GHI Installation Guide

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Spectrum Scale HPSS Interface Release 3.0.1

May 2019

About this book
The Spectrum Scale (formerly GPFS) to HPSS Interface (GHI) Installation Guide is for use at system installation time. It outlines the steps required to install and configure a GHI system.

Conventions Used in This Document
Example commands that should be typed at a command line will be proceeded by a percent sign (%):
% sample command

Example command output and example contents of ASCII files are presented in *italic* font:
sample file line 1
sample file line 2

Any text preceded by a pound sign ('#') is considered comment lines:
This is a comment

A byte is an eight-bit data octet.
A kilobyte, KB, is 1024 bytes (2^{10} bytes).
A megabyte, MB, is 1,048,576 bytes (2^{20} bytes).
A gigabyte, GB, is 1,073,741,824 bytes(2^{30} bytes).
A terabyte, TB, is 1,099,511,627,776 bytes (2^{40} bytes).
A petabyte, PB, is 1,125,899,906,842,624 bytes (2^{50} bytes).
An exabyte, EB, is 1,152,921,504,606,846,976 bytes (2^{60} bytes).

Chapter 1

Preparing for GHI Install

Prior to installing GHI, a series of planning meetings should have been conducted to discuss the customer's requirements in order to properly size and configure the GHI system. The planning process is not a subject of this Installation Guide and is done during the Proposal and/or System Engineering phase of the project. In addition, *refer to the GHI Management Guide* for planning considerations.

IBM recommends you install GHI on a Spectrum Scale cluster that has no other Hierarchical Storage Management (HSM) application running, for example, Tivoli Storage Manager (TSM). If another HSM managed file system is required, it must run on a separate cluster and be remotely mounted on the GHI managed cluster. GHI is dependent on timely Data Management Application Programming Interface (DMAPI) events from Spectrum Scale; therefore, there should not be two applications competing for events.

For systems installed with High Availability (HA) Core Server, it is critical to ensure that the required GHI components are installed on the backup or stand-by Core Server. These components include Db2 accounts creation and configuration, Db2 Server configuration and Independent Standalone HPSS TAR (ISHTAR).

GHI installation requires root or root-equivalent privileges, except where noted otherwise.

1.1 Prerequisites

Before installing GHI, review the GHI Release Notes on the HPSS Admin wiki for prerequisites, special notes, and possible known issues for the version you plan to install. The Release Notes define the software version of each prerequisite software:

- HPSS Core Server and Movers
- Operating system
  - memcached
  - libmemcache
- Python – not covered in this document
- IBM_db egg (Python support for Db2) – not covered in this document
- Spectrum Scale
- Db2 Client
- HPSS Client
  - hpss-lib
  - hpss-lib-devel
- GHI-ISHTAR
- GHI

1.2 Operating System

1.2.1 Set ulimits

Note
Change the default soft and hard core size from "0" to "unlimited". This will allow GHI to create a core dump file for debug purposes. The default inode scan bucket size is 1000. Increase the max open file descriptors limit to 65536 in /etc/security/limits.d/19-hpss.conf on all systems that will run GHI. Reboot each node to validate each change is correct and persistent.

Example:

```
% vi /etc/security/limits.d/19-hpss.conf
#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4
*               soft     core            unlimited  <- (add)
*               hard     core            unlimited  <- (add)
*               soft     nofile          65536  <- (add)
*               hard     nofile          65536  <- (add)
```

Validate each change by running:

```
$ ulimit -a
```

1.2.2 rsyslog configuration

Note
We recommend suppressing repeat messages and turning rate limiting off

Sites must evaluate policies and configuration needs for their own systems and determine what works best for their own needs. The below is an example:

1. In /etc/rsyslog.conf update or add the following lines:

   ```
   SystemLogRateLimitInterval 0
   SystemLogRateLimitBurst 0
   IMUXSockRateLimitInterval 0
   IMJournalRateLimitInterval 0
   IMJournalRateLimitBurst 0
   ```

2. In /etc/systemd/journald.conf update or add the following lines:

   ```
   RateLimitInterval=0
   RateLimitBurst=0
   Storage=volatile
   Compress=no
   RateLimitInterval=0
   MaxRetentionSec=5s
   ```

3. In /etc/rsyslog.d/hpss.conf update or add the following line:

```
RepeatedMsgReduction off
```

4. Restart the services for changes to take effect

```
systemctl restart systemd-journald
systemctl restart rsyslog
```

1.3 Memcached

Memcached is an in-memory key-value store for small chunks of arbitrary data. Memcached allows applications to take memory from parts of the system where it has more than it needs and make it accessible to areas where applications have less than they need.

GHI uses memcached to reduce the load on the HPSS metadata. Memcached improves the performance of GHI full backups, GHI image backup verification, ghiverifyfs, and ghi_ls. Install the memcached and libmemcached-devel RPMs from the RHEL software distribution on each machine you want memcached to run to improve the above operations.

1.3.1 Install memcached and libmemcached

1. Read the Release Notes to check prerequisites for appropriate version to use

```
% yum list available | grep memcached
% yum install memcached
% yum install libmemcached
```

2. Verify the packages and versions have been properly installed

```
% rpm -qa | grep memcached
```

1.3.2 Configure memcached

- Start the memcached daemon on each node.

```
% memcached -d -u root -m 1024
```

This will start the server with 1024 MB of memory. Adjust to the desired cache size. The rule of thumb is that 10,000 items in the cache will use approximately 16MB.

- Create directory /var/hpss/ghi/etc and file memcached.conf

```
On each GHI node:
% mkdir /var/hpss/ghi/etc/
% vi /var/hpss/ghi/etc/memcached.conf
```

- Add the following lines to memcached.conf:

```
--POOL-MIN=4
--POOL-MAX=32
--SERVER=<ghi node name>:11211
--SERVER=<ghi node name>:11211
```

- Verify memcache's port number in file /etc/services.
  If the port is not 11211, then substitute the correct port number in its place.

For RHEL 7 GHI Nodes, use the following steps:

- Run the following commands to enable, start, and status memcached.

```
% systemctl enable memcached.service
% systemctl start memcached.service
% systemctl status memcached.service
```

- Verify that memcached configuration files have been created.

```
% cd /usr/lib/systemd/system/
% ls | grep memcached
```

- If memcached.services does not exist, follow these steps below:
  - Create /usr/lib/systemd/system/memcached.service
  - Add the lines below to the file.

```
[Unit]
Description=Memcached
Before=httpd.service
After=network.target

[Service]
Type=simple
EnvironmentFile=-/etc/sysconfig/memcached
Restart=always
ExecStart=/usr/bin/memcached -u $USER -p $PORT -m $CACHESIZE -c $MAXCONN ↩
    $OPTIONS

[Install]
WantedBy=multi-user.target
```

  - Create the file /etc/sysconfig/memcached with contents:

```
PORT="11211"
USER="memcached"
MAXCONN="1024"
CACHESIZE="64"
OPTIONS=""
```

  - Check status and, if necessary, disable, enable, reload, and restart memcached.service

```
% systemctl list-unit-files | grep memcached
% systemctl enable memcached.service
% systemctl daemon-reload
% systemctl restart memcached.service
% systemctl status memcached.service
```

For RHEL 6 GHI Nodes, use the following steps:

- Create /etc/init/memcached.conf and add the lines below to the file.

```
stop on runlevel [012]
start on starting rc RUNLEVEL=[345]
respawn
exec /usr/bin/memcached -d -u root -m 1024
```

- Run the memcached command to start.

```
% memcached -d -u root -m 1024
```

1.4 GHI-ISHTAR

Before installing GHI-ISHTAR, verify that prerequisites hpss-lib, hpss-lib-devel, and hpss-clnt are installed.

```
% rpm -qa "hpss*"
```

- Install prerequisites if they are missing

```
% rpm -ivh hpss-lib-<version>*
% rpm -ivh hpss-lib-devel-<version>*
% rpm -ivh hpss-clnt-<version>*
```

After HPSS RPMs are installed, a message will appear letting the user know where the package directory is located. This directory path will be needed for the next step.

```
root@elayne /hpss_src/hpss753 > rpm -ivh hpss-clnt-7.5.3.0-0.el7.ppc64le.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:hpss-clnt-7.5.3.0-0.el7          ################################# [100%]
Files for package hpss-clnt installed under /hpss_src/hpss-7.5.3.0-0.el7
```

- Create /opt/hpss link to the directory where HPSS Client files are installed.

```
% ln -s /hpss_src/hpss-<version>* /opt/hpss
```

Example:

```
% ln -s /hpss_src/hpss-7.5.3.0-0.el7 /opt/hpss
```

1.5 Install GHI-ISHTAR

GHI-ISHTAR must be installed on all GHI IOM nodes. The GHI-ISHTAR RPM is compatible with GHI and is provided by IBM. Only the provided version should be used with GHI. Check the GHI Release Notes for the version of GHI-ISHTAR to install.

```
$ rpm -ivh ghi-ishtar*.rpm
```

Files for package ghi-ishtar installed under /var/hpss/hsi

Note
HPSS libraries must be installed on each GHI IOM node before GHI-ISHTAR can be installed.

Chapter 2

Spectrum Scale

2.1 Install Spectrum Scale

Contact your IBM Spectrum Scale customer support representative to obtain the Spectrum Scale software and install it according to instructions.

2.2 Configure Spectrum Scale

1. After Spectrum Scale is installed, make sure ssh or rsh is working between nodes in the cluster. If using ssh, be certain to complete additional configuration steps to allow for passwordless command execution (steps are covered in the Spectrum Scale documentation).
2. Enable threshold processing.
   Check to see if the requested configuration attributes are set.

   ```
   % mmlsconfig
   % mmchconfig enablelowspaceevents=yes
   ```

3. Configure NSD Multipath.
   If using multipath, follow the steps below to create NSDs.

   a. Create a /etc/multipath/bindings file. The file needs to match on all nodes using the NSD.
   b. Create an nsddevices script for NSD Device Discovery

      ```
      % cp /usr/lpp/mmfs/samples/nsddevices.sample /var/mmfs/etc/nsddevices
      ```

   c. Edit /var/mmfs/etc/nsddevices to look like the example below:

      ```
      osName=$(/bin/uname -s)

      if [[ $osName = Linux ]]
      then
      CONTROLLER_REGEX='mpath[a-z]+'
      for dev in $( /bin/ls /dev/mapper | egrep $CONTROLLER_REGEX )
      do
      # dmm vs. generic is used by Spectrum Scale to prioritize internal order
      # of searching through available disks, then later Spectrum Scale
      # discards other disk device names that it finds that match as the
      # same NSD device by a different path. For this reason,
      # dmm vs. generic is an important distinction if you are not
      ```

```
# explicitly producing the entire and exclusive set of disks
# that Spectrum Scale should use, as output from this
# script (nsddevices) and exiting this script with a "return 0".
echo mapper/$dev dmm
echo mapper/$dev generic
done
fi

# To bypass the Spectrum Scale disk
# discovery (/usr/lpp/mmfs/bin/mmdevdiscover),
return 0
# To continue with the Spectrum Scale disk discovery steps,
return 1
```

d. Ensure this script is executable.

```
% chmod +x /var/mmfs/etc/nsddevices
```

e. Execute /var/mmfs/etc/nsddevices.
Example:

```
% /var/mmfs/etc/nsddevices
mapper/mpatha dmm
mapper/mpathb dmm
mapper/mpathc dmm
```

2.3 Create SSH Trust

After Spectrum Scale is installed, create ssh trust between all nodes in each direction in each node and between each node. Be certain to complete additional configuration steps to allow for passwordless command execution.

2.4 Create a new GPFS Cluster

Only on the main node run the command to create a GPFS cluster. Upon successful completion of the mmcrcluster command, the /var/mmfs/gen/mmsdrfs and the /var/mmfs/gen/mmfsNodeData files are created on each node in the cluster.

1. Run mmcrcluster * Example and output:

```
% mmcrcluster -n /var/hpss/ghi/gpfs_config/node.conf -p ghi_server1 \
-r /usr/bin/ssh -R /usr/bin/scp
```

2. Check that mmsdrfs and mmfsNodeData files are created and the output shows success and completion.

```
% cat /var/mmfs/gen/mmsdrfs
% cat /var/mmfs/gen/mmfsNodeData
```

Output:

```
mmcrcluster: Performing preliminary node verification ...
mmcrcluster: Processing quorum and other critical nodes ...
mmcrcluster: Finalizing the cluster data structures ...
mmcrcluster: Command successfully completed
mmcrcluster: Warning: Not all nodes have proper GPFS license designations.
mmcrcluster: Propagating the cluster configuration data to all affected nodes.
This is an asynchronous process.
```

2.4.1 Configure license

The mmchlicense command designates appropriate GPFS licenses. Run mmchlicense to accept and configure licenses.

```
% mmchlicense server --accept -N all
```

Output:
The following nodes will be designated as possessing server licenses:

```
ghi_server2.clearlake.ibm.com
ghi_server1.clearlake.ibm.com
```

2.4.2 Create NSD (Network Shared Disks) on main the GHI node only

- On the primary GHI Node, create NSD configuration file(s) for each disk.

```
% cd /var/hpss/ghi/gpfs_config
% touch nsd.StanzaFile nsd.StanzaFile2 ... nsd.StanzaFileX
% vi nsd.StanzaFile
```

- Add the following lines:

```
%nsd:
device=/dev/sdb
nsd=nsd1
servers=ghi_server1
usage=dataAndMetadata
% vi nsd.StanzaFile2
```

- Add the following lines:

```
%nsd:
device=/dev/sdc
nsd=nsd2
servers=ghi_server1
usage=dataAndMetadata
```

Note
Create a block for each resource. Include all GHI nodes that see the disk separated by a comma. For example, if two servers share a disk resource "servers=" value, the line will contain both hostnames like this: servers=<node1 shortname>,<node2 shortname>

- Create NSD stanzas file that uses the multipath aliases.
  For systems using multipath, skip this step if you are not using multipath.
  Edit /var/hpss/ghi/gpfs_config/nsd.StanzaFile and insert the lines below.

```
%nsd: device=/dev/mapper/mpatha
nsd=nsd1
servers=ghi_server1,ghi_server2
usage=dataAndMetadata
```

- Enable DMAPI on the Spectrum Scale file system.

```
% mmchfs <file system> -z Yes
```

- Run the mmcrnsd command to create network shared disk (NSD) servers The option "-F" specifies the file containing the NSD stanzas for the disks to be created. The option "-v no" specifies that the disks are to be created irrespective of their previous state.

```
% mmcrnsd -F /var/hpss/ghi/gpfs_config/nsd.StanzaFile -v no

mmcrnsd: Processing disk sdb
mmcrnsd: Propagating the cluster configuration data to all affected nodes.
This is an asynchronous process.

% mmcrnsd -F /var/hpss/ghi/gpfs_config/nsd.StanzaFile2 -v no

mmcrnsd: Processing disk sdc
mmcrnsd: Propagating the cluster configuration data to all affected nodes.
This is an asynchronous process.
```

- Create GPFS file system

  First ensure all the GPFS nodes are active and then create the GPFS file system. Wait until the mmgetstate output shows that all nodes are active before issuing the mmcrfs command.

```
% mmgetstate -a

Node number  Node name        GPFS state

1      ghi_server1            active
2      ghi_server2           active
```

If the node state remains down, run *mmstartup -a* to start GPFS
If the node state remains down after *mmstartup*, check GPFS logs
If the node state is arbitrating, check GPFS logs
If the node needs to be recycled, run *mmshutdown -a*, and rerun mmstartup

---

Note
The GFPS log location is /var/mmfs/gen/mmfslog.

- Run mmcrfs to create the file system(s) with options to enable automount (-A yes), activate quotas automatically (-Q yes), enable DMAPI (-z yes), set blocksize (-B 256K), and specify disk to not belong to an existing file system (-v no).

```
% mmcrfs /ghi_server1_fs1 /dev/ghi_server1_fs1 -F /var/hpss/ghi/gpfs_config/nsd. ↩
   StanzaFile \
-A yes -Q yes -z yes -B 256K -v no

%  mmcrfs /ghi_server1_fs2 /dev/ghi_server1_fs2 -F /var/hpss/ghi/gpfs_config/nsd. ↩
   StanzaFile2 \
-A yes -Q yes -z yes -B 256K -v no
```

Note
If the user plans on having a GPFS file system without a GHI file system for image restores, the "temp space" GPFS file system should have DMAPI set to no (-z no).

Sample Output:

```
The following disks of ghi_server1_fs2 will be formatted on node
ghi_server2.clearlake.ibm.com:
    nsd2: size 153600 MB
Formatting file system ...
Disks up to size 1.51 TB can be added to storage pool system.
Creating Inode File
Creating Allocation Maps
Creating Log Files
Clearing Inode Allocation Map
Clearing Block Allocation Map
Formatting Allocation Map for storage pool system
Completed creation of file system /dev/ghi_server1_fs2.
mmcrfs: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

Note
Use mmlsfs to list the file system attributes. For example, if you want to check if DMAPI is enabled on all GPFS file systems, run: *mmlsfs all | grep DMAPI*

- Display the configuration data for a GPFS Cluster for each node

  Log in to main node:

  ```
  % root@ghi_server1 /var/mmfs > mmlsconfig
  ```

```
Configuration data for cluster ghi_server1.clearlake.ibm.com:

clusterName ghi_server1.clearlake.ibm.com
clusterId 16335425671093415616
autoload no
dmapiFileHandleSize 32
minReleaseLevel 5.0.2.0
ccrEnabled yes
cipherList AUTHONLY
adminMode central

File systems in cluster ghi_server1.clearlake.ibm.com:

/dev/ghi_server1_fs1
/dev/ghi_server1_fs2
```

Log in to all secondary nodes to check

```
% root@ghi_server2 /root > mmlsconfig
```

```
Configuration data for cluster ghi_server1.clearlake.ibm.com:

clusterName ghi_server1.clearlake.ibm.com
clusterId 16335425671093415616
autoload no
```

```
dmapiFileHandleSize 32
minReleaseLevel 5.0.2.0
ccrEnabled yes
cipherList AUTHONLY
adminMode central

File systems in cluster ghi_server1.clearlake.ibm.com:

/dev/ghi_server1_fs1
/dev/ghi_server1_fs2
```

2.4.3 NSD Multipath

1. Configure NSD Multipath.
   If using multipath, follow the steps below to create NSDs.

   a. Create a /etc/multipath/bindings file. The file needs to match on all nodes using the NSD.

   b. Create an nsddevices script for NSD Device Discovery

   ```
   % cp /usr/lpp/mmfs/samples/nsddevices.sample /var/mmfs/etc/nsddevices
   ```

   c. Edit /var/mmfs/etc/nsddevices to look like the example below:

   ```
   osName=$(/bin/uname -s)

   if [[ $osName = Linux ]]
   then
   CONTROLLER_REGEX='mpath[a-z]+'
   for dev in $( /bin/ls /dev/mapper | egrep $CONTROLLER_REGEX )
   do
   # dmm vs. generic is used by Spectrum Scale to prioritize internal
   # order of
   # searching through available disks, then later Spectrum Scale
   # discards other disk device names that it finds that match as the same
   # NSD device by a different path. For this reason, dmm vs. generic is an
   # important distinction if you are not explicitly producing the entire
   # and exclusive set of disks that Spectrum Scale should use,
   #  as output from
   # this script (nsddevices) and exiting this script with a "return 0".
   echo mapper/$dev dmm
   echo mapper/$dev generic
   done
   fi

   if [[ $osName = AIX ]]
   then:
   # Add function to discover disks in the AIX environment.
   fi

   # To bypass the Spectrum Scale disk discovery
   # (/usr/lpp/mmfs/bin/mmdevdiscover),
   return 0
   # To continue with the Spectrum Scale disk discovery steps,
   return 1
   ```

d. Ensure the script is executable.
Example:

```
% chmod +x /var/mmfs/etc/nsddevices
```

e. Execute /var/mmfs/etc/nsddevices.
Example:

```
# /var/mmfs/etc/nsddevices
mapper/mpatha dmm
mapper/mpathb dmm
mapper/mpathc dmm
```

f. Create NSD stanzas file that uses the multipath aliases.
Edit /var/hpss/ghi/gpfs_config/nsd.StanzaFile and insert the lines.

```
%nsd: device=/dev/mapper/mpatha
nsd=nsd1
servers=ghi_server1,ghi_server2
usage=dataAndMetadata
```

g. Continue with Creating NSDs.

2. Enable DMAPI on the Spectrum Scale file system.

```
% mmchfs <file system> -z Yes
```

Chapter 3

Db2

3.1 Users and Groups

GHI needs three users (hpss, hpssdb, hpssdmg) and two groups (hpss, hpsssrvr) on all GHI nodes that will have the HPSS client installed. The user and group ID numbers created on the GHI nodes must match the corresponding user and group ID numbers on the HPSS Core Server. User IDs hpss and hpssdb should exist after the HPSS Core Server has been installed and configured. The user ID hpssdmg will need to be created on the HPSS Core Server using hpssuser.

- Use the system command **id** to verify the required users and groups exist.

```
% id <user>
% id -g <user>
```

User	Primary Group	Home Directory:
hpss	hpss	/var/hpss
hpssdb	hpssdb	/db2data/db2_hpssdb
hpssdmg	hpsssrvr	/var/hpssdmg

- If any of the above users or groups do not exist, use the **useradd** system command to add them.
  The following shows the usage of the useradd command and an example adding hpssdb as a user and group:

```
% useradd -d <home directory> -g <group> -p password <user>

% useradd -d /db2data/db2_hpssdb -g 300 -p hpssdb hpssdb
```

- Check to make sure Core Server and GHI nodes have matching entries for users hpss, hpssdb, and hpssdmg in the /etc/passwd and /etc/group files.

```
% cat /etc/passwd | grep hpss
hpss:x:300:300:HPSS User:/var/hpss:/bin/bash
hpssdba:x:301:301::/db2data/db2_hpssdb:/bin/bash
hpssdmg:x:1001:302::/var/hpssdmg:/bin/bash

% cat /etc/group | grep hpss
hpss:x:300:hpss,hpssdba
hpssdba:x:301:root
hpsssrvr:x:302:hpssdmg
```

The hpssdmg user in /etc/passwd is in the primary group of hpsssrvr. Also notice that in /etc/group hpssdmg is a secondary group to hpsssrvr. Make sure all Core and GHI nodes have the same configuration and passwords.

3.1.1 Add hpssdmg with hpssuser tool

- On the HPSS Core Server, use **hpssuser** to add user hpssdmg with **Unix Authentication**.
  Skip to the next bullet if your site uses **Kerberos Authentication**.

```
$ /opt/hpss/bin/hpssuser -add hpssdmg -unix
User ID#: 1001
Primary group name: hpsssrvr
Enter password for hpssdmg: [hpssdmg]
Re-enter password to verify: [hpssdmg]
Full name: hpssdmg
Login shell: /bin/bash
Unix (local/system) home directory: /var/hpssdmg
[ adding unix user ]
[ added unix user ]
```

- On the HPSS Core Server, use **hpssuser** to add user hpssdmg with **Kerberos Authentication**.
  If your site uses Unix Authentication, go to the previous step.

```
$ /opt/hpss/bin/hpssuser -add hpssdmg -unix -krb -krbkeytab
/var/hpss/etc/hpss.keytab
User ID#: 1001
Primary group name: hpsssrvr
Enter password for hpssdmg: [hpssdmg]
Re-enter password to verify: [hpssdmg]
Full name: hpssdmg
Login shell: /bin/bash
Unix (local/system) home directory: /var/hpssdmg
[ adding unix user ]
[ added unix user ]
[ adding kerberos principal ]
INFO: Using kadmin.local for kerberos administrative actions
[ adding kerberos keytab entry to '/var/hpss/etc/hpss.keytab' ]
[ added kerberos keytab entry to '/var/hpss/etc/hpss.keytab' ]
[ added kerberos principal ]
```

- Check that hpssdmg has been added to /var/hpss/etc/passwd and to /var/hpss/etc/group under the group hpsssrvr.
  This step is valid only if you are using HPSS local password and group files. Otherwise, skip this step.

```
% cat /var/hpss/etc/passwd | grep hpssdmg
hpssdmg:x:1001:301:hpssdmg:/var/hpssdmg:/bin/bash
```

```
% cat /var/hpss/etc/group | grep hpssdmg
hpsssrvr:*:301:hpssmvr,hpsssd,hpssftp,hpssssm,hpsspvr,hpssgk,hpssmps,
hpssrait,hpsscore,hpsspvl,hpssfs,hpssls,hpssdmg
```

- Copy HPSS Core /var/hpss/etc/ to each GHI node with scp.

  On the core:

```
% cd /var/hpss/etc
% tar -cvzf /tmp/etcnew.tar.gz ./
% scp /tmp/etcnew.tar.gz root@<GHI NODE>:/var/hpss
```

- Move old /var/hpss/etc and make room for new one.
  On each GHI node:

```
% cd /var/hpss/
% mv etc etc.ori
% mkdir /var/hpss/etc
% cp /var/hpss/etcnew.tar.gz /var/hpss/etc
% cd /var/hpss/etc
% tar -xzvf etcnew.tar.gz
```

- Link /var/hpss/hpssdb to the hpssdb user's home directory.
  On each GHI Node:

```
$ ln -s /db2data/db2_hpssdb /var/hpss/hpssdb
```

3.2 Add hpssdmg to HPSS ACL on Core Server

Execute the HPSS ACL utility on your HPSS Core Server to add user hpssdmg, if necessary. Look at the output and choose the Account Validation Interface menu option. Next run the *show* command to determine if hpssdmg needs to be added. The order of ACL entries may differ from the following example:

```
% /opt/hpss/bin/hpss_server_acl

hsa> acl -t CORE

1) PVL Mount Notification Interface (v1) 007ff347-e533-1cc6-b22d-02608c2cedf4
2) Client Interface (v1) 32ba9692-4667-11d6-aa3a-0004ac49692b
3) Account Validation Interface (v1) 647f22a8-a1e9-11d3-a739-000001341966
4) Realtime Monitor Interface (v1) 80c9a256-2f13-11d3-a0c8-000001341966
Select an interface
Choose an item by number (RET to cancel):
> 3
hsa> show
```

```
perms - type - ID (name) - realm ID (realm)
=========================================
r--c--- - user - 302 (hpssftp) - 10000 (<core_server>.clearlake.ibm.com)
r--c--- - user - 306 (hpssfs) - 10000 (<core_server>.clearlake.ibm.com)
rw-c-dt - user - 307 (hpssmps) - 10000 (<core_server>.clearlake.ibm.com)
rw-c-d- - user - 312 (hpssssm) - 10000 (<core_server>.clearlake.ibm.com)
------t - any_other
```

```
hsa> add user hpssdmg rwc
hsa> show
```

```
perms - type - ID (name) - realm ID (realm)
=========================================
r--c--- - user - 302 (hpssftp) - 10000 (<core_server>.clearlake.ibm.com)
r--c--- - user - 306 (hpssfs) - 10000 (<core_server>.clearlake.ibm.com)
rw-c-dt - user - 307 (hpssmps) - 10000 (<core_server>.clearlake.ibm.com)
rw-c-d- - user - 312 (hpssssm) - 10000 (<core_server>.clearlake.ibm.com)
rw-c--- - user - 1001 (hpssdmg) - 10000 (<core_server>.clearlake.ibm.com)
------t - any_other
```

```
hsa> quit
```

3.3 Set up GHI Tablespace on HPSS Core Server

GHI should be configured to use the same Db2 storage group that is used in HPSS.

Important
GHI Tablespaces should be configured on HPSS Core Server only while the HPSS system is down. The actual configuration for Db2 should be determined during the system engineering planning phase of the deployment. The GHI Db2 mapping table has the potential to become very large and care should be taken in configuring Db2 to handle it.

Note
Repeat this section to set up the GHI tablespace on the HA Backup Core Server for proper failover operations.

3.3.1 Database using Single Partition

This configuration is performed only on the HPSS Core server while Db2 is running and HPSS servers are down.

1. Shut down all servers via HPSS GUI.
2. Find the number of partition(s)
   As hpssdb user, the following shows there is only one partition.

   ```
   % cat $HOME/sqllib/db2nodes.cfg
   0 <core_server>.clearlake.ibm.com 0
   ```

3. Source the database profile.

   ```
   % source ~hpssdb/sqllib/db2profile
   ```

4. Create the database.
   This is the default example for a one partition and two storage paths file systems.
   For systems that do not use the default, edit path partition names and storage path file systems to match your system configuration.
   The following examples show path names and partition expressions usage:

   ```
   % db2 "CREATE DATABASE HGHI ON \
   \'/db2data/p0000/stg0001', \
   \'/db2data/p0000/stg0002' \
   DBPATH on '/db2data/db2_hpssdb'"
   ```

   ```
   % db2 "CREATE DATABASE HGHI ON \
   \'/db2data/p \$4N /stg0001\', \
   \'/db2data/p \$4N /stg0002\' \
   DBPATH ON '/db2data/db2_hpssdb'"
   ```

3.3.2 CREATE DATABASE PARTITION GROUP

1. Connect to the HGHI database.

   ```
   % db2 CONNECT TO HGHI
   ```

2. For a single partition run the command:

```
% db2 "CREATE DATABASE PARTITION GROUP HPSS_GHI ON DBPARTITIONNUM (0)"
```

3. Check that a partition is created.

```
$ db2 list db partition groups
```

Example output:

```
DATABASE PARTITION GROUP
----------------------------
HPSS_GHI
IBMCATGROUP
IBMDEFAULTGROUP

3 record(s) selected.
```

1. Create the bufferpool used for GHI DB table space.

```
% db2 "CREATE BUFFERPOOL SMALLTABLES \
   DATABASE PARTITION GROUP HPSS_GHI SIZE 1000 AUTOMATIC \
   PAGESIZE 4 K"
```

2. Create the bufferpool used for GHI mapping tablespace.

```
% db2 "CREATE BUFFERPOOL bp32k SIZE 10000 PAGESIZE 32K \
   DATABASE PARTITION GROUP HPSS_GHI SIZE 1000 AUTOMATIC \
   PAGESIZE 32 K"
```

3. Create Db2 tablespaces.

   a. Create Db2 tablespace for GHIDB.

```
% db2 "CREATE LARGE TABLESPACE GHIDB \
 IN DATABASE PARTITION GROUP HPSS_GHI \
 PAGESIZE 4K \
 MANAGED BY AUTOMATIC STORAGE \
 AUTORESIZE YES \
 INITIALSIZE 32M \
 MAXSIZE NONE \
 EXTENTSIZE 128 \
 PREFETCHSIZE AUTOMATIC \
 BUFFERPOOL "SMALLTABLES" \
 OVERHEAD 7.500000  \
 TRANSFERRATE 0.060000 \
 NO FILE SYSTEM CACHING \
 DROPPED TABLE RECOVERY ON \
 DATA TAG NONE"
```

   b. Create Db2 tablespace for GHIMAPPING.

```
% db2 "CREATE LARGE TABLESPACE GHIMAPPING
 IN DATABASE PARTITION GROUP HPSS_GHI \
 PAGESIZE 32K \
 MANAGED BY AUTOMATIC STORAGE \
 AUTORESIZE YES \
 EXTENTSIZE 128 \
 PREFETCHSIZE AUTOMATIC \
 BUFFERPOOL BP32K \
 DATA TAG NONE \
 OVERHEAD 7.500000 \
 TRANSFERRATE 0.060000 \
 MAXSIZE NONE  \
 NO FILE SYSTEM CACHING \
 DROPPED TABLE RECOVERY ON"
```

3.3.3 Configure Logging on the HPSS Core Server

- Grant user hpss access to the database.

```
% db2 "grant connect on database to user hpss"
% db2 "grant createtab on database to user hpss"
% db2 "grant dbadm on database to user hpss"
```

- Configure the primary logs, secondary logs, log archives, log file size, and number of logs similar to the standard of the HPSS databases.

```
% mkdir /db2data/p0000/db2_log/hghi
% db2 "update db cfg for hghi using NEWLOGPATH <primary_log_path> hghi"
% db2 "update db cfg for hghi using NEWLOGPATH '/db2data/p0000/db2_log/hghi'"
% mkdir /db2data/p0000/db2_logmirror/hghi
% db2 "update db cfg for hghi using MIRRORLOGPATH <secondary_log_path> hghi"
% db2 "update db cfg for hghi using MIRRORLOGPATH
'/db2data/db2_logmirror/hghi'"
% db2 "update db cfg for hghi using AUTO_MAINT off"
% db2 "update db cfg for hghi using AUTO_RUNSTATS off"
% db2 "update db cfg for hghi using AUTO_TBL_MAINT off"
% mkdir /db2data/p0000/db2_logarchive1/hghi
% db2 "update db cfg for hghi using LOGARCHMETH1 \
DISK:/ <primary_log_archive_path>/hghi/"
% db2 "update db cfg for hghi using LOGARCHMETH1 \
DISK:/db2data/p0000/db2_logarchive1/hghi/"
% mkdir /db2data/p0000/db2_logarchive2/hghi
% db2 "update db cfg for hghi using LOGARCHMETH2 \
DISK:/<secondary_log_archive_path>/hghi/"
% db2 "update db cfg for hghi using LOGARCHMETH2 \
DISK:/db2data/p0000/db2_logarchive2/hghi/"

% db2 "update db cfg for hghi using LOGFILSIZ 25000"
% db2 "update db cfg for hghi using LOGPRIMARY 10"
% db2 "update db cfg for hghi using LOGSECOND -1"
```

Table 3.1: LOGBUFSZ

Machine Memory	LOGBUFSZ <Table Value>
< 16 GB RAM	4096
16 - 64 GB RAM	8192
> 64 GB RAM	16384

```
% db2 "update db cfg for hghi using LOGBUFSZ <table value>"

% db2 "update db cfg for hghi using DFT_QUERYOPT 2"
```

- Disconnect from the database.

  ```
  % db2 disconnect all
  ```

3.4 Install Db2 Client on all GHI Nodes

Install the Db2 Client on each Spectrum Scale quorum node (all nodes which include "quorum" in the "Designation" column from the **mmlscluster** command). Follow the Db2 Command Reference document to install the server.

3.5 Add Db2 Permanent License on all GHI Nodes

Add a permanent license on each Spectrum Scale quorum node that has the Db2 client installed.

- Add License.

  ```
  % cd /opt/ibm/db2/<version>/adm
  % ./db2licm -a <path name to Db2 generic license file>/db2aese_c.lic
  ```

Note
The generic Db2 license file ("*/db2/license/db2ese.lic") can be found on the Db2 Installation CD or image. It can also be obtained by contacting your HPSS customer support representative.

Tip
Refer to the Db2 Command Reference document for more information on how to use the db2licm utility to manage the Db2 license. Create the Db2 database connection on the GHI Session nodes which should already have the Db2 client installed per the prerequisites.

- Create an instance as root.

  ```
  % /opt/ibm/db2/<version>/instance/db2icrt -a CLIENT -s client -u hpssdb hpssdb
  ```

- Source db2profile system wide to establish database environment.
  As root, add lines to aliases.sh.

```
$ su - root
$ vim /etc/profile.d/aliases.sh
. ~hpssdb/sqllib/db2profile
```

- Set DB2COMM.
  As hpssdb

```
% su - hpssdb
% db2set DB2COMM=tcpip
```

- Verify that DB2COMM is set to tcpip

```
% db2set -all

[i] DB2COMM=TCPIP
[g] DB2SYSTEM=ghi_server1.clearlake.ibm.com
```

- Verify the local services in /etc/services file for Db2 support
  As root, copy the Db2 service entries from the Core Server /etc/services file. The number of entries will differ based on configuration.
  Example output:

```
# Local services
db2c_hpssdb    59999/tcp
DB2_hpssdba     60000/tcp
DB2_hpssdba_1   60001/tcp
DB2_hpssdba_2   60002/tcp
DB2_hpssdba_END 60003/tcp
```

- Catalog the database profile.

```
% db2 catalog tcpip node $NODE remote $HPSS_CORE server $PORT

% db2 catalog tcpip node ghi_server2 remote <HPSS_Core_server> server 59999
DB20000I  The CATALOG TCPIP NODE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
```

Where:

```
$NODE = unique name; recommend using short host name of current machine.
$HPSS_CORE = hostname of the HPSS Core server.
$PORT = port number acquired from the Core server /etc/services file
Steps to check hpssdb port on core server:
1) Source the database profile.
% . ~hpssdb/sqllib/db2profile
2) Run the command
% db2 get dbm cfg | grep SVCENAME
3) Look at the value for the SVCENAME
TCP/IP Service name             (SVCENAME) = db2_hpssdb
4) Cat the /etc/services file and grep for the SVCENAME from above.
% cat /etc/services | grep db2_hpssdb
5) Use the port number found from the grep of the /etc/services file for $PORT.
```

- Catalog the database hghi

```
% db2 catalog db hghi as hghi at node $NODE

% db2 catalog db hghi as hghi at node ghi_server2
Db20000I  The CATALOG DATABASE command completed successfully.
Db21056W  Directory changes may not be effective until the directory cache is
refreshed.
```

a. Cycle Db2 on the HPSS Core Server to pick up the changes.

```
% rc.hpss stop
% rc.hpss start
```

b. Verify that Db2 client can connect to the Db2 server on the HPSS core machine.

```
% /opt/hpss/bin/ghi_db_test --connect
```

Chapter 4

HPSS

The HPSS Core server must also be able to connect to the network configured for the Spectrum Scale configuration. For example, if the Spectrum Scale cluster is configured exclusively on a data network, HPSS must be able to connect to that data network, even if the Spectrum Scale nodes also have an additional network to connect to the HPSS Core Server.

4.1 Verify HPSS rpms on all GHI NODES

Verify that the following RPMs are installed on all the GHI nodes:

```
% rpm -qa | grep hpss
        hpss-clnt-<version>*
        hpss-lib-<version>*
        hpss-lib-devel-<version>*
```

These should exist when GHI-ISHTAR was previously installed.

4.2 Configure HPSS Client

1. Set up /var/hpss/etc on GHI client machines.
   a. Verify that /var/hpss/etc/* was copied from the HPSS Core Server to each GHI node.
   b. Add HPSS_API_HOSTNAME=<long hostname> to /var/hpss/etc/env.conf
   c. Add HPSS_PTHREAD_STACK=524288 to /var/hpss/etc/env.conf
2. Set up authentication.
   Copy the HPSS PAM module (/etc/pam.d/hpss) from the HPSS Core Server to /etc/pam.d/hpss on all GHI nodes.
3. Set up links.

   ```
   % /opt/ibm/db2 > ln -s /opt/ibm/db2/<version> /opt/ibm/db2/default
   % /opt/hpss/db2 > ln -s /opt/ibm/db2/<version> /opt/hpss/db2/default
   ```

If using Kerberos authentication, copy /etc/krb5.conf from the HPSS Core Server to all GHI nodes.

1. Specify the *HPSS_NET_FAMILY*. Ensure that the HPSS client configuration has the correct HPSS_NET_FAMILY in /var/hpss/etc/env.conf. The default value is ipv4_only.
   Examples:

```
ipv6_only
ipv4_only
ipv6
```

Chapter 5

GHI Installation & Configuration

5.1 Install GHI

1. Install the following RPMs on all GHI nodes:

```
% rpm -ivh ghi-lib-*<architecture>.rpm
% rpm -ivh  ghi-*<architecture>.rpm
```

GHI files will be installed under /hpss_src/hpss-*.architecture

1. Create /opt/hpss/src/ghi link

```
% ln -s /hpss_src/hpss-*.architecture /opt/hpss/src/ghi
```

2. Verify that the following directories exist:

```
/opt/hpss/src/ghi
/opt/hpss/bin
/opt/hpss/lib
/usr/share/man/cat7
/var/hpss/ghi
/var/hpss/ghi/policy
/var/hpss/ghi/config
/var/hpss/ghi/config/templates
```

3. Create /var/hpss/ghi/etc directory

```
% mkdir /var/hpss/ghi/etc
```

5.1.1 Configure GHI-ISHTAR

1. Copy the htar.ksh wrapper script to /var/hpss/hsi/bin.

```
% cd /var/hpss/hsi/bin
% cp htar.ksh.template htar.ksh
% edit htar.ksh  (Variables to modify are described with example below)
% /bin/chmod 755 htar.ksh
```

2. Modify the htar.ksh script to provide correct values for the following information:

Variable	Description	Example
TMPDIR	Location of the temporary files. The amount of space required is based on the size of an aggregate, plus temporary files created for the data files.	export TMPDIR=/<Spectrum Scale_mount_point>/scratch/.ghi
DEFAULT_REALM	Realm name for location of the HPSS Core server. This name must match what is set for "site name" in /var/hpss/etc/site.conf from the Core Server.	if ["$DEFAULT_REALM" = ""]; then DEFAULT_REALM=core_server.clearlake.ibm.com fi
HPSS_AUTH_METHOD	Set this variable for desired authentication type <Unix or Kerberos>. This variable will determine your keytab file to be used.	export HPSS_AUTH_METHOD=unix
HPSS_KEYTAB_PATH	Location of keytab. Set this variable when using UNIX authentication (ex. /var/hpss/etc/hpssdmg.unix.keytab).	export HPSS_KEYTAB_PATH= /var/hpss/etc/hpssdmg.unix.keytab
HPSS_HOSTNAME	Interface to be used for the data path.	export HPSS_HOSTNAME=ghi_server1

5.2 GHI Users and Groups

All authentication and authorization are done using the hpssdmg principal. The numeric IDs must match those on the HPSS Core Server, which may be obtained from the “/etc/passwd” file on your HPSS Core Server.

1. Verify hpssdmg User ID exists on each GHI node.
2. Verify Group ID hpsssrvr is set for hpssdmg.

If the user hpssdmg or group hpsssrvr do not exist, create them.

5.3 Configure GHI

GHI is configured using command-line tools. All of the GHI commands discussed in this section are fully documented in the GHI Management Guide.

These are the steps to configure GHI:

1. Create GHI cluster from the Spectrum Scale configuration.
2. Add Spectrum Scale file system for GHI to manage.
3. Add IOMs for each GHI managed Spectrum Scale file system.

5.4 Create GHI Cluster

Define the overall cluster configuration, including the nodes which will be known to GHI (not necessarily all nodes known to Spectrum Scale). This is accomplished via the ghicrcluster command. The ghicrcluster command must run on the session node that is designated as "Cluster manager node". Use mmlsmgr command to determine which node is the cluster manager.

```
root@ghi_server1 /var/hpss/hsi/bin > mmlsmgr
file system      manager node
---------------- ------------------
ghi_server1_fs2        192.168.221.199 (ghi_server1)
ghi_server1_fs1        192.168.221.200 (ghi_server2)

Cluster manager node: 192.168.221.199 (ghi_server1)
```

In addition, Spectrum Scale must be running when defining the cluster configuration to GHI.

All nodes that are designated as quorum with the **mmlscluster** command must be listed after the cluster manager. This will allow GHI to assign them as a manager node in the case of a failover.

```
% ghicrcluster [-v] {--ppc_64 | --ppc_64le | --x86_64 <node>} ... <GHI_node> ..
% ghicrcluster -N [-v] {--ppc_64 | --ppc_64le | --x86_64 <node>} ... < ↩
   nodelist_file>
```

Where:

```
<node> = This is the node the source code was compiled on (ghilscluster).
<GHI_node> | <nodelist_file> = Node list of machines from mmlscluster
which will have the designation of "manager" in the command mmlscluster.
```

The below command is an example.

```
% ghicrcluster -v --ppc_64 firefly firefly falcon
```

After **ghicrcluster** returns "Done.", restart Spectrum Scale and GHI.

```
% ghishutdown -G
% ghistartup -G
```

Note
If **ghicrcluster** fails during the configuration, retry the configuration step with the "-r" option after the errors from the failure are resolved (ghicrcluster -r [-v]).

5.5 Create Spectrum Scale file systems GHI will manage

Use the command **ghiaddfs** for each file system to be created, which may be issued from any node in the cluster. File systems to be defined must NOT be mounted in Spectrum Scale when the **ghiaddfs** command is issued. **ghiaddfs** will supply default values for the file system which can be updated or changed with the command **ghichfs**.

For each file system, the name and mount point are to be supplied by the user. The ports to be used by the associated SD and ED may also be user-supplied or left to their default values.

```
% ghiaddfs [-v] <FS_Name> [-c "# <comment>"] <Mount_Point> [<SD_Port> <ED_Port>]
```

Note
<FS_Name> = The same as the Spectrum Scale configuration name.
<Mount_Point> = The same as the Spectrum Scale configuration mount point.
<SD_Port> = The default Scheduler Daemon port is 80x0, where *x* is the order in which file systems were configured. For example, 8010 for the first configured file system. GHI will assign a port if one is not specified.
<ED_Port> = The default Event Daemon Port is 80x1, where *x* is the order in which file systems were configured. For example, 8011 for the first configured file system. GHI will assign a port if one is not specified.

The below command is an example.

```
% ghiaddfs firefly /firefly
```

5.6 Create IOMs for each GHI Managed File systems

If you have multiple GHI nodes, you should create one IOM per GHI file system on each node. Each file system will use the same IOM port number across all nodes.

The default port selected for an IOM is 80*x*2, where ***x*** is the order in which the file system was configured (8012 for the first configured file system, 8022 for the second configured file system, etc.). For more details about ports, see the GHI Management Guide.

Refer to the GHI Management Guide for more details about **ghiaddiom**.

```
% ghiaddiom [-vd|D] <File System name> [-c "# <comment>"] <IOM Node> \
<active_on_session_node> <estimate_transfer_rate> <chunksize>
```

Where:

```
<File System name> = Name of the file system added with the ghiaddfs command.
<IOM Node> = Name of node the IOM will run.
<active_on_session_node> = Active state of the IOM on the manager session node.
<estimate_transfer_rate> = Estimated data transfer rate.
<chunksize> = maximum # of bytes to transfer per non-aggregate HPSS I/O request.
```

Example:

```
% ghiaddiom -v firefly firefly:8012 TRUE 1GB 1TB
```

5.7 Modify xinetd.conf for number of IOMs

Note
This setting should be adjusted whether using RHEL 6.X or RHEL 7.X

```
% vi /etc/xinetd.conf*
```

change *cps = 50 10* to *cps = <IOM Thread Pool Size * number of IOMs> 10*

5.8 Information Lifecycle Management (ILM) Policies

GHI makes use of Spectrum Scale ILM policies. A policy is a plain-text file that describes files and directories to be included or excluded from processing. IBM provides templates which you may use as a starting point to configure custom policies. These templates can be found in the /var/hpss/ghi/policy directory. Below is a list of policy templates.

migrate.policy
This file can be placed in any directory in the system. The policy should have separate rules for aggregates and non-aggregates. The script, ghi_migrate, gets invoked from the policy engine and requires a "-a" option to process aggregates.

reset_incomplete_migration.policy
Use this policy to reset files for which a migration was started but never completed. Such files will show as "[incompletely-migrated]" when listed with "ghi_ls -h". They are "migrated enough" such that Spectrum Scale will not select them to be re-migrated, and the migration-reset process will result in the files being set back to "un-migrated" so that Spectrum Scale will select them in the next applicable migration policy run. This file can be placed in any directory in the system.

recall.policy
The recall policy does not use a bulk size. The policy generates on list. That list is parsed into aggregates and non-aggregates. The recall.policy file can be placed in any directory in the system.

tape_smart_migration.policy
This is an example used to migrate files in a tape smart manner. Files are migrated by HPSS file families and by path name. This policy can be used in combination with the –split-filelists-by-weight option for mmapplypolicy to generate file lists that contain elements with the same WEIGHT value.

backup_migration.policy
The migration policy will run a full Spectrum Scale scan and will attempt to migrate any files that are not stored currently in HPSS. The policy file should be updated to reflect the migration rules used for this file system. The policy should be able to select every file that has not been migrated to HPSS and exclude any file which should not be migrated. Verify that the backup migration policy matches what is being backed up in the backup_metadata.policy to ensure that files which have not been migrated are included in the metadata backup.

backup_metadata.policy
This policy is used by the Spectrum Scale SOBAR mmimgbackup command. Spectrum Scale file system namespace and file metadata are sent to GHI and HPSS.

Important
Do not change the backup_metadata policy without contacting IBM GHI support.

backup_error.policy
The backup error policy contains the rules that are used to validate the capture of a file system's metadata.

Important
Do not change the backup_error policy without contacting IBM GHI support.

threshold.policy
The Spectrum Scale ILM threshold policy provides the capability for GHI to space manage the Spectrum Scale file system. New and modified Spectrum Scale files are copied to HPSS on a periodic basis. When the Spectrum Scale file system reaches a predefined space threshold, the Spectrum Scale ILM threshold policy is executed to identify file candidates whose data can be removed from the file system. This file must be copied from the /var/hpss/ghi/policy directory to /var/hpss/ghi/policy/<file system> and modified to be file system specific. The script, ghi_migrate, is invoked from the policy engine requires a "-p" option to punch holes in the file system.

Chapter 6

Backup & Restore

6.1 Backups

To backup a Spectrum Scale file system, use the GHI ghi_backup command line interface. The backup interface uses the Spectrum Scale mmimgbackup command, which uses the ILM policy management engine.

GHI backups use the Spectrum Scale snapshot feature to take a point-in-time image of the file system.
When running a backup:

1. A snapshot of the Spectrum Scale namespace is saved after the backup migration policy and any other running migration policies have completed.
2. The state of each of the files is saved.

Each file system to be backed up uses its own copy of each of the following backup policy templates that reside in the /var/hpss/ghi/policy directory:

backup_migration.policy:
The backup migration policy contains the migration rules for the Spectrum Scale file system to be backed up. The rules can migrate files as aggregates or non-aggregates. The rules must select all the files to be backed up.

backup_metadata.policy:
The backup metadata policy contains the rules that previous GHI versions need to capture a file system's metadata. The new image backup feature does not require a metadata policy for metadata backup. The metadata is contained in the image generated by Spectrum Scale as part of the backup process.

backup_error.policy:
The backup error policy contains the rules that are used to validate the capture of the file system's metadata.

Tip
IBM recommends running a daily backup. Checking the backup logs daily to correct any errors is a good practice to ensure successful backups. The GHI backup option is image. Full non-image backup is deprecated. Details of backup is provided in the GHI Management Guide.

Most sites create a crontab entry to run a daily backup.

```
ghi_backup <filesys> <type> [-E <Fsets> [-U <user_args>]] | [-U] [<user_args>]
ghi_backup <filesys> <type> [-E -F <List of Fsets> [-U <user_args>]]
```

Where:

```
type  = one of the backup type:
image = image backup using Spectrum Scale SOBAR capability.
Fsets = filesets which need not be linked-in, either space-separated
List of Fsets or -F space-separated list of files containing list
of filesets user_args = args to be passed to policy.
```

Example command:

```
% ghi_backup firefly image
```

6.2 Restore

Refer to Backup and Recovery in the GHI Management Guide for more restore details.

Chapter 7

GHI Conversions

7.1 Conversion steps for 3.0.0 to 3.0.1

Note
The conversion process will use RPMs for GHI 3.0.1.

- Shutdown GHI.

```
% ghishutdown -g
```

- Verify all non-IOM GHI processes are not in the process list

```
% ps -ef | grep ghi | grep -v ghi_iom
```

Note
If any GHI processes remain in the list use, kill -9 <pid> to shut them down.

- Unmount the file system

```
% mmumount <file system> -a
```

- Remove old GHI RPMs

```
% rpm -qa | grep ghi - List installed RPMs
% rpm -e <ghi_rpms installed from above>
% rpm -qa | grep ghi - Verify no GHI RPMs are installed
```

- Install new GHI 3.0.1 RPMs

Note
RPMs deployed will vary. Contact GHI support rep before installing.

```
% cd <path to GHI 3.0.1 RPMs>
% rpm -ivh ghi-3.0.1.0-0.<arch>, ghi-lib-3.0.1.0-0.<arch>, ghi-ishtar-5.1.2.0-0.< ↩
   arch>
```

- Run ghiupdate

```
% ghiupdate -vT --<OS arch> <all nodes listed in ghilsnodes> - It is recommended ↩
   to use the -T option first.

% ghiupdate -v --<OS arch> <all nodes listed in ghilsnodes> - This will perform ↩
   actions.
```

- Convert the GHI garbage collection table on each file system. This step should be run for all file systems; the conversion of file system tables can be executed in parallel.
  - For each file system, modify /var/hpss/ghi/templates/ghimodifygc.ddl by replacing the template info (<GC_FILESYSTEM>) with your file system name throughout the file (e.g. if your file system name is "foo", the table name should be "GC_FOO").

```
% cat /var/hpss/ghi/templates/ghimodifygc.ddl

connect to hghi;
ALTER TABLE HPSS."GC_FOO_FS1"
 ADD COLUMN INODE
 BIGINT NOT NULL DEFAULT 0
 ADD COLUMN IGEN
 INTEGER NOT NULL DEFAULT 0;

reorg table HPSS."GC_CANAAN_FS1";
```

  - Run the modified ghimodifygc.ddl.

```
% db2 -svtf ghimodifygc.ddl
```

- Start new GHI

```
% ghistartup -g
```

- Verify that PM, CM, MD processes are running.

```
% ps -ef | grep ghi
```

- Mount GHI managed File Systems

```
% mmmount <file system> -a
```

- Verify SD, ED, and then IOMs are active

```
% tail /<file system>/scratch/mon/<mon_sd.out | mon_iom.out>
```

Note
The Pin Conversion process converts DMAPI PIN attributes into timestamp values for all file systems. If GPFS is restored using a backup prior to this conversion, the unconverted DMAPI PIN attributes will be restored as well, and it will be necessary to rerun these steps to convert the PIN attribute values again.

- Convert the DMAPI PIN attributes into timestamp values **for all file systems**
  - List all pinned files using the following policy:

```
# Template for listing old style pinned files
```

```
RULE EXTERNAL POOL 'hsm' EXEC '/opt/hpss/bin/ghi_migrate'

RULE 'Pinned Files' LIST 'files_pinned'
                    SHOW ('-s' FILE_SIZE)
                    WHERE XATTR('dmapi._GHI_PIN') LIKE 'TRUE%'
                    AND path_name NOT LIKE '%/%scratch%'
                    AND path_name NOT LIKE '%/%.snapshot%'

RULE 'Default' SET POOL 'system'
```

* Apply the above policy

```
% ghiapplypolicy <file system> -P <above template file> -I defer
```

– Generate a file list for all pinned files and convert them:

```
% cat <path/to/scratch/.ghi>/list.files_pinned | cut -d' ' -f7 > /tmp/ ↩
   pinned_files
% ghi_pin -f /tmp/pinned_files

# Policy template example

RULE EXTERNAL POOL 'hsm' EXEC '/opt/hpss/bin/ghi_migrate'
RULE 'UnPinned Files' LIST 'files_unpinned'
                    SHOW ('-s' FILE_SIZE)
                    WHERE XATTR('dmapi._GHI_PIN') LIKE 'FALSE%'
                    AND path_name NOT LIKE '%/%scratch%'
                    AND path_name NOT LIKE '%/%.snapshot%'
RULE 'Default' SET POOL 'system'
```

– Generate a file list for all unpinned files (files which were previously pinned, but were later unpinned using the ghi_pin tool) and convert them:

```
% cat <path/to/scratch/.ghi>/list.files_unpinned | cut -d' ' -f7 > /tmp/ ↩
   unpinned_files
% ghi_pin -u -f /tmp/unpinned_files
```

– Validate that all attributes have been migrated by listing the pinned files and comparing row counts - they should match. If not, list the pinned files again and rerun the ghi_pin tool against the resulting list.

```
% ghiapplypolicy -P /var/hpss/ghi/policy/pin_time_list.policy -I defer
% wc -l <path/to/scratch/.ghi>/list.files_pinned </path/to/scratch/.ghi>/list. ↩
   files_pinned_time
1000 <path/to/scratch/.ghi>/list.files_pinned
1000 <path/to/scratch/.ghi>/list.files_pinned_time
```

Note

It could take up to ten minutes for ED to connect to SD and fifteen minutes for the IOMs to connect to the ED. If after twenty minutes things are still not connecting, contact your GHI support rep.
