



# Installation Guide: OpenSAF

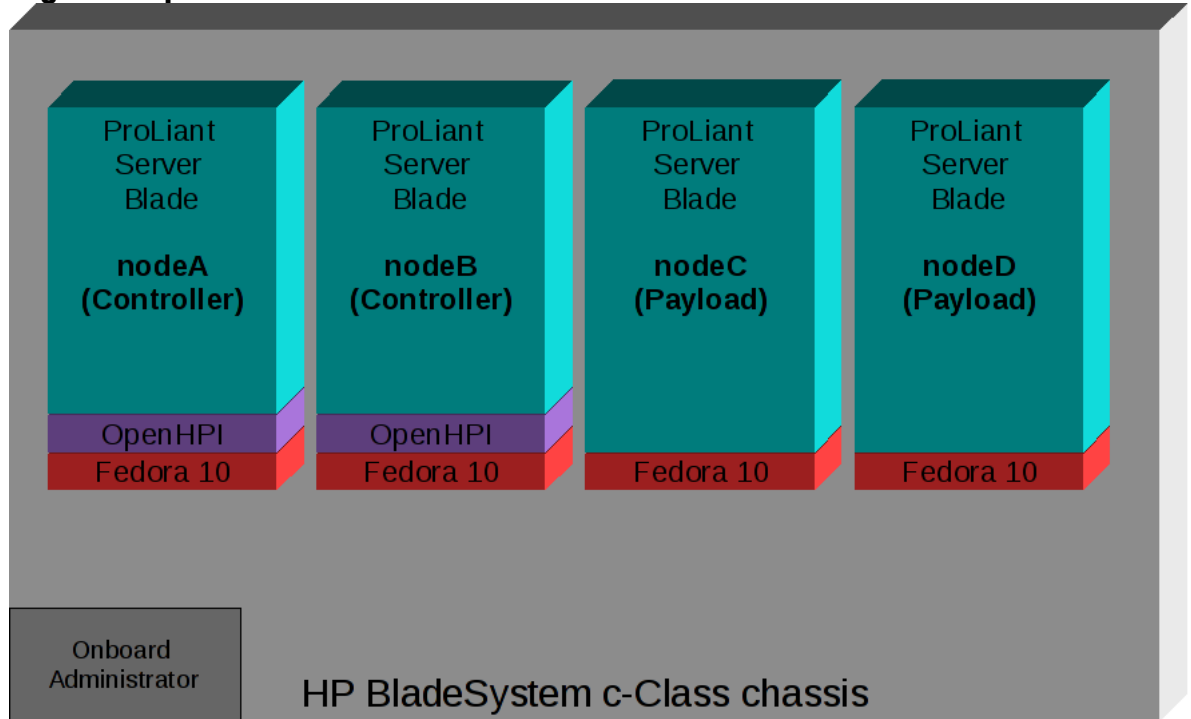
Abstract .....	2
1 Installing Prerequisites for OpenSAF .....	2
1-1 Installing OpenHPI .....	3
1-1-1 Configuring the HP BladeSystem Onboard Administrator .....	3
1-1-2 Configuring OpenHPI for HP BladeSystem c-Class .....	3
1-1-3 Starting the OpenHPI daemon .....	5
1-1-4 Verifying OpenHPI is communicating with the Onboard Administrator .....	5
1-2 Installing SNMP .....	5
1-3 Installing TIPC .....	6
1-3-1 Installing the required kernel headers and development tools .....	6
1-3-2 Installing the TIPC utilities .....	6
1-4 Installing Xerces-c .....	6
2 Installing and Configuring OpenSAF .....	7
2-1 Building and Installing OpenSAF .....	7
2-1-1 Downloading and Compiling OpenSAF RPMs .....	7
2-1-2 Installing the OpenSAF RPMs .....	7
2-2 Configuring OpenSAF .....	7
2-2-1 Configuring each node in the cluster .....	8
2-2-2 Controller node specific configuration .....	10
2.3 Running OpenSAF .....	11
2-3-1 Starting the OpenSAF daemon .....	11
2-3-2 Verifying OpenSAF is running .....	11

# Abstract

This installation guide provides instructions for system administrators installing a four node OpenSAF development cluster with a 2N redundancy model.

The following is a graphical representation of an OpenSAF cluster.

**Figure 1 OpenSA Cluster**



The installation instructions provided in this guide assume that you are using the following hardware and software configurations:

- HP BladeSystem c-Class chassis
- Dedicated ProLiant Server Blades for each node
- Each Blade/Node is installed with Fedora 10 Linux
- The Fedora “Everything” repository enabled via yum
- OpenHPI for hardware management
- A non-production cluster for development use and testing

---

**Note:**

State replication using DRBD is not covered in this document.

---

The OpenSAF version described in this installation guide is BETA software and subject to change. This installation guide was tested specifically with OpenSAF version 3.0.0-M3.

## 1 Installing Prerequisites for OpenSAF

The software described in this installation guide is distributed in the same standard package formats that are provided by the Linux distribution and that provide prerequisite information internally. If you

attempt to install a component that does not have all the necessary prerequisite software, the installation may abort and a list of missing prerequisites will be generated. If this occurs, consult the documentation provided by your Linux distribution for information on locating and installing the prerequisite software and retry the installation.

For full functionality, OpenSAF requires the following components:

- OpenSAF 3.0.0-M3
- OpenHPI 2.12 or later (optional, required for hardware management functionality)
- SNMP stack of the Linux distribution
- Xerces-C
- TIPC
- DRBD (optional, not covered by this installation guide)

## 1-1 Installing OpenHPI

You only need to install OpenHPI on nodes that are used as “controller” nodes; OpenHPI is not needed for “payload” only nodes. In the example configuration, this installation needs to be performed on nodeA and nodeB. To install OpenHPI, login as `root` and enter the following:

```
# yum install openhpi openhpi-devel
```

---

### Note:

Support for the HP BladeSystem c-Class was added to OpenHPI version 2.12.0. Ensure that you have version OpenHPI version 2.12.0 or greater installed.

---

### 1-1-1 Configuring the HP BladeSystem Onboard Administrator

You must configure and Set up a user account in the Onboard Administrator (OA) for each HP BladeSystem c-Class enclosure that you want to manage. The OA is configured at the factory with a default user name and password which can be found on the tag attached to the hardware. To setup or change the login and/or password, refer to the [HP BladeSystem Onboard Administrator User Guide](#). The user account for the plug-in on the OA must have administrator-level privileges. You must also use OA firmware version 2.02 or later.

### 1-1-2 Configuring OpenHPI for HP BladeSystem c-Class

After the installation process, OpenHPI is not automatically configured to use the necessary OA SOAP plug-in that allows OpenHPI to communicate with HP BladeSystem OA. To manually configure OpenHPI, you must disable the simulator plug-in and enable the OA SOAP plug-in by performing the following steps:

1. Copy over the example configuration file by logging in as `root` and entering the following:

```
# cp /usr/share/doc/openhpi-2.12.0/openhpi.conf.example \  
/etc/openhpi/openhpi.conf
```

2. To disable the simulator plug-in, edit the `/etc/openhpi/openhpi.conf` file as follows:

- a. Locate the following lines:

```
## copy must have a unique name.  
handler libsimulator {  
    entity_root = "{SYSTEM_CHASSIS,1}"  
    name = "simulator"
```

```
}
}
```

- b. Comment each line out by adding the pound character (#) to the front of every line. The lines should now look like the following:

```
## copy must have a unique name.
#handler libsimulator {
#   entity_root = "{SYSTEM_CHASSIS,1}"
#   name = "simulator"
#}
```

3. To enable the OA SOAP plug-in, edit the `/etc/openhpi/openhpi.conf` file as follows:

- a. Locate the following lines:

```
#handler liboa_soap {
#   entity_root = "{SYSTEM_CHASSIS,8}"
#   OA_User_Name = "user" # OA user name with admin
#   privileges
#   OA_Password = "passwd" # OA password for above user
#   (required)
#   ACTIVE_OA = "hostname" # Active OA hostname or IP address
#   STANDBY_OA = "hostname" # Standby OA hostname or IP
#   address
#}
```

- b. Uncomment the lines by removing the pound character (#) from the front of each line. The lines should now look like the following:

```
handler liboa_soap {
    entity_root = "{SYSTEM_CHASSIS,8}"
    OA_User_Name = "user" # OA user name with admin
    privileges
    OA_Password = "passwd" # OA password for above user
    (required)
    ACTIVE_OA = "hostname" # Active OA hostname or IP
    address (required)
    STANDBY_OA = "hostname" # Standby OA hostname or IP
    address
}
```

Table 1 provides a list of the OA SOAP Plug-in parameters that are contained in the `openhpi.conf` file, along with a description of each.

**Table 1. OpenHPI OA SOAP Plug-in Configuration Details**

Parameter	Description
<i>entity_root</i>	Indicates the entity root of the entity path. The entity path for the discovered resources are generated by adding the prefix <code>entity_root</code> to the location of the resource in the chassis.
<i>OA_User_Name</i>	Holds the OA user name. It is used for authentication with OA.
<i>OA_Password</i>	Holds the OA password. It is used for authentication with OA.
<i>ACTIVE_OA</i>	Holds the Active OA IP address.
<i>STANDBY_OA</i>	Holds the StandBy OA IP address. This parameter is optional.

4. Change the `SYSTEM_CHASSIS` ID from 8 to 2 on the `entity_root` line as follows:

```
entity_root = "{SYSTEM_CHASSIS,2}"
```

---

**Note:**

OpenSAF assumes a default `SYSTEM_CHASSIS` ID of 2. If you must change the `SYSTEM_CHASSIS` ID OpenSAF configuration and source code must be modified to support the change.

---

### 1-1-3 Starting the OpenHPI daemon

To start the `openhpid` daemon, login as `root` and enter the following:

```
# service openhpid restart
```

### 1-1-4 Verifying OpenHPI is communicating with the Onboard Administrator

The `hpi top` command can be used to traverse through the Resource Presence Table (RPT) and the Resource Data Records (RDR) to verify that the OA SOAP plug-in is correctly configured and able to communicate with the Onboard Administrator.

1. Enter the `hpi top` command.
2. Verify that the output looks similar to the following:

```
hpi top (rev 6571) - This program came with OpenHPI 2.12.0
SAF HPI Version B.02.01
{SYSTEM_CHASSIS,2}
|
+--- {SYSTEM_CHASSIS,2}
|   |__ Inventory Idr Num: 0, Num Areas: 3, Tag: BladeSystem c7000 Enclosure
|   |__ Sensor Num: 2, Type: TEMPERATURE, Category: THRESHOLD, Tag: Enclosure
Temperature
|
+--- {SYSTEM_CHASSIS,2}{SYSTEM_BLADE,1}
|   |__ Inventory Idr Num: 0, Num Areas: 2, Tag: ProLiant BL460c G1
|   |__ Sensor Num: 2, Type: TEMPERATURE, Category: THRESHOLD, Tag: Server Board
Temperature
```

If “BladeSystem c7000 Enclosure” and “ProLiant BL460c G1” are listed in the output, this indicates that OpenHPI is able to communicate with the OA.

## 1-2 Installing SNMP

SNMP is only required on the `nodeA` and `nodeB` controller nodes. To install `snmp`, login to each controller node as `root` and enter the following:

```
# yum install net-snmp net-snmp-devel net-snmp-libs net-snmp-utils
```

---

**Note:**

The proper configuration of SNMP is beyond the scope of this installation guide. Please refer to your distribution documentation for proper configuration instructions.

---

### 1-2-1 Starting the SNMP daemon

To start the openhpid daemon, login as root and enter the following:

```
# service snmpd restart
```

## 1-3 Installing TIPC

OpenSAF utilizes the Transparent Inter-Process Communication (TIPC) protocol for intra-cluster communication. There are two components required to use this module, the TIPC kernel module and the user space utilities. Fedora 10 and many other Linux distributions include the necessary kernel module.

### 1-3-1 Installing the required kernel headers and development tools

To install the required kernel headers and development tools login as root and enter the following:

```
# yum groupinstall 'Development Tools'
# yum install kernel-headers kernel-devel
```

### 1-3-2 Installing the TIPC utilities

To download the tipc utilities bundle, login as the root user and enter the following:

```
# cd /usr/src
# wget http://prdownloads.sourceforge.net/tipc/tipc-1.7.6-bundle.tar.gz?download
# tar zxvf tipc-1.7.6-bundle.tar.gz
# cd tipc-1.7.6-bundle
# tar xvf tipc-1.7.6.tar
# tar xvf tipcutils-1.1.8.tar
# cd tipcutils-1.1.8/tipc-config/
# make INCLUDE_PATH=../../include/
# cp tipc-config /usr/bin/
```

## 1-4 Installing Xerces-c

Xerces-c is only required on controller nodes, nodeA and nodeB. To install Xerces-c, login to each controller node as root and enter the following:

```
# yum install xerces-c xerces-c-devel
```

## 2 Installing and Configuring OpenSAF

This section includes advanced topics on the installation and configuration of OpenSAF.

### 2-1 Building and Installing OpenSAF

#### 2-1-1 Downloading and Compiling OpenSAF RPMs

You only need to build OpenSAF on a single system, the final compiled RPMs can then be copied to each node and installed. To download and build the OpenSAF RPMs, perform the following steps:

1. Login as `root` and enter the following commands:

```
# cd ~
# wget http://download.opensaf.org/releases/opensaf-3.0.M3.tar.gz
# tar zxvf opensaf-3.0.M3.tar.gz
# cd opensaf-3.0.M3
# ./bootstrap.sh
# ./configure --with-hpirev=B02
# make rpm
```

After the compilation is finished, all of the rpms are located in the `~/opensaf-3.0.M3/rpms/RPMS/<arch>/` directory.

2. You must copy the necessary RPMs to each node in the cluster. For example, login as `root` on `nodeA` and enter the following commands:

```
# mkdir ~/OpenSAF
# cp ~/opensaf-3.0.M3/rpms/RPMS/<arch>/ * ~/OpenSAF
# scp -r ~/OpenSAF root@nodeB:
# scp -r ~/OpenSAF root@nodeC:
# scp -r ~/OpenSAF root@nodeD:
```

#### 2-1-2 Installing the OpenSAF RPMs

To install the OpenSAF RPMs, perform the following steps:

1. Login as `root` and enter the following commands:

```
# cd ~/OpenSAF
# rpm -ivh opensaf-common-3.0.M3-1.<arch>.rpm
# rpm -ivh opensaf-libs-3.0.M3-1.<arch>.rpm
# rpm -ivh opensaf-controller-3.0.M3-1.<arch>.rpm
```

2. To install a payload on node, login as `root` and enter the following commands:

```
# cd ~/OpenSAF
# rpm -ivh opensaf-common-3.0.M3-1.<arch>.rpm
# rpm -ivh opensaf-libs-3.0.M3-1.<arch>.rpm
# rpm -ivh opensaf-payload-3.0.M3-1.<arch>.rpm
```

3. To update the dynamic linker runtime binding cache, login as `root` and enter the following:

```
# ldconfig
```

Optionally, you may choose to install the development and debug RPMs.

### 2-2 Configuring OpenSAF

This section describes the minimum configuration necessary to enable an OpenSAF cluster. Please refer to the `00-README.conf` file in the root directory of the OpenSAF source for more detailed configuration information.

## 2-2-1 Configuring each node in the cluster

By default, Fedora 10 enables a firewall that blocks the ports that OpenSAF uses to communicate. Proper firewall configuration is outside the scope of this installation guide. This installation guide only covers installation of a non-production development cluster where external security is not an issue, as this guide documents how to disable the firewall entirely, this is not recommended for a production or publicly connected server. Please see the appropriate distribution documentation on how to open ports 5003 (as defined in `/etc/opensaf/rde.conf`).

To disable the firewall, perform the following steps:

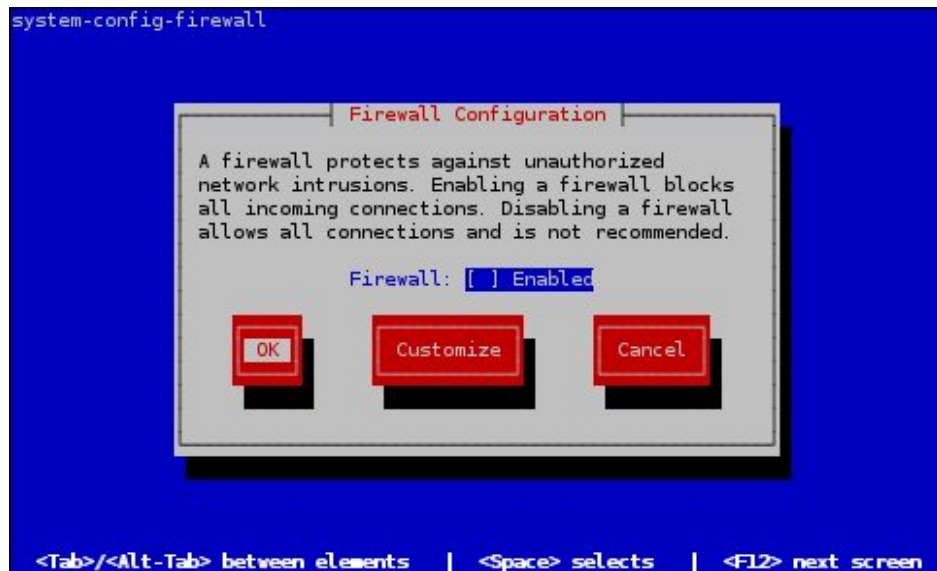
1. Login to every node in the cluster as `root` and enter the following:

```
# setup
```

2. Select "Firewall configuration". Next, select "Run Tool".



3. Highlight the "[\*] Enabled" line and press the spacebar to unmark the selection. Press "OK"



4. At the warning message prompt, select "Yes".

---

**Important:**

Completely disabling the Firewall should not be done on production or internet facing systems.

---



Every node in the cluster needs a unique slot ID, where slot ID of 1 and 2 specify that the node is a controller node and all other numbers specify the node is a payload.

To set the unique slot ID for each node, perform the following steps:

1. On nodeA , login as root and enter the following:

```
# echo 1 > /etc/opensaf/slot_id
```

2. On nodeB, login as root and enter the following:

```
# echo 2 > /etc/opensaf/slot_id
```

3. On nodeC, login as root and enter the following:

```
# echo 3 > /etc/opensaf/slot_id
```

4. On nodeD, login as root and enter the following:

```
# echo 4 > /etc/opensaf/slot_id
```

The TIPC network id is specified towards the end of the line with the default value of 1234. This should be modified to reflect the unique TIPC network ID of your cluster. If there are no other TIPC clusters on your network, the default value 1234 can be used. Otherwise choose a unique value between 1000 and 9999.

For each node, edit the `/etc/opensaf/nodeinit.conf` file and find the TIPC line:

```
.../nid_tipc:TIPC:S:.../nid_tipc:4000::2:1:start eth0 1234:stop
```

---

**Note:**

The TIPC ID must be unique to each cluster on the same SUBNET.

---

1. For every node, edit the `/etc/opensaf/script.conf` file and find the `OPENSAF_TARGET_SYSTEM_ARCH` line:

```
#####
# TARGET SYSTEM ARCHITECTURE
# Choices are:
#   ATCA (default)
#   HP_CCLASS
#   HP_PROLIANT
#####
export OPENSAF_TARGET_SYSTEM_ARCH="ATCA"
```

2. Change "ATCA" to "HP\_CCLASS". The file should look like the following:

```
#####
# TARGET SYSTEM ARCHITECTURE
# Choices are:
#   ATCA (default)
#   HP_CCLASS
#   HP_PROLIANT
#####
export OPENSAF_TARGET_SYSTEM_ARCH="HP_CCLASS"
```

### 2-2-2 Controller node specific configuration

To configure the controller nodes, use the following steps on controller nodeA and nodeB:

1. Edit the `/etc/opensaf/rde.conf` file and set `CONRTOLLER1` to the IP address of nodeA and `CONTROLLER2` to the IP address of nodeB. The file should look similar to the following:

```
export CONTROLLER1=10.232.92.160
export CONTROLLER2=10.232.92.206
export RDE_PORT_NUMBER=5003
```

2. To start the HPI Interface Service (HISv), use the included System BOM file for the HP BladeSystem c-Class. Login as root and enter the following:

```
# cp /etc/opensaf/NCSSystemBOM.xml.hp.c-class \
/etc/opensaf/NCSSystemBOM.xml
```

For OpenSAF CLI commands to work, the correct bay information must be configured in the section of the System BOM file that defines the hardware-deployment configuration. For example, nodeA is in bay 8 of the c-Class chassis, nodeB is in bay 9, nodeC is in bay 10 and nodeD is in bay 11. The hardware-deployment config section in `/etc/oeponsaf/NCSSystemBOM.xml` should look similar to the following:

```
<Hardware-Deployment-Config>
  <EntityDeploymentInstance>
    <Name>chassis_descriptor</Name>
    <EntityTypeInstanceName>CHASSIS</EntityTypeInstanceName>
    <NodeName>safNode=Chassis_Pc_Cluster</NodeName>
    <EntityLocation>2</EntityLocation>
    <HPIEntityType>SAHPI_ENT_SYSTEM_CHASSIS</HPIEntityType>
    <isActivationSourceNCS>1</isActivationSourceNCS>
    <contains>
      <EntityDeploymentInstance>
        <Name>Payload_3</Name>
        <EntityTypeInstanceName>PAYLOAD_PC</EntityTypeInstanceName>
        <NodeName>safNode=PL_2_3</NodeName>
```

```

        <EntityLocation>10</EntityLocation>
        <HPIEntityType>SAHPI_ENT_SYSTEM_BLADE</HPIEntityType>
        <isActivationSourceNCS>1</isActivationSourceNCS>
    </EntityDeploymentInstance>

    <EntityDeploymentInstance>
        <Name>Payload_4</Name>
        <EntityTypeInstanceName>PAYLOAD_PC</EntityTypeInstanceName>
        <NodeName>safNode=PL_2_4</NodeName>
        <EntityLocation>11</EntityLocation>
        <HPIEntityType>SAHPI_ENT_SYSTEM_BLADE</HPIEntityType>
        <isActivationSourceNCS>1</isActivationSourceNCS>
    </EntityDeploymentInstance>
    <EntityDeploymentInstance>
        <Name>Controller_1</Name>
        <EntityTypeInstanceName>CONTROLLER_1</EntityTypeInstanceName>
        <NodeName>safNode=SC_2_1</NodeName>
        <EntityLocation>8</EntityLocation>
        <HPIEntityType>SAHPI_ENT_SYSTEM_BLADE</HPIEntityType>
        <isActivationSourceNCS>1</isActivationSourceNCS>

        <diskBoot/>

    </EntityDeploymentInstance>

    <EntityDeploymentInstance>
        <Name>Controller_2</Name>
        <EntityTypeInstanceName>CONTROLLER_2</EntityTypeInstanceName>
        <NodeName>safNode=SC_2_2</NodeName>
        <EntityLocation>9</EntityLocation>
        <HPIEntityType>SAHPI_ENT_SYSTEM_BLADE</HPIEntityType>
        <isActivationSourceNCS>1</isActivationSourceNCS>

        <diskBoot/>

    </EntityDeploymentInstance>

</contains>
</EntityDeploymentInstance>

```

## 2.3 Running OpenSAF

### 2-3-1 Starting the OpenSAF daemon

To start the opensafd daemon, login as root and enter the following:

```
# service opensafd restart
```

### 2-3-2 Verifying OpenSAF is running

```

Fri Feb 13 13:42:22 MST 2009 - Starting Node Initial
Starting TIPC service... Done.
Starting RDF service... Done.
RDF-ROLE for this System Controller is: 0, ACTIVE
Starting HLFM service... Done.
Starting DTSV service... Done.
Starting MASV service... Done.
Starting PSSV service... Done.
Starting EDSV service... Done.
Starting SUBAGT service... Done.
Starting SCAP service... Done.
Node Initialization Successful.
SUCCESSFULLY SPAWNED ALL SERVICES!!!
Fri Feb 13 13:42:57 MST 2009 - OpenSAF Service Initialization Success

```

For errors and failures, check the log files located under `/var/lib/opensaf/stdout/*.log`. Pay special attention to `ncs_hisv.log` as it logs errors related to hardware management.

At the time this document was written, OpenSAF 3.0 was in active development and changing on a daily basis. The snapshot of Milestone 3 was used as a base for the examples in this document. Due to the rapid rate of development, it is recommended that you use the latest version of OpenSAF 3.0 that is currently available. Newer releases can be found at <http://download.opensaf.org/releases/>

For additional information please refer to the OpenSAF developers site at: <http://devel.opensaf.org> and the OpenSAF users mailing list at <http://list.opensaf.org/maillist/listinfo/users>.

© Copyright 2009 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Linux is a U.S. registered trademark of Linus Torvalds. Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation.

MPN #5900-0035

April 2009

