IBM DB2 Universal
Database for UNIX
and Windows:
New Log File
Management

Document Version 1.00 – October 13th, 2004
Terms for Included Open Source Software

This SAP software contains also the third party open source software products listed below. Please note that for these third party products the following special terms and conditions shall apply.

SAP License Agreement for STLport

SAP License Agreement for STLport
between

SAP Aktiengesellschaft
Systems, Applications, Products in Data Processing
Neurottstrasse 16
69190 Walldorf
Germany
( hereinafter: SAP )

and

you
( hereinafter: Customer )

1. Subject Matter of the Agreement
a. SAP grants Customer a non-exclusive, non-transferable, royalty-free license to use the STLport.org C++ library (STLport) and its documentation without fee.
b. By downloading, using, or copying STLport or any portion thereof Customer agrees to abide by the intellectual property laws, and to all of the terms and conditions of this Agreement.
c. The Customer may distribute binaries compiled with STLport (whether original or modified) without any royalties or restrictions.
d. Customer shall maintain the following copyright and permission notices on STLport sources and its documentation unchanged:
   Copyright 2001 SAP AG

e. The Customer may distribute original or modified STLport sources, provided that:
   • The conditions indicated in the above permission notice are met;
   • The following copyright notices are retained when present, and conditions provided in accompanying permission notices are met:

   Copyright 1994 Hewlett-Packard Company
   Copyright 1997 Moscow Center for SPARC Technology.
   Copyright 1999,2000 Boris Fomitchev
   Copyright 2001 SAP AG

   Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

   Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

   Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Moscow Center for SPARC Technology makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.
2. Support and Maintenance
SAP does not provide software maintenance for the STLport. Software maintenance of the STLport therefore shall be not included.
All other services shall be charged according to the rates for services quoted in the SAP List of Prices and Conditions and shall be subject to a separate contract.

3. Exclusion of warranty
As the STLport is transferred to the Customer on a loan basis and free of charge, SAP cannot guarantee that the STLport is error-free, without material defects or suitable for a specific application under third-party rights. Technical data, sales brochures, advertising text and quality descriptions produced by SAP do not indicate any assurance of particular attributes.

4. Limited Liability
a. Irrespective of the legal reasons, SAP shall only be liable for damage, including unauthorized operation, if this (i) can be compensated under the Product Liability Act or (ii) if caused due to gross negligence or intent by SAP or (iii) if based on the failure of a guaranteed attribute.

b. If SAP is liable for gross negligence or intent caused by employees who are neither agents or managerial employees of SAP, the total liability for such damage and a maximum limit on the scope of any such damage shall depend on the extent to which its occurrence ought to have been anticipated by SAP when concluding the contract, due to the circumstances known to it at that point in time representing a typical transfer of the software.

c. In the case of Art. 4.2 above, SAP shall not be liable for indirect damage, consequential damage caused by a defect or lost profit.

d. SAP and the Customer agree that the typical foreseeable extent of damage shall under no circumstances exceed EUR 5,000.

e. The Customer shall take adequate measures for the protection of data and programs, in particular by making backup copies at the minimum intervals recommended by SAP. SAP shall not be liable for the loss of data and its recovery, notwithstanding the other limitations of the present Art. 4 if this loss could have been avoided by observing this obligation.

f. The exclusion or the limitation of claims in accordance with the present Art. 4 includes claims against employees or agents of SAP.
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Example Text</em></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.</td>
</tr>
<tr>
<td><strong>Example Text</strong></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

## Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="caution-icon.png" alt="Caution" /></td>
<td>Caution</td>
</tr>
<tr>
<td><img src="example-icon.png" alt="Example" /></td>
<td>Example</td>
</tr>
<tr>
<td><img src="note-icon.png" alt="Note" /></td>
<td>Note</td>
</tr>
<tr>
<td><img src="recommendation-icon.png" alt="Recommendation" /></td>
<td>Recommendation</td>
</tr>
<tr>
<td><img src="syntax-icon.png" alt="Syntax" /></td>
<td>Syntax</td>
</tr>
</tbody>
</table>

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see Help on Help → General Information Classes and Information Classes for Business Information Warehouse on the first page of any version of SAP Library.
Contents

1 Introduction ....................................................................................................................... 7

2 DB2 Log File Management Concepts ................................................................. 8

2.1 New Components ........................................................................................................ 8
  2.1.2 Configuration ............................................................................................................. 10
  2.1.2 Log File Chains ........................................................................................................ 11
  2.1.3 Database Recovery Using the Recover Command .................................................. 12

2.2 DB2 Log Manager Backend Support ........................................................................ 13
  2.2.1 Disk ............................................................................................................................ 13
  2.2.2 Tivoli Storage Manager ............................................................................................ 15
  2.2.3 Other Storage Vendors ........................................................................................... 16
  2.2.4 User Exit .................................................................................................................... 17

2.3 History File ................................................................................................................. 17

2.4 Monitoring the DB2 Log Manager ........................................................................... 21

2.5 Tape Support .............................................................................................................. 22
  2.5.1 Configuration of the DB2 Tape Manager ................................................................. 22
  2.5.2 Tape Labeling .......................................................................................................... 23
  2.5.3 Physical Tape Layout ............................................................................................... 23
  2.5.4 DB2 Tape Manager Command Line Interface ......................................................... 25
  2.5.5 Archiving Log Files to Tape Using STORE and DOUBLE STORE ......................... 27
  2.5.7 Retrieving Log Files from Tape ............................................................................ 29
  2.5.8 Other Operations ..................................................................................................... 31
  2.5.9 Security Features .................................................................................................... 32
  2.5.10 Troubleshooting .................................................................................................... 32

3 Migration ...................................................................................................................... 35
  3.1 Differences Between the Log File Management Concepts ...................................... 35
  3.2 Performing the Migration ........................................................................................ 38
1 Introduction

This documentation describes the concepts of the new DB2 log file management and how you can migrate from the existing SAP DB2 log file management solution to the DB2 V8.2 log file management solution.

In the past, SAP delivered its own log file management tools for the administration of the DB2 transaction logs. With version 8.2, IBM DB2 UDB for UNIX and Windows introduces its own log file management. This new log file management is fully integrated into DB2 and provides features such as log file chaining, which is not available within the SAP DB2 log file management.

The SAP DB2 log file management is based on the user exit interface of DB2 for handling log files. DB2 V8.2 provides a compatibility mode for the user exit interface, thus the SAP DB2 log file management can still be used with IBM DB2 UDB for UNIX and Windows Version 8.2.

As of IBM DB2 UDB for UNIX and Windows Version 9.1 the compatibility will no longer be supported.

In addition, as of the release following SAP NetWeaver ’04 and for all products based on it, SAP will no longer support the SAP DB2 log file management tools.

The existing SAP DB2 log file management tools will not be supported on new upcoming platforms, such as Linux on AMD64 and pSeries as well as Windows on AMD64.

The SAP DB2 log file management tools are only supported for SAP applications that are based on SAP Web AS ABAP. You cannot use the SAP DB2 log file management tools for SAP products that are based on SAP Web AS Java or products like SAP BusinessOne.

However, the new DB2 log file management is independent from SAP products and thus you can use it with all SAP products that run on DB2 UDB for UNIX and Windows.
2 DB2 Log File Management Concepts

2.1 New Components

The following graphic shows the new concept of the log file management of DB2 Version 8.2 including the two new components, the DB2 log manager (db2logmgr) and the DB2 tape manager (db2tapemgr).

**DB2 Log Manager**

This is the central component for managing log files. The DB2 log manager (db2logmgr) is part of the DB2 engine and is responsible for archiving and retrieving log files.
The location of the log files is recorded in the DB2 history file. The DB2 log manager supports the following archiving media:

- **USEREXIT**: Used to provide the downward compatibility to the current user exit interface. The DB2 log manager will call the user exit executable for log file management like today.

- **TSM**: The DB2 log manager has built-in support for accessing the Tivoli Storage Manager (TSM).

- **DISK**: The DB2 log manager can use a disk location for archiving log files.

- **VENDOR**: DB2 provides a vendor API for the log file management, which is an extension to the existing backup API. Storage vendors can provide their own library to allow log file management with DB2.

The DB2 log manager supports two archiving locations (so that a log file can be stored on two different locations), for example, TSM, disk or vendor products. The DB2 log manager is configured using database configuration parameters.

If log files are retrieved, the DB2 log manager directly retrieves them from the backend and puts them in the DB2 transaction log directory (`log_dir`). From there the DB2 engine can read them to perform a database recovery or rollforward operation.

If log files cannot be archived to the designated destination, for example, due to a network outage, you can specify a local directory (`FAILARCHPATH`) that is used as intermediate storage for the log files. The DB2 log manager puts log files into the `FAILARCHPATH`, if the archiving destination is not available. If the archiving destination becomes available again, the DB2 log manager will move them to the archiving destination. This may help to avoid the problem of log dir full problem.

**DB2 Tape Manager**

This is an executable, which you can call from the command line. The DB2 tape manager (`db2tapemgr`) can be used to archive DB2 log files to tape. The DB2 log manager cannot directly handle the log files that are stored on tape. You have to call the DB2 tape manager (`db2tapemgr`) explicitly from the command line.

The log files, which are retrieved from the DB2 tape manager, are put in the DB2 overflow log path (`OVERFLOWLOGPATH`) and not directly in the transaction log directory (`log_dir`). From there, the DB2 engine can read them to perform a database recovery or rollforward operation. If log files are archived to tape, the DB2 tape manager (`db2tapemgr`) updates the history file. This will help you to identify the tapes that are needed for a database recovery.
## 2.1.2 Configuration

The following table lists the database configuration parameters, which control the DB2 log file management configuration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGARCHMETH1</td>
<td>Specifies the media type of the primary destination for archived log files. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• DISK:&lt;path&gt;</td>
</tr>
<tr>
<td></td>
<td>• TSM:&lt;TSM management class&gt;</td>
</tr>
<tr>
<td></td>
<td>• VENDOR:&lt;vendor library&gt;</td>
</tr>
<tr>
<td></td>
<td>• USEREXIT</td>
</tr>
<tr>
<td></td>
<td>• LOGRETAIN</td>
</tr>
<tr>
<td>LOGARCHMETH2</td>
<td>Specifies the media type of the secondary destination for archived log files. If this variable is</td>
</tr>
<tr>
<td></td>
<td>specified, log files are archived to both this destination and the destination that is specified</td>
</tr>
<tr>
<td></td>
<td>by the database configuration parameter LOGARCHMETH1.</td>
</tr>
<tr>
<td></td>
<td>Only the destinations DISK, TSM and VENDOR are allowed for this parameter.</td>
</tr>
<tr>
<td>LOGARCHOPTS1</td>
<td>Specifies the options for the primary destination specified in LOGARCHMETH1 for archived log files (if required).</td>
</tr>
<tr>
<td></td>
<td>You can use this parameter, for example, to specify an additional TSM parameter, for example, –fromnode &lt;node&gt; –fromowner &lt;owner&gt;.</td>
</tr>
<tr>
<td>LOGARCHOPTS2</td>
<td>Specifies the options for the secondary destination specified in LOGARCHMETH2 for archived log files (if required).</td>
</tr>
<tr>
<td>FAILARCHPATH</td>
<td>Intermediate location for log files that cannot be archived to either the primary or (if set) the secondary archiving destinations (because of a media problem affecting these destinations).</td>
</tr>
<tr>
<td></td>
<td>The specified path must reference a disk location.</td>
</tr>
<tr>
<td>NUMARCHRETRY</td>
<td>Specifies the number of times that DB2 tries to archive a log file to the primary or the secondary archiving directory before trying to archive the log file to the failover directory.</td>
</tr>
<tr>
<td></td>
<td>This parameter is only used if the FAILARCHPATH database configuration parameter is set. If NUMARCHRETRY is not set, DB2 continues to try archiving to the primary or the secondary log archiving destination.</td>
</tr>
</tbody>
</table>
2 DB2 Log File Management Concepts

| ARCHRETRYDELAY | Specifies the number of seconds DB2 has to wait after a failed archiving attempt before trying to archive the log file again.
| Subsequent retries only take effect if the value of the NUMARCHRETRY database configuration parameter is at least 1. |
| OVERFLOWLOGPATH | Specifies an additional location for DB2 to find log files that are needed for a rollforward operation. |

There is an additional DB2 registry variable called DB2_TAPEMGR_TAPE_EXPIRATION. This variable specifies when it is allowed to overwrite log file tapes. The value DB2_TAPEMGR_TAPE_EXPIRATION defines the number of days before the tape can be overwritten. Setting this variable avoids that you overwrite log files on tape that are still needed for a database recovery.

2.1.2 Log File Chains

The DB2 transaction log files have consecutive names from S0000000.LOG to S99999999.LOG. If a log file is full, DB2 creates a new log file with the next number.

In some special cases DB2 can create log files with the same name but different contents.

The following graphic describes a possible scenario with different log file chains.
As already mentioned, all the information of log file locations is stored in the history file. The picture above shows the content of the history file, if the following steps are performed:

1. An offline backup B1 is created.
2. Transactional work on the database creates log files 0 - 8 that belong to log file chain 0.
3. A database recovery to point-in time T1 is performed. This is done by using backup image 1 and applying log files 0 - 4.
4. Transactional work on the database creates log files 5 - 14, which belong to log file chain 1.
5. A database recovery to point-in time T3 is performed. This is done by using backup image 2 and applying log files 8 - 10 of log file chain 1.
6. Transactional work on the database creates log files 11 - 14, which belong to log file chain 2.
7. A database recovery to point-in time T2 is performed. This is done by using backup image 1 and applying log files 0 - 4 of log file chain 0 and log files 5 - 6 of log file chain 1.
8. Transactional work on the database creates log files 7 - 9, which belong to log file chain 3.

The example shows that the log file chaining ensures that you can recover the database to any point in time with the right set of log files.

### 2.1.3 Database Recovery Using the Recover Command

Up to DB2 Version 8.1 you had to select a backup image, perform the restore of the backup image and perform the database rollforward with the correct set of log files. In a multi-partitioned environment it was even more complex, because you had to perform this procedure for each database partition. In addition, you first had to perform the restore of the database partition where the system catalog resides.

The `recover` command, which is introduced with DB2 Version 8.2, simplifies the database recovery considerably. The `recover` command selects a suitable backup image and the log files required to recover the database to a specific point in time or to end of logs. The `recover` command performs the restore and the rollforward operation. This also applies in a multi-partitioned environment with one command call.

This approach always requires an up-to-date history file because, all information about backup images and log files is retrieved from there.

![Tip]

To get an up-to-date history file, create an empty tablespace and back up this tablespace on a regular basis to get regular backups of the history file with a small overhead. If you archive log files to tape, you will find a relatively new version of the history file on the latest tape.

The database recovery with the `recover` command is performed in two phases.

In the **first phase** the `recover` command selects a suitable backup from the history file and restores the backup image. This is done for all partitions.
In the **second phase** the recover command performs the database rollforward to the specified point-in-time. During the rollforward phase of the recover command the following search order applies:

1. Log directory or mirror log directory
2. `<OVERFLOWLOGPATH>/Szzzzzzzzz.LOG`
3. `<OVERFLOWLOGPATH>/NODExxxx/Cyyyyyyy/Szzzzzzzzz.LOG`
4. From the location, which is stored in the history file
5. From `LOGARCHMETH1` or `LOGARCHMETH2`

If you call the rollforward command separately, the log file search is different. In this case the log file from the latest chain (the newest log file with the same number) is always used to roll forward the database.

### 2.2 DB2 Log Manager Backend Support

The DB2 log manager supports the following backends:

- **Disk** [Page 13]
- **Tivoli Storage Manager (TSM)** [Page 15]
- **Other storage vendors** [Page 16]
- **User exit** [Page 17]

#### 2.2.1 Disk

You activate log file archiving to disk by using the prefix `DISK` for the database configuration variables `LOGARCHMETH1` and `LOGARCHMETH2`. For example, the general format of the value is `DISK:<log_archive>`.

⚠️ The directory `<log_archive>` **must** exist before you issue this command.

The following command sets the log archiving method 1 for the database PRD to directory `/db2/PRD/log_archive`:

```
db2 update db cfg for PRD using logarchmeth1 DISK:/db2/PRD/log_archive
```

The log files will be stored in a hierarchy of subdirectories under the path specified for the log archiving method `<log_archive>`. The hierarchy looks as follows:

```
<log_archive>/<instance>/<database>/NODExxxx/Cyyyyyyy/Szzzzzzzzz.LOG
```
This hierarchy avoids that log files are overwritten by other database instances.

<table>
<thead>
<tr>
<th>Subdirectory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;log_archive&gt;</td>
<td>Basis path specified by the database configuration parameter LOGARCHMETH1 or LOGARCHMETH2</td>
</tr>
<tr>
<td>&lt;instance&gt;</td>
<td>Name of the database instance</td>
</tr>
<tr>
<td>&lt;database&gt;</td>
<td>Name of the database identifier</td>
</tr>
<tr>
<td>NODExxxxx</td>
<td>Partition for which the log file was created. xxxx are digits from 0 - 9 and specify the partition number</td>
</tr>
<tr>
<td>Cyyyyyyy</td>
<td>Describes to which log file chain the log files belong. yyyyyy are digits from 0-9 and specify the log file chain number.</td>
</tr>
<tr>
<td>Szzzzzzz.LOG</td>
<td>Name of log file. Zzzzzzzz are digits from 0-9 and specify the log file number.</td>
</tr>
</tbody>
</table>

An entry in the history file for the log file that was archived to disk looks as follows:

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
X  D  20040721170957      1    D  S0000000.LOG C0000000
```

Comment:
Start Time: 20040721170957
End Time: 20040721170957
Status: A

EID: 2 Location: /db2/PRD/log_archive/db2prd/PRD/NODE0000/C0000000/S0000000.LOG

As you can see, field Type is 1 for LOGARCHMETH1, field Dev is D for disk and the Location shows the fully qualified path of the log file.
2.2.2 Tivoli Storage Manager

DB2 has integrated backend support for archiving backups to TSM and also provides this support for the new log file management. You activate log archiving to TSM by using the prefix TSM for the database configuration variables LOGARCHMETH1 or LOGARCHMETH2.

In addition, you can specify a TSM management class (the default is the default TSM management class) as well as additional TSM parameters by using the LOGARCHOPTS1 and LOGARCHOPTS2 database configuration variables.

```
   db2 update db cfg for PRD using logarchmeth1 TSM:LOGMGMTCLS
   db2 update db cfg for PRD using logarchopts1 "-fromnode prdsystem"
```

The log archiving method for TSM is set to the TSM management class LOGMGMTCLS. With the second command the fromnode TSM option will be set.

The specified TSM management class must have a TSM archive copy group defined. DB2 backups are stored to TSM using the TSM backup copy group. The DB2 log files are stored to TSM using the TSM archive copy group.

The log files are stored in TSM as follows:

- In the file space with the database name
- As high level name, the partition number (NODExxxx) is used.
- As log level name, the log file name including the log file chain (Szzzzzzzz_Cyyyyyyy.LOG) is used.

An entry in the history file for the log file stored in TSM looks as follows:

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
--- --- ------------------ ---- --- ------------ ------------ --------------
X  D  20040811112452      1    A  S0000006.LOG C0000003
```

Comment:
Start Time: 20040811112452
End Time: 20040811112503
Status: A

EID: 31 Location: LOGMGMTCLS

As you can see, field Type is 1 for LOGARCHMETH1, device field Dev is A for Adstar Storage Manager (the former name of TSM) and the Location shows the used TSM management class.

To query, extract or delete log files from TSM, you can use the DB2 tool db2adutl. The db2adutl tool is described in detail in the IBM documentation DB2 Command Reference.

However, if you want to delete log files in TSM, we strongly recommend that you use the DB2 prune command. Thus, the entries in the history file are also updated. For more information, see Deleting Entries in the History File [Page 21].
The log files are stored with the database instance owner user. Therefore, you always have to use the db2adutl tool as database instance owner. Otherwise, you will not see the log files in TSM.

### 2.2.3 Other Storage Vendors

Other vendors may provide log file management support by providing a vendor library. The API (Application Programming Interface), which the library has to implement, is described in the appendix of the IBM documentation *DB2 Administration API Reference*. The API for log file management extends the existing backup vendor API of DB2. You can find the required definitions in the header file `sqluvend.h`, which is part of the DB2 product.

You activate log archiving with a vendor library by using the prefix VENDOR and the path to the vendor library for the database configuration variables LOGARCHMETH1 or LOGARCHMETH2. In addition, you can specify vendor-specific options using database configuration variables LOGARCHOPTS1 and LOGARCHOPTS2. DB2 passes the options, which are set with LOGARCHOPTS1/2, on to calls to the vendor library.

To set LOGARCHMETH1 for archiving with a vendor library, enter the following command:

```sql
db2 update db cfg for PRD using LOGARCHMETH1 VENDOR: d:\sql\lib\bin\db2vendor.dll
```

A typical entry in the history file for log files that were archived with a vendor library looks as follows:

```
Op   Obj  Timestamp+Sequence Type  Dev  Earliest Log  Current Log  Backup ID
-----------------------------------------------
X    D   20040811122524      1    O  S0000007.LOG  C0000003  ------------------

Comment:
Start Time: 20040811122524
End Time: 20040811122525
Status: A

EID: 33 Location: d:\sql\bin\db2vendor.dll
```

As you can see, field Type is 1 for LOGARCHMETH1, field Dev is 0 for other vendor and the location field shows the used vendor library.
2.2.4 User Exit

For downward compatibility with the former user exit concept, you can specify value USEREXIT for variable LOGARCHMETH1. In this case, you cannot use the database configuration variable LOGARCHMETH2.

If you use this mode, the user exit program db2uext2 is called to archive and retrieve log files. The user exit concept does not support log file chains.

To set LOGARCHMETH1 for archiving with the user exit, enter the following command:

db2 update db cfg for PRD using LOGARCHMETH1 USEREXIT

A typical entry in the history file for a log file that was archived with the user exit looks as follows:

<table>
<thead>
<tr>
<th>Op</th>
<th>Obj</th>
<th>Timestamp+Sequence</th>
<th>Type</th>
<th>Dev</th>
<th>Earliest Log</th>
<th>Current Log</th>
<th>Backup ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>D</td>
<td>20040721143030</td>
<td>1</td>
<td>U</td>
<td>S0000329.LOG</td>
<td>C0000000</td>
<td></td>
</tr>
</tbody>
</table>

Comment:
Start Time: 20040721143030
End Time: 20040721152108
Status: A

As you can see, field Type is 1 for LOGARCHMETH1, field Dev is U for user exit.

2.3 History File

The history file contains information about the location of log files. Log file entries are created, if a new log file is used by the database during normal operation or if a log file is applied during a database rollforward.

To list the log file information about the command line, you can use the DB2 command list history:

db2 list history archive log all for <db>

The following sample output of this command shows two entries:

- The first entry refers to an archived log file.
- The second entry displays the result of the db2 archive log for db <db> command. ARCHIVE LOG in the Comment field and N in the Type field in the entry for the DB2 command archive log for db.
D:\>db2 list history archive log all for sample

List History File for sample

Number of matching file entries = 2

<table>
<thead>
<tr>
<th>Op</th>
<th>Obj</th>
<th>Timestamp+Sequence</th>
<th>Type</th>
<th>Dev</th>
<th>Earliest Log</th>
<th>Current Log</th>
<th>Backup ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>D</td>
<td>20040721170957</td>
<td>D</td>
<td>S</td>
<td>S0000000.LOG</td>
<td>C0000000</td>
<td></td>
</tr>
</tbody>
</table>

Comment:
Start Time: 20040721170957
End Time: 20040721170957
Status: A

EID: 2 Location: e:\log_archive\DB2\SAMPLE\NODE0000\C0000000\S0000000.LOG

<table>
<thead>
<tr>
<th>Op</th>
<th>Obj</th>
<th>Timestamp+Sequence</th>
<th>Type</th>
<th>Dev</th>
<th>Earliest Log</th>
<th>Current Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>D</td>
<td>20040721170957</td>
<td>N</td>
<td></td>
<td>S0000000.LOG</td>
<td></td>
</tr>
</tbody>
</table>

Comment: ARCHIVE LOG
Start Time: 20040721170957
End Time: 20040721170957
Status: A

EID: 3

For archived log file entries the following fields are used:

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Op (Operation)</td>
<td>Always X</td>
</tr>
<tr>
<td>Obj (Object)</td>
<td>Always D</td>
</tr>
<tr>
<td>Timestamp+Sequence</td>
<td>A 14-digit timestamp that indicates the log file creation time during normal operation. The sequence number is not used. If the log file entry was created during rollforward, the field contains the time when the log file was applied.</td>
</tr>
<tr>
<td>Type</td>
<td>The following types are possible:</td>
</tr>
<tr>
<td></td>
<td>• F (Primary LOGPATH)</td>
</tr>
<tr>
<td></td>
<td>• M (MIRRORLOGPATH)</td>
</tr>
<tr>
<td></td>
<td>• F (FAILARCHPATH)</td>
</tr>
<tr>
<td></td>
<td>• 1 (LOGARCHMETH1)</td>
</tr>
<tr>
<td></td>
<td>• 2 (LOGARCHMETH2)</td>
</tr>
</tbody>
</table>
The following devices are possible:
- D (Disk)
- A (TSM)
- O (Vendor)
- U (User exit)

Earliest Log
- Log file name, for example, S0000000.LOG

Current Log
- Chain number the log file belongs to. An eight-character string starting with C and the chain number, for example, C0000000

Comment
- Additional information about log file location or tape location

Starttime
- Same as Timestamp+Sequence

Endtime
- 14-digit timestamp that indicates when log file was archived.

Status
- Always A (active).

Location
- Information about log file location, depending on the archiving method

EID
- Unique identifier for the entry in the history file

If you have specified two archiving methods, the history file will contain two entries per log file.

The following graphic shows how the device type and the operation type are mapped to the locations of the log files.
Updating the History File

You update the history file using the `UPDATE HISTORY` command. This command can be used, for example, in the following situations:

- If you have moved log files or backups, you can update the location and device type.
- If a backup is no longer available, you can update the status to `inactive`. Thus, you make sure that the recover command does not try to use the backup for database recovery.

To update the status to `inactive`, enter the following command:

```
db2 update history eid 10 with status 'I'
```

The syntax diagram for the `UPDATE HISTORY` command looks as follows:

```
UPDATE HISTORY FOR object-part
  [EID eid]
  WITH LOCATION new-location
  DEVICE TYPE new-device-type
  COMMENT new-comment
  STATUS new-status
```
Deleting Entries in the History File

To delete entries in the history file, you use the `PRUNE HISTORY` command as shown in the following syntax diagram:

```
PRUNE HISTORY |timestamp| WITH FORCE OPTION | AND DELETE |
LOGFILE PRIOR TO | log-file-name |
```

If you use **AND DELETE** or **LOGFILE PRIOR TO**, the log files are deleted physically as well.

You **cannot** use this command to delete log files that are stored on tape.

If log file are stored in TSM or any other storage management system, the storage management system rules may automatically delete old log files. These deletions are not automatically reflected in the history file.

2.4 Monitoring the DB2 Log Manager

To get an overview which operations were performed by the DB2 log manager, you can use the new DB2 tool `db2diag`, which is introduced with DB2 Version 8.2, by calling the following command:

```
db2diag -gi "proc := db2logmgr"
```

A sample output looks as follows:

```
2004-08-11-15.43.18.509829+120 I1367G363          LEVEL: Warning
PID     : 7118                 TID  : 1024        PROC : db2logmgr
(L4D)
INSTANCE: db2l4d               NODE : 000
FUNCTION: DB2 UDB, data protection, sqlpgArchiveLogFile, probe:3180
MESSAGE : Successfully archived log file S0000330.LOG to USEREXIT from
          /data/db2/L4D/log_dir/NODE0000/.

2004-08-11-15.49.04.652665+120 I14249G363         LEVEL: Warning
PID     : 9435                 TID  : 1024        PROC : db2logmgr
(L4D)
INSTANCE: db2l4d               NODE : 000
FUNCTION: DB2 UDB, data protection, sqlpgArchiveLogFile, probe:3180
MESSAGE : Successfully archived log file S0000331.LOG to USEREXIT from
          /data/db2/L4D/log_dir/NODE0000/.
```

For more information about the new `db2diag` tool, see the IBM documentation *DB2 Command Reference*. 
2.5 Tape Support

The DB2 tape manager (db2tapemgr) archives log files to tape and retrieves them from tape.

For a database recovery, you must retrieve the log files from tape before the recovery procedure is started.

It updates the entries in the history file. In addition, the DB2 tape manager queries the required log files and their locations including the tapes that are required for a database recovery.

Before using the DB2 tape manager you should take the following information into consideration:

- You have to explicitly call the DB2 tape manager to archive or retrieve log files to or from tape.
- The DB2 tape manager does not need a database connection and thus the database can be deactivated when you call the DB2 tape manager. As a result the DB2 tape manager needs to be called as instance owner, because access to the history file on operating system level is required.
- The DB2 tape manager supports multi-partition environments. However, you have to call the tool for each partition separately and you must execute it on the system where the database partition resides.
- The DB2 tape manager supports the same set of tape devices as the standard DB2 backup.

2.5.1 Configuration of the DB2 Tape Manager

The following list describes configuration steps that you should perform before you call the DB2 tape manager:

- For tape support, you have to set the database configuration parameter LOGARCHMETH1 to a disk location. To archive log files, the DB2 tape manager checks the history file for entries that are related to parameter LOGARCHMETH1. Log files, that are stored to disk using the LOGARCHMETH2, cannot be stored on tape using the DB2 tape manager.

- You should also set the DB2 registry variable DB2_TAPEMGR_TAPE_EXPIRATION to a feasible duration in days. This avoids that log file tapes that contain relatively new log files, can accidentally be overwritten. For example, if you perform daily backups and you want to keep log files for at least one week, you have to set the variable DB2_TAPEMGR_TAPE_EXPIRATION to 7.

- To reduce the amount of parameters that you have to specify for calls to the DB2 tape manager, you can set the database configuration variable OVERFLOWLOGPATH and the environment variable or registry variable DB2DBDFT.
2.5.2 Tape Labeling

To help the administrator to manage the tapes, the DB2 tape manager supports a simple tape labeling.

Each tape receives a tape label that was either automatically generated or supplied on the command line during archiving operations.

The automated tape label consists of the database name and the current time as a 14-digit timestamp.

<dbalias><YYYYMMDDHHMMSS>, sample PRD20040805121303.

For easy identification this tape label can be written on the tape cover. In addition, the tape label is written on the tape as part of the tape header file DB2TAPEMGR.HEADER.

The tape label can be up to 22 characters long, has to be alphanumeric (characters A - Z and digits from 0 - 9) and is not case-sensitive.

2.5.3 Physical Tape Layout

Each file that the DB2 tape manager writes to tape is encapsulated in a cpio archive.

The cpio archive file format is a well known file format to UNIX administrators and can be read and written to with the UNIX command cpio. The advantage of writing each file into its own cpio archive file is that fast tape positioning is possible.

Between each cpio archive a file marker is created on the tape. This allows fast positioning on the tape. Using the cpio format for log files has the advantage that incomplete written files can be recognized and you can use standard UNIX tools to read the tape. The default block size, that is used for writing the cpio archives to tape, is 5120 bytes as with the cpio option -B.

Example for a Layout:

<table>
<thead>
<tr>
<th>DB2TAPEMGR.HEADER</th>
</tr>
</thead>
<tbody>
<tr>
<td>File marker</td>
</tr>
<tr>
<td>NODE0000/C0000000/S0000010.LOG</td>
</tr>
<tr>
<td>File marker</td>
</tr>
<tr>
<td>NODE0000/C0000000/S0000011.LOG</td>
</tr>
<tr>
<td>File marker</td>
</tr>
<tr>
<td>NODE0000/C0000000/S0000012.LOG</td>
</tr>
<tr>
<td>File marker</td>
</tr>
<tr>
<td>NODE0000/db2rhist.asc (the history file)</td>
</tr>
<tr>
<td>File marker</td>
</tr>
<tr>
<td>File marker (end of tape marker)</td>
</tr>
</tbody>
</table>
The following is an example content of the tape header file `DB2TAPEMGR.HEADER`:

```
label        :TAPE0
hostname     :PFERD
instance     :DB2PRD
database     :PRD
partition    :NODE0000
db version   :8.1.7.440
first used   :20040809183742
last modified:20040810173833
usage count  :18
contents     :
  0  DB2TAPEMGR.HEADER
  1  NODE0000\C0000000\S0000029.LOG
  2  NODE0000\C0000000\S0000030.LOG
  3  NODE0000\C0000000\S0000031.LOG
  4  NODE0000\C0000000\S0000032.LOG
  5  NODE0000\db2rhist.asc
```

The following table describes the content of the tape header file in detail:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label:</td>
<td>tape label</td>
<td>Tape label that was specified at the store operation</td>
</tr>
<tr>
<td>hostname:</td>
<td>host name</td>
<td>Host name of computer, where tape was created</td>
</tr>
<tr>
<td>instance:</td>
<td>instance name</td>
<td>Instance name</td>
</tr>
<tr>
<td>database:</td>
<td>database name</td>
<td>Database name</td>
</tr>
<tr>
<td>partition:</td>
<td>partition number</td>
<td>Specifies the partition, where tape has been created</td>
</tr>
<tr>
<td>db version:</td>
<td>database version</td>
<td>Database version</td>
</tr>
<tr>
<td>first used:</td>
<td>14 digit timestamp</td>
<td>Timestamp when the DB2 tape manager wrote the first time to this tape</td>
</tr>
<tr>
<td>last modified:</td>
<td>14 digit timestamp</td>
<td>Timestamp when the DB2 tape manager wrote the last time to this tape</td>
</tr>
<tr>
<td>usage count:</td>
<td>amount</td>
<td>Indicates, how often the DB2 tape manager has written to this tape</td>
</tr>
<tr>
<td>contents:</td>
<td>table containing:</td>
<td>Shows the tape content. Allows fast access to special log files even if the</td>
</tr>
</tbody>
</table>
2.5.4 DB2 Tape Manager Command Line Interface

The DB2 tape manager provides a large set of different options. The following section provides an overview about the command syntax.

**Command Syntax**

```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASE</td>
<td>DB source-database-alias</td>
</tr>
<tr>
<td>ON DBPARTITIONNUM n</td>
<td>Specifies the database partition number to work on. If no value is specified, DB2NODE is used. If DB2NODE is not set, 0 is the default value.</td>
</tr>
<tr>
<td>STORE ON tape-device</td>
<td>Log file is archived to tape and deleted from disk afterwards.</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DOUBLE STORE ON tape-device</strong></td>
<td>All log files, which had been archived only once, and those log files that had never been archived, are stored on tape. Only those log files that were stored twice to tape are deleted; others are kept on disk.</td>
</tr>
</tbody>
</table>
| **ON tape-device**             | Needs to be the name of a **non-rewind** tape-device. The tape-device names depend on the operating system platform as follows:  
  - **Windows:** `\\.\TAPE0`  
  - **AIX:** `/dev/rmt0.1`  
  - **HP-UX:** `/dev/rmt/0mn`  
  - **Solaris:** `/dev/rmt/0n`  
  - **Linux:** `/dev/nst0` |
| **TAPE LABEL tape-label**      | Specifies a label to be applied to the tape. If tape label is not specified, one will be generated automatically. For more information, see Tape Labeling [Page 23]. |
| **ALL LOGS**                   | Specifies that the command applies to all log files or a specified number of log files. |
| **n LOGS**                     | Specifies the block size for tape access. The default size is **5120**, and it must be a multiple of 512. The minimum is **512**. |
| **FORCE**                      | If the DB2 tape manager rejects writing to tape (because checks to avoid accidental overwrites of tapes failed) you can use the **FORCE** option to overwrite these checks. |
| **USING blocksize**            | Specifies that the tape is to be ejected after the operation completes. |
| **RETRIEVE [for rollforward clause]** | With this option the DB2 tape manager determines in the history file which tapes are required for a database recovery. You will be asked to insert the required tapes. |
| **RETRIEVE (ALL LOGS | LOGS n to m)** | With this option, the DB2 tape manager does not access the information in the history file. You can retrieve all or only some logs with this option even if you not have an up-to-date history file. |
| **RETRIEVE HISTORY FILE**      | With this option you can retrieve the history file from the tape which archived during the **STORE** operation. |
### 2.5.5 Archiving Log Files to Tape Using STORE and DOUBLE STORE

The DB2 tape manager supports the following two archiving operations when archiving log files to tape:

- **STORE**
  
  The **STORE** operation copies the log file to tape and then deletes the log files from disk.

- **DOUBLE STORE**
  
  The **DOUBLE STORE** operation copies log files to tape and then deletes only those log files, that were copied to tape the second time.

Since tapes are relatively unreliable (for example, when used too often they cannot be read anymore), we recommend that you use the **DOUBLE STORE** operation to avoid loss of data.

The following is an example for a **STORE** operation:

```
  db2tapemgr DB PRD STORE ON \..\TAPE0
```

The following is an example for a **DOUBLE STORE** operation:

```
  db2tapemgr DB PRD DOUBLE STORE ON \..\TAPE0
```
The DB2 tape manager performs the following steps during archiving operation:

1. If no log files were found for store, it stops the operation with the information “DBT2016I  No log files found for processing.”.

2. Reads the tape header file DB2TAPEMGR.HEADER.

3. Validates if writing to tape is acceptable. For more information, see Security Features [Page 32].

4. If the tape already contains log files, the history file is updated. Any entries in the history file for log files, that are related to the tape you are writing to are marked with a minus sign '-' in the Location field.

5. Writes a new tape header file DB2TAPEMGR.HEADER to the tape.

6. Copies the log files to tape. If the log files do not fit on the tape, the DB2 tape manager automatically reduces the number of log files to be stored on tape and starts writing to the tape again from the beginning. To avoid this time-consuming operation, we recommend that you limit the number of files written during an archiving operation by using the \texttt{n LOGS} option.

7. Updates the history file. The entries in the history file for the log files show the location on the tape.
   - For a \texttt{STORE} operation the history file entries are updated as follows:
     - The device type is changed to \texttt{T} and the location field is updated to contain the position on the tape in the following format: 
       \texttt{<tape label>:<pos>:<relative path>}
   - For a \texttt{DOUBLE STORE} operation the history file entries are updated as follows:
     - If the log file had not been stored to a tape before, the comment field in the history file is updated to contain the position on the tape in the following format: 
       \texttt{<tape label>:<pos>}
     - If the log file had already been stored on another tape, the device type is updated to \texttt{T} and the location field is updated to contain the position on the tape in the following format: 
       \texttt{<tape label>:<pos>:<relative path>}

The DB2 tape manager also updates the history field end time. The end time field is set to the time when the tape header file was created.

8. Deletes log files from disk
   - \texttt{STORE} operation: Deletes all archived log files
   - \texttt{DOUBLE STORE}: Deletes only those log files that were stored twice now

9. Writes the history file to the tape. If the history file does not fit on the tape, only a warning is displayed. You still have to keep the tape.

10. If you have specified the \texttt{EJECT} option, the tape is ejected from the tape drive.
The following table describes the different log file history entries that you might find in the history file when using the DB2 tape manager:

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Location</th>
<th>Comment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>&lt;path to log file on disk&gt;</td>
<td></td>
<td>The log file is still located on disk. The DB2 tape manager has not been called for that log file so far.</td>
</tr>
<tr>
<td>D</td>
<td>&lt;path to log file on disk&gt;</td>
<td>&lt;tape-label&gt;:&lt;pos&gt;</td>
<td>The log file is still located on disk, but has been stored to tape with a DOUBLE STORE operation.</td>
</tr>
<tr>
<td>T</td>
<td>&lt;tape-label&gt;:&lt;pos&gt;:NODExxxx x/Cyyyyyyyy/Szzzzzz.LOG</td>
<td></td>
<td>This log file is only located on tape.</td>
</tr>
<tr>
<td>T</td>
<td>&lt;tape-label&gt;:&lt;pos&gt;:NODExxxx x/Cyyyyyyyy/Szzzzzz.LOG</td>
<td>&lt;tape-label&gt;:&lt;pos&gt;</td>
<td>The log file is located on two tapes, after the second DOUBLE STORE operation for this log file.</td>
</tr>
</tbody>
</table>

If you find a minus sign "-" in the Location or Comment field, it indicates that either a DELETE TAPE LABEL operation was performed or a tape, that contains log files, was overwritten with new log files.

2.5.7 Retrieving Log Files from Tape

You can easily retrieve log files from tape by using the RETRIEVE FOR ROLLFORWARD TO option. You will be asked to insert any required tape for a database recovery.

The following example output describes the retrieve operation of the DB2 tape manager:

```
db2tapemgr DB PRD retrieve for rollforward to end of logs from \\TAPE0
DBT2065I Using database partition "NODE0000".
Scanning history.
Scanning history.
Required tapes ":".
TESTTAPE
Insert tape "TESTTAPE".
Press '9' to quit or any other key to continue.
Rewinding tape.
Reading tape header.
Retrieving log files from tape "TESTTAPE".
Reading log file "NODE0000\C0000000\S00000000.LOG" from tape.
Reading log file "NODE0000\C0000000\S00000001.LOG" from tape.
Reading log file "NODE0000\C0000000\S00000002.LOG" from tape.
Reading log file "NODE0000\C0000000\S00000003.LOG" from tape.
Reading log file "NODE0000\C0000000\S00000004.LOG" from tape.
```
Positioning tape.
Reading log file "NODE0000\C0000001\S0000005.LOG" from tape.
Reading log file "NODE0000\C0000001\S0000006.LOG" from tape.
Positioning tape.
Reading log file "NODE0000\C0000003\S0000007.LOG" from tape.
Reading log file "NODE0000\C0000003\S0000008.LOG" from tape.
Reading log file "NODE0000\C0000003\S0000009.LOG" from tape.
DBT2006I db2tapemgr completed successfully.

If you do not have a current version of the history file and you know on which tape you find the required log files, you can use the following options to retrieve log files from tape:

- RETRIEVE ALL LOGS
- RETRIEVE LOGS n TO m

In case of a disaster recovery and if you do not know where the log file are, you can use the DB2 tape manager to retrieve the history file from the tape that was stored at the end of the tape.

Always use the latest tape and the following command to retrieve the history file from tape:

db2tapemgr RETRIEVE HISTORY FILE FROM \\.\TAPE0 TO c: \temp

With this history file you can use the QUERY ... USING HISTORY FILE <hist-file> option to find the required tapes and the RETRIEVE ... USING HISTORY FILE <hist-file> to retrieve the required log files.

To find the required tapes, enter the following command:

db2tapemgr QUERY USING HISTORY FILE c:\temp\NODE0000\db2rhist.asc

To retrieve the required log files, enter the following command:

db2tapemgr RETRIEVE USING HISTORY FILE c:\temp\NODE0000\db2rhist.asc

If you do not specify a destination path using the TO <directory> option, the log files are retrieved to the overflow log path directory, which can be set in the database configuration.

The log files are created in a hierarchical manner, for example, <dir>/NODExxxx/Cyyyyyy/Szzzzzzz.LOG.

If the destination directory already contains the node directory NODExxxx, the log files are restored as <dir>/Cyyyyyy/Szzzzzzz.LOG.

Since the recover command searches the log files by default in the overflow log path directory, you should set the OVERFLOWLOGPATH parameter in the database configuration. This simplifies calls to the DB2 tape manager and the call for the recover command.

To avoid duplicate retrieval of log files from tape, the DB2 tape manager does not retrieve log files that it already finds in the destination path. In this case a warning is displayed.
2.5.8 Other Operations

You can also perform the following operations using the DB2 tape manager:

- **EJECT TAPE**
- **SHOW TAPE HEADER**
- **DELETE TAPE LABEL**

**EJECT TAPE**

If you want to eject the tape from the tape drive, enter the following command:

```
db2tapemgr EJECT TAPE /dev/rmt0.1
```

**SHOW TAPE HEADER**

This option displays the contents of the tape header file. You can use this information to check:

- Which tape is inserted in the tape drive.
- How often the tape has been used to archive log files
- Which log files are on the tape in case you lost the history file.

The following is an example output when the **SHOW TAPE HEADER** operation is performed:

```
db2tapemgr SHOW TAPE HEADER \./TAPE0

DBT2062I  Working on database "PRD".
DBT2065I  Using database partition "NODE0000".
Rewinding tape.
Reading tape header.
Tape header contents
label :TAPE0
hostname :PFERD
instance :DB2PRD
database :PRD
partition :NODE0000
db version :8.1.7.440
first used :20040809183742
last modified:20040810173833
usage count :18
contents :
0 DB2TAPEMGR.HEADER
1 NODE0000\C0000000\S00000029.LOG
2 NODE0000\C0000000\S00000030.LOG
3 NODE0000\C0000000\S00000031.LOG
4 NODE0000\C0000000\S00000032.LOG
5 NODE0000\db2rhist.asc

DBT2006I  db2tapemgr completed successfully.
```

**DELETE TAPE LABEL**

This option removes the location information from the log file entries in the history file that are related to the specified tape. If you have lost a tape or a tape is corrupt, you should use this option because this should be reflected in the history file.

To delete tape labels, enter the following command:

```
db2tapemgr DELETE TAPE LABEL TAPE0
```
2.5.9 Security Features

To prevent log files from being overwritten on the tape, the DB2 tape manager performs the following security checks:

- If the tape was used for archiving log files before by verifying that a `DB2TAPEMGR.HEADER` is on the tape. This avoids overwriting tapes with other contents (for example, backup tapes). The other way around is not safe. For example, the `db2 backup` command would overwrite tapes containing log files.

- The tape header contents is checked to make sure that:
  - The tape content has not expired. To achieve this, the DB2 tape manager calculates the difference between the `last modified` field from the tape header file with the current time and compares the difference against the value which was specified by the DB2 registry variable `DB2_TAPEMGR_TAPE_EXPIRATION`.
  - Log files are not overwritten by log files of another instance, database or partition.
  - The same tape label is not used multiple times for different tapes.

You can use the `FORCE` option to override these checks.

Before log files are retrieved from tape similar checks are performed. The DB2 tape manager checks if the tape header information about instance, database and partition fits. If discrepancies are found, the DB2 tape manager asks you if you want to proceed with the log file retrieval.

In some cases, for example, if you want to retrieve log files for a rollforward after a redirected restore, the check will fail, but you need to continue.

2.5.10 Troubleshooting

This section provides information about how to proceed if problems occur with the DB2 tape manager.

Be aware that the information in this section is not complete but will help you to resolve problems in some cases.

Tracing

To get detailed information about the cause of a problem, you have to create a trace file of a DB2 tape manager run. This trace file is essential for support purposes.

By adding the `TRACE` option to the `db2tapemgr` command, trace information is written to standard output and you can redirect it to a file.

The following command is an example of how to create a trace file:

```
db2tapemgr DB PRD STORE ON \\TAPE0 TRACE > db2tapemgr.trc
```
You can also use the standard DB2 tracing function as follows:

1. To activate the DB2 tracing function, enter the following command:

   ```
   db2trc on -f <filename>
   ```

   To limit the impact on the system performance, you can limit the traced DB2 components, when activating the trace. The component number of the DB2 tape manager is 143. For example, enter the following command:

   ```
   db2trc on -m "**.*.143.*.*" -f <filename>
   ```

2. Run the `db2tapemgr` command that failed. For example, enter the following command:

   ```
   db2tapemgr DB PRD STORE ON \\TAPE0
   ```

3. To deactivate the DB2 tracing function, enter the following command:

   ```
   db2trc off
   ```

### Problems with the Tape Drive

<table>
<thead>
<tr>
<th>Problem Description</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>On some Linux distributions, such as SuSE, not all users are allowed to read and write to tape devices.</td>
<td>Make sure that the user who calls the DB2 tape manager has sufficient authorizations to perform tape operations.</td>
</tr>
<tr>
<td>The tape drive that you are using, does not support the default block size of 5120 bytes.</td>
<td>Specify the block size using the <code>BLOCKSIZE</code> parameter.</td>
</tr>
<tr>
<td>You have problems accessing the tape with the DB2 tape manager.</td>
<td>To check if the tape drive is working correctly, use the UNIX tools <code>dd</code>, <code>cpio</code> and <code>mt</code>.</td>
</tr>
</tbody>
</table>

### Tape Loss

If you lost a tape or a tape became unreadable, use the `DELETE TAPE` option of the DB2 tape manager to remove the history entries of the log files from the history file to reflect that these log files do not exist on the tape anymore. In addition, you should check if your database is still recoverable. If not, you should perform a database backup immediately.

### Inconsistencies of the History File

If the history file is corrupt or not up-to-date and you want to use the interactive retrieval of log files using the DB2 tape manager, proceed as follows

1. Retrieve the history file from the latest tape. For example, enter the following command:

   ```
   db2tapemgr RETRIEVE HISTORY FILE FROM /dev/rmt0.1 TO /tmp
   ```

2. Query the required tapes for a database rollforward. For example, enter the following command:

   ```
   db2tapegmr QUERY USING HISTORY FILE /tmp/NODE0000/db2rhist.asc
   ```

3. Retrieve the required log files from tapes. For example, enter the following command:

   ```
   db2tapemgr RETRIEVE USING HISTORY FILE /tmp/NODE0000/db2rhist.asc FROM /dev/rmt0.1
   ```

If you know on which tape the required log files are located, you can also use the DB2 tape manager options `RETRIEVE LOGS n TO m` or `RETRIEVE ALL LOGS` to retrieve the required log files from tape.
Log Files to Be Archived Not Found

If you lose your history file, for example, because of a disk failure but there are still log files on the disk that need to be archived to tape, you cannot use the DB2 tape manager to archive these log files.

The DB2 tape manager cannot find these log files because the history files do not contain any information about them. The only way to archive these log files to tape is to use the UNIX tools cpio or tar. However, you have to remember later on that you archived these log files with UNIX tools and not using the DB2 tape manager.

Tape Header File Becomes Unreadable

If the tape header file cannot be read, the DB2 tape manager will not be able to retrieve log files from tape. Therefore, use the UNIX tool cpio to retrieve log files from tape. For example, enter the following command:

```
dd if=/dev/rmt0.1 bs=5120 | cpio –iduvB
```

As the cpio, dd and cpio commands are not available on Windows, this was tested with the Cygwin tool set for Windows. You can download the tool set at the following Internet address: www.cygwin.com.

Error Messages

The following table describes error messages that are not intuitive and might be displayed when using the DB2 tape manager:

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Reason</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanning history failed. Reason: &quot;SQL0_NLCK_PATH_ERROR: Unknown&quot;.</td>
<td>The user who has called the DB2 tape manager does not have access to the history file db2rhist.asc.</td>
<td>Start the DB2 tape manager as instance owner.</td>
</tr>
<tr>
<td>Reading tape header failed. Reason: &quot;1106: When accessing a new tape of a multivolume partition, the current block size is incorrect.&quot;</td>
<td>Tape header cannot be read from tape because you use a new tape on Windows.</td>
<td>Use the STORE or DOUBLE STORE operation with the FORCE option to write to the tape the first time.</td>
</tr>
</tbody>
</table>
3 Migration

The following sections provide information about differences between the current SAP DB2 log file management solution and the new DB2 log file management that you should be aware of before starting the migration as well as on the actual migration process.

SAP DB2 Log File Management

The following graphic is an overview of the current SAP DB2 log file management including all participating components. The SAP DB2 log file management tools are described in detail in the SAP documentation *Database Administration Guide: SAP on IBM DB2 Universal Database for UNIX and Windows*. This documentation is available on SAP Service Marketplace at service.sap.com/instguidesNW04 → Operation.

3.1 Differences Between the Log File Management Concepts

In the following section the differences between the new DB2 log file management and the SAP DB2 log file management is described for the following areas:

- Multi-partition support
- Backend support
- Standby database support
- CCMS support

Multi-Partition Support

As of DB2 Version 8.2 and the new DB2 log file management, you can configure the DB2 log manager for each database partition. With the SAP DB2 log file management tools, there was only one possible configuration for all partitions.
If you want to use the DB2 tape manager for tape operations, you must be aware that you must call the DB2 tape manager for each partition and on the system where the partition resides. With the SAP DB2 log file management, `brarchive` was able to store log files from each partition with the option `-node ALL`.

**Backend Support**

- **Tape**
  - With the new DB2 log file management you must call the DB2 tape manager as instance owner `db2<dbsid>`. The DB2 tape manager does not work under user `<sapsid>adm`. The SAP DB2 log file management tool `brarchive` could be run under user `<sapsid>adm` and `db2<dbsid>`.
  - The `brarchive` option `-ssd` will no longer exist. Instead, you can use the DB2 tape manager `DOUBLE STORE` operation. For more information, see Archiving Log Files to Tape Using STORE or DOUBLE STORE [Page 27].
  - No concurrency check is performed when you use the DB2 tape manager. With the SAP DB2 log file management tool, `brarchive` prevented an invocation in parallel.

  To avoid loss of data, do **not** call the DB2 tape manager in parallel on the same tape device.

- **TSM**
  - With the SAP DB2 log file management solution you have two choices, that is indirect archiving (`brarchive`) and direct archiving. With the DB2 log file management this choice is no longer available. You can only **directly** archive log files using the DB2 log manager.
  - The SAP DB2 log file management uses a TSM `archive copy group` for storing log files by default. If no TSM `archive copy group` is available, but a TSM `backup copy group`, it will use the TSM `backup copy group`.

  The DB2 log file manager only uses the TSM `archive copy group` and will fail if no TSM `archive copy group` exists. Therefore, you should always make sure that you provide a TSM `archive copy group` before you archive log files using the DB2 tape manager.

- **Vendor**
  - Both, the DB2 log file management and the SAP DB2 log file management provide support for vendor libraries. However, the SAP DB2 log file management vendor libraries are not compatible with the DB2 log file management libraries. Contact your vendor, for a compatible library that you can use for the DB2 log file management.

**Standby Database Support**

DB2 provides **standby databases**. To keep standby databases up-to-date, you can ship log files from the source database to the standby database and apply the log files to the standby database.

In the SAP DB2 log file management, you set the `DB2DB6_STANDBY_DIR` configuration variable to create an extra copy of the log files, which can be moved to the standby system.

As of DB2 UDB for UNIX and Windows Version 8.2, the high availability data replication (HADR) function is introduced, which provides log record shipping. If you still want to use log file shipping over a disk location, you can configure two archiving locations and use one of them as the standby directory.
For more information about HADR, see the DB2 documentation.

**CCMS Support**

As of SAP Netweaver ’05 or higher, only the DB2 log file management will be supported. If you intend to use DB2 log file management with previous SAP system releases, you have to take the following constraints into consideration.

**Transaction DB12:**

- **SAP R/3 Release 4.5B and below:**
  You can no longer display log file information in CCMS. We recommend that you use the following DB2 command to display the log file information:
  
  `db2 list history archive log all for database <dbsid>`

- **SAP R/3 Release 4.6B, 4.6C and SAP Web AS 6.10:**
  If a specific SAP Support Package is applied, you can display log file information. For more information about the required Support Package level, see SAP Note 300828.
  In addition, the latest version of program `dmdb6rdi` is required. For the latest version of program `dmdb6rdi`, see SAP Note 768817.

- **SAP Web AS 6.20 and higher:**
  If the latest version of program `dmdb6rdi` is applied, you can display log file information in CCMS. For the latest version of program `dmdb6rdi`, see SAP Note 768817.

- **SAP R/3 Releases 4.6B, 4.6C, SAP Web AS 6.10 and higher:**
  On the Backup Overview screen, the `brarchive` tab becomes obsolete when using the new DB2 log file management. When choosing the User Exit tab, information about all log files is displayed no matter where the log files are stored. In the Target column of the table on the User Exit tab, detailed information about the log file location is displayed with the following syntax:

  `<Media>,<Type>,<Status>[:<Tape Name>|<TSM management class>]`

  For example, target string `TSM,1,A:MCLASS_LOG` means that the log file was archived with log archiving method 1 (LOGARCHMETH1) to TSM and there to TSM management class MCLASS_LOG. The status of the log files is *active*.
If log files were stored to tape, the tape name is displayed after the colon. If log files were stored to disk, only the media name and the log archive method are displayed, for example DISK, 2, A. The reason is a length limitation of 20 characters in the Target column.

Possible values for <Type> are:
- 1 (LOGARCHMETH1)
- 2 (LOGARCHMETH2)
- F (FAILARCHPATH)

For more information about the database configuration parameters LOGARCHMETH1, LOGARCHMETH2 and FAILARCHPATH, see Configuration [Page 10].

Transaction DB13:
With the new DB2 log file management, you can no longer use the jobs Archive inactive log files into ADSM, Archive inactive log files onto device and Initialize Tape because these jobs are based on program brarchive (which is part if the SAP DB2 log file management tools).

### 3.2 Performing the Migration

The following procedure describes the steps you have to perform if you are currently using the SAP DB2 log file management and your database instance is now running with DB2 Version 8.2.

After the installation database software of DB2 Version 8.2 the new DB2 log file management is not yet active. The database configuration variable LOGARCHMETH1 is set to USEREXIT and the db2uext2 executable of the SAP DB2 log file management is still used.

**Procedure**

1. To be able to display log file information in CCMS when the DB2 log file management is enabled, perform the following steps:
   a. If scheduled in transaction DB13, delete the jobs Archive inactive log files into ADSM, Archive inactive log files onto device and Initialize Tape.
   b. If scheduled in transaction SM37, delete the job that executes the program sddb6mir.
   c. For SAP system Releases 4.6B, 4.6C and SAP Web AS 6.10:
      Apply the correct SAP Support Package level according to SAP Note 300828.
   d. For SAP system Release 4.6B and higher:
      Install the latest version of program dmdb6rdi, which is the interface between DB2 and CCMS, with the correct patch level according to SAP Note 768817.

2. To enable the DB2 log file management, adjust the log file management database configuration variables as described in DB2 Log Manager Backend Support [Page 13].
3. Perform an offline backup of your database.

4. If you have used indirect archiving so far, use `brarchive` to store the remaining log files, which were copied from the transaction log directory to the log archive directory with the SAP DB2 log file management user exit.

5. Deinstall the SAP DB2 log file management tools.

   To do so, enter the following command:

   ```
   sddb6ins -d
   ```

   The relevant executables, the `init<DBSID>.db6` configuration file and the Admin DB will be deinstalled. For more information, see the SAP documentation *Database Administration Guide: SAP on IBM DB2 Universal Database for UNIX and Windows*. You can find this documentation on SAP Service Marketplace at `service.sap.com/instguidesnw04 → Operation`.

   **For SAP system Releases 4.0B and 4.5B only:**

   To avoid ABAP short dumps in SAP transaction DB12, you must recreate some tables in the SAP database, which were dropped when using the command `sddb6ins -d`.

   You can recreate the tables using the option `sddb6ins -im`. Option `-im` is a new option that was implemented with Admin Tools Patch 29. For more information, see `SAP Note 768809`.

6. If you have installed the SAP DB2 Control Center Extensions, you can remove them as follows:

   - **Windows:**
     
     To deinstall the DB2 Control Center Extensions, remove the `db2plug.zip` file from the following DB2 directories:
     
     - `DB2 UDB for UNIX and Windows Version 8:`
       
       `<DB2 install directory>/sqllib/tools`
     
     - `DB2 UDB for UNIX and Windows Version 7:`
       
       `<DB2 install directory>/sqllib/cc`

   - **UNIX:**
     
     To deinstall the DB2 Control Center Extensions, remove the `db2plug.zip` file from the following DB2 directories:
     
     - `DB2 UDB for UNIX and Windows Version 8:`
       
       `<DB2 install directory>/sqllib/tools`
     
     - `DB2 UDB for UNIX and Windows Version 7:`
       
       `<DB2 install directory>/sqllib/cc`