◆ ◆ ◆ CHAPTER 8

Administering IPMP

This chapter provides tasks for administering interface groups with IP network multipathing (IPMP). The following major topics are discussed:

IPMP Administration Task Maps

In this Solaris release, the `ipmpstat` command is the preferred tool to use to obtain information about IPMP group information. In this chapter, the `ipmpstat` command replaces certain functions of the `ifconfig` command that were used in previous Solaris releases to provide IPMP information.

For information about the different options for the `ipmpstat` command, see "Monitoring IPMP Information" on page 143.

This following sections provide links to the tasks in this chapter.

IPMP Group Creation and Configuration (Task Map)

Task	Description	For Instructions
Plan an IPMP group.	Lists all ancillary information and required tasks before you can configure an IPMP group.	"How to Plan an IPMP Group" on page 124
Configure an IPMP group by using DHCP.	Provides an alternative method to configure IPMP groups by using DHCP.	"How to Configure an IPMP Group by Using DHCP" on page 125
Configure an active-active IPMP group.	Configures an IPMP group in which all underlying interfaces are deployed to host network traffic.	"How to Manually Configure an Active-Active IPMP Group" on page 128
Configure an active-standby IPMP group.	Configures an IPMP group in which one underlying interface is kept inactive as a reserve.	"How to Manually Configure an Active-Standby IPMP Group" on page 131

IPMP Group Maintenance (Task Map)

Task	Description	For Instructions
Add an interface to an IPMP group.	Configures a new interface as a member of an existing IPMP group.	"How to Add an Interface to an IPMP Group" on page 134
Remove an interface from an IPMP group.	Removes an interface from an IPMP group.	"How to Remove an Interface From an IPMP Group" on page 135
Add IP addresses to or remove IP addresses from an IPMP group.	Adds or removes addresses for an IPMP group.	"How to Add or Remove IP Addresses" on page 135
Change an interface's IPMP membership.	Moves interfaces among IPMP groups.	"How to Move an Interface From One IPMP Group to Another Group" on page 136
Delete an IPMP group.	Deletes an IPMP group that is no longer needed.	"How to Delete an IPMP Group" on page 137
Replace cards that failed.	Removes or replaces failed NICs of an IPMP group.	"How to Replace a Physical Card That Has Failed" on page 141

Probe-Based Failure Detection Configuration (Task Map)

Task	Description	For Instructions
Manually specify target systems	Identifies and adds systems to be targeted for probe-based failure detection.	"How to Manually Specify Target Systems for Probe-Based Failure Detection" on page 139
Configure the behavior of probe-based failure detection.	Modifies parameters to determine the behavior of probe-based failure detection.	"How to Configure the Behavior of the IPMP Daemon" on page 139

IPMP Group Monitoring (Task Map)

Task	Description	For Instructions
Obtain group information.	Displays information about an IPMP group.	"How to Obtain IPMP Group Information" on page 144
Obtain data address information.	Displays information about the data addresses that are used by an IPMP group.	"How to Obtain IPMP Data Address Information" on page 145
Obtain IPMP interface information.	Displays information about the underlying interfaces of IPMP interfaces or groups.	"How to Obtain Information About Underlying IP Interfaces of a Group" on page 145
Obtain probe target information.	Displays information about targets of probe-based failure detection.	"How to Obtain IPMP Probe Target Information" on page 147
Obtain probe information.	Displays real-time information about ongoing probes in the system.	"How to Observe IPMP Probes" on page 148
Customize the information display for monitoring IPMP groups.	Determines the IPMP information that is displayed.	"How to Customize the Output of the `ipmpstat` Command in a Script" on page 149

Configuring IPMP Groups

This section provides procedures that are used to plan and configure IPMP groups.

▼ How to Plan an IPMP Group

The following procedure includes the required planning tasks and information to be gathered prior to configuring an IPMP group. The tasks do not have to be performed in sequence.

1 Determine the general IPMP configuration that would suit your needs.

Your IPMP configuration depends on what your network needs to handle the type of traffic that is hosted on your system. IPMP spreads outbound network packets across the IPMP group's interfaces, and thus improves network throughput. However, for a given TCP connection, inbound traffic normally follows only one physical path to minimize the risk of processing out-of-order packets.

Thus, if your network handles a huge volume of outbound traffic, configuring multiple interfaces into an IPMP group can improve network performance. If instead, the system hosts heavy inbound traffic, then the number of interfaces in the group does not necessarily improve performance by load spreading traffic. However, having multiple interfaces helps to guarantee network availability during interfaces failure.

2 For SPARC based systems, verify that each interface in the group has a unique MAC address.

To configure a unique MAC address for each interface in the system, see "SPARC: How to Ensure That the MAC Address of an Interface Is Unique" on page 38.

3 Ensure that the same set of STREAMS modules is pushed and configured on all interfaces in the IPMP group.

All interfaces in the same group must have the same STREAMS modules configured in the same order.

a. Check the order of STREAMS modules on all interfaces in the prospective IPMP group.

You can print a list of STREAMS modules by using the `ifconfig` *interface* `modlist` command. For example, here is the `ifconfig` output for an `hme0` interface:

```
# ifconfig hme0 modlist
    0 arp
    1 ip
    2 hme
```

Interfaces normally exist as network drivers directly below the IP module, as shown in the output from `ifconfig hme0 modlist`. They should not require additional configuration.

However, certain technologies insert themselves as a STREAMS module between the IP module and the network driver. If a STREAMS module is stateful, then unexpected behavior

can occur on failover, even if you push the same module onto all of the interfaces in a group. However, you can use stateless STREAMS modules, provided that you push them in the same order on all interfaces in the IPMP group.

b. Push the modules of an interface in the standard order for the IPMP group.

`ifconfig` *interface* **`modinsert`** *module-name*`@`*position*

```
ifconfig hme0 modinsert vpnmod@3
```

4 Use the same IP addressing format on all interfaces of the IPMP group.

If one interface is configured for IPv4, then all interfaces of the group must be configured for IPv4. For example, if you add IPv6 addressing to one interface, then all interfaces in the IPMP group must be configured for IPv6 support.

5 Determine the type of failure detection that you want to implement.

For example, if you want to implement probe-based failure detection, then you must configure test addresses on the underlying interfaces. For related information, see"Types of Failure Detection in IPMP" on page 108.

6 Ensure that all interfaces in the IPMP group are connected to the same local network.

For example, you can configure Ethernet switches on the same IP subnet into an IPMP group. You can configure any number of interfaces into an IPMP group.

Note – You can also configure a single interface IPMP group, for example, if your system has only one physical interface. For related information, see "Types of IPMP Interface Configurations" on page 106.

---

7 Ensure that the IPMP group does not contain interfaces with different network media types.

The interfaces that are grouped together should be of the same interface type, as defined in `/usr/include/net/if_types.h`. For example, you cannot combine Ethernet and Token ring interfaces in an IPMP group. As another example, you cannot combine a Token bus interface with asynchronous transfer mode (ATM) interfaces in the same IPMP group.

8 For IPMP with ATM interfaces, configure the ATM interfaces in LAN emulation mode.

IPMP is not supported for interfaces using Classical IP over ATM.

▼ How to Configure an IPMP Group by Using DHCP

In the current IPMP implementation, IPMP groups can be configured with Dynamic Host Configuration Protocol (DHCP) support.

A multiple-interfaced IPMP group can be configured with active-active interfaces or active-standby interfaces. For related information, see "Types of IPMP Interface Configurations" on page 106. The following procedure describes steps to configure an active-standby IPMP group by using DHCP.

Before You Begin Make sure that IP interfaces that will be in the prospective IPMP group have been correctly configured over the system's network data links. For procedures to configure links and IP interfaces, see "Data Link and IP Interface Configuration (Tasks)" on page 37. For information about configuring IPv6 interfaces, see "Configuring an IPv6 Interface" in *System Administration Guide: IP Services*.

Additionally, if you are using a SPARC system, configure a unique MAC address for each interface. For procedures, see "SPARC: How to Ensure That the MAC Address of an Interface Is Unique" on page 38.

Finally, if you are using DHCP, make sure that the underlying interfaces have infinite leases. Otherwise, in case of a group failure, the test addresses will expire and the IPMP daemon will then revert to link-based failure detection. Such circumstances would trigger errors in the manner the group's failure detection behaves during interface recovery. For more information about configuring DHCP, refer to Chapter 12, "Planning for DHCP Service (Tasks)," in *System Administration Guide: IP Services*.

1 On the system on which you want to configure the IPMP group, assume the Primary Administrator role, or become superuser.

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 Create an IPMP interface.

```
# ifconfig ipmp-interface ipmp [group group-name]
```

Note – To configure IPv6 IPMP interfaces, use the same command syntax for configuring IPv6 interfaces by specifying `inet6` in the `ifconfig` command, for example:

```
# ifconfig ipmp-interface inet6 ipmp [group group-name]
```

This note applies to all configuration procedures that involve IPv6 IPMP interfaces.

---

ipmp-interface Specifies the name of the IPMP interface. You can assign any meaningful name to the IPMP interface. As with any IP interface, the name consists of a string and a number, such as `ipmp0`.

group-name Specifies the name of the IPMP group. The name can be any name of your choice. Assigning a group name is optional. By default, the name of the IPMP interface also becomes the name of the IPMP group. Preferably, retain this default setting by not using the *group-name* option.

Note – The syntax in this step uses the preferred explicit method of creating an IPMP group by creating the IPMP interface.

An alternative method to create an IPMP group is implicit creation, in which you use the syntax `ifconfig` *interface* `group` *group-name*. In this case, the system creates the lowest available `ipmp`*N* to become the group's IPMP interface. For example, if `ipmp0` already exists for group `acctg`, then the syntax `ifconfig ce0 group fieldops` causes the system to create `ipmp1` for group `fieldops`. All `UP` data addresses of `ce0` are then assigned to `ipmp1`.

However, implicit creation of IPMP groups is not encouraged. Support for implicit creation is provided only to have compatible implementation with previous Solaris releases. Explicit creation provides optimal control over the configuration of IPMP interfaces.

---

3 Add underlying IP interfaces that will contain test addresses to the IPMP group, including the standby interface.

```
# ifconfig interface group group-name -failover [standby] up
```

4 Have DHCP configure and manage the data addresses on the IPMP interface.

You need to plumb as many logical IPMP interfaces as data addresses, and then have DHCP configure and manage the addresses on these interfaces as well.

```
# ifconfig ipmp-interface dhcp start primary
# ifconfig ipmp-interface:n plumb
# ifconfig ipmp-interface:n dhcp start
```

5 Have DHCP manage the test addresses in the underlying interfaces.

You need to issue the following command for each underlying interface of the IPMP group.

```
# ifconfig interface dhcp start
```

Example 8–1 Configuring an IPMP Group With DHCP

This example shows how to configure an active-standby IPMP group with DHCP. This example is based on Figure 7–1, which contains the following information:

- Three underlying interfaces, `subitops0`, `subitops1`, and `subitops2` are designated members of the IPMP group.
- The IPMP interface `itops0` shares the same name with the IPMP group.
- `subitops2` is the designated standby interface.

- To use probe-based failure detection, all the underlying interfaces are assigned test addresses.

```
# ifconfig itops0 ipmp

# ifconfig subitops0 plumb group itops0 -failover up
# ifconifg subitops1 plumb group itops0 -failover up
# ifconfig subitops2 plumb group itops0 -failover standby up

# ifconfig itops0 dhcp start primary
# ifconfig itops0:1 plumb
# ifconfig itops0:1 dhcp start

# ifconfig subitops0 dhcp start
# ifconfig subitops1 dhcp start
# ifconfig subitops2 dhcp start
```

To make the test address configuration persistent, you would need to type the following commands:

```
# touch /etc/dhcp.itops0 /etc/dhcp.itops0:1
# touch /etc/dhcp.subitops0 /etc/dhcp.subitops1 /etc/dhcp.subitops2

# echo group itops0 -failover up > /etc/hostname.subitops0
# echo group itops0 -failover up > /etc/hostname.subitops1
# echo group itops0 -failover standby up > /etc/hostname.subitops2
# echo ipmp > /etc/hostname.itops0
```

▼ How to Manually Configure an Active-Active IPMP Group

The following procedure describes steps to manually configure an active-active IPMP group.

Before You Begin Make sure that IP interfaces that will be in the prospective IPMP group have been correctly configured over the system's network data links. For procedures to configure links and IP interfaces, see "Data Link and IP Interface Configuration (Tasks)" on page 37. For information about configuring IPv6 interfaces, see "Configuring an IPv6 Interface" in *System Administration Guide: IP Services*.

Additionally, if you are using a SPARC system, configure a unique MAC address for each interface. For procedures, see "SPARC: How to Ensure That the MAC Address of an Interface Is Unique" on page 38.

1 **On the system on which you want to configure the IPMP group, assume the Primary Administrator role, or become superuser.**

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 **Create an IPMP interface.**

```
# ifconfig ipmp-interface ipmp [group group-name]
```

ipmp-interface Specifies the name of the IPMP interface. You can assign any meaningful name to the IPMP interface. As with any IP interface, the name consists of a string and a number, such as `ipmp0`.

group-name Specifies the name of the IPMP group. The name can be any name of your choice. Any nun-null name is valid, provided that the name does not exceed 31 characters. Assigning a group name is optional. By default, the name of the IPMP interface also becomes the name of the IPMP group. Preferably, retain this default setting by not using the *group-name* option.

Note – The syntax in this step uses the preferred explicit method of creating an IPMP group by creating the IPMP interface.

An alternative method to create an IPMP group is implicit creation, in which you use the syntax `ifconfig` *interface* `group` *group-name*. In this case, the system creates the lowest available `ipmp`*N* to become the group's IPMP interface. For example, if `ipmp0` already exists for group `acctg`, then the syntax `ifconfig ce0 group fieldops` causes the system to create `ipmp1` for group `fieldops`. All `UP` data addresses of `ce0` are then assigned to `ipmp1`.

However, implicit creation of IPMP groups is not encouraged. Support for implicit creation is provided only to have compatible implementation with previous Solaris releases. Explicit creation provides optimal control over the configuration of IPMP interfaces.

---

3 **Add underlying IP interfaces to the group.**

```
# ifconfig ip-interface group group-name
```

Note – In a dual-stack environment, placing the IPv4 instance of an interface under a particular group automatically places the IPv6 instance under the same group as well.

---

4 **Add data addresses to the IPMP interface.**

```
# ifconfig plumb ipmp-interface ip-address up
# ifconfig ipmp-interface addif ip-address up
```

For additional options that you can use with the `ifconfig` command while adding addresses, refer to the `ifconfig(1M)` man page.

5 Configure test addresses on the underlying interfaces.

```
# ifconfig interface -failover ip-address up
```

Note – You need to configure a test address only if you want to use probe-based failure detection on a particular interface.

All test IP addresses in an IPMP group must use the same network prefix. The test IP addresses must belong to a single IP subnet.

---

6 (Optional) Preserve the IPMP group configuration across reboots.

To configure an IPMP group that persists across system reboots, you would edit the `hostname` configuration file of the IPMP interface to add data addresses. Then, if you want to use test addresses, you would edit the `hostname` configuration file of one of the group's underlying IP interface. Note that data and test addresses can be both IPv4 and IPv6 addresses. Perform the following steps:

a. Edit the `/etc/hostname.`*ipmp-interface* file by adding the following lines:

```
ipmp group group-name data-address up

addif data-address
...
```

You can add more data addresses on separate `addif` lines in this file.

b. Edit the `/etc/hostname.`*interface* file of the underlying IP interfaces that contain the test address by adding the following line:

```
group group-name -failover test-address up
```

Follow this same step to add test addresses to other underlying interfaces of the IPMP group.

Caution – When adding test address information on the `/etc/hostname.`*interface* file, make sure to specify the `-failover` option before the `up` keyword. Otherwise, the test IP addresses will be treated as data addresses and would cause problems for system administration. Preferably, set the `-failover` option before specifying the IP address.

How to Manually Configure an Active-Standby IPMP Group

For more information about standby interfaces, see "Types of IPMP Interface Configurations" on page 106. The following procedure configures an IPMP group where one interface is kept as a reserve. This interface is deployed only when an active interface in the group fails.

1 On the system on which you want to configure the IPMP group, assume the Primary Administrator role, or become superuser.

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 Create an IPMP interface.

```
# ifconfig ipmp-interface ipmp [group group-name]
```

ipmp-interface Specifies the name of the IPMP interface. You can assign any meaningful name to the IPMP interface. As with any IP interface, the name consists of a string and a number, such as `ipmp0`.

group-name Specifies the name of the IPMP group. The name can be any name of your choice. Any nun-null name is valid, provided that the name does not exceed 31 characters. Assigning a group name is optional. By default, the name of the IPMP interface also becomes the name of the IPMP group. Preferably, retain this default setting by not using the *group-name* option.

Note – The syntax in this step uses the preferred explicit method of creating an IPMP group by creating the IPMP interface.

An alternative method to create an IPMP group is implicit creation, in which you use the syntax `ifconfig` *interface* `group` *group-name*. In this case, the system creates the lowest available `ipmp`*N* to become the group's IPMP interface. For example, if `ipmp0` already exists for group `acctg`, then the syntax `ifconfig ce0 group fieldops` causes the system to create `ipmp1` for group `fieldops`. All `UP` data addresses of `ce0` are then assigned to `ipmp1`.

However, implicit creation of IPMP groups is not encouraged. Support for implicit creation is provided only to have compatible implementation with previous Solaris releases. Explicit creation provides optimal control over the configuration of IPMP interfaces.

---

3 Add underlying IP interfaces to the group.

```
# ifconfig ip-interface group group-name
```

Note – In a dual-stack environment, placing the IPv4 instance of an interface under a particular group automatically places the IPv6 instance under the same group as well.

4 Add data addresses to the IPMP interface.

```
# ifconfig plumb ipmp-interface ip-address up
# ifconfig ipmp-interface addif ip-address up
```

For additional options that you can use with the `ifconfig` command while adding addresses, refer to the `ifconfig(1M)` man page.

5 Configure test addresses on the underlying interfaces.

- To configure a test address on an active interface, use the following command:

  ```
  # ifconfig interface -failover ip-address up
  ```

- To configure a test address on a designated standby interface, use the following command:

  ```
  # ifconfig interface -failover ip-address standby up
  ```

Note – You need to configure a test address only if you want to use probe-based failure detection on a particular interface.

All test IP addresses in an IPMP group must use the same network prefix. The test IP addresses must belong to a single IP subnet.

6 (Optional) Preserve the IPMP group configuration across reboots.

To configure an IPMP group that persists across system reboots, you would edit the `hostname` configuration file of the IPMP interface to add data addresses. Then, if you want to use test addresses, you would edit the `hostname` configuration file of one of the group's underlying IP interface. Note that data and test addresses can be both IPv4 and IPv6 addresses. Perform the following steps:

a. Edit the `/etc/hostname.`*ipmp-interface* file by adding the following lines:

```
ipmp group group-name data-address up
addif data-address
...
```

You can add more data addresses on separate `addif` lines in this file.

b. Edit the `/etc/hostname.`*interface* file of the underlying IP interfaces that contain the test address by adding the following line:

```
group group-name -failover test-address up
```

Follow this same step to add test addresses to other underlying interfaces of the IPMP group. For a designated standby interface, the line must be as follows:

```
group group-name -failover test-address standby up
```

Caution – When adding test address information on the `/etc/hostname.`*interface* file, make sure to specify the `-failover` option before the `up` keyword. Otherwise, the test IP addresses will be treated as data addresses and would cause problems for system administration. Preferably, set the `-failover` option before specifying the IP address.

Example 8–2 Configuring an Active-Standby IPMP Group

This example shows how to manually create the same persistent active-standby IPMP configuration that is provided in Example 8–1.

```
# ifconfig itops0 ipmp

# ifconfig subitops0 group itops0
# ifconfig subitops1 group itops0
# ifconfig subitops2 group itops0

# ifconfig itops0 192.168.10.10/24 up
# ifconfig itops0 addif 192.168.10.15/24 up

# ifconfig subitops0 -failover 192.168.85.30/24 up
# ifconfig subitops1 -failover 192.168.86.32/24 up
# ifconfig subitops2 -failover 192.168.86.34/24 standby up

# ipmpstat -g
GROUP     GROUPNAME   STATE      FDT        INTERFACES
itops0    itops0      ok         10.00s     subitops0 subitops1 (subitops2)

# ipmpstat -t
INTERFACE      MODE     TESTADDR        TARGETS
subitops0      routes   192.168.10.30   192.168.10.1
subitops1      routes   192.168.10.32   192.168.10.1
subitops2      routes   192.168.10.34   192.168.10.5

# vi /etc/hostname.itops0
ipmp group itops0 192.168.10.10/24 up
addif 192.168.10.15/24 up

# vi /etc/hostname.subitops0
group itops0 -failover 192.168.10.30/24 up
```

```
# vi /etc/hostname.subitops1
group itops0 -failover 192.168.10.32/24 up

# vi /etc/hostname.subitops2
group itops0 -failover 192.168.10.34/24 standby up
```

Maintaining IPMP Groups

This section contains tasks for maintaining existing IPMP groups and the interfaces within those groups. The tasks presume that you have already configured an IPMP group, as explained in "Configuring IPMP Groups" on page 124.

▼ How to Add an Interface to an IPMP Group

Before You Begin Make sure that the interface that you add to the group matches all the constraints to be in the group. For a list of the requirements of an IPMP group, see "How to Plan an IPMP Group" on page 124.

1 On the system with the IPMP group configuration, assume the Primary Administrator role or become superuser.

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 Add the IP interface to the IPMP group.

```
# ifconfig interface group group-name
```

The interface specified in *interface* becomes a member of IPMP group *group-name*.

Example 8–3 Adding an Interface to an IPMP Group

To add the interface `hme0` to the IPMP group `itops0`, you would type the following command:

```
# ifconfig hme0 group itops0
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT       INTERFACES
itops0  itops0      ok         10.00s    subitops0 subitops1 hme0
```

How to Remove an Interface From an IPMP Group

1 **On the system with the IPMP group configuration, assume the Primary Administrator role or become superuser.**

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 **Remove the interface from the IPMP group.**

```
# ifconfig interface group ""
```

The quotation marks indicate a null string.

Example 8–4 Removing an Interface From a Group

To remove the interface `hme0` from the IPMP group `itops0`, you would type the following command:

```
# ifconfig hme0 group ""
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT       INTERFACES
itops0  itops0      ok         10.00s    subitops0 subitops1
```

How to Add or Remove IP Addresses

You use the `ifconfig addif` syntax to add addresses or the `ifconfig removeif` command to remove addresses from interfaces. In the current IPMP implementation, test addresses are hosted on the underlying IP interface, while data addresses are assigned to the IPMP interface. The following procedures describes how to add or remove IP addresses that are either test addresses or data addresses.

1 **Assume the role of Primary Administrator, or become superuser.**

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 **Add or remove data addresses.**

- To add data addresses to the IPMP group, type the following command:

  ```
  # ifconfig ipmp-interface addif ip-address up
  ```

- To remove an address from the IPMP group, type the following command:

```
# ifconfig ipmp-interface removeif ip-address
```

3 Add or remove test addresses.

- To assign a test address to an underlying interface of the IPMP group, type the following command:

  ```
  # ifconfig interface addif -failover ip-address up
  ```

- To remove a test address from an underlying interface of the IPMP group, type the following command:

  ```
  # ifconfig interface removeif ip-address
  ```

Example 8–5 Removing a Test Address From an Interface

The following example uses the configuration of `itops0` in Example 8–2. The step removes the test address from the interface `subitops0`.

```
# ipmpstat -t
INTERFACE      MODE     TESTADDR        TARGETS
subitops0      routes   192.168.10.30   192.168.10.1

# ifconfig subitops0 removeif 192.168.85.30
```

▼ How to Move an Interface From One IPMP Group to Another Group

You can place an interface in a new IPMP group when the interface belongs to an existing IPMP group. You do not need to remove the interface from the current IPMP group. When you place the interface in a new group, the interface is automatically removed from any existing IPMP group.

1 On the system with the IPMP group configuration, assume the Primary Administrator role or become superuser.

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 Move the interface to a new IPMP group.

```
# ifconfig interface group group-name
```

Placing the interface in a new group automatically removes the interface from any existing group.

Example 8–6 Moving an Interface to a Different IPMP Group

This example assumes that the underlying interfaces of your group are subitops0, subitops1, subitops2, and hme0. To change the IPMP group of interface hme0 to the group cs-link1, you would type the following:

```
# ifconfig hme0 group cs-link1
```

This command removes the `hme0` interface from IPMP group `itops0` and then puts the interface in the group `cs-link1`.

How to Delete an IPMP Group

Use this procedure if you no longer need a specific IPMP group.

1. **Assume the role of Primary Administrator, or become superuser.**

   The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2. **Identify the IPMP group and the underlying IP interfaces.**

   ```
   # ipmpstat -g
   ```

3. **Delete all IP interfaces that currently belong to the IPMP group.**

   ```
   # ifconfig ip-interface group ""
   ```

   Repeat this step for all the IP interfaces that belong to the group.

   Note – To successfully delete an IPMP interface, no IP interface must exist as part of the IPMP group.

4. **Delete the IPMP interface.**

   ```
   # ifconfig ipmp-interface unplumb
   ```

   After you unplumb the IPMP interface, any IP address that is associated with the interface is deleted from the system.

5. **To make the deletion persistent, perform the following additional steps:**

   a. **Delete the IPMP interface's corresponding `hostname` file.**

      ```
      # rm /etc/hostname.ipmp-interface
      ```

b. Remove the "`group`" keywords in the `hostname` files of the underlying interfaces.

Example 8–7 Deleting an IPMP Interface

To delete the interface `itops0` that has the underlying IP interface `subitops0` and `subitops1`, you would type the following commands:

```
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT        INTERFACES
itops0  itops0      ok         10.00s     subitops0 subitops1

# ifconfig subitops0 group ""
# ifconfig subitops1 group ""
# ifconfig itops0 unplumb
# rm /etc/hostname.itops0
```

You would then edit the files `/etc/hostname.subitops0` and `/etc/hostname.subitops1` to remove "`group`" entries in those files.

Configuring for Probe-Based Failure Detection

Probe-based failure detection involves the use of target systems, as explained in "Probe-Based Failure Detection" on page 109. In identifying targets for probe-based failure detection, the `in.mpathd` daemon operates in two modes: router target mode or multicast target mode. In the router target mode, the multipathing daemon probes targets that are defined in the routing table. If no targets are defined, then the daemon operates in multicast target mode, where multicast packets are sent out to probe neighbor hosts on the LAN.

Preferably, you should set up host targets for the `in.mpathd` daemon to probe. For some IPMP groups, the default router is sufficient as a target. However, for some IPMP groups, you might want to configure specific targets for probe-based failure detection. To specify the targets, set up host routes in the routing table as probe targets. Any host routes that are configured in the routing table are listed before the default router. IPMP uses the explicitly defined host routes for target selection. Thus, you should set up host routes to configure specific probe targets rather than use the default router.

To set up host routes in the routing table, you use the route command. You can use the `-p` option with this command to add persistent routes. For example, `route -p add` adds a route which will remain in the routing table even after you reboot the system. The `-p` option thus allows you to add persistent routes without needing any special scripts to recreate these routes every system startup. To optimally use probe-based failure detection, make sure that you set up multiple targets to receive probes.

The sample procedure that follows shows the exact syntax to add persistent routes to targets for probe-based failure detection. For more information about the options for the `route` command, refer to the `route(1M)` man page.

Consider the following criteria when evaluating which hosts on your network might make good targets.

- Make sure that the prospective targets are available and running. Make a list of their IP addresses.
- Ensure that the target interfaces are on the same network as the IPMP group that you are configuring.
- The netmask and broadcast address of the target systems must be the same as the addresses in the IPMP group.
- The target host must be able to answer ICMP requests from the interface that is using probe-based failure detection.

How to Manually Specify Target Systems for Probe-Based Failure Detection

1. **Log in with your user account to the system where you are configuring probe-based failure detection.**

2. **Add a route to a particular host to be used as a target in probe-based failure detection.**

   ```
   $ route -p add -host destination-IP gateway-IP -static
   ```

   where *destination-IP* and *gateway-IP* are IPv4 addresses of the host to be used as a target. For example, you would type the following to specify the target system `192.168.10.137`, which is on the same subnet as the interfaces in IPMP group `itops0`:

   ```
   $ route -p add -host 192.168.10.137 192.168.10.137 -static
   ```

   This new route will be automatically configured every time the system is restarted. If you want to define only a temporary route to a target system for probe-based failure detection, then do not use the `-p` option.

3. **Add routes to additional hosts on the network to be used as target systems.**

How to Configure the Behavior of the IPMP Daemon

Use the IPMP configuration file `/etc/default/mpathd` to configure the following system-wide parameters for IPMP groups.

- FAILURE_DETECTION_TIME
- TRACK_INTERFACES_ONLY_WITH_GROUPS
- FAILBACK

1 On the system with the IPMP group configuration, assume the Primary Administrator role or become superuser.

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 Edit the `/etc/default/mpathd` file.

Change the default value of one or more of the three parameters.

a. Type the new value for the `FAILURE_DETECTION_TIME` parameter.

```
FAILURE_DETECTION_TIME=n
```

where *n* is the amount of time in seconds for ICMP probes to detect whether an interface failure has occurred. The default is 10 seconds.

b. Type the new value for the `FAILBACK` parameter.

```
FAILBACK=[yes | no]
```

- *yes*– The *yes* value is the default for the failback behavior of IPMP. When the repair of a failed interface is detected, network access fails back to the repaired interface, as described in "Detecting Physical Interface Repairs" on page 111.
- *no* – The *no* value indicates that data traffic does not move back to a repaired interface. When a failed interfaces is detected as repaired, the `INACTIVE` flag is set for that interface. This flag indicates that the interface is currently not to be used for data traffic. The interface can still be used for probe traffic.

  For example, the IPMP group `ipmp0` consists of two interfaces, `ce0` and `ce1`. In the `/etc/default/mpathd` file, the *FAILBACK=no* parameter is set. If `ce0` fails, then it is flagged as `FAILED` and becomes unusable. After repair, the interface is flagged as `INACTIVE` and remains unusable because of the *FAILBACK=no* setting.

  If `ce1` fails and only `ce0` is in the `INACTIVE` state, then `ce0`'s `INACTIVE` flag is cleared and the interface becomes usable. If the IPMP group has other interfaces that are also in the `INACTIVE` state, then any one of these `INACTIVE` interfaces, and not necessarily `ce0`, can be cleared and become usable when `ce1` fails.

c. Type the new value for the `TRACK_INTERFACES_ONLY_WITH_GROUPS` parameter.

```
TRACK_INTERFACES_ONLY_WITH_GROUPS=[yes | no]
```

Note – For information about this parameter and the anonymous group feature, see "Failure Detection and the Anonymous Group Feature" on page 110.

- *yes*– The *yes* value is the default for the behavior of IPMP. This parameter causes IPMP to ignore network interfaces that are not configured into an IPMP group.
- *no* – The *no* value sets failure and repair detection for *all* network interfaces, regardless of whether they are configured into an IPMP group. However, when a failure or repair is detected on an interface that is not configured into an IPMP group, no action is triggered in IPMP to maintain the networking functions of that interface. Therefore, the*no* value is only useful for reporting failures and does not directly improve network availability.

3 **Restart the `in.mpathd` daemon.**

```
# pkill -HUP in.mpathd
```

Recovering an IPMP Configuration With Dynamic Reconfiguration

This section contains procedures that relate to administering systems that support dynamic reconfiguration (DR).

▼ How to Replace a Physical Card That Has Failed

This procedure explains how to replace a physical card on a system that supports DR. The procedure assumes the following conditions:

- You assigned administratively chosen names to the data links over which you configured the IP interfaces. These interfaces are `subitops0` and `subitops1`.
- Both interfaces belong to the IPMP group, `itops0`.
- The interface `subitops0` contains a test address.
- The interface `subitops0` has failed, and you need to remove `subitops0`'s underlying card, `ce`.
- You are replacing the `ce` card with a `bge` card.
- The configuration files correspond to the interfaces and use the interfaces' customized link names, thus `/etc/hostname.subitops0` and `/etc/hostname.subitops1`.

Before You Begin The procedures for performing DR vary with the type of system. Therefore, make sure that you complete the following:

- Ensure that your system supports DR.
- Consult the appropriate manual that describes DR procedures on your system. For Sun hardware, all systems that support DR are servers. To locate current DR documentation on Sun systems, search for "dynamic reconfiguration" on `http://docs.sun.com`.

Note – The steps in the following procedure refer only to aspects of DR that are specifically related to IPMP and the use of link names. The procedure does not contain the complete steps to perform DR. For example, some layers beyond the IP layer require manual configuration steps, such as for ATM and other services, if the configuration is not automated. Follow the appropriate DR documentation for your system.

---

1 On the system with the IPMP group configuration, assume the Primary Administrator role or become superuser.

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 Perform the appropriate DR steps to remove the failed NIC from the system.

- If you are removing the card without intending to insert a replacement, then skip the rest of the steps after you remove the card.
- If you are replacing a card, then proceed to the subsequent steps .

3 Make sure that the replacement NIC is not being referenced by other configurations in the system.

For example, the replacement NIC you install is `bge0`. If a `/etc/hostname.bge0` file exists on the system, remove that file.

```
# rm /etc/hostname.bge0
```

4 Replace the default link name of the replacement NIC with the link name of the failed card.

By default, the link name of the `bge` card that replaces the failed `ce` card is `bge`*n*, where *n* is the instance number, such as `bge0`.

```
# dladm rename-link bge0 subitops0
```

This step transfers the network configuration of `subitops0` to `bge0`.

5 Attach the replacement NIC to the system.

6 Complete the DR process by enabling the new NIC's resources to become available for use.

For example, you use the `cfgadm` command to perform this step. For more information, see the `cfgadm(1M)` man page.

After this step, the new interface is configured with the test address, added as an underlying interface of the IPMP group, and deployed either as an active or a standby interface, all depending on the configurations that are specified in `/etc/hostname.subitops0`. The kernel can then allocate data addresses to this new interface according to the contents of the `/etc/hostname.`*ipmp-interface* configuration file.

About Missing Interfaces at System Boot

Certain systems might have the following configurations:

- An IPMP group is configured with underlying IP interfaces
- A `/etc/hostname.`*interface* file exists for one underlying IP interface.
- The physical hardware that is associated with the `/etc/hostname` file is missing.

With the new IPMP implementation where data addresses belong to the IPMP interface, recovering the missing interface becomes automatic. During system boot, the boot script constructs a list of failed interfaces, including interfaces that are missing. Based on the `/etc/hostname` file of the IPMP interface as well as the `hostname` files of the underlying IP interfaces, the boot script can determine to which IPMP group an interface belongs. When the missing interface is subsequently dynamically reconfigured on the system, the script then automatically adds that interface to the appropriate IPMP group and the interface becomes immediately available for use.

Monitoring IPMP Information

The following procedures use the `ipmpstat` command, enabling you to monitor different aspects of IPMP groups on the system. You can observe the status of the IPMP group as a whole or its underlying IP interfaces. You can also verify the configuration of data and test addresses for the group. Information about failure detection is also obtained by using the `ipmpstat` command. For more details about the `ipmpstat` command and its options, see the PLACEHOLDER IPMPSTAT MAN PAGE.

By default, host names are displayed on the output instead of the numeric IP addresses, provided that the host names exist. To list the numeric IP addresses in the output, use the `-n` option together with other options to display specific IPMP group information.

Note – In the following procedures, use of the `ipmpstat` command does not require system administrator privileges, unless stated otherwise.

▼ How to Obtain IPMP Group Information

Use this procedure to list the status of the various IPMP groups on the system, including the status of their underlying interfaces. If probe-based failure detection is enabled for a specific group, the command also includes the failure detection time for that group.

- **Display the IPMP group information.**

```
$ ipmpstat -g
GROUP   GROUPNAME   STATE      FDT        INTERFACES
itops0  itops0      ok         10.00s     subitops0 subitops1
acctg1  acctg1      failed     --         [hme0 hme1]
field2  field2      degraded   20.00s     fops0 fops3 (fops2) [fops1]
```

`GROUP` Specifies the IPMP interface name. In the case of an anonymous group, this field will be empty. For more information about anonymous groups, see the `in.mpathd(1M)` man page.

`GROUPNAME` Specifies the name of the IPMP group. In the case of an anonymous group, this field will be empty.

`STATE` Indicates a group's current status, which can be one of the following:

- `ok` indicates that all underlying interfaces of the IPMP group are usable.
- `degraded` indicates that some of the underlying interfaces in the group are unusable.
- `failed` indicates that all of the group's interfaces are unusable.

`FDT` Specifies the failure detection time, if failure detection is enabled. If failure detection is disabled, this field will be empty.

`INTERFACES` Specifies the underlying interfaces that belong to the group. In this field, active interfaces are listed first, then inactive interfaces, and finally unusable interfaces.The status of the interface is indicated by the manner in which it is listed:

- *interface* (without parentheses or brackets) indicates an active interface. Active interfaces are those interfaces that being used by the system to send or receive data traffic.
- *(interface)* (with parentheses) indicates a functioning but inactive interface. The interface is not in use as defined by administrative policy.
- *[interface]* (with brackets) indicates that the interface is unusable because it has either failed or been taken offline.

How to Obtain IPMP Data Address Information

Use this procedure to display data addresses and the group to which each address belongs. The displayed information also includes which address is available for use, depending on whether the address has been toggled by the `ifconfig [up/down]` command. You can also determine on which inbound or outbound interface an address can be used.

- **Display the IPMP address information.**

```
$ ipmpstat -an
ADDRESS         STATE    GROUP      INBOUND     OUTBOUND
192.168.10.10   up       itops0     subitops0   subitops0 subitops1
192.168.10.15   up       itops0     subitops1   subitops0 subitops1
192.0.0.100     up       acctg1     --          --
192.0.0.101     up       acctg1     --          --
128.0.0.100     up       field2     fops0       fops0 fops3
128.0.0.101     up       field2     fops3       fops0 fops3
128.0.0.102     down     field2     --          --
```

`ADDRESS` Specifies the hostname or the data address, if the `-n` option is used in conjunction with the `-a` option.

`STATE` Indicates whether the address on the IPMP interface is `up`, and therefore usable, or `down`, and therefore unusable.

`GROUP` Specifies the IPMP IP interface that hosts a specific data address.

`INBOUND` Identifies the interface that receives packets for a given address. The field information might change depending on external events. For example, if a data address is down, or if no active IP interfaces remain in the IPMP group, this field will be empty. The empty field indicates that the system is not accepting IP packets that are destined for the given address.

`OUTBOUND` Identifies the interface that sends packets that are using a given address as a source address. As with the `INBOUND` field, the `OUTBOUND` field information might also change depending on external events. An empty field indicates that the system is not sending out packets with the given source address. The field might be empty either because the address is down, or because no active IP interfaces remain in the group.

How to Obtain Information About Underlying IP Interfaces of a Group

Use this procedure to display information about an IPMP group's underlying IP interfaces. For a description of the corresponding relationship between the NIC, data link, and IP interface, see "Overview of the Networking Stack" on page 13.

- **Display the IPMP interface information.**

```
$ ipmpstat -i
INTERFACE   ACTIVE   GROUP      FLAGS      LINK       PROBE      STATE
subitops0   yes      itops0     --mb---    up         ok         ok
subitops1   yes      itops0     -------    up         disabled   ok
hme0        no       acctg1     -------    unknown    disabled   offline
hme1        no       acctg1     is-----    down       unknown    failed
fops0       yes      field2     --mb---    unknown    ok         ok
fops1       no       field2     -i-----    up         ok         ok
fops2       no       filed2     -------    up         failed     failed
fops3       yes      field2     --mb---    up         ok         ok
```

`INTERFACE` Specifies each underlying interface of each IPMP group.

`ACTIVE` Indicates whether the interface is functioning and is in use (`yes`) or not (`no`).

`GROUP` Specifies the IPMP interface name. In the case of anonymous groups, this field will be empty. For more information about anonymous groups, see the `in.mpathd(1M)` man page.

`FLAGS` Indicates the status of the underlying interface, which can be one or any combination of the following:

- `i` indicates that the `INACTIVE` flag is set for the interface and therefore the interface is not used to send or receive data traffic.
- `s` indicates that the interface is configured to be a standby interface.
- `m` indicates that the interface is designated by the system to send and receive IPv4 multicast traffic for the IPMP group.
- `b` indicates that the interface is designated by the system to receive broadcast traffic for the IPMP group.
- `M` indicates that the interface is designated by the system to send and receive IPv6 multicast traffic for the IPMP group.
- `d` indicates that the interface is down and therefore unusable.
- `h` indicates that the interface shares a duplicate physical hardware address with another interface and has been taken offline. The `h` flag indicates that the interface is unusable.

`LINK` Indicates the state of link-based failure detection, which is one of the following states:

- `up` or `down` indicates the availability or unavailability of a link.
- `unknown` indicates that the driver does not support notification of whether a link is `up` or `down` and therefore does not detect link state changes.

`PROBE` Specifies the state of the probe–based failure detection for interfaces that have been configured with a test address, as follows:

- `ok` indicates that the probe is functional and active.
- `failed` indicates that probe-based failure detection has detected that the interface is not working.
- `unknown` indicates that no suitable probe targets could be found, and therefore probes cannot be sent.
- `disabled` indicates that no IPMP test address is configured on the interface. Therefore probe-based failure detection is disabled.

`STATE` Specifies the overall state of the interface, as follows:

- `ok` indicates that the interface is online and working normally based on the configuration of failure detection methods.
- `failed` indicates that the interface is not working because either the interface's link is down, or the probe detection has determined that the interface cannot send or receive traffic.
- `offline` indicates that the interface is not available for use. Typically, the interface is switched offline under the following circumstances:
  - The interface is being tested.
  - Dynamic reconfiguration is being performed.
  - The interface shares a duplicate hardware address with another interface.
- `unknown` indicates the IPMP interface's status cannot be determined because no probe targets can be found for probe-based failure detection.

▼ How to Obtain IPMP Probe Target Information

Use this procedure to monitor the probe targets that are associated with each IP interface in an IPMP group.

- **Display the IPMP probe targets.**

```
$ ipmpstat -nt
INTERFACE   MODE          TESTADDR        TARGETS
subitops0   routes        192.168.85.30   192.168.85.1 192.168.85.3
subitops1   disabled      --              --
hme0        disabled      --              --
hme1        routes        192.1.2.200     192.1.2.1
fops0       multicast     128.9.0.200     128.0.0.1 128.0.0.2
fops1       multicast     128.9.0.201     128.0.0.2 128.0.0.1
fops2       multicast     128.9.0.202     128.0.0.1 128.0.0.2
fops3       multicast     128.9.0.203     128.0.0.1 128.0.0.2
```

`INTERFACE` Specifies the underlying interfaces of the IPMP group.

`MODE` Specifies the method for obtaining the probe targets.

- `routes` indicates that the system routing table is used to find probe targets.
- `mcast` indicates that multicast ICMP probes are used to find targets.
- `disabled` indicates that probe-based failure detection has been disabled for the interface.

`TESTADDR` Specifies the hostname or, if the `-n` option is used in conjunction with the `-t` option, the IP address that is assigned to the interface to send and receive probes. This field will be empty if a test address has not been configured.

Note – If an IP interface is configured with both IPv4 and IPv6 test addresses, the probe target information is displayed separately for each test address.

`TARGETS` Lists the current probe targets in a space-separated list. The probe targets are displayed either as hostnames or IP addresses, if the `-n` is used in conjunction with the `-t` option.

How to Observe IPMP Probes

Use this procedure to observe ongoing probes. When you issue the command to observe probes, information about probe activity on the system is continuously displayed until you terminate the command with `Ctrl-C`. You must have Primary Administrator privileges to run this command.

1 Assume the role of Primary Administrator, or become superuser.

The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2 Display the information about ongoing probes.

```
# ipmpstat -pn
TIME    INTERFACE   PROBE   TARGET        NETRTT   RTT      RTTAVG     RTTDEV
0.11s   subitops0   589     192.168.85.1  0.51ms   0.76ms   0.76ms     --
0.17s   hme1        612     192.1.2.1     --       --       --         --
0.25s   fops0       602     128.0.0.1     0.61ms   1.10ms   1.10ms     --
0.26s   fops1       602     128.0.0.2     --       --       --         --
0.25s   fops2       601     128.0.0.1     0.62ms   1.20ms   1.00ms     --
0.26s   fops3       603     128.0.0.1     0.79ms   1.11ms   1.10ms     --
1.66s   hme1        613     192.1.2.1     --       --       --         --
1.70s   subitops0   603     192.168.85.3  0.63ms   1.10ms   1.10ms     --
^C
```

TIME	Specifies the time a probe was sent relative to when the ipmpstat command was issued. If a probe was initiated prior to ipmpstat being started, then the time is displayed with a negative value, relative to when the command was issued.
PROBE	Specifies the identifier that represents the probe.
INTERFACE	Specifies the interface on which the probe is sent.
TARGET	Specifies the hostname or, if the -n option is used in conjunction with -p, the target address to which the probe is sent.
NETRTT	Specifies the total network round-trip time of the probe and is measured in milliseconds. NETRTT covers the time between the moment when the IP module sends the probe and the moment the IP module receives the ack packets from the target. If the in.mpathd daemon has determined that the probe is lost, then the field will be empty.
RTT	Specifies the total round-trip time for the probe and is measured in milliseconds. RTT covers the time between the moment the daemon executes the code to send the probe and the moment the daemon completes processing the ack packets from the target. If the in.mpathd daemon has determined that the probe is lost, then the field will be empty. Spikes that occur in the RTT which are not present in the NETRTT might indicate that the local system is overloaded.
RTTAVG	Specifies the probe's average round-trip time over the interface between local system and target. The average round-trip time helps identify slow targets. If data is insufficient to calculate the average, this field will be empty.
RTTDEV	Specifies the standard deviation for the round-trip time to the target over the interface. The standard deviation helps identify jittery targets whose ack packets are being sent erratically. For jittery targets, the in.mpathd daemon is forced to increase the failure detection time. Consequently, the daemon would take a longer time before it can detect such a target's outage. If data is insufficient to calculate the standard deviation, this field will be empty.

▼ How to Customize the Output of the ipmpstat Command in a Script

When you use the ipmpstat, by default, the most meaningful fields that fit in 80 columns are displayed. In the output, all the fields that are specific to the option that you use with the ipmpstat command are displayed, except in the case of the ipmpstat -p syntax. If you want to specify the fields to be displayed, then you use the -o option in conjunction with other options that determine the output mode of the command. This option is particularly useful when you issue the command from a script or by using a command alias

- **To customize the output, issue one of the following commands:**
  - To display selected fields of the `ipmpstat` command, use the `-o` option in combination with the specific output option. For example, to display only the `GROUPNAME` and the `STATE` fields of the group output mode, you would type the following:

    ```
    $ ipmpstat -g -o groupname,state

    GROUPNAME  STATE
    itops0     ok
    accgt1     failed
    field2     degraded
    ```

  - To display all the fields of a given `ipmpstat` command, use the following syntax:

    ```
    # ipmpstat -o all
    ```

▼ How to Generate Machine Parseable Output of the `ipmpstat` Command

You can generate machine parseable information by using the `ipmpstat -P` syntax. The `-P` option is intended to be used particularly in scripts. Machine-parseable output differs from the normal output in the following ways:

- Headers are omitted.
- Fields are separated by colons (`:`).
- Fields with empty values are empty rather than being filled with the double dash (`--`).
- In the case of multiple fields being requested, if a field contains a literal colon (`:`) or back slash (`\`), these can be escaped or excluded by prefixing these characters with a back slash (`\`) .

To correctly use the `ipmpstat -P` syntax, observe the following rules:

- Use the `-o` *option fields* together with the `-P` option.
- Never use `-o all` with the `-P` option.

Ignoring either one of these rules will cause `ipmpstat -P` to fail.

- **To display in machine parseable format the group name, the failure detection time, and the underlying interfaces, you would type the following:**

```
$ ipmpstat -P -o -g groupname,fdt,interfaces
itops0:10.00s:subitops0 subitops1
acctg1::[hme0 hme1]
field2:20.00s:fops0 fops3 (fops2) [fops1]
```

The group name, failure detection time, and underlying interfaces are group information fields. Thus, you use the `-o -g` options together with the `-P` option.

Example 8–8 Using `ipmpstat -P` in a Script

This sample script displays the failure detection time of a particular IPMP group.

```
getfdt() {
         ipmpstat -gP -o group,fdt | while IFS=: read group fdt; do
             [[ "$group" = "$1" ]] && { echo "$fdt"; return; }
         done
     }
```