-200

E

II

4169479

11.1.0.6

dbverify can be slow

II

6673755

10.2.0.5

DBVerify dumps when run against ASM

-

4360183

10.2.0.1

DBV-102 file IO error with using DBV on ASM files over 4gb

-

3981329

9.2.0.8, 10.1.0.5, 10.2.0.1

DBV-102 accessing file being written in O\_DIRECT mode by DBWR

-

3571244

9.2.0.7, 10.1.0.4, 10.2.0.1

dbverify does not report "marked corrupt" blocks in its summary

E

-

3457571

9.2.0.6, 10.1.0.3, 10.2.0.1

Enhancement to DBV to show highest SCN in a file / show blocks above an SCN

-

2469314

9.2.0.3, 10.1.0.2

DBVERIFY may dump when reporting a block corruption

-

2348277

9.2.0.3, 10.1.0.2

DBV/RMAN may incorrectly report completely zero blocks as corrupt

-

2326948

9.0.1.4, 9.2.0.2, 10.1.0.2

DBV-111 using DBVERIFY SEGMENT\_ID=... on a bitmap space managed segment

-

2373145

9.2.0.2

DBVERIFY does not identify fractured blocks

-

2432864

9.2.0.2, 10.1.0.2

DBVERIFY does not work properly on RAW or for different BLOCKSIZE

-

2070167

8.1.7.4, 9.0.1.3, 9.2.0.1

DBV spins reporting same block corrupt if block type is bad

-

1847798

9.0.1.1, 9.2.0.1

DBV may spin when SEGMENT\_ID & USERID specified

-

1656136

8.1.7.3, 9.0.1.2, 9.2.0.1

DBVERIFY may incorrectly report TEMPORARY tablespace blocks as corrupt

I

1456397

9.2.0.1

DBV: DBV-100 using DBVERIFY against a file with no filename extension

I

1565578

8.1.7.2, 9.0.1.0

DBV reports spurious errors for Oracle 7 format UNDO blocks

-

1275263

8.1.7.0

DBVERIFY does not error on completely zero data blocks

-

749600

8.0.6.0

Large file support for CORE code (DBV etc..)

-

569962

7.3.4.3, 8.0.3.0

DBVERIFY incorrectly reports blocktype 0 as corrupt

- '\*' indicates that an alert exists for that issue.

- '+' indicates a particularly notable issue / bug.

- See Note:1944526.1 for details of other symbols used

Example Output

~~~~~~~~~~~~~~

  $ dbv file=users01.dbf blocksize=2048

  DBVERIFY - Verification starting : FILE = users01.dbf

  kdbchk: a row ends in the middle of another

  tab=0 slot=1 begin=0x7a0 len=0x14

  Page 3 failed with check code 5

  Page 10 is marked software corrupt

  Page 12 is marked software corrupt

  DBVERIFY - Verification complete

  Total Pages Examined : 512

  Total Pages Processed (Data) : 1

  Total Pages Failing (Data) : 1

  Total Pages Processed (Index): 0

  Total Pages Failing (Index): 0

  Total Pages Empty : 507

  Total Pages Marked Corrupt : 2

  Total Pages Influx : 0

  If the utility reports any pages to be 'Marked Corrupt' or 'Failing' then

  re-run the command to see if the problem is transient or not. If there

  are still corruptions reported then contact your local support centre

  for advice.

Interpreting the Output

~~~~~~~~~~~~~~~~~~~~~~~

  If any pages report an error then contact Oracle support with the output.

  This section gives a brief overview of the meaning of the main output

  lines from the above output.

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  "Page N failed with check code Y" errors

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  This implies the block wrapper is correct but the content of the block

  failed one of the many internal consistency checks.

  The "Page" number in the output is the database block number within the

  file.

  eg: kdbchk: a row ends in the middle of another

  tab=0 slot=1 begin=0x7a0 len=0x14

  Page 3 failed with check code 5

  This sort of corruption on a block is most likely to cause one of the

  following problems when accessed:

  - ORA-600 errors

  - Core dump (ORA-7445)

  - Corrupt data to be returned

  If block checking is enabled on the database then an update to the

  block may mark the block as corrupt (ORA-1578) or may just crash

  the session.

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  "Page N is marked software corrupt" or

  "Page N is marked media corrupt" or

  "Page N is marked corrupt"

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  This means the block wrapper is incorrect.

  Oracle7 does not dump any additional information so the form of the

  corruption is difficult to tell just from DBV. Oracle8 dumps details

  of the cache wrapper.

  The "Page" number in the output is the database block number within the

  file.

  eg: Page 10 is marked corrupt

  \*\*\*

  Corrupt block relative dba: 0x04c0000a file=0. blocknum=10.

  Bad header found during dbv:

  Data in bad block - type:6. format:2. rdba:0x04c000ff

  last change scn:0x056c.ce87bf25 seq:0x1 flg:0x00

  consistency value in tail 0xbf250601

  check value in block header: 0x0, check value not calculated

  spare1:0x0, spare2:0x0, spare2:0x0

  This sort of corruption on a block will show up as an ORA-1578

  when the block is accessed.

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  "Block Checking: DBA = 67108867, Block Type = Undo data block"

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  This form of error can be reported by DBV from 8.1.6 onwards if it

  sees corrupt rollback segment blocks. If the database has been

  migrated from Oracle7 then these errors may be spurious if DBV

  sees an Oracle7 format rollback segment block.

  See Bug:1359160 and Note:118008.1 for more details.

  If this is a real problem (and not due to an Oracle7 to 8 migration

  as described above) then this sort of corruption is most likely to

  cause one of the following problems when accessed:

  - ORA-600 errors

  - Core dump (ORA-7445)

  - Corrupt data

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  Summary lines

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  When DBV completes it outputs a summary of the form below:

  Output text Meaning

  ~~~~~~~~~~~ ~~~~~~~

  Total Pages Examined : 512 Number of blocks looked at

  Total Pages Processed (Data) : 1 Number of TABLE blocks seen

  Total Pages Failing (Data) : 1 Number of TABLE blocks with

  internal inconsistencies

  Total Pages Processed (Index): 0 Number of INDEX blocks seen

  Total Pages Failing (Index): 0 Number of INDEX block with

  internal inconsistencies

  Total Pages Empty : 507 Number of unused blocks seen

  Total Pages Marked Corrupt : 2 Number of blocks with corrupt

  cache wrappers

  Total Pages Influx : 0 Number of pages we re-read

  as the page looked like it

  was being modified when it

  was first read.

What to do if DBV shows problems on a file

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  If DBV reports any pages as 'Marked Corrupt' or 'Failing' then it is

  advisable to re-run the command to see if the problem is transient or not.

  Continual transient problems are often caused by a faulty disk controller

  and to the first step should be the check the disk subsystem.

  If DBV consistently reports errors in the same location then the file

  contains corrupt block/s:

  If there are very many errors it usually best to assume the

  file is bad and look for backups of the file which could be recovered.

  If there are only a few bad blocks then you need to note down:

  - The filename

  - The absolute file number of this file

  ( Use "SELECT file#, name FROM V$DATAFILE;" to find this )

  - The block number of the bad block/s

  ( This is the same as the "Page" number in the DBV output )

  - The type of error on the block

  If the block would signal an ORA-1578 error in Oracle

  there are more options to get around the corruption but

  the content of the block is lost. If the block is

  internally inconsistent there are less options to get

  around the corruption but some rows in the block may

  be accessible.

  Once you know the above details see Note:28814.1 which describes

  how to determine which object is corrupt and what you can do about

  it. Use the absolute FILE# for <F> and the page number for <B> in

  that article. Note that it is quite possible that the corrupt block

  does not belong to any current object in which case the corruption

  can be ignored.

  \*\*\* IMPORTANT \*\*\* If the file you have DBVed is from a backup

  then you cannot use the current data dictionary

  to determine which object/s have a problem as

  the block may have been reused by a different

  object. (eg: Original table dropped)

Example

~~~~~~~

1) Log into svrmgrl to find out block size for your databae and datafile

  locations and names:

  $ svrmgrl

  Oracle Server Manager Release 3.1.5.0.0 - Production

  (c) Copyright 1997, Oracle Corporation. All Rights Reserved.

  Oracle8i Enterprise Edition Release 8.1.5.0.0 - Production

  With the Partitioning and Java options

  PL/SQL Release 8.1.5.0.0 - Production

  SVRMGR> connect internal

  Connected.

  SVRMGR> show parameter db\_block\_size

  NAME TYPE VALUE

  ----------------------------------- ------- -----------------------

  db\_block\_size integer 2048

  SVRMGR> select file#,name,bytes/2048 from v$datafile;

  FILE# NAME BYTES/2048

  ----- --------------------------------------- ----------

  1 /u02/oradata/R815/oradata/R815/system01.dbf 40960

  2 /u02/oradata/R815/oradata/R815/rbs01.dbf 7680

  3 /u02/oradata/R815/oradata/R815/temp01.dbf 5120

  4 /u02/oradata/R815/oradata/R815/users01.dbf 67454

  5 /u02/oradata/R815/oradata/R815/indx01.dbf 5120

  ...

2) Run dbv against any files you want to check:

  $ dbv file=/u02/oradata/R815/oradata/R815/users01.dbf blocksize=2048 logfile=users01\_dbv.log feedback=100

  DBVERIFY: Release 8.1.5.0.0 - Production on Tue Mar 21 15:05:35 2000

  (c) Copyright 1999 Oracle Corporation. All rights reserved.

  ...................................................................................

  $ cat users01\_dbv.log

  DBVERIFY: Release 8.1.5.0.0 - Production on Tue Mar 21 15:05:35 2000

  (c) Copyright 1999 Oracle Corporation. All rights reserved.

  DBVERIFY - Verification starting : FILE = /u02/oradata/R815/oradata/R815/users01.dbf

  DBVERIFY - Verification complete

  Total Pages Examined : 67454

  Total Pages Processed (Data) : 29310

  Total Pages Failing (Data) : 0

  Total Pages Processed (Index): 3425

  Total Pages Failing (Other): 294

  Total Pages Empty : 34425

  Total Pages Marked Corrupt : 0

  Total Pages Influx : 0

References:

~~~~~~~~~~~

Note:139962.1 DBVERIFY enhancement - How to scan an object/segment

  Oracle Server Utilities Guide