

AIX Version 7.2

Cluster management

IBM

AIX Version 7.2

Cluster management

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 21.

This edition applies to AIX Version 7.2 and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2015, 2017.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this document v

Highlighting	v
Case-sensitivity in AIX	v
ISO 9000.	v

Cluster management 1

Cluster Aware concepts	1
Cluster Aware AIX ports	2
Cluster repository	2
Cluster system architecture flow	3
Naming a cluster	3
Cluster communication	3
Deadman switch	4
Linked cluster	5
Asymmetric topology reconciliation.	5
Unicast communication.	6
Rolling upgrade and coexistence with prior AIX technology levels	6
IPv6 support	6
Configuring Cluster Aware	7
Setting up cluster SAN communication	7
Configuring cluster security	8

CAA licensing	9
Managing clusters with commands	9
Managing cluster events	10
Programming cluster sockets	11
Troubleshooting Cluster Aware	11
Troubleshooting with the snap command	11
Troubleshooting with node maintenance mode	12
Troubleshooting with component trace	12
Sample output for cluster commands.	13
clcmd date command sample output	13
lscluster -d command sample output	13
lscluster -i command sample output	14
lscluster -m command sample output.	15
lscluster -s command sample output	15
nodeState cluster event sample output	16
Code samples for cluster events	16
Cluster events using AHAFS sample code	16

Notices 21

Privacy policy considerations	23
Trademarks	23

About this document

The Cluster Aware function is part of the AIX operating system. Using Cluster Aware AIX[®], you can create a cluster of AIX nodes and build a highly available architectural solution for a data center.

Highlighting

The following highlighting conventions are used in this document:

Bold	Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
<i>Italics</i>	Identifies parameters whose actual names or values are to be supplied by the user.
Monospace	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

Case-sensitivity in AIX

Everything in the AIX operating system is case-sensitive, which means that it distinguishes between uppercase and lowercase letters. For example, you can use the **ls** command to list files. If you type **LS**, the system responds that the command is not found. Likewise, **FILEA**, **FiLea**, and **filea** are three distinct file names, even if they reside in the same directory. To avoid causing undesirable actions to be performed, always ensure that you use the correct case.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Cluster management

The Cluster Aware function is part of the AIX operating system. Using Cluster Aware AIX you can create a cluster of AIX nodes and build a highly available and an ideal architectural solution for a data center.

Cluster Aware concepts

When you create a cluster of a single node or multiple nodes, the interconnected set of nodes can leverage the Cluster Aware capabilities and services that are built into the AIX operating system.

Cluster Aware has the following capabilities:

- Clusterwide event management
 - Communication and storage events
 - Node UP and node DOWN
 - Network adapter UP and DOWN
 - Network address change
 - Point-of-contact UP and DOWN
 - Disk UP and DOWN
 - Predefined and user-defined events
- Clusterwide storage naming service
- Clusterwide command distribution
- Clusterwide communication making use of networking and storage communication

Applications can build on the tools and service capabilities that are provided when you create a cluster of nodes to make the application highly available and resilient.

Each node that is added to a cluster by using Cluster Aware must have common storage devices available, either through the storage area network (SAN) or through the serial-attached SCSI (SAS) subsystems. These storage devices are used for the cluster repository disk and for any clustered shared disks. The storage naming service does not provide a global device view.

A multicast address is used for cluster communications between the nodes in the cluster. Therefore, you need to review any network considerations before you create a cluster.

Each node must have at least one IP version 4 address configured on its network interface. The IP version 4 address is used as a basis for creating an IP version 4 multicast address, which the cluster communications uses for internal communications. You can configure IP version 6 addresses on any node or nodes in the cluster. These nodes support cluster monitoring of events and cluster configuration attributes.

Scalable reliable multicasting is implemented in the cluster with a special gossip protocol over the multicast address. The gossip protocol determines the node configuration and then transmits the gossip packets over all available networking and storage communication interfaces. If no storage communication interfaces are configured, only the traditional networking interfaces are used.

Using Cluster Aware you can monitor communications and network topology changes at various levels for all available services. With cluster monitoring, you can sense that a node is down, and a cluster can detect that a specific adapter is down or that a specific interface on an adapter is down.

A point-of-contact indicates that a node has actually received communication packets across this interface from another node. This communication process allows the application that is monitoring the health of a node to make discrete actions based on near real-time event notification. You can also monitor the storage devices to provide UP events and DOWN events for any recovery actions that are identified as necessary by the monitoring application.

Cluster Aware AIX ports

CAA uses certain ports for network communication. The ports must not be blocked on any nodes.

The CAA layer needs the following ports on all nodes for network communication:

- 4098 (for multicast)
- 6181
- 16191
- 42112

Cluster repository

The cluster repository disk is used as the central repository for the cluster configuration data.

The cluster repository disk must be accessible from all nodes in the cluster. The minimal size of the repository is largely dependent upon the cluster configuration. A minimal disk size of 10 GB is preferred. For VIOS, PowerHA pureScale cluster, see the respective release notes for the minimal size.

The cluster repository disk is backed up by a redundant and highly available storage configuration.

The cluster repository disk should be configured for RAID to accommodate the requirements of the data center.

The cluster repository disk is a special device for the cluster. The use of LVM commands are not supported when used on the cluster repository disk. The AIX LVM commands are single node administrative commands, and are not applicable in a clustered configuration.

Due to the special device characteristics required by the cluster repository disk, a raw section of the disk and a section of the disk that contains a special volume group and special logical volumes are used during cluster operations.

When CAA is configured with `repos_loss` mode set to `assert` and CAA loses access to the repository disk, the system automatically shuts down.

Reservation policy for repository disk

The following is an explanation of the reservation policy used in Cluster Aware.

All storage area network (SAN) provisioned disks must be zoned to all Fibre Channel adapters on the Virtual I/O Servers that will be members of the shared storage pool cluster.

The disks must have the reserve policy set to `no_reserve`. One disk with a minimum of 1 GB is used as the repository disk for the cluster.

Notes:

- Cluster Aware AIX (CAA) opens the repository disk, and CAA sets the ODM reserve attribute to `no_reserve` for all storage types.
- For nonrepository disks, use the `chdev` command to change the attribute to `no_reserve`.
- The cluster repository disk must be compliant with the 512 byte block size.

Related information:

chdev Command

Cluster system architecture flow

When you use Cluster Aware to create a cluster it is important that you understand the process of the clustering subsystem.

Note: Cluster Aware AIX (CAA) is not used as a stand-alone package. It is used with PowerHA[®] SystemMirror[®] or with Shared Storage Pool. PowerHA SystemMirror or Shared Storage Pool describes how to create a CAA cluster through its own commands. Refer to these products' respective documentation, including IBM[®] Redbooks[®] publications and release notes.

The following list describes the process of the clustering subsystem:

- The cluster is created by issuing the **mkcluster** command.
- The cluster configuration is written to the raw section of the cluster repository disk.
- Special volume groups and logical volumes are created on the cluster repository disk.
- Cluster file systems are created on the special volume group.
- Cluster services are made available to other functions in the operating system, such as Reliable Scalable Cluster Technology (RSCT) and PowerHA SystemMirror.
- Storage framework register lists are created on the cluster repository disk.
- A global device namespace is created and interaction with LVM starts for handling associated volume group events.
- A clusterwide multicast address is established.
- The node discovers all of the available communication interfaces.
- The cluster interface monitoring starts.
- The cluster interacts with Autonomic Health Advisory File System (AHAFS) for clusterwide event distribution.
- The cluster exports cluster messaging and cluster socket services to other functions in the operating system, such as Reliable Scalable Cluster Technology (RSCT) and PowerHA SystemMirror.

Related information:

PowerHA SystemMirror 7.1



PowerVM Virtualization Introduction and Configuration



Power HA Redbook



PowerVM Virtualization Managing and Monitoring

Naming a cluster

When you are naming a cluster you must follow specific guidelines.

The only acceptable ASCII characters you can use when naming a cluster are A - Z, a - z, 0 - 9, - (hyphen), . (period), and _ (underscore). The first character of the cluster name and domain name cannot be a hyphen. The maximum length of a cluster name is 63 characters.

Cluster communication

Cluster communication takes advantage of traditional networking interfaces, such as IP based network communications and storage interface communication through Fibre Channel and SAS adapters.

When you use both the IP-based network communications and the storage interface communications, all nodes in the cluster can always communicate with any other nodes in the cluster configuration. Having clusters in this configuration eliminates "split brain" incidents.

You must complete the Fibre Channel setup before the cluster can use the storage interfaces as an alternative communication path. The SAS adapter does not require special setup.

During Storage Area Network port configuration you must verify that your server interfaces are connected to the SAN fabric ports in the same zone.

Related concepts:

"Setting up cluster SAN communication" on page 7

You must complete the following setup before creating a cluster that uses storage communication interfaces.

Defining a virtual Ethernet adapter

Additional procedures for cluster communications.

During storage area network (SAN) port configuration you must verify that your server interfaces are connected to the SAN fabric ports in the same zone.

To configure the VLAN to establish SAN communication when the storage adapters are virtualized through VIOS, complete the following steps

1. Enable the target mode enabled (TME) attribute on VIOS Fibre Channel adapters as the padmin, by entering the following commands.

```
chdev -dev fcs0 -attr tme=yes -perm  
shutdown -restart
```
2. On the Hardware Management Console (HMC), add a virtual Ethernet adapter to the profile of each PowerHA SystemMirror virtual client node that has a VLAN ID of 3358.
3. Reactivate the partition by using the new profile. The new profile will boot, and then display a new *entX*. To display the interface status, enter the command `lscluster -i`

Notes:

1. VLAN 3358 must be created on the virtual client LPARs and VIOS servers.
2. VLAN 3358 is the only value that CAA uses. The VLAN tag of *sfw0* must not be changed.
3. The *entX* adapter that is associated with VLAN 3358 does not require an *enX* interface or an IP address.
4. VLAN 3358 must not be bridged to the Shared Ethernet Adapter (SEA).
5. When SAN communication is configured properly, the `lscluster -m` command shows the status of the *sfwcom* (storage framework communication) interface as up.
6. The VIOS fcs adapter that serves the repository disk through N_Port ID Virtualization (NPIV) can also be used for SAN communication. However, this configuration represents a single point of failure and therefore, different VIOS fcs adapters must be used for the repository and SAN communication.

Related information:

Creating a virtual Ethernet adapter using HMC

Deadman switch

A deadman switch is an action that occurs when Cluster Aware AIX (CAA) detects that a node has become isolated in a multinode environment. This setting occurs when nodes are not communicating with each other via the network and the repository disk.

The AIX operating system can react differently depending on the deadman switch setting or the `deadman_mode` which is tunable. The deadman switch mode can be set to either force a system shut down or generate an Autonomic Health Advisor File System (AHAFS) event.

Related information:

`clctrl` Command

Linked cluster

IBM AIX 7.1 with Technology Level 2 Cluster Aware AIX (CAA) introduces the concept of linked cluster.

Linked cluster provides the reliable exchange of data and control messages between two or more nodes that are part of the same cluster but that are separated by geographical boundaries. Each location is called a site. The AIX 7 with 7100-02 CAA supports up to two sites.

The only mode of communication between nodes that are in two sites is through TCP/IP. There is no Storage Area Network (SAN) or disk communication.

The nodes within a site share a common repository. The repositories between sites are synchronized by CAA. When sites are divided or merge, CAA provides a mechanism to reconcile the two repositories. The reconciliation can be done either through a reboot (of all the nodes on site whose repository needs to be updated) or through an application programming interface (API) implemented exclusively for Reliable Scalable Cluster Technology (RSCT).

Disks are not shared across sites. Therefore, the addition or deletion of disks is limited to a single site.

Autonomic Health Advisor File System (AhaFS) events are propagated across the entire linked cluster to provide a consistent view across the sites and links. Similarly, the `lscluster` command displays the clusterwide information, that is, the command includes information from both the sites.

The suggested way of creating a two-site cluster is to first create a single-site cluster by issuing the `mkcluster` command and then to add the remote site and node by issuing the `chcluster` command. The nodes and the site can be removed by issuing the `rmcluster` command.

You can upgrade an existing AIX 7 with 7100-01 or AIX 7 with 7100-01 SP4 of CAA, which does not support clusters with sites to a cluster with site support associated through the process of rolling upgrade.

Several tunable parameters are provided to tune the rate of exchange of heartbeat messages between nodes at different sites.

Related concepts:

“IPv6 support” on page 6

IBM AIX 7.1 with Technology Level 2 Cluster Aware AIX (CAA) introduces support for Internet Protocol version 6 (IPv6) for network-based communications.

Asymmetric topology reconciliation

Partial loss of connectivity of a node or nodes in one site to a node or nodes in another site or within the same site can lead to an asymmetric view of the topology among the nodes. When a partial loss of connectivity happens, nodes within a cluster do not have a consistent view of the cluster. Such loss of a symmetric view can create cluster operability problems. Clusterwide locks are potentially erroneously granted. This action also creates confusion among the other users of CAA such as RSCT, VIOS, and PowerHA SystemMirror.

A CAA algorithm safeguards against this condition. This action begins when all the nodes in a cluster are at the AIX 7 with 7100-02 of CAA.

Nodes keep exchanging their views of the cluster until a node or nodes recognize a partial view of the cluster. Nodes are then rebooted selectively until a consistent view of the cluster is reached.

Unicast communication

Cluster Aware AIX (CAA) uses multicast communications for heartbeat and other protocol messages, which might require an additional network setup at customer site. The unicast cluster provides a new capability to CAA to support clustering with simultaneous unicasting of CAA protocol messages, instead of multicasting. It is applied to all sites within the CAA cluster.

The communication mode of the cluster can be toggled at run time by using the `clctrl -tune` command and changing the value of the `communication_mode` tunable parameter, between `u` (for unicast) and `m` (for multicast). The CAA default value is `m` but it can vary depending on product. For example, VIOS SSP defaults to the unicast mode.

Rolling upgrade and coexistence with prior AIX technology levels

With IBM AIX 7.1 with Technology Level 2 Cluster Aware AIX (CAA) you can upgrade without a total cluster outage.

AIX 7 with 7100-02 (CAA) no longer require a total cluster outage to upgrade the cluster nodes to AIX 7 with 7100-02.

A *rolling upgrade* of a cluster is done by taking a node offline and upgrading it to a new AIX technology level, while the other nodes remain active. After a node is upgraded, the node is rebooted and brought online by issuing the `clctrl` command. This process is repeated until all the nodes are upgraded.

In a mixed cluster environment, nodes running AIX 7 with 7100-02 (CAA) maintain compatibility with nodes that are still running prior AIX technology levels by running at the lowest effective level. New features are not enabled until all the cluster nodes are upgraded to the new technology level.

For example, AIX 7 with 7100-02(CAA) introduces support for IPv6 networks and multiple sites. This support is not available until the entire cluster is upgraded to AIX 7 with 7100-02 (CAA).

Rolling upgrade and coexistence support are not provided for nodes running AIX 7.1 or AIX 7.1 SP5 (CAA) unless the mandatory APARs are installed. Nodes that have AIX 7.1 must have APAR IV16481. If your nodes do not have the required APARs, a total cluster outage is still required. In that situation, you must remove your cluster, install AIX 7 with 7100-02 (CAA) on all of your nodes, and then re-create your cluster.

Note: Applying the mandatory APARs also requires a total cluster outage, so it is worthwhile to install the mandatory APARs, if you immediately plan to install AIX 7 with 7100-02 (CAA).

If you are running other clustering software, such as PowerHA SystemMirror, on top of your CAA cluster, see the documentation for that software for additional information and instructions for upgrading your cluster.

IPv6 support

IBM AIX 7.1 with Technology Level 2 Cluster Aware AIX (CAA) introduces support for Internet Protocol version 6 (IPv6) for network-based communications.

With this support, nodes are now able to participate in homogeneous IPv6 and heterogeneous IPv4 and IPv6 network environments.

Network interfaces configured with IPv6 are automatically detected and used by the CAA kernel communications services. Network interfaces configured with both IPv4 and IPv6 maintain heartbeat and communicate over both versions of IP.

The **lscluster** command has been updated to support IPv6:

- IPv6 addresses configured over monitored network interfaces will be displayed.
- The IP protocol for each network-based point-of-contact will be displayed.

The IPv6 multicast group is of site-local scope and is generated by using the IPv4 multicast group that was either manually specified or automatically generated. Specifically, the IPv4 multicast group occupies the bottom 32-bit word of a standard IPv6 site-local multicast address. The AIX 7 with 7100-02 CAA does not allow you to specify or change the IPv6 multicast group used for the cluster. The multiple-site feature introduced in AIX 7 with 7100-02 CAA requires that each site have its own unique multicast group. The site multicast group is either specified or automatically generated when the site is created. The ability to directly define a site's IPv6 multicast group is not supported.

You can upgrade an existing AIX 7 with 7100-01 or AIX 7 with 7100-01 SP4 release of a CAA cluster that does not have support for IPv6 to an AIX 7 with 7100-02 release of a CAA cluster that does have support for IPv6 through the process of a rolling upgrade. Additionally, for clusters that you plan to run IPv6 exclusively over their network topology, you need to specify the IPv6 capabilities flag during cluster creation to indicate that IPv6 support is required on all nodes to create the cluster.

VLAN pseudoadapter support

IBM AIX 7 with 7100-02 release of a Cluster Aware AIX (CAA) supports VLAN pseudoadapters for participation in VLAN networks. Network interfaces configured over VLAN pseudoadapters are automatically detected and used for CAA kernel communications services.

Related concepts:

“Linked cluster” on page 5

IBM AIX 7.1 with Technology Level 2 Cluster Aware AIX (CAA) introduces the concept of linked cluster.

Configuring Cluster Aware

The following information deals with the configuring of the cluster

Setting up cluster SAN communication

You must complete the following setup before creating a cluster that uses storage communication interfaces.

The following information applies only to Fibre Channel adapters. You do not need to set up the serial-attached SCSI (SAS) adapters or configure storage area network (SAN) communication to deploy and to manage Cluster Aware AIX (CAA) clusters or PowerHA SystemMirror clusters.

SAN communication is supported on Fibre Channel adapters that support the target mode enabled (TME) attribute. The following adapters support SAN communication:

- 4 GB Single-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 1905; CCIN 1910)
- 4 GB Single-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 5758; CCIN 280D)
- 4 GB Single-Port Fibre Channel PCI-X Adapter (FC 5773; CCIN 5773)
- 4 GB Dual-Port Fibre Channel PCI-X Adapter (FC 5774; CCIN 5774)
- 4 Gb Dual-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 1910; CCIN 1910)
- 4 Gb Dual-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 5759; CCIN 5759)
- 4-Port 8 Gb PCIe2 FH Fibre Channel Adapter (FC 5729)

- 8 Gb PCI Express Dual Port Fibre Channel Adapter (FC 5735; CCIN 577D)
- 8 Gb PCI Express Dual Port Fibre Channel Adapter 1Xe Blade (FC 2B3A; CCIN 2607)
- 3 Gb Dual-Port SAS Adapter PCI-X DDR External (FC 5900 and 5912; CCIN 572A)

Note: For the most current list of supported Fibre Channel adapters, contact your IBM representative.

For the adapter to be supported, it must have target mode support.

The target mode enabled (TME) attribute for a supported adapter is only present when the minimum AIX level for CAA is installed.

To configure the Fibre Channel adapters that will be used for cluster storage communications, complete the following steps:

Note: In the following steps the *X* in *fcsX* represents the number of your Fibre Channel adapters, for example, *fcs1*, *fcs2*, or *fcs3*.

1. Run the following command:

```
rmdev -Rl fcsX
```

Note: If you booted from the Fibre Channel adapter, you do not need to complete this step.

2. Run the following command:

```
chdev -l fcsX -a tme=yes
```

Note: If you booted from the Fibre Channel adapter, add the *-P* flag.

3. Run the following command:

```
chdev -l fcsXiX -a dyntrk=yes -a fc_err_recov=fast_fail
```

4. Run the **cfgmgr** command.

Note: If you booted from the Fibre Channel adapter and used the *-P* flag, you must reboot.

5. Verify the configuration changes by running the following command:

```
lsdev -C | grep sfwcom
```

The following is an example of the output displayed from the **lsdev -C | grep sfwcom** command:

```
lsdev -C | grep sfwcom
sfwcomm0      Available 01-00-02-FF Fiber Channel Storage Framework Comm
sfwcomm1      Available 01-01-02-FF Fiber Channel Storage Framework Comm
```

After you create the cluster, you can list the cluster interfaces and view the storage interfaces by running the following command:

```
lscluster -i
```

Related concepts:

“Cluster communication” on page 3

Cluster communication takes advantage of traditional networking interfaces, such as IP based network communications and storage interface communication through Fibre Channel and SAS adapters.

Configuring cluster security

Cluster security secures the core communication between nodes of the cluster. Message security is achieved by encryption mechanism.

Cluster Security supports the following types of encryption keys for message encryption:

- Message Digest 5 (MD5) with Data Encryption Standard (DES)
- MD5 with Triple DES

- MD5 with Advanced Encryption Standard (AES).

Select an encryption algorithm that is compatible with the security methodology used by your organization. You can configure the security options and options for distributing encryption keys using the SMIT interface or the **clctrl** command.

The smitty fast path for the cluster security is:

```
smitty clustsec
```

Related information:

clctrl Command

CAA licensing

A list of product versions for which CAA is licensed.

The following table lists the product versions for which CAA is licensed:

CAA licensed	AIX 6.1			AIX 7.1		
	Express	Standard	Enterprise	Express	Standard	Enterprise
PowerHA	Yes	Yes	Yes	Yes	Yes	Yes
VIOS SSP (shared storage pools)	Yes	Yes	Yes	Yes	Yes	Yes
External consumer	No	No	No	No	Yes	Yes

Managing clusters with commands

You can use commands to manage a set of cluster nodes.

Use the following commands to manage clusters:

mkcluster

Use this command to create a cluster. The following example creates a multinode cluster:

```
mkcluster -n mycluster -m nodeA,nodeB,nodeC -r hdisk7 -d hdisk20,hdisk21,hdisk22
```

chcluster

Use this command to change the cluster configuration. The following example adds a node to the cluster configuration:

```
chcluster -n mycluster -m +nodeD
```

rmcluster

Use this command to remove the cluster configuration. The following example removes the cluster configuration:

```
rmcluster -n mycluster
```

lscluster

Use this command to list cluster configuration information. The following example lists the cluster configuration for all nodes:

```
lscluster -m
```

clcmd Use this command to distribute a command to a set of nodes that are members of a cluster. The following example lists the date for all the nodes in the cluster:

```
clcmd date
```

Related concepts:

“Sample output for cluster commands” on page 13

You can view sample output for the **lscluster -d** command, the **lscluster -i** command, the **lscluster -m** command, and the **lscluster -s** command.

Related information:

- chcluster command
- clcmd command
- lscluster command
- mkcluster command
- rmcluster command

Managing cluster events

AIX event management is implemented using a pseudofile system architecture. The use of the pseudofile system allows you to use existing application programming interfaces (APIs) to program the monitoring of events, such as a select () call or a blocking read () call.

The Autonomic Health Advisory File System (AHAFS) is an in-memory file system that is used to store the necessary objects to manage the configuration and use of the file monitoring facilities.

When you are monitoring for events in a cluster configuration, you must specify the CLUSTER=YES attribute to write to the monitor file. The cluster information for node number, node ID, and cluster ID is available in the results from a cluster event.

The AHAFS file system is automatically mounted when you create the cluster. If the AHAFS file system is already mounted by another application before the cluster is created, the original mount point is used by the cluster configuration.

Table 1. Cluster events

Cluster events	Description
nodeList	Monitors changes in cluster membership
clDiskList	Monitors changes in cluster disk membership
nodeContact	Monitors the last contact status of the node in a cluster
nodeState	Monitors the state of the node in the cluster
nodeAddress	Alias is added or removed from a network interface
networkAdapterState	Monitors the network interface of a node in the cluster
clDiskState	Monitors clustered disks
repDiskState	Monitors the repository disk
diskState	Monitors the local disk changes
vgState	Verifies the status of the volume group on a disk

The following steps display the process for event handling:

1. Create a monitor file based on the /aha directory.
2. Write the required information to the monitor file to represent the wait type, either a select call or blocking read call, and when the event should be triggered. For example, a state change of node down.
3. Wait in a select () call or a blocking read () call.
4. Read from the monitor file to obtain the event data.

Related concepts:

“nodeState cluster event sample output” on page 16

Related information:

Programming cluster sockets

Cluster communications can operate over the traditional networking interfaces (IP-based) or using the storage interfaces (Fibre Channel or SAS).

When cluster communications is configured over both transports the redundancy and high availability of the underlying cluster node software and hardware configuration can be maximized by using all the paths for communications. In case of network interface failures, you can use the storage framework (Fibre Channel or SAS) to maintain communication between the cluster nodes. Cluster communications is achieved by exploiting the multicast capabilities of the networking and storage subsystems.

Example: Using a socksimple program

The following cluster socket program example uses a pinglike interface to send and receive packets over the cluster communications. The example program uses the local cluster as the scope of nodes that can send or receive information.

The example environment has a three-node cluster of nodeA, nodeB, and nodeC.

To start the socksimple program as the receiver on node 1 (nodeA), run the following command:

```
./socksimple -r -a 1
```

Note: To find the node number, view the output from the **lscluster -m** command. For the cluster shorthand ID, you can also use the **get_clusterid** function.

To start the socksimple program as the sender on node 3 (nodeC), run the following command:

```
./socksimple -s -a 1
```

Note: The **-a (address)** option sends the packets to node 1 in this local cluster.

The following code is output from running the `socksimple -s -a 1` command:

```
./socksimple -s -a 1
socksimple version 1.2
socksimple 1/12 with ttl=1:

1275 bytes from cluster host id = 1: seqno=1275 ttl=1 time=0.411 ms
1276 bytes from cluster host id = 1: seqno=1276 ttl=1 time=0.275 ms
1277 bytes from cluster host id = 1: seqno=1277 ttl=1 time=0.287 ms
1278 bytes from cluster host id = 1: seqno=1278 ttl=1 time=0.284 ms
--- socksimple statistics ---
4 packets transmitted, 4 packets received
round-trip min/avg/max = 0.267/0.291/0.411 ms
```

Troubleshooting Cluster Aware

You can review troubleshooting tips for using the **snap** command, and the cluster maintenance mode.

Troubleshooting with the snap command

The clustering subsystem provides a **snap** script that you can use to help you collect logs and data configurations that you can use to help troubleshoot problems.

Run the following command to execute the **snap** script:

```
snap caa
```

The following structure is an example of the data files collected during the snap script execution for Cluster Aware AIX:

```
/tmp/ibmsupt
|-- caa
    |-- Data
        |-- 20100817215934 (For example, a timestamp at which "snap caa" was run)
            |-- nodeA.austin.ibm.com.tar.gz
            |-- ...
            |-- nodeB.austin.ibm.com.tar.gz
            |-- ...
            |-- nodeC.austin.ibm.com.tar.gz
        |-- ... (For example, more timestamp directories to distinguish separate "snap caa" invocations)
```

Related information:

snap command

Troubleshooting with node maintenance mode

Maintenance of the cluster, nodes, and disks are not needed under normal operation. If maintenance is necessary, you can use the **clctrl -stop** command to place a node or set of nodes in maintenance mode.

The **clctrl -stop** command quiesces cluster services on one or more nodes. You may make cluster configuration changes as long as one node in the cluster is in normal operation. If all nodes in the cluster are stopped, you cannot make cluster configuration changes.

Nodes that have been stopped do not participate in cluster configuration or communications and are seen by the other nodes as down. The stopped state is persistent. Nodes that have been stopped must be explicitly started via the **clctrl -start** command before they can resume cluster participation.

To set a node in maintenance mode, run the following command:

```
clctrl -stop -n mycluster -m nodeA
```

To set all nodes in maintenance mode, run the following command:

```
clctrl -stop -n mycluster -a
```

To set a node to normal operation, run the following command:

```
clctrl -start -n mycluster -m nodeA
```

To set all nodes to normal operation, run the following command:

```
clctrl -start -n mycluster -a
```

Troubleshooting with component trace

The cluster subsystem uses component trace, which is controlled by the **ctctrl** command.

The hierarchy is as follows:

```
cluster      : Base parent component for CAA
.config      : Component for configuration
.lock        : Component for locking
  .ahafs     : Component for AHAFS
  .comm      : Parent component for communication
  .disk      : Subcomponent for disk communication
  .net       : Subcomponent for network communication
  .san       : Subcomponent for SAN communication
```

AHAFS – Autonomic Health Advisor File System

Related information:

clctrl Command

Sample output for cluster commands

You can view sample output for the **lscluster -d** command, the **lscluster -i** command, the **lscluster -m** command, and the **lscluster -s** command.

Related concepts:

“Managing clusters with commands” on page 9

You can use commands to manage a set of cluster nodes.

clcmd date command sample output

```
-----  
NODE nodeA.austin.ibm.com  
-----
```

```
Fri Jul 30 08:00:00 CDT 2010  
  
-----
```

```
NODE nodeB.austin.ibm.com  
-----
```

```
Fri Jul 30 08:00:00 CDT 2010  
  
-----
```

```
NODE nodeC.austin.ibm.com  
-----
```

```
Fri Jul 30 08:00:00 CDT 2010
```

lscluster -d command sample output

Storage Interface Query

```
Cluster Name: mycluster  
Cluster uuid: 15f90c7e-e651-11e1-84be-00145e76c700  
Number of nodes reporting = 2  
Number of nodes expected = 2
```

```
Node nodeA.austin.ibm.com  
Node uuid = 1602a950-e651-11e1-84be-00145e76c700  
Number of disk discovered = 2  
  disk6  
    State : UP  
    uDid : 200B75DC891480507210790003IBMfcp  
    uUId : 447dac46-c779-c5ff-ca46-7f885ec6f742  
    Site uUId : 51735173-5173-5173-5173-517351735173  
    Type : CLUSDISK  
  disk7:  
    State : UP  
    uDid : 200B75DC891480607210790003IBMfcp  
    uUId : 3e77c6b6-5624-d27a-01d9-9b291c5e8437  
    Site uUId : 51735173-5173-5173-5173-517351735173  
    Type : REPDISK
```

```
Node nodeB.austin.ibm.com  
Node UUID = ebc9b154-e70b-11e1-a379-00145e76c700  
Number of disks discovered = 2  
  disk6:  
    State : UP  
    uDid : 200B75DC891480507210790003IBMfcp  
    uUId : 447dac46-c779-c5ff-ca46-7f885ec6f742  
    Site uUId : 51735173-5173-5173-5173-517351735173  
    Type : CLUSDISK  
  disk7:
```

State : UP
uDid : 200B75DC891480607210790003IBMfcp
uUid : 3e77c6b6-5624-d27a-01d9-9b291c5e8437
Site uUid : 51735173-5173-5173-5173-517351735173
Type : REPDISK

lscluster -i command sample output

```
# lscluster -i  
Network/Storage Interface Query:
```

```
Cluster Name: mycluster  
Cluster uuid: 15f90c7e-e651-11e1-84be-00145e76c700  
Number of nodes reporting = 2  
Number of nodes stale = 0  
Number of nodes expected = 2
```

```
Node nodeA.austin.ibm.com  
Node uuid = 1602a950-e651-11e1-84be-00145e76c700  
Number of interfaces discovered = 2  
  Interface number 1 en0  
    NDD type = 7 (NDD_ISO88023)  
    MAC address length = 6  
    MAC address = 00:14:5E:E7:01:F1  
    Smoothed RTT across interface = 8  
    Mean deviation in network RTT across interface = 3  
    Probe interval for interface = 110 ms  
    IFNET flags for interface = 0x1E080863  
    NDD flags for interface = 0x0061081B  
    Interface state = UP  
    Number of regular addresses configured on interface = 1  
    IPv4 ADDRESS: 10.3.207.183 broadcast 10.3.207.255 netmask 255.255.255.0  
    Number of cluster multicast addresses configured on interface = 1  
    IPv4 MULTICAST ADDRESS: 228.3.207.179  
  Interface number 2, dpcom  
    IFNET type = 0 (none)  
    NDD type = 305 (NDD_PINGCOMM)  
    Smoothed RTT across interface = 330  
    Mean deviation in network RTT across interface = 214  
    Probe interval for interface = 5440 ms  
    IFNET flags for interface = 0x00000000  
    NDD flags for interface = 0x00000009  
    Interface state = UP RESTRICTED AIX_CONTROLLED
```

```
Node nodeB.austin.ibm.com  
Node UUID = 6bdfd974-e651-11e1-a546-00145e76c700  
Number of interfaces discovered = 2  
  Interface number 1, en0  
    IFNET type = 6 (IFT_ETHER)  
    NDD type = 7 (NDD_ISO88023)  
    MAC address length = 6  
    MAC address = 00:14:5E:E7:2C:B1  
    Smoothed RTT across interface = 7  
    Mean deviation in network RTT across interface = 3  
    Probe interval for interface = 100 ms  
    IFNET flags for interface = 0x1E080863  
    NDD flags for interface = 0x0061081B  
    Interface state = UP  
    Number of regular addresses configured on interface = 1  
    IPv4 ADDRESS: 10.3.207.197 broadcast 10.3.207.255 netmask 255.255.255.0  
    Number of cluster multicast addresses configured on interface = 1  
    IPv4 MULTICAST ADDRESS: 228.3.207.179
```

```

Interface number 2, dpcom
  IFNET type = 0 (none)
  NDD type = 305 (NDD_PINGCOMM)
  Smoothed RTT across interface = 701
  Mean deviation in network RTT across interface = 413
  Probe interval for interface = 11140 ms
  IFNET flags for interface = 0x00000000
  NDD flags for interface = 0x00000009
  Interface state = UP RESTRICTED AIX_CONTROLLED

```

Iscluster -m command sample output

Calling node query for all nodes
Node query number of nodes examined: 2

```

Node name: nodeA.austin.ibm.com
Cluster shorthand id for node: 1
UUID for node: 1602a950-e651-11e1-84be-00145e76c700
State of node: UP
Smoothed rtt to node: 7
Mean Deviation in network rtt to node: 3
Number of clusters node is a member in: 1
CLUSTER NAME      SHID      UUID
cluster_test      0         15f90c7e-e651-11e1-84be-00145e76c700
SITE NAME         SHID      UUID
LOCAL             1         51735173-5173-5173-5173-517351735173

```

Points of contact for node: 2

Interface	State	Protocol	Status
dpcom	DOWN	none	RESTRICTED
en0	UP	IPv4	none

```

Node name: nodeB.austin.ibm.com
Cluster shorthand id for node: 2
UUID for node: 468fdcfa-e651-11e1-98bb-00145e76c700
State of node: UP NODE_LOCAL
Smoothed rtt to node: 0
Mean Deviation in network rtt to node: 0
Number of clusters node is a member in: 1
CLUSTER NAME      SHID      UUID
cluster_test      0         15f90c7e-e651-11e1-84be-00145e76c700
SITE NAME         SHID      UUID
LOCAL             1         51735173-5173-5173-5173-517351735173

```

Points of contact for node: 0

Iscluster -s command sample output

Cluster Network Statistics:

```

pkts seen: 7720796           passed: 2413542
IP pkts: 6637134            UDP pkts: 5322951
gossip pkts sent: 568435    gossip pkts recv: 1683869
cluster address pkts: 0      CP pkts: 5307254
bad transmits: 1            bad posts: 3
short pkts: 0               multicast pkts: 4686518
cluster wide errors: 1      bad pkts: 0

```

dup pkts: 2418	pkt fragments: 479
fragments queued: 0	fragments freed: 0
pkts pulled: 0	no memory: 0
rxmit requests rcv: 95	requests found: 79
requests missed: 30	ooo pkts: 301
requests reset sent: 30	reset rcv: 82
remote tcpsock send: 0	tcpsock rcv: 0
rxmit requests sent: 151	
alive pkts sent: 0	alive pkts rcv: 0
ahafs pkts sent: 8	ahafs pkts rcv: 26
nodedown pkts sent: 0	nodedown pkts rcv: 5
socket pkts sent: 1944	socket pkts rcv: 1975
cwide pkts sent: 819965	cwide pkts rcv: 1231139
socket pkts no space: 0	pkts rcv notforhere: 338933
Pseudo socket pkts sent: 0	Pseudo socket pkts rcv: 0
Pseudo socket pkts dropped: 0	
arp pkts sent: 11	arp pkts rcv: 10
stale pkts rcv: 0	other cluster pkts: 2
storage pkts sent: 1	storage pkts rcv: 1
disk pkts sent: 2919	disk pkts rcv: 9150
unicast pkts sent: 617527	unicast pkts rcv: 636433
out-of-range pkts rcv: 0	
IPv6 pkts sent: 0	IPv6 pkts rcv: 2443
IPv6 frags sent: 0	IPv6 frags rcv: 0
Unhandled large pkts: 0	

nodeState cluster event sample output

aha/cluster/nodeState.monFactory/nodeStateEvent.mon

```
BEGIN_EVENT_INFO
TIME_tvsec=1280597380
TIME_tvnsec=591097152
SEQUENCE_NUM=4
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=NODE_DOWN
NODE_NUMBER=1
NODE_ID=0xDCE3A808999111DFAA800245C0004002
CLUSTER_ID=0x22A3BFAE9CC611DFA9B80245C0002004
END_EVPROD_INFO
END_EVENT_INFO
```

Related concepts:

“Managing cluster events” on page 10

AIX event management is implemented using a pseudofile system architecture. The use of the pseudofile system allows you to use existing application programming interfaces (APIs) to program the monitoring of events, such as a select () call or a blocking read () call.

Code samples for cluster events

You can view code samples for cluster events by using AHAFS and cluster socket programming.

Cluster events using AHAFS sample code

The sample program code, test_prog, is executed by using the following arguments:

```
./test_prog /aha/cluster/nodeState.monFactory/nodeStateEvent.mon "CHANGED=YES;CLUSTER=YES" 10 /tmp/nodestateevent
```

The following is the code for test_prog:

```
#include <stdio.h>
#include <string.h> /* for strcmp() */
#include <fcntl.h>
#include <errno.h>
#include <sys/time.h>
```



```

#include <sys/select.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <libgen.h>
#include <usersec.h>

#define MAX_WRITE_STR_LEN    255

void    syntax(char *prog);
int     ahaMonFile(char *str);
static int mk_parent_dirs (char *path);
void    read_data (int fd,int outfd);

char    *monFile;

test_prog :: main

int main (int argc, char *argv[])
{
    int    fd,outfd, rc,i=0,cnt=0;
    fd_set readfds;
    char *outputFile;
    char   wrStr[MAX_WRITE_STR_LEN+1];
    char   waitInRead[] = "WAIT_TYPE=WAIT_IN_READ";
    if (argc < 5)
        syntax( argv[0]);
    monFile = argv[1];
    if ( ! ahaMonFile(monFile) ) /* Not a .mon file under /aha */
        syntax( argv[0]);
    /* Create intermediate directories of the .mon file */
    rc = mk_parent_dirs(monFile);
    if (rc)
    {
        fprintf (stderr,
            "Could not create intermediate directories of the file %s !\n", monFile);
        return(-1);
    }
    printf("Monitor file name: %s\n", monFile);
    sprintf (wrStr, "%s", argv[2]);
    cnt = atoi(argv[3]);
    printf("Write String : %s\n", wrStr);
    outputFile = argv[4];
    fd = open (monFile, O_CREAT|O_RDWR);
    if (fd < 0)
    {
        fprintf (stderr,"Could not open the file %s; errno = %d\n", monFile,errno);
        exit (1);
    }
    outfd = open (outputFile, O_CREAT|O_RDWR);
    if (outfd < 0)
    {
        fprintf (stderr, "Could not open the file %s; errno = %d !\n", monFile, errno);
        return(-1);
    }
    write(fd, wrStr, strlen(wrStr));

    for(i = 0; i < cnt; i++)
    {
        if (strstr(wrStr, waitInRead) == NULL)
        {
            FD_ZERO(&readfds);
            FD_SET(fd, &readfds);
            printf(

```

```

        "Entering select() to wait till the event corresponding to the AHA node %s occurs.\n",
        monFile);
printf("Please issue a command from another window to trigger this event.\n");
rc = select (fd+1, &readfds, NULL, NULL, NULL);
printf("\nThe select() completed. \n");
if (rc <= 0) /* No event occurred or an error was found. */
{
    fprintf (stderr, "The select() returned %d.\n", rc);
    perror ("select: ");
    return (-1);
}
if(! FD_ISSET(fd, &readfds))
    goto end;
printf("The event corresponding to the AHA node %s has occurred.\n", monFile);
}
else
{
    printf(
        "Entering read() to wait till the event corresponding to the AHA node %s occurs.\n",
        monFile);
    printf("Please issue a command from another window to trigger this event.\n");
}
read_data(fd,outfd);
}
end:
close(fd);
close(outfd);
}

```

test_prog :: syntax

```

/* ----- */
void syntax(char *prog)
{
    printf("\nSYNTAX: %s <aha-monitor-file> [<key1>=<value1>[;<key2>=<value2>;...]] <count> <outfile> \n",prog);
    exit (1);
}

```

test_prog :: ahaMonFile

```

/* -----
 * PURPOSE: To check whether the file provided is an AHA monitor file.
 */
int ahaMonFile(char *str)
{
    char cwd[PATH_MAX];
    int len1=strlen(str), len2=strlen(".mon");
    int rc = 0;
    struct stat    sbuf;

    /* Make sure /aha is mounted. */
    if ((stat("/aha", &sbuf) < 0) ||
        (sbuf.st_flag != FS_MOUNT))
    {
        printf("ERROR: The filesystem /aha is not mounted!\n");
        return (rc);
    }

    /* Make sure the path has .mon as a suffix. */
    if ((len1 <= len2) ||
        (strcmp ( (str + len1 - len2), ".mon")))
        )
        goto end;

    if (! strncmp (str, "/aha",4)) /* The given path starts with /aha */
        rc = 1;
}

```

```

else /* It could be a relative path */
{
    getcwd (cwd, PATH_MAX);
    if ((str[0] != '/') && /* Relative path and */
        (! strncmp (cwd, "/aha",4)) /* cwd starts with /aha . */
        )
        rc = 1;
}
end:
if (!rc)
    printf("ERROR: %s is not an AHA monitor file !\n", str);
return (rc);
}

```

test_prog :: mk_parent_dirs

```

/*-----
* NAME: mk_parent_dirs()
* PURPOSE: To create intermediate directories of a .mon file if
*          they are not created.
*/
static int
mk_parent_dirs (char *path)
{
    char s[PATH_MAX];
    char *dirp;
    struct stat buf;
    int rc=0;

    dirp = dirname(path);
    if (stat(dirp, &buf) != 0)
    {
        sprintf(s, "/usr/bin/mkdir -p %s", dirp);
        rc = system(s);
    }
    return (rc);
}

```

test_prog :: read_data

```

/*-----
* PURPOSE: To parse and print the data received at the occurrence
*          of the event.
*/
void
read_data (int fd,int outfd)
{
#define READ_BUF_SIZE 3072
    char data[READ_BUF_SIZE];
    char *p, *line;
    char cmd[64];
    time_t sec, nsec;
    pid_t pid;
    uid_t uid, luid;
    gid_t gid;
    char curTm[64];
    int n;
    int stackInfo = 0;
    char uname[64], lname[64], gname[64];

    bzero((char *)data, READ_BUF_SIZE);
    /* Read the info from the beginning of the file. */
    n=pread(fd, data,READ_BUF_SIZE, 0);
    p = data;
    printf("%s\n",p);
    write(outfd, data, n);
}

```

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as the customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at <http://www.ibm.com/privacy> and IBM’s Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at www.ibm.com/legal/copytrade.shtml.



Printed in USA