HPSS Conversion Guide

High Performance Storage System, version 9.3.0.0.0, 13 December 2021

HPSS Conversion Guide

High Performance Storage System, version 9.3.0.0.0, 13 December 2021

Table of Contents

List of Tables

About this Book. The HPSS Conversion Guide is for use when upgrading HPSS. It will guide system administrators through the planning and execution of upgrading an HPSS system from their current version to their target version.

Chapter 1 provides a process for converting HPSS from a big-endian to little-endian system.

Chapter 2 gives HPSS-specific instructions on upgrading DB2 and Linux

Chapter 3 provides troubleshooting procedures for problems administrators may encounter.

In addition, a glossary of terminology and abbreviations is provided.

Each conversion process is broken up into three sections:

1. The first section provides general information about the conversion process.

2. The second section provides information about which metadata changes are required and guidance.

3. The third section provides the conversion steps.

Conventions Used in This Book. Example commands that should be typed at a command line will be proceeded by a percent sign ("%") if the command may be executed as a user other than the local superuser root and be presented in a courier font:

```
% sample command
```

or with a pound sign ("#") if the command must be executed as the local superuser **root**:

```
# sample command
```

Example command output and example contents of ASCII files will be presented in a courier font without a prompt:

```
sample file line 1

sample file line 2
```

In contexts other than providing command execution instruction, any text preceded by a pound sign ("#") should be considered a comment:

```
# This is a comment
```

Chapter 1. Conversion Overview

1.1. Introduction

This conversion guide provides background and the necessary upgrade steps to upgrade HPSS from any *supported* version up to the latest HPSS version. The functionality to perform each of the metadata upgrades/conversions is included in the target version of HPSS.

As HPSS software versions become unsupported, they will be removed from this guide so long as they are not required for any supported conversion path.

In order to upgrade from a version several releases back, it is **necessary** to perform each metadata upgrade/conversion one level at a time. For example, to upgrade from 7.5.3 to 9.1, one **must** execute the metadata upgrade/conversions from:

*   7.5.3 to 8.3

*   8.3 to 9.1

This ensures that metadata changes are applied in the proper order, .

1.2. Contents of This Guide

This guide provides general guidance on conversions. For version specific guidance, refer to the conversion guidance from HPSS Support.

1.3. Preparation

The following steps are recommended prior to performing any HPSS upgrade:

1.3.1. Gather Data On Existing System

Gather information on the existing system using the **lshpss**, **dump_sspvs**, and **lsdb2** tools, and save the results. These can be used later to validate the conversion.

```
% lshpss -all
% dump_sspvs -i#        (for each subsystem)
% dump_sspvs -i# -d     (for each subsystem)
% lsdb2              (run as hpssdb)
```

1.3.2. Benchmark Existing System

If desired, it may be beneficial to benchmark the existing system prior to upgrading so that it may be compared with the performance of the upgraded system. This allows a site to determine whether any performance degradation has taken place and provides support representatives with information to help them determine the cause.

1.3.3. Backup Existing System

Back up the existing database tables using a full backup.

```
% db2 backup db hcfg to <backup_dir>
% db2 backup db hsubsys1 to <backup_dir>
```

Note: All subsystem databases should be backed up if multiple subsystems are configured.

Now **cd** to the backup dir and run:

```
% db2ckbkpd
```

All images should be successfully verified.

1.4. hpss_managetables

This program is used by **mkhpss** to create database tables and indexes when new systems are created. It is designed to be operated by both **mkhpss** and by an HPSS administrator performing conversions. It is used in a number of conversions within this guide.

It is vitally important to understand that this program, when used from a command line by an administrator, has the ability to destroy an HPSS system. It can create and delete any database table in an HPSS system. It can create and delete any index, view or referential integrity constraint on any database table. It should also be understood that this program alone is the source of the HPSS database schema. Therefore, it will be necessary to use this program from time to time as changes to the schema are required.

hpss_managetables is a keyboard interactive program that operates in a "manual commit" mode. This means that database transactions are committed manually from the keyboard, not automatically. The administrator uses the program to make a connection to the database, then issues commands that make changes to the database. When the connection is made, the program creates an empty database transaction. When any command is issued that makes a change to the database, the change is contained in the transaction and remains uncommitted. Additional commands can be issued which are added to the transaction. The transaction remains uncommitted until a "commit" command is entered at which time the changes are committed to the database. If an "abort" command is entered instead, the changes are rolled back and no net change takes place. If an uncommitted transaction exists when the "quit" command is entered, the program reports the existence of the transaction and will not quit.

When performing conversions, even through multiple releases, use the **hpss_managetables** for the final target release. For example, if upgrading from X.1 to Y.1, use the **hpss_managetables** for the target version (e.g. Y.1). By doing this you'll ensure that the metadata updates you are making are as current as possible.

If you follow the instructions carefully and refrain from experimenting with **hpss_managetables**, you won't do any damage to your system. If something unexpected happens, use the "abort" command and "quit" and seek advice from HPSS support.

hpss_managetables must be run by a user ID that has DBADM authority on the databases. We recommend running as "hpssdb". The program can be found in the HPSS `bin` directory.

hpss_managetables recognizes two environment variables, **HPSS_MM_SCHEMA_NAME** and **HPSS_GRP_NAME_SERVER**. Both of these variables have appropriate defaults set by HPSS. **HPSS_MM_SCHEMA_NAME** is set to "HPSS" and **HPSS_GRP_NAME_SERVER** is set to "hpsssrvr". If for some reason these values don't apply to your system, adjust them in the environment in which **hpss_managetables** will be run, or in the HPSS `env.conf` file before starting the program.

More information on **hpss_managetables** can be found in the **hpss_managetables** man page (`man/hpss_managetables.7`).

Chapter 2. Endian Conversion

This chapter summarizes HPSS metadata conversion from big-endian to little-endian. HPSS currently does not support little-endian to big-endian conversion.

2.1. General Information

The conversion from big-endian to little-endian manipulates a number of metadata field which store data in a endian-dependent manner. The process is automated through the use of provided conversion scripts. In general, two types of fields must be converted, UUIDs and SOIDs:

2.1.1. UUIDs

For UUID types the following byte order swapping is necessary:

- Bytes 1-4 are reversed (4-3-2-1)

- Bytes 5-6 are reversed (6-5)

- Bytes 7-8 are reversed (8-7)

- Bytes 9-16 are not altered

```
struct hpss_uuid {
    unsigned32       time_low;
    unsigned16       time_mid;
    unsigned16       time_hi_and_version;
    unsigned8        clock_seq_hi_and_reserved;
    unsigned8        clock_seq_low;
    byte          node[6];
};
```

2.1.2. SOIDs

For SOID types the following byte order swapping is necessary:

- Bytes 1-4 are reversed (4-3-2-1)

- Bytes 5-6 are reversed (6-5)

- Bytes 7-8 are reversed (8-7)

- Bytes 9-16 are not altered

- Bytes 17-20 are reversed (20-19-18-17)

- Bytes 21-22 are reversed (22-21)

- Bytes 23-24 are reversed (24-23)

- Bytes 25-32 are not altered

```
struct hpssoid_s {
      hpss_uuid_t    ObjectID;           /* unique object ID        */
      unsigned32     ServerDep1;          /* Piece of server uuid     */
      unsigned16     ServerDep2;
      unsigned16     ServerDep3;
      byte           ServerDep4;         /* Piece of server uuid     */
      byte           ServerDep5;         /* Piece of server uuid     */
      byte           SecurityLevel[2];   /* security info (0's)      */
      byte           Reserved[2];        /* Reserved[0] used by BFS in COS */
                                         /* change processing. Reserved 1 */
                                         /* should be 0.    */
      byte           SubType;             /* Disk/Tape object indicator */
      byte           Type;                /* type of object identified  */
};
```

2.1.3. Conversion Components

The conversion requires the execution of four scripts:

1. **hpss.hcfg.export.db2** - This script performs the necessary byte swaps for metadata stored in the HPSS configuration database. This script requires a connection to the source HPSS configuration database (big-endian).

2. **hpss.hsubsys.export.db2** - This script perform the necessary byte swaps for metadata stored in an HPSS subsystem database. It must be executed once for each subsystem database. This script requires a connection to the source HPSS subsystem database (big-endian).

3. **hpss.hcfg.load.db2** - This script loads the data generated by the **hpss.hcfg.export.db2** script. This script requires a connection to the target HPSS configuration database (little-endian).

4. **hpss.hsubsys.load.db2** - This script loads the data generated by the **hpss.hsubsys.export.db2** script. This script requires a connection to the target HPSS subsystem database (little-endian), and must be executed once for each subsystem.

2.2. Metadata Changes

This section describes what metadata will be changed in order to convert from big-endian to little-endian.

2.2.1. Metadata Change Detail

2.2.1.1. Configuration Tables

Table 2.1. Configuration Tables

Table Name	Change
AUTHZACL	Convert SERVER_ID, INTERFACE_ID.

Table Name	Change
CARTRIDGE	Convert PVR_ID.
CORE	Convert SERVER_ID.
DMG	Convert SERVER_ID.
GATEKEEPER	Convert GKID.
GLOBAL	Convert ROOT_CORE_SERVER_ID.
LOGCLIENT	Convert LOGC_SERVER_ID.
LOGDAEMON	Convert LOGD_SERVER_ID.
LSPOLICY	Convert HPSS_ID.
MOVERDEVICE	Convert MVR_ID.
MOVER	Convert MVR_ID.
MPS	Convert MPSID.
NFS	Convert NFS_ID.
NSGLOBALFILESET	Convert CORE_SERVER_UUID, GATEWAY_UUID.
PVLACTIVITY	Convert CLIENT.
PVLDRIVE	Convert MVR_ID, PVR.
PVLPV	Convert ALLOC_CLIENT_ID.
PVR	Convert PVR_ID.
SERVERINTERFACES	Convert INTERFACE_ID, SERVER_ID.
SERVER	Convert SERVER_ID.
SITE	Convert HPSS_ID.
STORAGESUBSYS	Convert GATEKEEPER_ID.

2.2.1.2. Subsystem Tables

Table 2.2. Subsystem Tables

Table Name	Change
BFCOSCHANGE	Convert BFID field.
BFDISKALLOCREC	Convert BFID and SSEG_ID fields.
BFMIGRREC	Convert BFID.
BFPURGEREC	Convert BFID.
BFSSEGCHKPT	Convert BFID, SSEG_ID.
BFTAPESEG	Convert BFID, SSEG_ID.
BITFILE	Convert BFID.
DISKSEGUNLINK	Convert SSEGID.
DMGFILESET	Convert MOUNT_PT_CORE_SERVER_ID, MOUNT_PT_HANDLE_CORE_UUID,

Table Name	Change
	RPC_ENDPOINT_INTERFACE_ID, SECURITY_SERVER_ID, SERVER_ID.
NSFILESETATTR	Convert FILESET_HANDLE_CORE_UUID.
NSOBJECT	Convert BITFILE_ID.
NSOBJECT	Convert SUB_TREE_HANDLE_CORE_UUID.
SSPVDISK	Convert VVID.
SSPVTAPE	Convert VVID.
STORAGEMAPDISK	Convert VVID.
STORAGEMAPTAPE	Convert CURRENT_SSID, VVID.
STORAGESEGAUX	Convert SSID.
STORAGESEGDISKEXTENTS	Convert SSID, VVID.
STORAGESEGDISK	Convert SSID, VVID.
STORAGESEGTAPEABSADDR	Convert SSID.
STORAGESEGTAPE	Convert SSID, VVID.
TAPESEGUNLINK	Convert SSEGID.
VVDISK	Convert VVID.
VVTAPEABSADDR	Convert VVID.
VVTAPE	Convert VVID.

2.3. Introduction

The following section will guide you through the process of converting the system from big-endian to little-endian. This process should only be attempted after discussions with IBM support. Though the process is fairly straightforward and automated, it is recommended that IBM support guide administrators through the process as mistakes can potentially be extremely dangerous and difficult to recover from.

2.3.1. Prepare Environment

- Log into the system as the HPSS DB2 instance owner.

- Create the following directory structure on the source system (big-endian) to hold the exported/ converted metadata. This should be done on a file system with sufficient space. It will take approximately 1,200 bytes of storage per bitfile in the HPSS subsystem. This number will vary depending on HPSS configuration. Things like dual tape copies will require more space. This new directories should either be owned by the DB2 instance owner, or the instance owner should have sufficient privileges to create and write metadata export files.

  - `<path>/endian_conversion`

  - `<path>/endian_conversion/hcfg`

- For each subsystem:

    - `<path>/endian_conversion/<subsystem database name>`

- The target system will require a similar directory structure as was created on the source system.

- Follow the general guidelines for converting a system - shut down HPSS and take a full offline backup.

2.3.2. Export HCFG

- Execute the following statements:

```
% cd into <path>/endian_conversion/hcfg
% db2 connect to <HPSS configuration database name>
% db2 -tvf <path to export
scripts>/hpss.hcfg.export.db2 >
hpss.hcfg.export.db2.log
% db2 connect reset
```

Validate the number of rows exported matches the source.

2.3.3. Export Subsystems

- For each subsystem, execute the following statements:

```
% cd into <path>/endian_conversion/<subsystem database name>
% db2 connect to <subsystem database name>
 % db2 -tvf <path to export
scripts>/hpss.hsubsys.export.db2 >
hpss.hsubsys.export.db2.log
% db2 connect reset
```

Validate the number of rows exported matches the source table.

- Transfer .ixf files to the target system.

    - It is recommended that the ixf files be checksummed on source and destination machines to validate their integrity after the transfer.

2.3.4. Import HCFG

- Log in as the DB2 instance owner on the target system.

- Execute the following commands:

```
% cd into <path>/endian_conversion/hcfg
% db2 connect to <HPSS configuration database name>
% db2 -tvf <path to load scripts>/ hpss.hcfg.load.db2 >
hpss.hcfg.load.db2.log
% db2 connect reset
```

Validate the number of rows added to each table matches the source.

Perform an offline backup of the HPSS configuration database.

2.3.5. Import Subsystems

- For each subsystem, execute the following commands:

```
% cd <path>/endian_conversion/<HPSS subsystem database name>
% db2 connect to <HPSS subsystem database name>
% db2 -tvf <path to load scripts>/
hpss.hsubsys.load.db2 > hpss.hsubsys.load.db2.log
% db2 connect reset
```

Validate the number of rows added to each table matches the source.

Perform an offline backup of the HPSS subsystem database.

2.4. Security

When installing HPSS on a new system, **mkhpss**, the HPSS installer, will create the **hpss** user with the home directory as `/var/hpss`. Beginning in 9.2, it becomes critical that the home directory point to the location of the HPSS configuration files, `/var/hpss` by default. If either the configuration directory is moved or the HPSS user has a different home directory, HPSS will fail to start. If possible, ensure that the *hpss* user home directory matches the location of the HPSS config data.

There is also an alternate method to specify this configuration home, for those who may be on a global user directory such as NIS or ActiveDirectory, and whose user homes cannot be customized. A symbolic link can be created from `/etc/hpss` to the location of the HPSS configuration directory base. For example:

```
ln -s /var/hpss /etc/hpss
```

Check to make sure `/etc/pam.d/hpss` contains a reference to the shadow file. If not, delete `/etc/pam.d/hpss` and run `config/setup_pam.pm` to recreate it. The `config/setup_pam.pm` program does not overwrite existing files in the `/etc/pam.d` directory, because it can't create in-place backups— all files in the `/etc/pam.d` directory are significant to PAM and identify security methods. Thus, `setup_pam.pm` chooses the safe alternative of leaving existing files alone. So it is important to remove the existing file in order to rebuild it.

The RPM installs of HPSS place the binaries within subdirectories of `/hpss_src`, and setting a symlink from `/opt/hpss` to the latest RPM installation. In order for Perl modules within the config directory, including `setup_pam.pm`, to find their dependent modules, they find their own absolute path into the config subdirectory. Because of this, the PAM stack that `setup_pam.pm` sets up uses this absolute path for its module. This creates a requirement to rerun `setup_pam.pm` after every upgrade. However, if the HPSS_ROOT environment variable is set to `/opt/hpss`, and assuming that `/opt/hpss` is set to the correct installation, `setup_pam.pm` will create the PAM stack to refer to `/opt/hpss`, which makes it somewhat future-proof. Therefore, it is advisable to run the setup_pam.pm command as:

```
rm -f /etc/pam.d/hpss
HPSS_ROOT=/opt/hpss /opt/hpss/config/setup_pam.pm
```

Chapter 3. Upgrading DB2 and Linux

3.1. Overview

This chapter provides guidelines and considerations to help the administrator complete software upgrades of DB2 and Linux to meet HPSS prerequisites.

3.2. Upgrading Db2

Make sure that you are using the 64-bit installation images for Db2.

Review the software requirements for the version of Db2 you are installing.

Once the software and hardware prerequisites are satisfied, the DB2 upgrade may begin. Review the HPSS release notes and release roadmap for supported Db2 versions for your version of HPSS. Fix Packs and smaller Db2 increments are not individually qualified or listed in the HPSS release notes.

Sites should familiarize themselves with the DB2 guide for performing this upgrade, the DB2 Quick Beginnings Guide and DB2 Migration Guide for Linux, Unix, and Windows. The documents contain details not covered in this section. Some pre-migration requirements are:

On Linux:

* SWAP space twice the size of available memory

* 100MB of freespace in `/tmp`

* At least 1GB of free space in the Db2 install path.

* The Java SDK requirement stated in the Db2 migration guide.

Take a full offline backup of each HPSS database at this point to allow for reversion if necessary. To complete the full offline backup, execute:

```
% db2 force applications all
% db2 backup database <dbname> to <outdir>
```

Check the validity of the backup images by invoking db2ckbkp with the backup file name as the input parameter. For example:

```
% db2ckbkp HSUBSYS1.0.hpssdb.NODE0000.CATN0000.20080709112544.001
```

If the image is valid, the final output should be:

```
Image Verification Complete - successful.
```

The following steps allow for restoration of the instance and databases if necessary and are optional but recommended. The following example is on a test system with 2 subsystems with databases HSUBSYS1 and HSUBSYS2 and a configuration database name HCFG.

Save database DDL:

```
% db2support <output-directory> -d <database-name> -cl 0
% db2support /tmp/hcfg -d hcfg -cl 0
% db2support /tmp/hsubsys1 -d hsubsys1 -cl 0
% db2support /tmp/hsubsys2 -d hsubsys2 -cl 0
```

Save package information:

```
% db2 connect to hcfg
% db2 list packages for schema hpss show detail > /tmp/hcfg/packages.out
% db2 terminate
% db2 connect to hsubsys1
% db2 list packages for schema hpss show detail >
/tmp/hsubsys1/packages.out
% db2 terminate
% db2 connect to hsubsys2
% db2 list packages for schema hpss show detail >
/tmp/hsubsys2/packages.out
% db2 terminate
```

Save database manager and database configurations:

```
% db2 get dbm cfg > /tmp/db2.dbm.cfg.out
% db2 connect to hcfg; db2 get db cfg for hcfg show detail > \
/tmp/hcfg/db2.db.cfg.out
% db2 connect to hsubsys1; db2 get db cfg for hsubsys1 show detail > \
/tmp/hsubsys1/db2.db.cfg.out
% db2 connect to hsubsys2; db2 get db cfg for hsubsys2 show detail > \
/tmp/hsubsys2/db2.db.cfg.out
```

Save database information with db2look:

```
% db2look -d hcfg -e -o db2look.hcfg.out -l -x
% db2look -d hsubsys1 -e -o db2look.hsubsys1.out -l -x
% db2look -d hsubsys2 -e -o db2look.hsubsys2.out -l -x
```

Save DB2 registry settings:

```
% db2set -all > /tmp/db2set.out
```

At this point, enough information is retained to restore DB2 and the HPSS instance and databases to system with DB2 software installed.

For most HPSS installations, databases already have adequate settings to support migration, but consider the following in preparation for migration. Consider increasing the number of primary logs available to each database (LOGPRIMARY in the database configuration settings). Consider increasing the size of the system temporary tablespaces for each database to allow for twice as much free pages as used pages. Consider increasing the size of the system catalog tablespace to have at least 50% free space prior to conversion.

On Linux, if sites are using raw devices for DB2 logging they must change to using block devices as this feature is no longer available with DB2. See the DB2 Migration Guide for details on how to accomplish this.

The instance is now ready for migration. To perform the instance migration for an HPSS instance called hpssdb perform the following as the instance owner/user:

```
% db2 force applications all
% db2stop
% slibclean
% ipclean
```

As root, execute:

```
% /opt/ibm/db2/default/instance/db2imigr -u franky hpssdb
```

Where -u argument is the username of a user for DB2 fenced procedures. If you don't have one, just enter a user that won't do any useful work with DB2 (e.g. not root, or hpss or hpssdb).

To validate that the instance is at the correct level as the instance owner execute:

```
% db2start
% db2level
```

If there is an existing DAS, now is the time to migrate it. If migrating the DAS is problematic, sites may also simply drop the existing DAS and recreate a new one (see the DB2 Migration Guide). Migrating the DAS is optional and some sites may just want to skip to migrating the HPSS databases in the next step. To migrate the DAS:

As DAS owner (e.g. dasusr1):

```
% db2admin stop
```

As root:

```
% /opt/ibm/db2/default/instance/dasmigr dasusr1
```

Now migrate each HPSS database. Prior to migrating the databases, rename the current DB2 diagnostic log to provide a clean separation between different versions of DB2.

```
% mv /var/hpss/hpssdb/sqllib/db2dump/db2diag.log \
/var/hpss/hpssdb/sqllib/db2dump/db2diag.log.db2v8
```

where /var/hpss/hpssdb is the path to the DB2 instance at your site.

To migrate each HPSS database as the instance owner (e.g. hpssdb):

```
% db2 migrate database <dbname> [ user xxxx using xxxx ]
```

where dbname is each HPSS database name and xxxx is the database user and password at your site if necessary for connection. For example:

```
% db2 migrate database hcfg
```

Now sites need to perform some post migration steps. Begin by activating each migrated database as the instance owner (hpssdb):

```
% db2 activate database <dbname>
```

Attempt to connect to each migrated database to ensure it's active and available:

```
% db2 connect to <dbname>
```

If auditing is enabled, grant the audit user the new SECADM authority on each database by connecting to it and issuing:

```
% db2 grant SECADM on database to user <user>
% db2 "select * from syscat.auditpolicies a where \
a.auditpolicyname=DB2AUDIT_CFG_MIGR"
```

Rebind any packages for each migrated database by executing the following as instance owner:

```
% db2rbind <dbname> -l /tmp/db2rbind.out
```

Where -l argument is the output file desired.

Congratulations, your instance and databases are now migrated.

Sites can use the DB2 Migration Guide on post-migration steps to make additional useful configuration changes.

3.3. Upgrading Linux

Refer to documentation specific to Linux to perform the upgrade. The only specific consideration for DB2 and HPSS on Linux is to be aware that after the Linux upgrade often the DB2 devices used for tablespaces used by HPSS have their permissions or ownership modified as a result of the operating system upgrade. The problem is often not noticed until HPSS servers are started and SSM Alarms & Events shows errors accessing tables or metadata. The problem can be corrected by shutting down HPSS and DB2, correcting the permissions on the devices to allow access by the DB2 instance owner (e.g. hpssdb), and restarting DB2 and HPSS. HPSS should be rebuilt following the upgrade.

Chapter 4. Troubleshooting

This section covers warnings and errors that may occur during the conversion process and presents help in resolving the problem.

4.1. DBI and DBD

The following section provides information and troubleshooting when installing Perl's DBI module or when installing DB2's Perl DBD module. DBI was available for Linux for PowerPC platforms, but the DB2 DBD did not successfully install on Linux for PowerPC.

4.1.1. DBI and DBD - References

- DBI Mail List Archives

  http://groups.google.com/groups?group=perl.dbi.users

  http://www.xray.mpe.mpg.de/mailing-lists/dbi/

  http://www.mail-archive.com/dbi-users%40perl.org/

- Driver and Database Documentation (especially links at the bottom of the page):

  http://search.cpan.org/~timb/DBI/DBI.pm#Driver_and_Database_Documentation

- DBI Frequently Asked Questions:

  http://dbi.tiddlyspot.com/

4.2. DB2 Upgrade Errors

Problem:

Various errors upon installing the DB2 software.

Description:

If the installation is not performed as the root user and without sourcing the db2profile of any existing instances of DB2, the software may produce different errors due to trying to allow you to have multiple instances of HPSS on the same system.

Resolution:

Uninstall the DB2 software. Reinstall the DB2 software and follow the migration guide or instructions included in this document as the root user and without sourcing the db2profile of any existing DB2 instance (e.g. ldap, hpssdb).

Problem:

Tablespace access not allowed upon completion of the migration under linux.

Description:

Sites have reported a common problem with migrating databases under Linux is that the software updates the permissions on the underlying devices for database containers to exclude the HPSS instance owner from accessing them with rwx permissions.

Resolution:

Update the permissions on the database container devices (e.g. `/dev`) to allow rwx permissions by the HPSS instance owner (e.g. hpssdb).

Problem:

The **db2ckmig** program hangs.

Description:

The **db2ckmig** program should exit after reporting whether the instance and databases are ready for migration or not. This also caused the **db2imigr** program to hang as it calls **db2ckmig** initially.

Resolution:

CTRL-C the **db2ckmig** program after it reports whether the instance and databases are ready for migration or not. Also rename the program and create an empty (exit 0) **db2ckmig** in its place to proceed with the **db2imigr** program.

Appendix A. Glossary of Terms

ASN   Autonomous System Number, referring to a series of messages reserved for DB2 replication

BFID   HPSS bitfile identifier

DAS   DB2 Administration Server, used for remote administration of DB2 or for the GUI Control Center

DB2   Database 2

DBD   Database Driver, database specific implementation of DBI

DBI   Database Interface, Perl open source application programming interface for database interactions

DDL   Data Definition Language, used to define the table structure to DB2

DMS   Database Managed Storage, uses raw devices for tablespace containers

HPSS   High Performance Storage System software

Instance   Refers to a DB2 instance which is named according to the user that owns it

OBJECT_ID   HPSS namespace object identifier

RHEL   Red Hat Enterprise Linux

SLES   SUSE Linux Enterprise Server Instance

SMS   System Managed Space, uses filesystem facilities for tablespace containers

VVID   HPSS virtual volume identifier

Appendix B. References

1. **HPSS Installation Guide**

2. **HPSS Management Guide**

3. **HPSS Managetables Man Page (hpss_managetables.7)**