HPSS Storage Broker Installation Guide

HPSS Storage Broker Installation Guide

Table of Contents

About this book. The HPSS Storage Broker (HSB) Installation Guide is for use during system installation as well as throughout the lifetime of the system. It will guide system administrators through the planning and installation of a new HPSS Storage Broker (HSB) system. It also guides system administrators through the conversion process to upgrade existing HSB systems. It serves as a reference whenever the system is reconfigured by the addition, deletion, or modification of hosts, tape libraries, devices, or other components.

Chapter 1 discusses HPSS Storage Broker changes for the current release.

Chapter 2 gives an overview of HPSS Storage Broker technology.

Chapters 3-5 guide administrators of new HPSS Storage Broker systems through planning, system preparation, HPSS Storage Broker software installation, and configuration of the HPSS Storage Broker infrastructure.

Conventions used in this book. Example commands that should be typed at a command line will be proceeded by a percent sign ("%") and be presented in a courier font:

```
% sample command
```

Example command output and example contents of ASCII files will be presented in a courier font:

```
sample file line 1

sample file line 2
```

Any text preceded by a pound sign ("#") should be considered comment lines:

```
# This is a comment
```

Chapter 1. Release 1

This is the initial release (1.1) of HPSS Storage Broker (HSB).

In future versions of this document, this chapter will summarize HPSS Storage Broker changes in four categories:

1. New features
2. Retired features
3. Deprecated features
4. Changed features

Chapter 2. HPSS Storage Broker (HSB) basics

2.1. Introduction

The HPSS Storage Broker (HSB) is an on-premise service that uses Apache HTTPD and Db2. Users can interface with the system using either a web browser, the HSB Client GUI, or a command line tool. The HSB supports the ability for administrators to configure POSIX files systems, HPSS systems, S3 Clouds, and OpenStack Swift Clouds as accessible storage. In HSB terminology, these are known as storage endpoints. The HSB supports the ability for administrators to configure policies that define the storage endpoints and access behaviors to use when creating managed datasets.

Users can select files, objects, directories and containers from static storage endpoints (or from dynamic storage endpoints) and create managed datasets from that data, using a configured storage policy. Users can recall groups or individual files, objects, directories and containers from managed datasets. They can also optionally associate user-defined metadata (in XML format) with a managed dataset to facilitate quick and efficient searches.

The storage policy defines the storage endpoints to use, the fragmentation and parity characteristic, and the number of copies to generate when creating managed datasets.

All HSB requests are scheduled in a common queue based on the location of the requested data and the current availability of media.

The HSB server is the HSB component responsible for handling API calls, request queue management, and resource management.

The HSB can transfer data between storage endpoints using parallel streams. The default and maximum levels of parallelism are configured as part of the storage policy configuration. The parallelism is accomplished by aggregating all specified files or objects into a single data stream and then fragmenting the stream. The fragments are written to the storage endpoints in parallel.

The HSB copiers are the HSB components responsible for all data movement, including: managed dataset creation, recall and movement (for example, between storage endpoints).

Chapter 3. Planning

3.1. Prerequisites

3.1.1. Software

See the HSB Software prerequisites.

3.1.2. Hardware

3.1.2.1. HSB server

Architecture:

- Intel x86_64 systems
- Power ppc64LE systems

Storage:

- The **treefrog_setup** tool will extract the installation binaries (Db2 install image, wheels, RPMs) to a temporary directory in `/tmp`. The current size of these binaries is approximately 2 GiB.

3.1.2.2. HSB copier

Architecture:

- Intel x86_64 systems
- Power ppc64LE systems

3.1.2.3. HSB client GUI

Architecture:

Hardware architectures supporting the operating systems described in HSB Software prerequisites.

3.2. Packaging

The installation material will include three components.

HSB server executable tar file
: Installed on the HSB server system.

HSB copier RPM
: Installed on each system that will run a HSB copier.

HSB client RPM
installed on the HSB server system.

Systems that will run the HSB client GUI will install the necessary infrastructure from the HSB Admin GUI.

3.3. HSB server components

Components created during the execution of **treefrog_setup**.

3.3.1. User account

The **treefrog_setup** tool will create the user `treefrog` on the installation system:

1. **treefrog_setup** will prompt user for a password and password verification.
   - The home directory will be set to `/var/treefrog`
2. The `treefrog` account will be used as the database instance owner.
   - The database instance will be created under `/var/treefrog/sqllib`
3. The web services temp directory will be created under the home directory: `/var/treefrog/tmp`
   - This directory will hold the temporary dataset catalog files and user metadata.

The **treefrog_setup** tool will set up the **zookeeper** directories and configuration:

- The zookeeper configuration information will be created in `/var/treefrog/zk`
- The zookeeper logs and snapshots will be stored in `/var/treefrog/zk/data`

The following HSB server processes run as user **treefrog**:

- wsgi:treefrog
- treefrog_request_processor
- treefrog_cred_expirer

3.3.2. HSB server

SSL and certificate authorities

- SSL directory for the **treefrog** user account `/var/treefrog/ssl`.
  - Used for the SSL certificates and keys.
- **treefrog_setup** will create private keys for the HSB Certificate Authority(CA) and server with user input:

- Set the CA subject name
- Set the server subject name

Syslog path for the HSB server

- HSB logging path: `/var/log/treefrog`
- **treefrog_setup** will enable syslog for HSB servers

3.3.3. Database instance

- Name: **treefrog**
  - The configured Db2 instances can be listed with the following command (as root):

    `(> /var/treefrog/db2/V11.5/instance/db2ilist)`
- Instance owner account: **treefrog**
- Instance home directory: `/var/treefrog`
  - The instance is installed under `/var/treefrog/sqllib`
- Type: **ese**
  - ese: Enterprise Server Edition
- Authentication type: **SERVER**
  - This parameter specifies and determines how and where authentication of a user takes place.
- Port name: **db2c_treefrog**
  - This port information will be added to `/etc/services` as part of the instance creation.
  - This parameter contains the name of the TCP/IP port which a database server will use to await communications from client nodes. This name must be reserved for use by the database manager.

3.3.4. HSB copier

When installing a copier the following directories and files will be created.

- `/var/lib/treefrog/tomcat` - The copier is a Spring Boot application running on an embedded web server and this directory is defined as the Tomcat base directory for the embedded Tomcat web server.
- `/var/lib/treefrog/catalogs` - This directory serves as a temporary cache location used during the creation of the catalog file on create dataset operations.
- `/etc/treefrog/copier.conf` - The configuration file for the copier application.

- `/etc/treefrog/ssl` - The location of the copier keystore and truststore files.
- `/var/treefrog/ssl` - The location of the copier certificate and key files.
- `/var/log/treefrog/copier.log` - The location of the copier log file.
- `/usr/bin/treefrog_copier` - The script that controls the startup of the copier application.

3.3.5. Zookeeper

- Installation path: `/opt/zookeeper-3.5.4-beta`
- Host: `127.0.0.1`
- Port: `2281`

3.3.6. ABRT configuration

HSB relies on the ABRT (Automatic Bug detection and Reporting Tool) to collect crash information. ABRT is a set of tools to help users detect and report application crashes. The abrt and report tools are included and enabled as part of the core RHEL installation.

The abort configuration file for the abrtd application is `/etc/abrt/abrt.conf`. HSB uses the standard abrt configuration.

3.4. Configuration considerations

3.4.1. Database configuration

By default the HSB setup utility **treefrog_setup** will automatically configure a database with the name **treefrog**. By default all database storage will be configured under `/var/treefrog`. Size the `/var/treefrog` directory to support the database or use a custom database configuration.

For information on how to set up a custom database configuration, contact HPSS support.

Datebase Sizing

A database for 1 million datasets containing 100 million entries total (100 objects or files per dataset), stored with 2 data fragments and 1 parity fragment will take approximately 71 GiB of space.

Refer to Appendix B for a <detailed Db2 sizing breakdown>

Changes to `/etc/services`

The file `/etc/services` will be updated with the port entries used by the Db2 instance when the instance is created. The instance will be created by the treefrog setup utility **treefrog_setup**. For example:

```
db2c_treefrog   50000/tcp
```

```
#FCM Ports
DB2_treefrog    60000/tcp
DB2_treefrog_1  60001/tcp
DB2_treefrog_2  60002/tcp
DB2_treefrog_3  60003/tcp
DB2_treefrog_4  60004/tcp
DB2_treefrog_END        60005/tcp
```

By default, the first port (50000) is reserved for connection requests, and the first available six ports above 60000 are reserved for Fast Communication Manager (FCM) communication. One port is for the instance-owning database partition server and five ports are reserved for future use.

Database Path

The database path is the location where the database's hierarchical directory structure is created. The structure holds the files that are needed for the operation of the database.

The HSB setup tool will automatically configure DB2 to use the following settings:

Base path

- The base path is created under the database instance home directory `/var/treefrog/database`
- It will contain all of the directories created to support the HSB database.

Base storage path

- The base storage path is created under the base path: `/var/treefrog/database/storage`
- The number of storage paths determines how many paths are created for the database.
- The default number of storage paths is 4
- All of the storage paths created in the base storage path will start with **stg** and will have the path number appended:
  - `/var/treefrog/database/storage/stg0001`
  - `/var/treefrog/database/storage/stg0002`
  - `/var/treefrog/database/storage/stg0003`
  - `/var/treefrog/database/storage/stg0004`

Transaction log path

- This directory is created under database base path: `/var/treefrog/database/log`
- It will contain the active/primary database transaction log files.

Transaction mirror log path

- This directory is created under database base path: `/var/treefrog/database/mirrorlog`
- It will contain the database transaction log file mirrors.

First archive log path

- This directory is created under database base path: `/var/treefrog/database/logarchive1`
- It will contain the first copy of archived database transaction log files.

Second archive log path

- This directory is created under database base path: `/var/treefrog/database/logarchive2`
- It will contain the second copy of archived database transaction log files.

Database directory path

- This directory is created under database base path: `/var/treefrog/database/dbpath`
- The database path is the location where a hierarchical directory structure is created. The structure holds the following files that are needed for the operation of the database:
  - Buffer pool information
  - Table space information
  - Storage path information
  - Database configuration information
  - History file information regarding backups, restores, loading of tables, reorganization of tables, altering of table spaces, and other database changes
  - Log control files with information about active logs

Database backup directory

- This path is created under database base path: `/var/treefrog/database/backups`
- Initial backups of the database taken during the setup process will be stored here.

Configuration Parameter Settings

See the "Db2 Database Configuration Parameters" appendix for details.

3.4.1.1. Zookeeper configuration considerations

The Zookeeper configuration will be installed at `/var/treefrog/zk/zookeeper.cfg`. Zookeeper will be configured to use a single node.

The following configuration options will be set in `/var/treefrog/zk/zookeeper.cfg`:

```
dataDir: /var/treefrog/zk/data
dataLogDir: Not set
```

The Zookeeper log directory default configuration is `/var/log/treefrog`.

For assistance in constructing custom Zookeeper configuration, contact HPSS support.

3.5. HSB users and passwords

Executing **treefrog_setup** will create users, keystores, and truststores. It requires the administator to give names and passwords for the following items:

HSB user password
The password to use for the HSB Linux user named **treefrog**. The **treefrog_setup** will create the user using the supplied password. The password for this user can be changed from a Linux terminal using the **passwd** command.

HSB primary admin user and password
The HSB equivalent to the root user in Linux. It will be created by **treefrog_setup** and the password will be set as part of the creation. The password for this user can be changed from the HSB Admin GUI or using the treefrog_setup tool.

Flask password
The Flask password is used by the **treefrog_setup** tool to encrypt certain HSB database entries and session authentication information. The password should never be changed.

HSB keystore password
The **treefrog_setup** tool will create an HSB keystore to store private key and identity certificates for the HSB server and copiers. The HSB keystore password is used to encrypt the keystore. The password should never be changed.

HSB truststore password
The **treefrog_setup** tool will create an HSB truststore to store certificates from HSB servers and copiers. The HSB truststore password is used to encrypt the truststore. The password should never be changed.

Chapter 4. Preparation

4.1. Installing prerequisites

This section describes the hardware and software prerequisites needed by the HSB server, client GUI, and admin GUI.

4.1.1. Operation system

The HSB server and copier both require:

- RHEL 7.6 or greater

The HSB copier also requires the following:

- Java OpenJDK 1.8 or greater

HSB currently only supports OpenJDK. While in theory the IBM JDK will work, extra configuration is required to enable TLSv1.2.

The HSB server also requires the following:

- Python 2.7.5
- httpd
- mod_ssl
- mod_wsgi
- python2-cryptography
- xorg-x11-xauth
- java-1.8.0-openjdk-headless (or greater)

These packages can be installed via yum.

In order to do an offline install of these packages you can mount a DVD containing RHEL 7.6 or later installation ISO and create a local yum repository.

To avoid conflicts disable any non-Redhat repositories.

1. Create directory to mount to:

```
> mkdir -p  /mnt/disc
```

2. Mount the DVD or ISO:

DVD

```
> mount /dev/sr0  /mnt/disc
```

ISO

```
> mount -o loop RHEL7.1.iso /mnt/disc
```

3. Copy the media.repo file from the root of the mounted directory to `/etc/yum.repos.d/` and set the permissions to 0644 or another similar permissions set::

```
> cp /mnt/disc/media.repo /etc/yum.repos.d/rhel7dvd.repo
> chmod 644 /etc/yum.repos.d/rhel7dvd.repo
```

4. Edit the new repo file, changing the gpgcheck=0 setting to 1 and adding the: following 3 lines

```
> vi /etc/yum.repos.d/rhel7dvd.repo

enabled=1
baseurl=file:///mnt/disc/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

Example of a modified repo file:

```
[InstallMedia]
name=DVD for Red Hat Enterprise Linux 7.1 Server
mediaid=1359576196.686790
metadata_expire=-1
gpgcheck=1
cost=500
enabled=1
baseurl=file:///mnt/disc/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

5. Clear the cache and check whether you can get the packages list from the repo:

```
> yum clean all
> yum repolist enabled
```

It should look like the following:

```
> yum repolist enabled
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscrip
repo id
InstallMedia                                                                    Re
repolist: 5,229
```

6. Install the packages required by HSB:

```
> yum install httpd
> yum install mod_ssl
> yum install mod_wsgi
> yum install python2-cryptography
> yum install xorg-x11-xauth
> yum install java-1.8.0-openjdk-headless.x86_64
```

Or *java-1.7.0-openjdk-headless.ppc64le* for ppc64le architectures

4.1.2. Admin GUI prerequisites

The HSB Admin GUI requires a web browser: Mozilla Firefox or Google Chrome are recommended

4.1.3. Installation package

The executable installation package contains the HSB server binary distribution, the python prerequisites of the server, and a Db2 installation package.

Contents of the install package

Chapter 5. Installation

5.1. HSB installation steps

1. Install the HSB prerequisites
2. Install the HSB system components using **treefrog_server_installer**
3. Perform the HSB setup using **treefrog_setup**
4. Configure the HSB copiers
5. Configure the HSB client GUIs

5.2. Install HSB system components

On the host where you want to run the HSB system, locate the self-extracted file `treefrog_server_installer` and run it. Once it has completed the installer will install the HSB server, HSB copier, and HSB client GUI RPMs.

```
cd <path to installer>
> ./treefrog_server_installer
```

You may want to 'tee' the output to a file:

```
> ./treefrog_server_installer 2>&1 | tee /tmp/install.log
```

The following messages may be displayed during the installation of Db2. The 32-bit libraries referenced below are not required.

```
Requirement not matched for DB2 database "Server" . Version: "11.5.6.0".

Summary of prerequisites that are not met on the current system:

   DBT3514W  The db2prereqcheck utility failed to find the following 32-bit
   library file: "/lib/libpam.so*".


DBT3514W  The db2prereqcheck utility failed to find the following 32-bit library
file: "libstdc++.so.6".


Requirement not matched for DB2 database "Server" . Version: "11.5.6.0".

Summary of prerequisites that are not met on the current system:

   DBT3514W  The db2prereqcheck utility failed to find the following 32-bit
   library file: "/lib/libpam.so*".


DBT3514W  The db2prereqcheck utility failed to find the following 32-bit
library file: "libstdc++.so.6".
```

```
The execution completed successfully.

For more information see the DB2 installation log at
"/tmp/db2_install.log.8943".
```

5.3. HSB system setup

This will set up the infrastructure necessary to run a HSB system and the HSB admin GUI.

5.3.1. Features of the setup tool

The HSB setup tool (**treefrog_setup**) will perform the following steps:

1. Create the HSB user account.
2. Create an SSL directory for the HSB account (`/var/treefrog/ssl`).
3. Create an etc path for HSB account (`/etc/treefrog`).
4. Create a ssl path for the HSB account (`/etc/treefrog/ssl`).
5. Create the logging directory for HSB (`/var/log/treefrog`).
6. Set up the HSB Db2 database environment:
   a. Create the Db2 instance.
   b. Create the directory infrastructure supporting the Db2 database.
7. Create the Db2 database and supporting objects including table definitions.
   a. Set the database configuration parameters to recommended values.
   b. Installs the **db2_fullbackup.ksh** script to `/var/treefrog/bin`.
8. Add the following command to the root user's crontab:

```
0 0 * * * /var/treefrog/bin/db2_fullbackup.ksh -p /var/treefrog/database/backups \
                                               -s /var/treefrog/database/backups2 \
                                               -i treefrog
```

9. Enable syslog for TCP/IP support.
10. Configure X.509 certificates.
11. Configure the system services necessary to run all HSB components.

5.3.2. Setup tool command

To set up the HSB system and admin GUI execute the following command:

```
> treefrog_setup setup
```

This tool is "restartable". If a failure occurs during execution, it can be executed again and will pick up where it left off. Steps that have already been executed will either be re-executed, or skipped if the necessary components are in place from the previous execution.

Running the above command will require the following input:

1. Enter the HSB user password when prompted as follows:

```
HSB user password>
```

2. Verify the HSB user password:

```
Please Verify the Password>
```

3. The HSB user password will be required again later in the process:

```
        HSB user password>
```

4. Answer the certificate authority prompts:

```
        Certificate Authority Distinguished Name:
                Country[US]:
                State[TX]:
                Locality[Houston]:
                Organization[IBM]:
                Organizational Unit[HPSS Treefrog CA @ icsb2.clearlake.ibm.com]:
        CA Certificate Period Days[3650]:
```

5. Answer the server certificate prompts:

```
        Server Certificate Distinguished Name:
                Country[US]:
                State[TX]:
                Locality[Houston]:
                Organization[IBM]:
                Organizational Unit[HPSS Treefrog]:
                Canonical Name[icsb2.clearlake.ibm.com]:
        Server Certificate Period Days[1825]:
```

6. Enter the Flask secret key:

```
        Flask secret key>
        Please Verify the Password>
```

7. Specify the language:

```
        Language locale[en]:
```

8. Specify the HSB system httpd information:

```
        HSB server httpd processes[5]:
        HSB server httpd threads[10]:
```

9. Enter the primary administrator information:

```
        Primary administrator username[admin]:
        Primary administrator first name[Root]:
        Primary administrator last name[Admin]:
        Primary administrator email[rootadmin@yourdomain.com]:
```

```
        Primary administrator password>
        Please Verify the Password>
        Primary administrator account would be set to username = admin; firstname = Ro
```

10.Enter the HSB keystore password:

```
        HSB keystore password>
        Please Verify the Password>
```

11.Enter the HSB truststore password:

```
        HSB truststore password>
        Please Verify the Password>
```

The output from the **treefrog_setup** command will be directed to `/tmp/treefrog_setup.log`.

5.3.3. Usage

Usage: **treefrog_setup setup|resetadminpassword** [**-h**] [**-d**] [**-c**] [**--outputfile** *OUTPUTFILE*]

5.3.4. Positional arguments:

setup
: Set up a new treefrog system. All HSB system configuration components will be created. This is intended to be the primary entry point for this setup tool.

resetadminpassword
: Allows the root user only to reset the password for the default HSB administrator account *admin*.

5.3.5. Optional arguments:

-h | --help
: show this help message and exit

-d | --debug
: Enable additional debug output

-c | --console
: Log to console and log file

--outputfile *OUTPUTFILE*
: Output file name and path

5.3.6. Examples

1. **Setup a new system**

```
> treefrog_setup setup
HSB user password>
Please Verify the Password>
HSB user password>
Certificate Authority Distinguished Name:
```

```
        Country[US]:
        State[TX]:
        Locality[Houston]:
        Organization[IBM]:
        Organizational Unit[HPSS Treefrog CA @ icsb2.clearlake.ibm.com]:
CA Certificate Period Days[3650]:
Server Certificate Distinguished Name:
        Country[US]:
        State[TX]:
        Locality[Houston]:
        Organization[IBM]:
        Organizational Unit[HPSS Treefrog]:
        Canonical Name[icsb2.clearlake.ibm.com]:
Server Certificate Period Days[1825]:
Flask secret key>
Please Verify the Password>
Language locale[en]:
HSB server httpd processes[5]:
HSB server httpd threads[10]:
Primary administrator password>
Please Verify the Password>
Primary administrator account would be set to username = admin;
firstname = Root; lastname = Admin; email = rootadmin@yourdomain.com;
Do you want to continue? (Yy/Nn): y
Created symlink from
/etc/systemd/system/treefrogd.service.wants/treefrog_zkd.service to
/usr/lib/systemd/system/treefrog_zkd.service.
HSB keystore password>
Please Verify the Password>
HSB truststore password>
Please Verify the Password>
```

2. **Reset HSB admin user account password**

```
> treefrog_setup resetadminpassword
New password for user "admin">>
Please Verify the Password>
```

5.4. Install HSB copier

Follow these steps on each non-HSB server system and each host that will run a HSB copier:

1. Locate the HSB copier RPM in the installation media: `TreefrogCopier-1.1.0-0.noarch.rpm`

The copier RPM is installed automatically as part of the server installation; however, if the system will use parity, the copier ECC (Error Correction Code) will still need to be installed.

```
> cd <path to TreefrogCopier RPM>
### Change HSB version / Treefrog version below
> rpm -Uhv TreefrogCopier-1.1.0-0.noarch.rpm
Preparing...                          ########################################
Updating / installing...
### Change HSB version / Treefrog version below
TreefrogCopier-1.1.0-0                ########################################
Created symlink from
/etc/systemd/system/multi-user.target.wants/treefrog_copier.service to
```

```
/usr/lib/systemd/system/treefrog_copier.service.
```

To see the files installed by the HSB copier RPM:

```
> rpm -q --filesbypkg TreefrogCopier
TreefrogCopier            /etc/logrotate.d/treefrog-copier
TreefrogCopier            /etc/rsyslog.d/treefrog-copier.conf
TreefrogCopier            /etc/treefrog
TreefrogCopier            /etc/treefrog/copier.conf
TreefrogCopier            /etc/treefrog/ssl
TreefrogCopier            /usr/bin/treefrog_copier
TreefrogCopier            /usr/bin/treefrog_copier_syslog_plugin
TreefrogCopier            /usr/lib/systemd/system/treefrog_copier.service
### Change HSB version / Treefrog version below
TreefrogCopier            /usr/share/java/treefrog_copier-1.1.0.jar
TreefrogCopier            /var/lib/treefrog
TreefrogCopier            /var/lib/treefrog/catalogs
TreefrogCopier            /var/log/treefrog
```

2. Configure the copier using the admin GUI. Refer to the "HSB Administrator’s Guide", Section 3.2.2.1 Add copier.

3. For each copier, download the copier certification bundle and execute it. Refer to the section titled "Download copier certificate bundle" located in Section 3.2.2.3 of the "HSB Administrator’s Guide".

5.4.1. HSB copier logging

The HSB copier stores its current log file, `copier.log` as well as archived copier logs under the directory `/var/log/treefrog`. See the "HSB Administrator’s Guide" for more information on copier logging and log file rotation.

5.4.2. HSB copier ECC generation provider installation

If your configuration will include storage policies requiring the generation of parity fragments, the following Error Correction Code (ECC) provider, Intel ® Intelligent Storage Acceleration (ISA), must be installed on all the systems running copiers supporting those storage policies.

5.4.2.1. Intel ® Intelligent Storage Acceleration (ISA) library provider

The ISA library requires the following prerequisites be built and installed before installing the ISA library.

- NASM (Netwide Assembler)
- YASM (Yasm Modular Assembler Project)

NASM Installation

Follow these steps to build and install NASM:

1. As the *root* user, download the tar file for the latest stable release from `https://www.nasm.us`
2. Then build and install the library as follows:

```
> tar -xvf nasm-2.14.02.tar.gz
> cd nasm-2.14.02
> ./configure
> make
> make install
```

YASM installation

Follow these steps to build and install YASM:

1. As the *root* user, download the tar file for the latest stable release from `https://yasm.tortall.net/`
2. Then build and install the library as follows:

```
> tar -xvf yasm-1.3.0.tar.gz
> cd yasm-1.3.0
> ./configure
> make
> make install
```

ISA installation

Follow these steps to build and install the ISA library:

1. As the *root* user, download the zip file from `https://github.com/intel/isa-l`
2. Then build and install the library as follows:

```
> unzip isa-l-master.zip
> cd isa-l-master
> ./autogen.sh
> ./configure --prefix=/usr --libdir=/usr/lib64
> make
> make install
```

3. Locate the ISA ECC provider RPM in the installation media: `TreefrogCopierISAECC-1.0-0.x86_64.rpm`

```
> rpm -v --install TreefrogCopierISAECC-1.0-0.x86_64.rpm
Preparing packages...
TreefrogCopierISAECC-1.0-0.x86_64
```

To see the files installed by the ISA ECC provider RPM:

```
> rpm -q --filesbypkg TreefrogCopierISAECC-1.0-0
TreefrogCopierISAECC      /usr/lib/libISALerasureJNI.so
```

5.5. Install HSB client GUI

The preferred method for installing the HSB client GUI is the client installer package. Refer to Section 3.4.3 "Download Linux client installer" in the "HSB Administrator's Guide" for instructions on how to download and run the client installer.

The HSB client GUI RPM is automatically installed on the HSB server. If you need to install the GUI on other systems do so using either the option to download the Linux client installer from the "Tools" menu on the HSB admin GUI or by running the **treefrog_setup** tool with the **genclinstbundle** option on the server and providing the user with the resulting client installer package to execute on their system. Downloading the installer from the "Tools" menu is the easiest and preferred method of installing the client GUI on systems other than the HSB server. Refer to the "HSB Administrator's Guide" for more information.

If you need to install the client RPM without using the client installer the instructions are:

Installing the client RPM will not configure the client. The client installer package will still need to be run so the client GUI can connect to the HSB server.

1. Locate the HSB client GUI RPM: `TreefrogClient-1.1.0-0.noarch.rpm`
2. Perform the following steps:

```
> cd <path to TreefrogClient rpm>
### Change HSB version / Treefrog version below
> rpm -Uhv TreefrogClient-1.1.0-0.noarch.rpm
Preparing...                          ########################################
Updating / installing...
### Change HSB version / Treefrog version below
TreefrogClient-1.1.0-0                ########################################
```

   To see the files installed by the RPM:

```
rpm -q --filesbypkg TreefrogClient
TreefrogClient            /bin/treefrog_gui
### Change HSB version / Treefrog version below
TreefrogClient            /usr/share/java/treefrog_gui-1.1.0.jar
```

3. Download and run the client installer. The installer will create and configure the `~/.treefrog/app.conf` file and it will import the HSB server certificate into the keystore.

5.5.1. HSB client GUI logging

The HSB client GUI stores its current log file, `client.log`, as well as archived client logs under the directory `~/.treefrog/logs`. See the "HSB Administrator's Guide" for more information on client logging and log file rotation.

Appendix A. Db2 database configuration parameters

Configuration parameters that are not included in the following list are set to their default values. For additional detail on Db2 database configuration parameters go to the Db2 Knowledge Center: https://www.ibm.com/support/knowledgecenter/SSEPGG_11.5.0/com.ibm.db2.luw.admin.config.doc/doc/c0060795.html

For custom database configurations contact HPSS support.

A.1. NEWLOGPATH

Default value: `/var/treefrog/database/log`

This parameter allows you to specify a string of up to 242 bytes to change the location where the log files are stored.

The **NEWLOGPATH** and **MIRRORLOGPATH** should reference different storage devices.

A.2. MIRRORLOGPATH

Default value: `/var/treefrog/database/mirrorlog`

This parameter allows you to specify a string of up to 242 bytes to change the location where the log files are stored.

A.3. LOGPRIMARY

Default value: `10`

This parameter allows you to specify the number of primary log files to be preallocated. The primary log files establish a fixed amount of storage allocated to the recovery log files.

When the database is first activated the primary log files are created in the log paths specified in NEWLOGPATH and MIRRORLOGPATH. Each logfile will have space pre-allocated from the file system. The amount of space pre-allocated depends on the configuration parameter LOGFILSIZ.

A.4. LOGSECOND

Default value: `-1`

This parameter specifies the number of secondary log files that are created and used for recovery log files. The secondary log files are created only as needed.

A value of `-1` indicates the database is configured with infinite active log space. There is no limit on the size or the number of in-flight transactions running on the database. LOGPRIMARY and LOGFILSIZ still control the number and size of log files the database manager should keep in the active log path.

A.5. LOGFILSIZ

Default value: `25000`

Unit of measure: `4KiB page`

This parameter defines the size of each primary and secondary log file. The default size of each log file created will be: 25000 * 4KiB = 102400000 bytes

A.6. LOGBUFSZ

Default value: `16384`

This parameter allows you to specify the amount of the database heap (defined by the dbheap parameter) to use as a buffer for log records before writing these records to disk.

A.7. LOGARCHMETH1

Default value: `/var/treefrog/database/logarchive1`

This parameter specifies the media type and location of the primary destination for logs that are archived from the current log path.

The filesystem supporting this path should have enough space allocated to handle the number of transaction log files expected to be generated and archived over the period of time defined by the site. This should coincide with the age and number of database backups kept on the system for a timely recovery. For example, if the goal is to keep a week’s worth of database backups online, then the archived log files that have occurred since the oldest backup should be kept online as well. This number will vary and depends on the transaction workload the database is performing. More insert, update, or delete operations generate more log files.

A.8. LOGARCHMETH2

Default value: `/var/treefrog/database/logarchive2`

This parameter specifies the media type and location of the secondary destination for logs that are archived from either the current log path or the mirror log path. See LOGARCHMETH1 for file system details.

A.9. LOGARCHCOMPR1

Default value: `ON`

This parameter specifies whether the log files written to the primary archive destination for logs are compressed.

A.10. LOGARCHCOMPR2

Default value: `ON`

This parameter specifies whether the log files written to the secondary archive destination for logs are compressed.

A.11. NUM_DB_BACKUPS

Default value: 4

This parameter specifies the number of full database backups to retain for a database.

A.12. REC_HIS_RETENTN

Default value: `0`

This parameter specifies the number of days that historical information on backups are retained.

When set to "0" and **AUTO_DEL_REC_OBJ** is set to **ON**, automated history file pruning and recovery object deletion are carried out based on the timestamp of the oldest backup maintained by the **NUM_DB_BACKUPS** database configuration parameter.

A.13. AUTO_DEL_REC_OBJ

Default value: `ON`

This parameter specifies whether database log files, backup images, and load copy images should be deleted when their associated recovery history file entry is pruned.

A.14. SELF_TUNING_MEM

Default value: `ON`

This parameter determines whether the memory tuner will dynamically distribute available memory resources as required between memory consumers that are enabled for self-tuning.

A.15. PCKCACHESZ

Default value: `AUTOMATIC`

This parameter is allocated out of the database shared memory, and is used for caching of sections for static and dynamic SQL and XQuery statements on a database.

A.16. LOCKLIST

Default value: `AUTOMATIC`

This parameter indicates the amount of storage that is allocated to the lock list. There is one lock list per database and it contains the locks held by all applications concurrently connected to the database.

A.17. MAXLOCKS

Default value: `AUTOMATIC`

This parameter defines a percentage of the lock list held by an application that must be filled before the database manager performs lock escalation.

A.18. SHEAPTHRES_SHR

Default value: `AUTOMATIC`

This parameter represents a soft limit on the total amount of shared sort memory reservation available to sort heap-based operations.

A.19. SORTHEAP

Default value: `AUTOMATIC`

This parameter defines the maximum number of private or shared memory pages that an operation that requires sort heap memory allocates.

A.20. DATABASE_MEMORY

Default value: `AUTOMATIC`

This parameter specifies the size of the database memory set.

A.21. AUTO_REVAL

Default value: `DEFERRED`

This configuration parameter controls the revalidation and invalidation semantics.

A.22. AUTO_MAINT

Default value: `ON`

This parameter is the parent of all the other automatic maintenance database configuration parameters set during HSB setup:

AUTO_DB_BACKUP
Default Value: `OFF`

This automated maintenance parameter enables or disables automatic backup operations for a database.

AUTO_TBL_MAINT
Default Value: `ON`

This parameter is the parent of table maintenance parameters:

AUTO_RUNSTATS
Default value: `ON`

This automated table maintenance parameter enables or disables automatic table RUNSTATS operations for a database.

It is a parent to **AUTO_STMT_STATS**.

AUTO_STMT_STATS
Default value: `ON`

This parameter enables and disables the collection of real-time statistics. It is a child of the auto_runstats configuration parameter.

AUTO_REORG
Default value: `OFF`

This automated table maintenance parameter enables or disables automatic table and index reorganization for a database.

AUTO_STATS_VIEW
Default value: `ON`

This parameter enables and disables automatic statistic collection on statistical views.

AUTO_SAMPLING
Default value: `ON`

This parameter controls whether automatic statistics collection uses sampling when collecting statistics for a large table.

Appendix B. Database storage estimation tips

B.1. Space requirements

Database Sizing Factors

The configuration metadata is a very small percentage of the overall space requirements and is not factored into the space requirements. Storage space required by the database varies by installation, but will be driven by the following factors:

1. The number of managed datasets.
2. The number of copies defined in the storage policies used by the managed datasets. This affects the number of rows in the **MDS_CATALOG** table. There is a row in this table for every copy of a managed dataset.
3. The number of fragments defined in the storage policies used by the managed datasets. This affects the number of rows in the **MDS_FRAG_CHUNK** table. There is a row in this table for every copy of a managed dataset with addition rows based on the fragmentation settings defined by the storage policy copy.
4. The amount of user metadata ingested.

   User metadata is stored in Db2 as an XML object. Each XML object can be as large as 2 GiB and multiple XML objects can be stored per managed dataset.

   Additional indexes can be defined to optimize site-specific searching.

   

   User Metadata can significantly increase the storage requirements and workload on the database. If you expect to store significant amounts of user metadata or plan on running significant or frequent searches on user metadata, contact HPSS support. Support personnel can review storage and query expectations and suggest potential configuration optimizations.
5. The number of directories, containers, files, or objects stored in the managed datasets. This affects the number of rows in the **MDS_MANIFEST** table. There is a row in this table for every directory, container, file, and object stored in each managed dataset.
6. The retention period defined for manifest associated with the managed dataset. Manifest rows can be purged from the database to reduce the space requirements and limit growth. The manifest data is also stored in catalog files at the storage endpoints and can be retrieved/reloaded into the database if needed after being purged. This does not include user metadata associated with the managed dataset. The rention period is a site-specific setting.
7. Tablespace density:

   a. The tablespaces are the objects that actually store table data.

b. They are managed using a database concept called "Managed By Automatic Storage" where the database manager extends tablespace storage containers as needed.

c. When the available space in a tablespace is used up, the amount of space used per object is lower (higher density) than immediately following a tablespace expansion request (lower density).

   i. The variance in tablespace density before and after expansion will depend on the expansion size.

   ii. The larger the table, the less important this becomes when trying to estimate space usage.

8. Database compression efficiency:

   a. The data in the larger tables is compressed using the database's compression logic algorithm. The database uses three forms of compression to reduce the storage requirements of the database data:

      Classic row compression
      A dictionary-based compression algorithm to replace recurring strings with shorter symbols within data rows.

      Adaptive compression
      Improves upon the compression rates that can be achieved using **Classic Row Compression** by itself. Adaptive compression incorporates classic row compression; however, it also works on a page-by-page basis to further compress data.

      Index compression
      The degree of compression achieved will vary based on the type of index you are creating, as well as the data the index contains. For example, the database manager can compress an index with a large number of duplicate keys by storing an abbreviated format of the record identifier (RID) for the duplicate keys. In an index where there is a high degree of commonality in the prefixes of the index keys, the database manager can apply compression based on the similarities in prefixes of index keys.

   b. The compressed data will remain compressed as it is retrieved from storage (Tablespaces) and placed in memory (Bufferpools) and when the row updates are recorded in the Transaction Logs. See the Transaction Logging Requirements

For details on table data compression visit: https://www.ibm.com/support/knowledgecenter/SSEPGG_11.5.0/com.ibm.db2.luw.admin.dbobj.doc/doc/c0007306.html

For details on index compression visit: https://www.ibm.com/support/knowledgecenter/SSEPGG_11.5.0/com.ibm.db2.luw.admin.dbobj.doc/doc/c0054539.html

Example Example

The following tables are provided as examples to illustrate the space used by randomly generated test data. Each site's storage requirements will vary:

Configuration related metadata:

Table	Row Count	Index Space (KiB)	Data Space (KiB)	Total Space (KiB)	Space per (bytes)
Copier	1,000	512	384	896	917
Storage Endpoints	1,000	640	640	1,280	1,310
Copier Endpoints	100,000	4,992	2,816	7,808	79
Storage Policies	1,000	384	384	768	786
Storage Policy Copies	5,000	1,024	1,152	2,176	445
Storage Policy Copy Endpoints	46,183	2,560	1,536	4,096	90
Users	10,000	1,152	3,072	4,224	432
User Project Assignment	99,940	3,712	2,816	6,528	66

Managed Data Set related metadata:

Table	Row Count	Index Space (KiB)	Data Space (KiB)	Total Space (KiB)	Space per (bytes)
Projects	10,000	7,168	6,144	13,312	891
Project Storage Policy Assignment	99,486	4,096	3,072	7,168	73
Managed Datasets	998,837	97,280	60,416	157,696	161
Managed Dataset Catalogs	4,994,185	100,352	241,664	342,016	70
Managed Dataset Fragments	49,838,687	3,299,328	4,685,824	7,985,152	164
Managed Dataset Manifests	100,000,171	22,378,496	17,212,416	39,590,912	405

Database space allocation and utilization:

Total space allocated for data/ indexes (KiB)	Total space used (KiB)	Percent Utilized

70,944,412	48,424,188	68%

Space allocated to the database will increase automatically as the space allocated is consumed (when percent utilized reaches 100%). The increase in amount of space allocated can vary with each space request made by the database.

Event message tables are not shown in the example data above.

B.2. Transaction logging

Database logging is an important part of a highly available database solution design. Database logs make it possible to recover from a failure. These logs keep a record of database changes. If a database needs to be restored to a point beyond the last full offline backup, logs are required to roll the data forward to the point of failure.

By default the HSB setup utility **treefrog_setup** configures the database for **archival** logging.

For details on archival logging see: https://www.ibm.com/support/knowledgecenter/SSEPGG_11.5.0/com.ibm.db2.luw.admin.ha.doc/doc/c0051344.html

See the following database configuration parameter descriptions for details on configuring database transaction logging and space allocations:

- NEWLOGPATH
- MIRRORLOGPATH
- LOGPRIMARY
- LOGSECOND
- LOGFILSIZ
- LOGBUFSZ
- LOGARCHMETH1
- LOGARCHMETH2
- LOGARCHCOMPR1
- LOGARCHCOMPR2
- NUM_DB_BACKUPS
- AUTO_DEL_REC_OBJ
- REC_HIS_RETENTN

Transaction logging space requirements can be affected by the database compression efficiency.

Appendix C. Db2 database backup tool

This tool provides the capability to back up the Db2 database managing the HSB application metadata.

C.1. Features

- **Online** or **Offline** backups
- Backup image compression
- Local or remote secondary backup image location
- Uses syslog
- File system space utilization check

C.2. Usage

Usage: db2_fullbackup.ksh

- Required:
  - -i <DB2 instance owner>
  - -p <primary backup directory>
- Optional:
  - -s <secondary backup directory>

Examples:

```
> db2_fullbackup.ksh -p /db2_backup1 -s /db2_backup2 -i treefrog
```

```
db2_fullbackup.ksh -p /db2_backup1 -s hm2:/db2_backup2 -i treefrog
```

```
db2_fullbackup.ksh -p /db2_backup1 -i treefrog
```

All databases in the database directory will be backed up.

If a database has been activated an online backup will be performed, otherwise an offline backup will be performed.

If the secondary backup is on another host, use <host>:<directory> format. The **scp** command will be used to copy the database backup image from the primary backup

location to the secondary location on the remote host. It is recommended that the HSB instance owner user id have password-less ssh configured between the HSB server and the remote host.

C.3. Installation location

```
/var/treefrog/bin/*db2_fullbackup.ksh*
```

Appendix D. Install package contents

D.1. x86_64 Architecture

The contents include for x86_64 architectures:

HSB server wheel

- `TreefrogServer-1.1-py27-none-any.whl`

Db2 server and license files

- `v11.1.4fp4a_linuxx64_universal_fixpack.tar.gz`
- license files:
  - `db2ese_c.lic` - Db2 Enterprise Server license
  - `db2hc.lic` - Db2 High Capacity license
  - `iidr.lic` - Infosphere Data Replication license

HSB server RPM files

- `python2-pip-8.1.2-5.el7.noarch.rpm`

Zookeeper installation material

- `zookeeper-3.5.4-beta.tar.gz`

HSB server prerequisite wheels

- `Babel-2.6.0-py2.py3-none-any.whl`
- `certifi-2017.7.27.1-py2.py3-none-any.whl`
- `cffi-1.11.5-cp27-cp27mu-manylinux1_x86_64.whl`
- `chardet-3.0.4-py2.py3-none-any.whl`
- `click-6.7-py2.py3-none-any.whl`
- `clickclick-1.2.2-py2.py3-none-any.whl`
- `colorama-0.3.9-py2.py3-none-any.whl`
- `enum34-1.1.6-py2-none-any.whl`
- `flake8-2.6.0-py2.py3-none-any.whl`

- Flask_Babel-0.11.2-py2.py3-none-any.whl
- Flask_WTF-0.14.2-py2.py3-none-any.whl
- funcsigs-1.0.2-py2.py3-none-any.whl
- future-0.17.1-cp27-none-any.whl
- idna-2.8-py2.py3-none-any.whl
- jsonschema-2.6.0-py2.py3-none-any.whl
- kazoo-2.6.0-py2.py3-none-any.whl
- mccabe-0.5.3-py2.py3-none-any.whl
- mock-2.0.0-py2.py3-none-any.whl
- pbr-5.1.1-py2.py3-none-any.whl
- pip-9.0.1-py2.py3-none-any.whl
- pycodestyle-2.0.0-py2.py3-none-any.whl
- pycparser-2.19-py2.py3-none-any.whl
- pycrypto-2.6.1-cp27-cp27mu-linux_x86_64.whl
- pyflakes-1.2.3-py2.py3-none-any.whl
- PyJWT-1.7.1-py2.py3-none-any.whl
- python_dateutil-2.7.2-py2.py3-none-any.whl
- requests-2.21.0-py2.py3-none-any.whl
- setuptools-28.8.0-py2.py3-none-any.whl
- swagger_spec_validator-2.4.1-py2.py3-none-any.whl
- typing-3.6.2-py2-none-any.whl
- urllib3-1.24.1-py2.py3-none-any.whl
- Werkzeug-0.12.2-py2.py3-none-any.whl
- PyYAML-3.13-cp27-cp27mu-linux_x86_64.whl
- blinker-1.4-py2-none-any.whl
- Flask_SQLAlchemy-2.0-py2-none-any.whl
- Jinja2-2.7.2-py2-none-any.whl

- `Flask_AppBuilder-1.8.1-py2-none-any.whl`
- `ibm_db-2.0.8-cp27-cp27mu-linux_x86_64.whl`
- `ibm_db_sa-0.3.2-py2-none-any.whl`
- `Flask_Login-0.2.11-py2-none-any.whl`
- `strict-rfc3339-0.7.tar.gz`
- `Flask_OpenID-1.2.5-py2-none-any.whl`
- `Flask_Testing-0.7.1-py2-none-any.whl`
- `WTForms-2.1-py2.py3-none-any.whl`
- `SQLAlchemy-1.1.10-cp27-cp27mu-linux_x86_64.whl`
- `python-openid-2.2.5.tar.gz`
- `connexion-1.0.129-py2-none-any.whl`
- `Flask-0.10.1-py2-none-any.whl`
- `Flask_Mail-0.9.1-py2-none-any.whl`
- `itsdangerous-0.22-py2-none-any.whl`
- `MarkupSafe-0.11.tar.gz`
- `functools32-3.2.3-2.tar.gz`
- `pathlib-1.0.1.tar.gz`
- `ijson-3.1.tar.gz`

D.2. ppc64le Architecture

HSB server wheel

- `TreefrogServer-1.1-py27-none-any.whl`

Db2 server and license files

- `v11.1.4fp4a_linuxppc64le_universal_fixpack.tar.gz`
- license files:
  - `db2ese_c.lic` - Db2 Enterprise Server license
  - `db2hc.lic`- Db2 High Capacity license
  - `iidr.lic` - Infosphere Data Replication license

HSB server RPM files

- `python2-pip-8.1.2-5.el7.noarch.rpm`

XLC

- `XL_C_C_FOR_LINUX_13.1.6_PRODUCT.tar.gz`

Zookeeper installation material

- `zookeeper-3.5.4-beta.tar.gz`

HSB server prerequisite wheels

- `Babel-2.6.0-py2.py3-none-any.whl`
- `certifi-2017.7.27.1-py2.py3-none-any.whl`
- `cffi-1.12.3-cp27-cp27mu-linux_ppc64le.whl`
- `chardet-3.0.4-py2.py3-none-any.whl`
- `click-6.7-py2.py3-none-any.whl`
- `clickclick-1.2.2-py2.py3-none-any.whl`
- `colorama-0.3.9-py2.py3-none-any.whl`
- `enum34-1.1.6-py2-none-any.whl`
- `flake8-2.6.0-py2.py3-none-any.whl`
- `Flask_Babel-0.11.2-py2.py3-none-any.whl`
- `Flask_WTF-0.14.2-py2.py3-none-any.whl`
- `funcsigs-1.0.2-py2.py3-none-any.whl`
- `future-0.17.1-cp27-none-any.whl`
- `idna-2.8-py2.py3-none-any.whl`
- `jsonschema-2.6.0-py2.py3-none-any.whl`
- `kazoo-2.6.0-py2.py3-none-any.whl`
- `mccabe-0.5.3-py2.py3-none-any.whl`
- `mock-2.0.0-py2.py3-none-any.whl`
- `pbr-5.1.1-py2.py3-none-any.whl`
- `pip-9.0.1-py2.py3-none-any.whl`
- `pycodestyle-2.0.0-py2.py3-none-any.whl`

- pycparser-2.19-py2.py3-none-any.whl
- pycrypto-2.6.1-cp27-cp27mu-linux_ppc64le.whl
- pyflakes-1.2.3-py2.py3-none-any.whl
- PyJWT-1.7.1-py2.py3-none-any.whl
- python_dateutil-2.7.2-py2.py3-none-any.whl
- requests-2.21.0-py2.py3-none-any.whl
- setuptools-28.8.0-py2.py3-none-any.whl
- swagger_spec_validator-2.4.1-py2.py3-none-any.whl
- typing-3.6.2-py2-none-any.whl
- urllib3-1.24.1-py2.py3-none-any.whl
- Werkzeug-0.12.2-py2.py3-none-any.whl
- PyYAML-3.13-cp27-cp27mu-linux_ppc64le.whl
- blinker-1.4-py2-none-any.whl
- Flask_SQLAlchemy-2.0-py2-none-any.whl
- Jinja2-2.7.2-py2-none-any.whl
- Flask_AppBuilder-1.8.1-py2-none-any.whl
- ibm_db-2.0.8-cp27-cp27mu-linux_ppc64le.whl
- ibm_db_sa-0.3.2-py2-none-any.whl
- Flask_Login-0.2.11-py2-none-any.whl
- strict-rfc3339-0.7.tar.gz
- Flask_OpenID-1.2.5-py2-none-any.whl
- Flask_Testing-0.7.1-py2-none-any.whl
- WTForms-2.1-py2.py3-none-any.whl
- SQLAlchemy-1.1.10-cp27-cp27mu-linux_ppc64le.whl
- python-openid-2.2.5.tar.gz
- connexion-1.0.129-py2-none-any.whl
- Flask-0.10.1-py2-none-any.whl
- Flask_Mail-0.9.1-py2-none-any.whl

- `itsdangerous-0.22-py2-none-any.whl`
- `MarkupSafe-0.11.tar.gz`
- `functools32-3.2.3-2.tar.gz`
- `pathlib-1.0.1.tar.gz`
- `ijson-3.1.tar.gz`

Appendix E. Glossary of terms and acronyms

ABRT	Automatic Bug detection and Reporting Tool
Catalog	A complete list of the files, objects, directories, containers and chunks that comprise a Managed Dataset. This list is stored in a catalog file.
Class of Service	A set of storage system characteristics used to group HPSS bitfiles with similar logical characteristics and performance requirements together. A Class of Service is supported by an underlying hierarchy of storage classes.
Chunk	Contiguous data within a managed dataset. Fragments are composed of one or more chunks. Fragments can be broken into multiple chunks to facilitate device storage capacity limitations.
Copier	Component of the HSB service that creates and recalls managed datasets and lists contents of storage endpoints.
COS	Class of Service
CRC	Cyclic Redundancy Check
Credential expirer	Component of the HSB server that monitors endpoint credentials and notifies the user when credentials are older than the configured expiration period.
Db2	A relational database system, a product of IBM Corporation, used by HSB to store and manage HSB system metadata.
Directory, Container	The container components of a file system and object store, respectively.
DNS	Domain Name Service
DOE	Department of Energy
ECC	Error Correction Code
EOM	End of Media
File, Object	Data components of a file system and object store, respectively.
File family	An attribute of an HPSS file that is used to group a set of files on a common set of tape virtual volumes.
Fragment	Logically contiguous data within a managed dataset. Managed datasets can be fragmented into some number of relatively equal pieces to facilitate increased transfer performance via concurrent, parallel transfers and provide redundancy via the generation of parity fragments.
FTP	File Transfer Protocol
GUI	Graphical User Interface
HADR	Db2 High Availability Disaster Recovery
HPSS	High Performance Storage System

HPSS bitfile	A file stored in HPSS, represented as a logical string of bits unrestricted in size or internal structure. HPSS imposes a size limitation in 8-bit bytes, based upon the maximum size in bytes that can be represented by a 64-bit unsigned integer.
HPSS Storage Broker (HSB)	High Performance Storage System Storage Broker
HSB service	On-premise service that allows users to copy data between defined storage systems in a high-performance manner.
HTTP	Hyper Text Transmission Protocol
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
Instance	Instance of a HSB project name space.
I/O	Input/Output
IP	Internet Protocol
ISA	Intel ® Intelligent Storage Acceleration
JRE	Java Runtime Environment
LAN	Local Area Network
LANL	Los Alamos National Laboratory
LBP	Logical Block Protection
LDAP	Lightweight Directory Access Protocol
LLNL	Lawrence Livermore National Laboratory
LTO	Linear Tape-Open. A half-inch open tape technology developed by IBM, HP, and Seagate.
Manifest	A listing of the files and objects that comprise a managed dataset.
MAC	Mandatory Access Control
Managed Data Set (MDS)	Immutable collection of files or objects managed by the HSB service. Managed data sets are intended as a mechanism to group logically associated data and emphasize the desirable characteristics of high-latency, high-capacity storage. User-defined metadata can be associated with each managed dataset to facilitate efficient location and retrieval.
Name space	An organization of projects and managed datasets, so that these components can be referred to by name.
NASA	National Aeronautics and Space Administration
NASM	Netwide Assembler is an assembler for the x86 CPU architecture.
NERSC	National Energy Research Supercomputer Center
NIS	Network Information Service
NLS	National Language Support
NSL	National Storage Laboratory
ORNL	Oak Ridge National Laboratory
PFTP	Parallel extensions to File Transfer Protocol supported by HPSS

PFTPD	PFTP Daemon
POSIX	Portable Operating System Interface (for computer environments).
Project	Used to group managed datasets in the HSB name space and provide access control. Users can be granted specific permissions for a project and by association permissions on managed datasets in the project. Projects represent a static name space container. All managed datasets associated with a project are located within the project's HSB name space container.
RAO	Recommended Access Order
Repository storage endpoint	Storage system that can be used by the HSB service as a target for managed dataset creation requests or a source for managed dataset recall requests. These endpoints are represented in HSB service configuration data.
Request processor	Component of the HSB service that manages resources and schedules requests.
Server	HSB server is the component of the HSB service that orchestrates data transfers, provides metadata management, and graphical user interfaces for administrators and users.
SNIA	Storage Networking Industry Association
SNL	Sandia National Laboratories
Storage Policy	Defines storage characteristics and retrieval behavior for Managed Datasets, including repository storage endpoints, number of copies, fragmentation, parity, preferred recall copy, and recall priority.
Static storage endpoint	Storage endpoint that is represented in the system by configuration data that is used to connect to and access the storage system. These endpoints are used as data sources for managed dataset create requests and targets for managed dataset recall requests.
TCP/IP	Transmission Control Protocol/Internet Protocol
UDA	User-Defined Attribute
URI	Uniform Resource Identifier
User	An identity registered with the HSB system.
UUID	Universal Unique Identifier
Web services	Web service component providing HTTP communication with the HSB service.
XML	Extensible Markup Language
YASM	The Yasm Modular Assembler Project. A complete rewrite of the NASM assembler under the new BSD license.

Appendix F. References

1. **HPSS Error Messages Reference Manual**, current release
2. **HPSS User's Guide**, current release
3. **HPSS Storage Broker User's Guide**, current release
4. **HPSS Storage Broker Installation Guide**, current release
5. **HPSS Storage Broker Administrator's Guide**, current release
6. **SNIA Self-contained Information Retention Format (SIRF v1.0)**
7. R.W. Watson and R.A. Coyne, "**The Parallel I/O Architecture of the High-Performance Storage System (HPSS)**," from the 1995 IEEE MSS Symposium, courtesy of the IEEE Computer Society Press.

Appendix G. Developer acknowledgments

HPSS Storage Broker is a product of a government-industry collaboration. The project approach is based on the premise that no single company, government laboratory, or research organization has the ability to confront all of the system-level issues that must be resolved for significant advancement in high-performance storage system technology.

HPSS Storage Broker development was performed jointly by IBM Worldwide Government Industry, Lawrence Berkeley National Laboratory, Lawrence Livermore National Laboratory, Los Alamos National Laboratory, NASA Langley Research Center, Oak Ridge National Laboratory, and Sandia National Laboratories.