

## Novell IPX Commands

---

Novell Internet Packet Exchange (IPX) is derived from the Xerox Network Systems (XNS) Internet Datagram Protocol (IDP). One major difference between IPX and XNS is that they do not always use the same Ethernet encapsulation format. A second difference is that IPX uses Novell's proprietary Service Advertisement Protocol (SAP) to advertise special network services.

Our implementation of Novell's IPX protocol has been certified as providing full IPX communication server functionality. Our communication server connects Ethernet, Token Ring, and FDDI networks, either directly or through high-speed serial lines (56 kbps to T1 speeds), X.25, or Frame Relay. The Cisco X.25 and T1 support currently is not compatible with Novell. This means that our communication servers must be used on both ends of T1 and X.25 circuits.

Use the commands in this chapter to configure and monitor Novell IPX networks. For IPX configuration information and examples, refer to the "Configuring Novell IPX" chapter in the *Access and Communication Servers Configuration Guide*.

---

**Note** For all commands that previously had the keyword **novell**, this keyword has been changed to **ipx**. However, you can still use the keyword **novell** in all commands.

---

## access-list (standard)

To define a standard IPX access list, use the standard version of the **access-list** global configuration command. To remove a standard access list, use the **no** form of this command.

```
access-list access-list-number {deny | permit} source-network [source-node
[source-node-mask]] [destination-network] [destination-node [destination-node-mask]]
no access-list access-list-number {deny | permit} source-network [source-node
[source-node-mask]] [destination-network] [destination-node [destination-node-mask]]
```

### Syntax Description

<i>access-list-number</i>	Number of the access list. This is a decimal number from 800 to 899.
<b>deny</b>	Denies access if the conditions are matched.
<b>permit</b>	Permits access if the conditions are matched.
<i>source-network</i>	Number of the network from which the packet is being sent. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. A network number of 0 matches the local network. A network number of -1 matches all networks. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA.
<i>source-node</i>	(Optional) Node on <i>source-network</i> from which the packet is being sent. This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers ( <i>xxxx.xxx.xxx</i> ).
<i>source-node-mask</i>	(Optional) Mask to be applied to <i>source-node</i> . This is a 48-bit value represented as a dotted triplet of four-digit hexadecimal numbers ( <i>xxxx.xxx.xxx</i> ). Place ones in the bit positions you want to mask.
<i>destination-network</i>	(Optional) Number of the network to which the packet is being sent. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. A network number of 0 matches the local network. A network number of -1 matches all networks. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA.
<i>destination-node</i>	(Optional) Node on <i>destination-network</i> to which the packet is being sent. This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers ( <i>xxxx.xxx.xxx</i> ).
<i>destination-node-mask</i>	(Optional) Mask to be applied to <i>destination-node</i> . This is a 48-bit value represented as a dotted triplet of four-digit hexadecimal numbers ( <i>xxxx.xxx.xxx</i> ). Place ones in the bit positions you want to mask.

### Default

No access lists are predefined.

## Command Mode

Global configuration

## Usage Guidelines

Standard IPX access lists filter on the source network. All other parameters are optional.

Use the **ipx access-group** command to assign an access list to an interface. You can apply only one extended or one standard access list to an interface. The access list filters all outgoing packets on the interface.

To delete a standard access list, specify the minimum number of keywords and arguments needed to delete the proper access list. For example, to delete the entire access list, use the following command:

```
no access-list access-list-number
```

To delete the access list for a specific network, use the following command:

```
no access-list access-list-number {deny | permit} source-network
```

## Examples

The following example denies access to traffic from all IPX networks (-1) to destination network 2:

```
access-list 800 deny -1 2
```

The following example denies access to all traffic from IPX address 1.0000.0c00.1111:

```
access-list 800 deny 1.0000.0c00.1111
```

The following example denies access from all nodes on network 1 that have a source address beginning with 0000.0c:

```
access-list 800 deny 1.0000.0c00.0000 0000.00ff.ffff
```

The following example denies access from source address 1111.1111.1111 on network 1 to destination address 2222.2222.2222 on network 2:

```
access-list 800 deny 1.1111.1111.1111 0000.0000.0000 2.2222.2222.2222 0000.0000.0000
```

or

```
access-list 800 deny 1.1111.1111.1111 2.2222.2222.2222
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**access-list** (extended)

**ipx access-group**

**ipx input-network-filter**

**ipx output-network-filter**

**ipx router-filter**

**priority-list protocol** †

## access-list (extended)

To define an extended Novell IPX access list, use the extended version of the **access-list** global configuration command. To remove an extended access list, use the **no** form of this command.

```
access-list access-list-number { deny | permit } protocol [source-network [source-node
source-network-mask].source-node-mask] source-socket
[destination-network [destination-node
destination-network-mask.destination-node-mask] destination-socket]]
no access-list access-list-number { deny | permit } protocol [source-network [source-node
source-network-mask].source-node-mask] source-socket
[destination-network [destination-node
destination-network-mask.destination-node-mask] destination-socket]]
access-list permit [source-network.source-node source-network-mask.source-node-mask
source-socket] [destination-network.destination-node destination-network-mask
destination-node-mask destination-socket]
no access-list permit[source.network][[source-node] source-node-mask]] [source-socket]
[destination.network][[destination-node] destination-node-mask]] [destination-socket]
```

### Syntax Description

<i>access-list-number</i>	Number of the access list. This is a decimal number from 900 to 999.
<b>deny</b>	Denies access if the conditions are matched.
<b>permit</b>	Permits access if the conditions are matched.
<i>protocol</i>	Number of an IPX protocol type, in decimal. This also is sometimes referred to as the packet type. Table 19-1 in the “Usage Guidelines” section lists some IPX protocol numbers.
<i>source-network</i>	(Optional) Number of the network from which the packet is being sent. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. A network number of 0 matches the local network. A network number of -1 matches all networks. You do not need to specify leading zeros in the network number, for example, for the network number 000000AA, you can enter just AA.
<i>source-node</i>	(Optional) Node on <i>source-network</i> from which the packet is being sent. This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers ( <i>xxxx.xxxx.xxxx</i> ).
<i>source-network-mask</i>	(Optional) Mask to be applied to <i>source-network</i> . This is an eight-digit hexadecimal mask. Place ones in the bit positions you want to mask. The mask must immediately be followed by a period, which must in turn immediately be followed by <i>source-node-mask</i> .

<i>source-node-mask</i>	(Optional) Mask to be applied to <i>source-node</i> . This is a 48-bit value represented as a dotted triplet of four-digit hexadecimal numbers ( <i>xxxx.xxxx.xxxx</i> ). Place ones in the bit positions you want to mask.
<i>source-socket</i>	Socket number from which the packet is being sent, in hexadecimal. Table 19-2 in the “Usage Guidelines” section lists some IPX socket numbers.
<i>destination-network</i>	(Optional) Number of the network to which the packet is being sent. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. A network number of 0 matches the local network. A network number of -1 matches all networks. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA.
<i>destination-node</i>	(Optional) Node on <i>destination-network</i> to which the packet is being sent. This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers ( <i>xxxx.xxxx.xxxx</i> ).
<i>destination-network-mask</i>	(Optional) Mask to be applied to <i>destination-network</i> . This is an eight-digit hexadecimal mask. Place ones in the bit positions you want to mask. The mask must immediately be followed by a period, which must in turn immediately be followed by <i>destination-node-mask</i> .
<i>destination-node-mask</i>	(Optional) Mask to be applied to <i>destination-node</i> . This is a 48-bit value represented as a dotted triplet of four-digit hexadecimal numbers ( <i>xxxx.xxxx.xxxx</i> ). Place ones in the bit positions you want to mask.
<i>destination-socket</i>	(Optional) Socket number to which the packet is being sent, in hexadecimal. Table 19-2 in the “Usage Guidelines” section lists some IPX socket numbers.

**Default**

No access lists are predefined.

**Command Mode**

Global configuration

**Usage Guidelines**

Extended IPX access lists filter on protocol type. All other parameters are optional.

If a network mask is used, all other fields are required.

Use the **ipx access-group** command to assign an access list to an interface. You can apply only one extended or one standard access list to an interface. The access list filters all outgoing packets on the interface.

---

**Note** For some versions of NetWare, the protocol type field is not a reliable indicator of the type of packet encapsulated by the IPX header. In these cases, use the source and destination socket fields to make this determination. For additional information, contact Novell.

---

Table 19-1 lists some IPX protocol numbers. Table 19-2 lists some IPX socket numbers. For additional information about IPX protocol numbers and socket numbers, contact Novell.

**Table 19-1 IPX Protocol Numbers**

IPX Protocol Number (Decimal)	Protocol (Packet Type)
-1	Wildcard; matches any packet type
0	Could be any protocol; refer to the socket number to determine the packet type
1	Routing Information Protocol (RIP)
4	Service Advertisement Protocol (SAP)
5	Sequenced Packet Exchange (SPX)
17	NetWare Core Protocol (NCP)
20	IPX NetBIOS

**Table 19-2 IPX Socket Numbers**

IPX Socket Number (Hexadecimal)	Socket
0	All sockets, wild card used to match all sockets
451	NetWare Core Protocol (NCP) process
452	Service Advertisement Protocol (SAP) process
453	Routing Information Protocol (RIP) process
455	Novell NetBIOS process
456	Novell diagnostic packet
457	Novell serialization socket
4000-7FFF	Dynamic sockets; used by workstations for interaction with file servers and other network servers
8000-FFFF	Well-known sockets as assigned by Novell

To delete an extended access list, specify the minimum number of keywords and arguments needed to delete the proper access list. For example, to delete the entire access list, use the following command:

**no access-list** *access-list-number*

To delete the access list for a specific protocol, use the following command:

**no access-list** *access-list-number* {**deny** | **permit**} *protocol*

## Examples

The following example denies access to all RIP packets (protocol number 1) from socket 453 (RIP process socket) on source network 1 that are destined for socket 453 on network 2. It permits all other traffic.

```
access-list 900 deny 1 453 2 453
access-list 900 permit 0 -1 0 -1 0
```

The following example permits type 2 packets from any socket on network 10 to access any sockets on any nodes on networks 1000 through 100F. It denies all other traffic (with an implicit deny all):

---

**Note** This type is chosen only as an example. The actual type to use depends on the specific application.

---

```
access-list 910 permit 20 10.0000.0C00.0000 0000.0000.FFFF 0
1000.0000.0000.0000 F.FFFF.FFFF.FFFF 0
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**access-list** (standard) †

**ipx access-group**

**ipx input-network-filter**

**ipx output-network-filter**

**ipx router-filter**

**priority-list protocol** †

## access-list (SAP filtering)

To define an access list for filtering Service Advertisement Protocol (SAP) requests, use the SAP filtering form of the **access-list** global configuration command. To remove the access list, use the **no** form of this command.

```
access-list access-list-number { deny | permit } network [.node] [network.node-mask]
[service-type [server-name]]
no access-list access-list-number { deny | permit } network [.node] [network.node-mask]
[service-type [server-name]]
```

### Syntax Description

<i>access-list-number</i>	Number of the SAP access list. This is a decimal number from 1000 to 1099.
<b>deny</b>	Denies access if the conditions are matched.
<b>permit</b>	Permits access if the conditions are matched.
<i>network</i>	Network number. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. A network number of 0 matches the local network. A network number of -1 matches all networks. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA.
<i>node</i>	(Optional) Node on <i>network</i> . This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers ( <i>xxx.xxxx.xxx</i> ).
<i>network.node-mask</i>	(Optional) Mask to be applied to <i>network</i> and <i>node</i> . Place ones in the bit positions to be masked.
<i>service-type</i>	(Optional) Service type on which to filter. This is a hexadecimal number. A value of 0 means all services. Table 19-3 in the “Usage Guidelines” section lists examples of service types.
<i>server-name</i>	(Optional) Name of the server providing the specified service type. This can be any contiguous string of printable ASCII characters. Use double quotation marks (“ ”) to enclose strings containing embedded spaces. You can use an asterisk (*) at the end of the name as a wildcard to match one or more trailing characters.

### Default

No access lists are predefined.



## Command Mode

Global configuration

## Usage Guidelines

When configuring SAP filters for NetWare 3.11 and later servers, use the server's internal network and node number (the node number is always 0000.0000.0001) as its address in the **access-list** command. Do not use the *network.node* address of the particular interface board.

Table 19-3 lists some sample IPX SAP types. For more information about SAP types, contact Novell.

**Table 19-3 Sample IPX SAP Services**

Service Type (Hexadecimal)	Description
0	All SAP services; IPX defines server type 0 to be an unknown service, which means that you cannot define an access list to permit or deny unknown services
1	User
2	User group
3	Print server queue
4	File server
5	Job server
7	Print server
9	Archive server
A	Queue for job servers
21	NAS SNA gateway
2D	Time Synchronization VAP
2E	Dynamic SAP
47	Advertising print server
4B	Btrieve VAP 5.0
4C	SQL VAP
7A	TES—NetWare for VMS
98	NetWare access server
9A	Named Pipes server
9E	Portable NetWare—UNIX
111	Test server
166	NetWare management (Novell's Network Management Station [NMS])
26A	NetWare management (NMS console)
FFFF	Wildcard (any SAP service)

To delete a SAP access list, specify the minimum number of keywords and arguments needed to delete the proper access list. For example, to delete the entire access list, use the following command:

```
no access-list access-list-number
```

To delete the access list for a specific network, use the following command:

```
no access-list access-list-number {deny | permit} network
```

### Example

The following access list blocks all access to a file server (service type 4) on the directly attached network by resources on other Novell networks, but allows access to all other available services on the interface:

```
access-list 1001 deny -1 4  
access-list 1001 permit -1
```

### Related Commands

A dagger (†) indicates that the command is documented in another chapter.

```
ipx input-sap-filter  
ipx output-gns-filter  
ipx output-sap-filter  
ipx router-sap-filter  
priority-list protocol †
```

## clear ipx accounting

To delete all entries in the active accounting database when IPX accounting is enabled, use the **clear ipx accounting** EXEC command.

```
clear ipx accounting [checkpoint]
```

### Syntax Description

**checkpoint** (Optional) Clears the checkpointed database.

### Command Mode

EXEC

### Usage Guidelines

If you omit the **checkpoint** argument, all the entries in the active database are deleted. You can also delete all entries in the checkpointed database by issuing the **clear ipx accounting** command twice in succession.

### Example

The following example clears all entries in the active database:

```
clear ipx accounting
```

### Related Commands

**ipx accounting**  
**ipx accounting-list**  
**ipx accounting-threshold**  
**ipx accounting-transits**  
**show ipx accounting**

## clear ipx cache

To delete entries from the IPX fast-switching cache, use the **clear ipx cache** EXEC command.

**clear ipx cache**

### Syntax Description

This command has no arguments or keywords.

### Command Mode

EXEC

### Usage Guidelines

The **clear ipx cache** command clears entries used for fast switching.

### Example

The following example deletes all entries from the IPX fast-switching cache:

```
clear ipx cache
```

### Related Commands

**ipx route-cache**

**show ipx cache**

## clear ipx route

To delete routes from the IPX routing table, use the **clear ipx route** EXEC command.

```
clear ipx route [network | *]
```

### Syntax Description

*network*

(Optional) Number of the network whose routing table entry you want to delete. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA.

\*

(Optional) Deletes all routes in the routing table.

### Command Mode

EXEC

### Example

The following example clears the entry for network 3 from the IPX routing table:

```
clear ipx route 3
```

### Related Command

**show ipx route**

## distribute-list in

To filter networks received in updates, use the **distribute-list in** router configuration command. To change or cancel the filter, use the **no** form of this command.

**distribute-list** *access-list-number* **in** [*type number*]  
**no distribute-list** *access-list-number* **in** [*type number*]

### Syntax Description

<i>access-list-number</i>	Standard IPX access list number in the range 800 to 899. The list explicitly specifies which networks are to be received and which are to be suppressed.
<i>type</i>	(Optional) Interface type.
<i>number</i>	(Optional) Interface number.

### Default

Disabled

### Command Mode

Router configuration

### Example

The following example causes only two networks—network 2 and network 3—to be accepted by an IP Enhanced IGRP routing process:

```
access-list 800 permit 2
access-list 800 permit 3
access-list 800 deny -1
!
ipx router eigrp 100
network 3
distribute-list 800 in
```

### Related Commands

**access-list**  
**distribute-list out**  
**redistribute**

## distribute-list out

To suppress networks from being advertised in updates, use the **distribute-list out** router configuration command. To cancel this function, use the **no** form of this command.

```
distribute-list access-list-number out [interface-name | routing-process]  
no distribute-list access-list-number out [interface-name | routing-process]
```

### Syntax Description

<i>access-list-number</i>	Standard IPX access list number in the range 800 to 899. The list explicitly specifies which networks are to be sent and which are to be suppressed in routing updates.
<i>interface-name</i>	(Optional) Interface on which the access list should be applied to outgoing updates. If no interface is specified, the access list is applied to all outgoing updates.
<i>routing-process</i>	(Optional) Name of a particular routing process ( <b>rip</b> or <b>eigrp</b> autonomous system number).

### Default

Disabled

### Command Mode

Router configuration

### Usage Guidelines

When redistributing networks, a routing process name can be specified as an optional trailing argument to the **distribute-list out** command. This causes the access list to be applied to only those routes derived from the specified routing process. After the process-specific access list is applied, any access list specified by a **distribute-list out** command without a process name argument is applied. Addresses not specified in the **distribute-list out** command are not advertised in outgoing routing updates.

### Example

The following example causes only one network—network 3—to be advertised by an IPX Enhanced IGRP routing process:

```
access-list 800 permit 3  
access-list 800 deny -1  
!  
ipx router eigrp 100  
network 3  
distribute-list 800 out
```

**Related Commands**

- access-list**
- istribute-list in**
- redistribute**



## ipx access-group

To apply a generic output filter to an interface, use **ipx access-group** interface configuration command. To remove the access list, use the **no** form of this command.

```
ipx access-group access-list-number  
no ipx access-group access-list-number
```

### Syntax Description

<i>access-list-number</i>	Number of the access list. All outgoing packets defined with either standard or extended access lists and forwarded through the interface are filtered by the entries in this access list. For standard access lists, <i>access-list-number</i> is a decimal number from 800 to 899. For extended access lists, <i>access-list-number</i> is a decimal number from 900 to 999.
---------------------------	--

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

Generic filters control which packets are sent out an interface based on the packet's source and destination addresses, IPX protocol type, and source and destination socket numbers. You use the standard **access-list** and extended **access-list** commands to specify the filtering conditions.

You can apply only one generic filter to an interface.

### Example

In the following example, access list 801 is applied to Ethernet interface 1:

```
interface ethernet 1  
  ipx access-group 801
```

### Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**access-list** (standard)  
**access-list** (extended)  
**priority-list protocol** †

## ipx accounting

To enable IPX accounting, use the **ipx accounting** interface configuration command. To disable IPX accounting, use the **no** form of this command.

**ipx accounting**  
**no ipx accounting**

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

IPX accounting allows you to collect information about IPX packets and the number of bytes that are switched through the communication server. You collect information based on the source and destination IPX address. Accounting tracks only IPX traffic that is passing out of the communication server; it does not track traffic generated by or terminating at the communication server.

IPX accounting statistics will be accurate even if IPX fast switching is enabled or if IPX access lists are being used. However, IPX accounting does not keep statistics if autonomous switching is enabled.

The communication server software maintains two accounting databases: an active database and a checkpointed database.

### Example

The following example enables IPX accounting on Ethernet interface 0:

```
interface ethernet 0
 ipx accounting
```

### Related Commands

**clear ipx accounting**  
**ipx accounting-list**  
**ipx accounting-threshold**  
**ipx accounting-transits**  
**show ipx accounting**

## ipx accounting-list

To filter the networks for which IPX accounting information is kept, use the **ipx accounting-list** global configuration command. To remove the filter, use the **no** form of this command.

```
ipx accounting-list network mask  
no ipx accounting-list network mask
```

### Syntax Description

<i>network</i>	Network number. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA you can enter just AA.
<i>mask</i>	Network mask.

### Default

No filters are predefined.

### Command Mode

Global configuration

### Usage Guidelines

The source and destination addresses of each IPX packet are logically ANDed with the mask and compared with the network number. If there is a match, accounting information about the IPX packet is entered into the accounting database. If there is no match, the IPX packet is considered to be a transit packet and may be counted, depending on the setting of the **ipx accounting-transits** global configuration command.

### Example

The following example adds all networks with IPX network numbers beginning with 1 to the list of networks for which accounting information is kept:

```
ipx accounting-list 1 0000.0000.0000
```

### Related Commands

```
clear ipx accounting  
ipx accounting  
ipx accounting-threshold  
ipx accounting-transits  
show ipx accounting
```

## ipx accounting-threshold

To set the maximum number of accounting database entries, use the **ipx accounting-threshold** global configuration command. To restore the default, use the **no** form of this command.

```
ipx accounting-threshold threshold  
no ipx accounting-threshold threshold
```

### Syntax Description

*threshold* Maximum number of entries (source and destination address pairs) that the communication server can accumulate

### Default

512 entries

### Command Mode

Global configuration

### Usage Guidelines

The accounting threshold defines the maximum number of entries (source and destination address pairs) that the communication server accumulates. The threshold is designed to prevent IPX accounting from consuming all available free memory. This level of memory consumption could occur in a communication server that is switching traffic for many hosts. To determine whether overflows have occurred, used the **show ipx accounting EXEC** command.

### Example

The following example sets the IPX accounting database threshold to 500 entries:

```
ipx accounting-threshold 500
```

### Related Commands

```
clear ipx accounting  
ipx accounting  
ipx accounting-list  
ipx accounting-transits  
show ipx accounting
```

## ipx accounting-transits

To set the maximum number of transit entries that will be stored in the IPX accounting database, use the **ipx accounting-transits** global configuration command. To disable this function, use the **no** form of this command.

```
ipx accounting-transits count  
no ipx accounting-transits
```

### Syntax Description

<i>count</i>	Number of transit entries that will be stored in the IPX accounting database
--------------	--

### Default

0 entries

### Command Mode

Global configuration

### Usage Guidelines

Transit entries are those that do not match any of the filters specified by **ipx accounting-list** global configuration commands. If you have not defined any filters, no transit entries are possible.

To maintain accurate accounting totals, the communication server software maintains two accounting databases: an active database and a checkpointed database.

### Example

The following example specifies a maximum of 100 transit records to be stored in the IPX accounting database:

```
ipx accounting-transits 100
```

### Related Commands

```
clear ipx accounting  
ipx accounting  
ipx accounting-list  
ipx accounting-threshold  
show ipx accounting
```

## ipx backup-server-query-interval

To change the time between successive queries of each IPX Enhanced IGRP neighbor's backup server table, use the **ipx backup-server-query-interval** global configuration command. To restore the default time, use the **no** form of this command.

```
ipx backup-server-query-interval interval  
no ipx backup-server-query-interval
```

### Syntax Description

<i>interval</i>	Minimum time, in seconds, between successive queries of each Enhanced IGRP neighbor's backup server table. The default is 15 seconds.
-----------------	---

### Default

15 seconds

### Command Mode

Global configuration

### Usage Guidelines

A lower interval may use more CPU resources, but may cause lost server information to be retrieved from other servers' tables sooner.

### Example

The following example changes the server query time to 5 seconds:

```
ipx backup-server-query-interval 5
```

## ipx default-output-rip-delay

To set the default interpacket delay for Routing Information Protocol (RIP) updates sent on all interfaces, use the **ipx default-output-rip-delay** global configuration command. To return to the initial default delay value, use the **no** form of this command.

```
ipx default-output-rip-delay delay  
no ipx default-output-rip-delay [delay]
```

### Syntax Description

*delay* Delay, in milliseconds, between packets in a multiple-packet RIP update. With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms. Novell recommends a delay of 55 ms.

### Default

With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms (that is, no additional delay between routing update packets).

### Command Mode

Global configuration

### Usage Guidelines

The interpacket delay is the delay between the individual packets sent in a multiple-packet routing update. The **ipx default-output-rip-delay** command sets a default interpacket delay for all interfaces.

The system uses the delay specified by the **ipx default-output-rip-delay** command for periodic and triggered routing updates when no delay is set for periodic and triggered routing updates on an interface. When you set a delay for triggered routing updates, the system uses the delay specified by the **ipx default-output-rip-delay** command for only the periodic routing updates sent on all interfaces.

To set a delay for triggered routing updates, see the **ipx triggered-rip-delay** or **ipx default-triggered-rip-delay** commands.

Novell recommends a delay of 55 ms for compatibility with older and slower IPX machines. These machines may lose RIP updates because they process packets more slowly than the router sends them. The delay imposed by this command forces the router to pace its output to the slower-processing needs of these IPX machines.

The default delay on a NetWare 3.11 server is about 100 ms.

This command is also useful on limited bandwidth point-to-point links or X.25 and Frame Relay multipoint interfaces.

### Example

The following example sets a default interpacket delay of 55 ms for RIP updates sent on all interfaces:

```
ipx default-output-rip-delay 55
```

**Related Command**

`ipx default-triggered-rip-delay`

`ipx output-rip-delay`

`ipx triggered-rip-delay`

|



## ipx default-output-sap-delay

To set a default interpacket delay for Service Advertisement Protocol (SAP) updates sent on all interfaces, use the **ipx default-output-sap-delay** global configuration command. To return to the initial default delay value, use the **no** form of this command.

```
ipx default-output-sap-delay delay  
no ipx default-output-sap-delay [delay]
```

### Syntax Description

*delay* Delay, in milliseconds, between packets in a multiple-packet SAP update. With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms. Novell recommends a delay of 55 ms.

### Default

With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms (that is, no additional delay between update packets).

### Command Mode

Global configuration

### Usage Guidelines

The interpacket delay is the delay between the individual packets sent in a multiple-packet SAP update. The **ipx default-output-sap-delay** command sets a default interpacket delay for all interfaces.

The system uses the delay specified by the **ipx default-output-sap-delay** command for periodic and triggered SAP updates when no delay is set for periodic and triggered updates on an interface. When you set a delay for triggered updates, the system uses the delay specified by the **ipx default-output-sap-delay** command only for the periodic SAP updates sent on all interfaces.

To set a delay for triggered updates, see the **ipx triggered-sap-delay** or **ipx default-triggered-sap-delay** commands.

Novell recommends a delay of 55 ms for compatibility with older and slower IPX servers. These servers may lose SAP updates because they process packets more slowly than the router sends them. The delay imposed by this command forces the router to pace its output to the slower-processing needs of these servers.

The default delay on a NetWare 3.11 server is about 100 ms.

This command is also useful on limited bandwidth point-to-point links or X.25 interfaces.

### Example

The following example sets a default interpacket delay of 55 ms for SAP updates sent on all interfaces:

```
ipx default-output-sap-delay 55
```

**Related Command**

**ipx default-triggered-sap-delay**

**ipx output-sap-delay**

**ipx triggered-sap-delay**

|

## ipx default-triggered-rip-delay

To set the default interpacket delay for triggered Routing Information Protocol (RIP) updates sent on all interfaces, use the **ipx default-triggered-rip-delay** global configuration command. To return to the system default delay, use the **no** form of this command.

```
ipx default-triggered-rip-delay delay  
no ipx default-triggered-rip-delay [delay]
```

### Syntax Description

*delay* Delay, in milliseconds, between packets in a multiple-packet RIP update. With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms. Novell recommends a delay of 55 ms.

### Default

With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms (that is, no additional delay between routing update packets).

### Command Mode

Global configuration

### Usage Guidelines

The interpacket delay is the delay between the individual packets sent in a multiple-packet routing update. A triggered routing update is one that the system sends in response to a “trigger” event, such as a request packet, interface up/down, route up/down, or server up/down.

The **ipx default-triggered-rip-delay** command sets the default interpacket delay for triggered routing updates sent on all interfaces. On a single interface, you can override this global default delay for triggered routing updates using the **ipx triggered-rip-delay** interface command.

The global default delay for triggered routing updates overrides the delay value set by the **ipx output-rip-delay** or **ipx default-output-rip-delay** command for triggered routing updates.

If the delay value set by the **ipx output-rip-delay** or **ipx default-output-rip-delay** command is high, then we strongly recommend a low delay value for triggered routing updates so that updates triggered by special events are sent in a more timely manner than periodic routing updates.

Novell recommends a delay of 55 ms for compatibility with older and slower IPX machines. These machines may lose RIP updates because they process packets more slowly than the router sends them. The delay imposed by this command forces the router to pace its output to the slower-processing needs of these IPX machines.

The default delay on a NetWare 3.11 server is about 100 ms.

When you do not set the interpacket delay for triggered routing updates, the system uses the delay specified by the **ipx output-rip-delay** or **ipx default-output-rip-delay** command for both periodic and triggered routing updates.

When you use the **no** form of the **ipx default-triggered-rip-delay** command, the system uses the delay set by the **ipx output-rip-delay** or **ipx default-output-rip-delay** command for triggered RIP updates, if set. Otherwise, the system uses the initial default delay as described in the “Default” section.

This command is also useful on limited bandwidth point-to-point links or X.25 and Frame Relay multipoint interfaces.

**Example**

The following example sets an interpacket delay of 55 ms for triggered routing updates sent on all interfaces:

```
ipx default-triggered-rip-delay 55
```

**Related Command**

**ipx default-output-rip-delay**

**ipx output-rip-delay**

**ipx triggered-rip-delay**

## ipx default-triggered-sap-delay

To set the default interpacket delay for triggered Service Advertisement Protocol (SAP) updates sent on all interfaces, use the **ipx default-triggered-sap-delay** global configuration command. To return to the system default delay, use the **no** form of this command.

```
ipx default-triggered-sap-delay delay
no ipx default-triggered-sap-delay [delay]
```

### Syntax Description

<i>delay</i>	Delay, in milliseconds, between packets in a multiple-packet SAP update. With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms. Novell recommends a delay of 55 ms.
--------------	---

### Default

With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms (that is, no additional delay between update packets).

### Command Mode

Global configuration

### Usage Guidelines

The interpacket delay is the delay between the individual packets sent in a multiple-packet SAP update. A triggered SAP update is one that the system sends in response to a “trigger” event, such as a request packet, interface up/down, route up/down, or server up/down.

The **ipx default-triggered-sap-delay** command sets the default interpacket delay for triggered SAP updates sent on all interfaces. On a single interface, you can override this global default delay for triggered updates using the **ipx triggered-sap-delay** interface command.

The global default delay for triggered updates overrides the delay value set by the **ipx output-sap-delay** or **ipx default-output-sap-delay** command for triggered updates.

If the delay value set by the **ipx output-sap-delay** or **ipx default-output-sap-delay** command is high, then we strongly recommend a low delay value for triggered updates so that updates triggered by special events are sent in a more timely manner than periodic updates.

Novell recommends a delay of 55 ms for compatibility with older and slower IPX servers. These servers may lose SAP updates because they process packets more slowly than the router sends them. The delay imposed by this command forces the router to pace its output to the slower-processing needs of these IPX servers.

The default delay on a NetWare 3.11 server is about 100 ms.

When you do not set the interpacket delay for triggered SAP updates, the system uses the delay specified by the **ipx output-sap-delay** or **ipx default-output-sap-delay** command for both periodic and triggered SAP updates.

When you use the **no** form of the **ipx default-triggered-sap-delay** command, the system uses the delay set by the **ipx output-sap-delay** or **ipx default-output-sap-delay** command for triggered SAP updates, if set. Otherwise, the system uses the initial default delay as described in the “Default” section.

This command is also useful on limited bandwidth point-to-point links or X.25 and Frame Relay multipoint interfaces.

**Example**

The following example sets an interpacket delay of 55 ms for triggered SAP updates sent on all interfaces:

```
ipx default-triggered-sap-delay 55
```

**Related Command**

**ipx default-output-sap-delay**

**ipx output-sap-delay**

**ipx triggered-sap-delay**

## ipx delay

To set the tick count, use the **ipx delay** interface configuration command. To reset the default increment in the delay field, use the **no** form of this command.

```
ipx delay ticks  
no ipx delay
```

### Syntax Description

*ticks* Number of IBM clock ticks of delay to use. One clock tick is 1/18th of a second (approximately 55 milliseconds).

### Default

The default delay is determined from the delay configured on the interface with the **delay** command. It is  $(\text{interface delay} + 333) / 334$ . Therefore, unless you change the delay by a value greater than 334, you will not notice a difference.

### Command Mode

Interface configuration

### Usage Guidelines

The **ipx delay** command sets the count used in the IPX RIP delay field, which is also known as the ticks field.

Leaving the delay at its default value is sufficient for most interfaces.

If you do not specify the **ipx delay** command on an interface and you have not changed the interface delays with the **interface delay** interface configuration command, all LAN interfaces have a delay of 1 and all WAN interfaces have a delay of 6. The preferred method of adjusting delays is to use the **ipx delay** command, not the **interface delay** command. The **show ipx interface EXEC** command display only the delay value configured with the **ipx delay** command.

With IPXWAN, if you change the interface delay with the **interface delay** command, the **ipx delay** command uses that delay when calculating a delay to use. Also, when changing delays with IPXWAN, the changes affect only the link's calculated delay on the side considered to be the master.

### Example

The following example changes the delay for serial interface 0 to 10 ticks:

```
interface serial 0  
ipx delay 10
```

### Related Commands

A dagger (†) indicates that the command is documented in another chapter.

```
delay †  
ipx maximum-paths  
ipx output-network-filter  
ipx output-rip-delay
```

## ipx down

To administratively shut down an IPX network, use the **ipx down** interface configuration command. To restart the network, use the **no** form of this command.

```
ipx down network  
no ipx down
```

### Syntax Description

*network*

Number of the network to shut down. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

The **ipx down** command administratively shuts down the specified network. The network still exists in the configuration, but is not active. When shutting down, the network sends out update packets informing its neighbors that it is shutting down. This allows the neighboring systems to update their routing, SAP, and other tables without having to wait for routes and services learned via this network to time out.

### Example

The following example administratively shuts down network AA on Ethernet interface 0:

```
interface ethernet 0  
ipx down AA
```



## ipx gns-response-delay

To change the delay when responding to Get Nearest Server (GNS) requests, use the **ipx gns-response-delay** global configuration command. To return to the default delay, use the **no** form of this command.

```
ipx gns-response-delay [milliseconds]  
no ipx gns-response-delay
```

### Syntax Description

*milliseconds*

(Optional) Time, in milliseconds, that the communication server waits after receiving a GNS request from an IPX client before responding with a server name to that client. The default time is 0 milliseconds. A value of zero indicates no delay.

### Default

0 (no delay)

### Command Mode

Global configuration

### Usage Guidelines

In certain topologies, you might want to delay the communication server response to client GNS requests so that any local IPX server can respond first. The client typically takes the first response, not the best, and the best response in these end-host server systems is the one from the local server, not our router.

If you are using NetWare 2.x, be sure to set a GNS delay. This version of NetWare has a problem with dual-connected servers in parallel with a router. We recommend setting a delay of 500 milliseconds.

In situations in which servers are always located across routers from their clients, there is no need for a delay to be imposed.

### Example

The following example sets the delay in responding to GNS requests to 500 milliseconds (0.5 second):

```
ipx gns-response-delay 500
```

## ipx gns-round-robin

To rotate using a round-robin selection method through a set of eligible servers when responding to Get Nearest Server (GNS) requests, use the **ipx gns-round-robin** global configuration command. To use the most recently learned server, use the **no** form of this command.

**ipx gns-round-robin**  
**no ipx gns-round-robin**

### Syntax Description

The command has no arguments or keywords.

### Default

The most recently learned, eligible server is used.

### Command Mode

Global configuration

### Usage Guidelines

In the normal server selection process, requests for service are responded to with the most recently learned, closest server. If you enable the round-robin method, the communication server maintains a list of the nearest servers eligible to provide specific services. It uses this list when responding to Get Nearest Server (GNS) requests. Responses to requests are distributed in a round-robin fashion across all active IPX interfaces on the communication server.

Eligible servers are those that satisfy the “nearest” requirement for a given request and that are not filtered either by a SAP filter or by a GNS filter.

### Example

The following example responds to GNS requests using a round-robin selection method from a list of eligible nearest servers:

```
ipx gns-round-robin
```

### Related Commands

**ipx output-gns-filter**  
**ipx output-sap-filter**

## ipx hello-interval eigrp

To configure the interval between IPX Enhanced IGRP hello packets, use the **ipx hello-interval eigrp** interface configuration command. To restore the default interval, use the **no** form of this command.

```
ipx hello-interval eigrp autonomous-system-number seconds  
no ipx hello-interval eigrp autonomous-system-number seconds
```

### Syntax Description

<i>autonomous-system-number</i>	IPX Enhanced IGRP autonomous system number. It can be a decimal integer from 1 to 65535.
<i>seconds</i>	Interval between hello packets, in seconds. The default interval is 5 seconds, which is one-third of the default hold time.

### Default

5 seconds

### Command Mode

Interface configuration

### Usage Guidelines

If the current value for the hold time is less than two times the interval between hello packets, the hold time will be reset.

### Example

The following example changes the hello interval to 10 seconds:

```
interface ethernet 0  
ipx network 10  
ipx hello-interval eigrp 4 10
```

### Related Command

**ipx hold-time eigrp**

## ipx helper-address

To forward broadcast packets (except type 20 propagation packets) to a specified server, use the **ipx helper-address** interface configuration command. To disable this function, use the **no** form of this command.

```
ipx helper-address network.node  
no ipx helper-address network.node
```

### Syntax Description

<i>network</i>	Network on which the target IPX server resides. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. A network number of -1 indicates all-nets flooding. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA you can enter just AA.
<i>node</i>	Node number of the target Novell server. This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers (xxx.xxx.xxx). A node number of FFFF.FFFF.FFFF matches all servers.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

Communication servers normally block all broadcast requests and do not forward them to other network segments. This is done to prevent the degradation of performance over the entire network. The **ipx helper-address** command allows broadcasts to be forwarded to other networks (except type 20 propagation packets). This is useful when a network segment does not have an end-host capable of servicing a particular type of broadcast request. This command lets you forward the broadcasts to a server, network, or networks that can process them. Incoming unrecognized broadcast packets that match the access list created with the **ipx helper-list** command, if it is present, are forwarded.

Note that type 20 propagation packet handling is controlled by a separate mechanism. See the discussion of the **ipx type-20-propagation** command for more information.

You can specify multiple **ipx helper-address** commands on a given interface.

Our communication servers support all-networks flooded broadcasts (sometimes referred to as *all-nets flooding*). These are broadcast messages that are forwarded to all networks. To configure the all-nets flooding, define the IPX helper address for an interface as follows:

```
ipx helper-address -1.FFFF.FFFF.FFFF
```

On systems configured for IPX routing, this helper address is displayed as follows (via the **show ipx interface** command):

```
FFFFFFF.FFFF.FFFF.FFFF
```

Although our communication servers take care to keep broadcast traffic to a minimum, some duplication is unavoidable. When loops exist, all-nets flooding can propagate bursts of excess traffic that will eventually age out when the hop count reaches its limit (16 hops). Use all-nets flooding carefully and only when necessary. Note that you can apply additional restrictions by defining a helper list.

### Example

In the following example, all-nets broadcasts on Ethernet interface 0 (except type 20 propagation packets) are forwarded to IPX server 00b4.23cd.110a on network bb:

```
interface ethernet 0
ipx helper-address bb.00b4.23cd.110a
```

### Related Commands

**ipx helper-list**

**ipx type-20-propagation**

## ipx helper-list

To assign an access list to an interface to control broadcast traffic (including type 20 propagation packets), use the **ipx helper-list** interface configuration command. To remove the access list from an interface, use the **no** form of this command.

```
ipx helper-list access-list-number  
no ipx helper-list access-list-number
```

### Syntax Description

<i>access-list-number</i>	Number of the access list. All outgoing packets defined with either standard or extended access lists are filtered by the entries in this access list. For standard access lists, <i>access-list-number</i> is a decimal number from 800 to 899. For extended access lists, it is a decimal number from 900 to 999.
---------------------------	---

### Default

No access list is preassigned.

### Command Mode

Interface configuration

### Usage Guidelines

The **ipx helper-list** command specifies an access list to use in forwarding broadcast packets. One use of this command is to prevent client nodes from discovering services they should not use.

Because the destination address of a broadcast packet is by definition the broadcast address, this command is useful only for filtering based on the source address of the broadcast packet.

The helper list, if present, is applied to both all-nets broadcast packets and type 20 propagation packets.

The helper list on the input interface is applied to packets before they are output via either the helper address or type 20 propagation packet mechanism.

You should filter IPX broadcasts on dial-on-demand routing (DDR) and other similar interfaces, because IPX sends broadcast messages very regularly.

### Example

The following example assigns access list 900 to Ethernet interface 0 to control broadcast traffic:

```
interface ethernet 0  
ipx helper-list 900
```

### Related Commands

**access-list** (standard)  
**access-list** (extended)  
**ipx helper-address**  
**ipx type-20-propagation**

## ipx hold-time eigrp

To specify the length of time a neighbor should consider Enhanced IGRP hello packets valid, use the **ipx hold-time eigrp** interface configuration command. To restore the default time, use the **no** form of this command.

```
ipx hold-time eigrp autonomous-system-number seconds  
no ipx hold-time eigrp autonomous-system-number seconds
```

### Syntax Description

<i>autonomous-system-number</i>	Enhanced IGRP autonomous system number. It can be a decimal integer from 1 to 65535.
<i>seconds</i>	Hold time, in seconds. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is 15 seconds, which is three times the hello interval.

### Default

15 seconds

### Command Mode

Interface configuration

### Usage Guidelines

If the current value for the hold time is less than two times the interval between hello packets, the hold time will be reset to three times the hello interval.

If a router does not receive a hello packet within the specified hold time, routes through the router are considered available.

Increasing the hold time delays route convergence across the network.

### Example

The following example changes the hold time to 45 seconds:

```
interface ethernet 0  
ipx network 10  
ipx hold-time eigrp 4 45
```

### Related Command

**ipx hello-interval eigrp**

## ipx input-network-filter

To control which networks are added to the communication server's routing table, use the **ipx input-network-filter** interface configuration command. To remove the filter from the interface, use the **no** form of this command.

```
ipx input-network-filter access-list-number  
no ipx input-network-filter access-list-number
```

### Syntax Description

<i>access-list-number</i>	Number of the access list. All incoming packets defined with either standard or extended access lists are filtered by the entries in this access list. For standard access lists, <i>access-list-number</i> is a decimal number from 800 to 899. For extended access lists, it is a decimal number from 900 to 999.
---------------------------	---

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

The **ipx input-network-filter** command controls which networks are added to the routing table based on the networks learned in incoming IPX routing updates (RIP updates) on the interface.

You can issue only one **ipx input-network-filter** command on each interface.

### Examples

In the following example, access list 876 controls which networks are added to the routing table when IPX routing updates are received on Ethernet interface 1. Routing updates for network 1b will be accepted. Routing updates for all other networks are implicitly denied and are not added to the routing table.

```
access-list 876 permit 1b  
interface ethernet 1  
ipx input-network-filter 876
```

The following example is a variation of the preceding that explicitly denies network 1a and explicitly allows updates for all other networks:

```
access-list 876 deny 1a  
access-list 876 permit -1
```

### Related Commands

- access-list** (standard)
- access-list** (extended)
- ipx output-network-filter**
- ipx router-filter**



## ipx input-sap-filter

To control which services are added to the communication server's SAP table, use the **ipx input-sap-filter** interface configuration command. To remove the filter, use the **no** form of this command.

```
ipx input-sap-filter access-list-number  
no ipx input-sap-filter access-list-number
```

### Syntax Description

<i>access-list-number</i>	Number of the SAP access list. All incoming packets are filtered by the entries in this access list. The argument <i>access-list-number</i> is a decimal number from 1000 to 1099.
---------------------------	--

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

The **ipx input-sap-filter** command filters all incoming service advertisements received by the communication server. This is done prior to a communication server's accepting information about a service.

You can issue only one **ipx input-sap-filter** command on each interface.

When configuring SAP filters for NetWare 3.11 and later servers, use the server's internal network and node number (the node number is always 0000.0000.0001) as its address in the SAP **access-list** command. Do not use the *network.node* address of the particular interface board.

### Example

The following example denies service advertisements about the server at address 3c.0800.89a1.1527, but accepts information about all other services on all other networks:

```
access-list 1000 deny 3c.0800.89a1.1527  
access-list 1000 permit -1  
interface ethernet 0  
ipx input-sap-filter 1000
```

### Related Commands

**access-list** (SAP filtering)  
**ipx output-sap-filter**  
**ipx router-sap-filter**

## ipx ipxwan

To configure the IPXWAN protocol on a serial interface, use the **ipx ipxwan** interface configuration command. To disable the IPXWAN protocol, use the **no** form of this command.

```
ipx ipxwan local-node network local-server [retry-interval] [retry-attempts]  
no ipxwan
```

### Syntax Description

<i>local-node</i>	Primary network number of the communication server. This is an IPX network number that is unique across the entire internet. On NetWare 3.x servers, the primary network number is called the internal network number. The communication server with the higher number is determined to be the link Master.
<i>network</i>	The IPX network number to be used for the link if this communication server is the one determined to be the link master. The number is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA.
<i>local-server</i>	Name of the local communication server. It can be up to 47 characters long, and can contain uppercase letters, digits, underscores (_), hyphens (-), and at signs (@). On NetWare 3.x servers, the local server is called the communication server name.
<i>retry-interval</i>	(Optional) Retry interval, in seconds. This interval defines how often the communication server will retry failed negotiation attempts. It can be a value from 1 through 600. The default is 20 seconds.
<i>retry-attempts</i>	(Optional) Maximum number of times the communication server will retry failed negotiation attempts. It can be a value from 1 through 100. The default is 3. The communication server intentionally ignores the IPXWAN retry counter and continues to send TIMER_REQUEST packets until it receives a TIMER_RESPONSE packet.

### Default

IPXWAN is disabled.

When enabled, the command has the following defaults:

*retry-interval*: 20 seconds

*retry-attempts*: 3

### Command Mode

Interface configuration

## Usage Guidelines

IPXWAN is a startup end-to-end options negotiations protocol. When a link comes up, the first IPX packets sent across are IPXWAN packets negotiating the options for the link. When the IPXWAN options have been successfully determined, normal IPX traffic starts. The three options negotiated are the link IPX network number, Ethernet network number, and link delay (ticks) characteristics. The side of the link with the higher local-node number (internal network number) gives the IPX network number and delay to use for the link to the other side. Once IPXWAN finishes, no IPXWAN packets are sent unless link characteristics change or the connection fails. For example, if the IPX delay is changed from the default setting, an IPXWAN restart will be forced.

To enable the IPXWAN protocol on a serial interface, you must not have configured an IPX network number (using the **ipx network** interface configuration command) on that interface.

If you want to control the delay on a link, use the **ipx delay** interface configuration command. If you issue this command when the serial link is already up, the state of the link will be reset and renegotiated.

## Example

The following example enables IPXWAN on serial interface 1 on device CHICAGO-CS. When the link comes up, CHICAGO-CS will be the master because it has a larger internal network number. It will give the IPX number 100 to NYC-CS to use as the network number for the link. The link delay, in ticks, will be determined by the exchange of packets between the two communication servers.

On the local communication server (CHICAGO-CS):

```
interface serial 1
no ipx network
encapsulation ppp
ipx ipxwan 6666 100 CHICAGO-CS
```

On the remote router (NYC-CS):

```
interface serial 0
no ipx network
encapsulation ppp
ipx ipxwan 1000 101 NYC-CS
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**encapsulation ppp** †

**ipx delay**

**ipx network**

**show ipx interfaces**

## ipx maximum-paths

To set the maximum number of equal-cost paths the communication server uses when forwarding packets, use the **ipx maximum-paths** global configuration command. To restore the default value, use the **no** form of this command.

```
ipx maximum-paths paths  
no ipx maximum-paths
```

### Syntax Description

*paths* Maximum number of equal-cost paths which the communication server will use. The argument *paths* can be a value from 1 to 512. The default value is 1.

### Default

1 path

### Command Mode

Global configuration

### Usage Guidelines

The **ipx maximum-paths** command is designed to increase throughput by allowing the communication server to choose among several equal-cost, parallel paths. (Note that when paths have differing costs, the communication server chooses lower-cost routes in preference to higher-cost routes.) IPX does load sharing on a packet-by-packet basis in round-robin fashion, regardless of whether you are using fast switching or process switching. That is, the first packet is sent along the first path, the second packet along the second path, and so on. When the final path is reached, the next packet is sent to the first path, the next to the second path, and so on.

Limiting the number of equal-cost paths can save memory on communication servers with limited memory or very large configurations. Additionally, in networks with a large number of multiple paths and systems with limited ability to cache out-of-sequence packets, performance might suffer when traffic is split between many paths.

### Example

In the following example, the communication server uses up to three parallel paths:

```
ipx maximum-paths 3
```

### Related Commands

```
ipx delay  
show ipx route
```

## ipx netbios input-access-filter

To control incoming IPX NetBIOS messages, use the **ipx netbios input-access-filter** interface configuration command. To remove the filter, use the **no** form of this command.

```
ipx netbios input-access-filter {host | bytes} name
no ipx netbios input-access-filter {host | bytes} name
```

### Syntax Description

<b>host</b>	Indicates that the following argument is the name of a NetBIOS access filter previously defined with one or more <b>netbios access-list host</b> commands.
<b>bytes</b>	Indicates that the following argument is the name of a NetBIOS access filter previously defined with one or more <b>netbios access-list bytes</b> commands.
<i>name</i>	Name of a NetBIOS access list.

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

You can issue only one **ipx netbios input-access-filter host** and one **ipx netbios input-access-filter bytes** command on each interface.

These filters apply only to IPX NetBIOS packets. They have no effect on LLC2 NetBIOS packets.

### Example

The following example filters packets arriving on Token Ring interface 1 using the NetBIOS access list "engineering":

```
netbios access-list host engineering permit eng*
netbios access-list host engineering deny manu*
interface token 1
ipx netbios input-access-filter engineering
```

### Related Commands

```
ipx netbios output-access filter
netbios access-list
show ipx interface
```

## ipx netbios output-access-filter

To control outgoing NetBIOS messages, use the **ipx netbios output-access-filter** interface configuration command. To remove the filter, use the **no** form of this command.

```
ipx netbios output-access-filter {host | bytes} name  
no ipx netbios output-access-filter {host | bytes} name
```

### Syntax Description

<b>host</b>	Indicates that the following argument is the name of a NetBIOS access filter previously defined with one or more <b>netbios access-list host</b> commands.
<b>bytes</b>	Indicates that the following argument is the name of a NetBIOS access filter previously defined with one or more <b>netbios access-list bytes</b> commands.
<i>name</i>	Name of a previously defined NetBIOS access list.

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

You can issue only one **ipx netbios output-access-filter host** and one **ipx netbios output-access-filter bytes** command on each interface.

These filters apply only to IPX NetBIOS packets. They have no effect on LLC2 NetBIOS packets.

### Example

The following example filters packets leaving Token Ring interface 1 using the NetBIOS access list “engineering”:

```
netbios access-list bytes engineering permit 20 AA**04  
interface token 1  
ipx netbios output-access-filter bytes engineering
```

### Related Commands

```
ipx netbios input-access filter  
netbios access-list  
show ipx interface
```

## ipx network

To enable IPX routing on a particular interface and to optionally select the type of encapsulation (framing), use the **ipx network** interface configuration command. To disable IPX routing, use the **no** form of this command.

```
ipx network network [encapsulation encapsulation-type [secondary]]
no ipx network network [encapsulation encapsulation-type]
```

### Syntax Description

*network*

Network number. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA you can enter just AA.

**encapsulation**  
*encapsulation-type*

(Optional) Type of encapsulation. It can be one of the following values:

- **arpa** (for Ethernet interfaces only)—Use Novell’s Ethernet\_II encapsulation. This encapsulation is recommended for networks that handle both TCP/IP and IPX traffic.
- **hdlc** (for serial interfaces only)—Use HDLC encapsulation.
- **novell-ether** (for Ethernet interfaces only)—Use Novell’s “Ethernet\_802.3” encapsulation. This encapsulation consists of a standard 802.3 MAC header followed directly by the IPX header with a checksum of FFFF. It is the default encapsulation used by NetWare Version 3.11.
- **sap** (for Ethernet interfaces)—Use Novell’s Ethernet\_802.2 encapsulation. This encapsulation consists of a standard 802.3 MAC header followed by an 802.2 LLC header. This is the default encapsulation used by NetWare Version 4.0.
  - (for Token Ring interfaces)—This encapsulation consists of a standard 802.5 MAC header followed by an 802.2 LLC header.
  - (for FDDI interfaces)—This encapsulation consists of a standard FDDI MAC header followed by an 802.2 LLC header.
- **snap** (for Ethernet interfaces)—Use Novell Ethernet\_Snap encapsulation. This encapsulation consists of a standard 802.3 MAC header followed by an 802.2 SNAP LLC header.
  - (for Token Ring and FDDI interfaces)—This encapsulation consists of a standard 802.5 or FDDI MAC header followed by an 802.2 SNAP LLC header.

**secondary**

(Optional) Indicates an additional network configured after the first (primary) network.

## Default

IPX routing is disabled.

Encapsulation types:

For Ethernet: **novell-ether**

For Token Ring: **sap**

For FDDI: **snap**

## Command Mode

Interface configuration

## Usage Guidelines

The extended **ipx network** command allows you to configure more than one logical network on the same physical network (network cable segment). Each network on a given interface must have a different encapsulation type. The first network you configure on an interface is considered to be the primary network. Any additional networks are considered to be secondary networks; these must include the **secondary** keyword. You also can use this command to configure a single logical network on a physical network.

You can configure an IPX network on any supported interface as long as all the networks on the same physical interface use a distinct encapsulation type. For example, you can configure up to four IPX networks on a single Ethernet cable because Ethernet supports four encapsulation types.

The interface processes only packets with the correct encapsulation and the correct network number. IPX networks using other encapsulations can be present on the physical network. The only effect on the communication server is that it uses some processing time to examine packets to determine whether they have the correct encapsulation.

If you use the standard **ipx network** and **ipx encapsulation** commands on an interface that supports more than one logical network, you can use them only to define the first, or primary, network on the interface. Use the extended **ipx network** command to define additional logical networks.

All logical networks on an interface share the same set of configuration parameters. For example, if you change the IPX RIP update time on an interface, you change it for all networks on that interface.

This command is useful when migrating from one type of encapsulation to another. If you are using it for this purpose, you should define the new encapsulation on the primary network.

To delete all networks on an interface, use the following command:

```
no ipx network
```

Deleting the primary network with the following command also deletes all networks on that interface. The argument *network* is the number of the primary network.

```
no ipx network network
```

To delete a secondary network on an interface, use one of the following commands. The argument *network* is the number of a secondary network.

```
no ipx network network
```

```
no ipx network network encapsulation encapsulation-type
```



The following two commands also allow you to enable IPX routing on an interface and specify the encapsulation. These commands can be used on interfaces that support a single network or when enabling the primary subinterface on an interface that supports multiple networks. They are supported in this release of the software to provide compatibility with older versions of the communication server software.

**ipx network**  
**ipx encapsulation** *encapsulation-type*

### Example

The following example configures an interface that has four logical networks:

```
interface ethernet 0
ipx network 0123
ipx encapsulation snap
ipx network 0234 encapsulation sap secondary
ipx network 0345 encapsulation arpa secondary
ipx network 0456 encapsulation novell-ether secondary
```

### Related Command

**ipx routing**

## ipx output-gns-filter

To control which servers are included in the Get Nearest Server (GNS) responses sent by the communication server, use the **ipx output-gns-filter** interface configuration command. To remove the filter from the interface, use the **no** form of this command.

```
ipx output-gns-filter access-list-number  
no ipx output-gns-filter access-list-number
```

### Syntax Description

<i>access-list-number</i>	Number of the SAP access list. All outgoing GNS packets are filtered by the entries in this access list. The argument <i>access-list-number</i> is a decimal number from 1000 to 1099.
---------------------------	--

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

You can issue only one **ipx output-gns-filter** command on each interface.

### Example

The following example excludes the server at address 3c.0800.89a1.1527 from GNS responses sent on Ethernet interface 0, but allows all other servers:

```
access-list 1000 deny 3c.0800.89a1.1527  
access-list 1000 permit -1  
ipx routing  
interface ethernet 0  
ipx output-gns-filter 1000
```

### Related Commands

**access-list** (SAP filtering)  
**ipx gns-round-robin**

## ipx output-network-filter

To control the list of networks included in routing updates sent out an interface, use the **ipx output-network-filter** interface configuration command. To remove the filter from the interface, use the **no** form of this command.

```
ipx output-network-filter access-list-number  
no ipx output-network-filter access-list-number
```

### Syntax Description

*access-list-number*

Number of the access list. All outgoing packets defined with either standard or extended access lists are filtered by the entries in this access list. For standard access lists, *access-list-number* is a decimal number from 800 to 899. For extended access lists, it is a decimal number from 900 to 999.

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

The **ipx output-network-filter** command controls which networks the communication server advertises in its IPX routing updates (RIP updates).

You can issue only one **ipx output-network-filter** command on each interface.

### Example

In the following example, access list 896 controls which networks are specified in routing updates sent out the serial interface 1. This configuration causes network 2b to be the only network advertised in Novell routing updates sent on the specified serial interface.

```
access-list 896 permit 2b  
interface serial 1  
ipx output-network-filter 896
```

### Related Commands

**access-list** (standard)

**access-list** (extended)

**ipx input-network-filter**

**ipx router-filter**

## ipx output-rip-delay

To set the interpacket delay for Routing Information Protocol (RIP) updates sent on a single interface, use the **ipx output-rip-delay** interface configuration command. To return to the default value, use the **no** form of this command.

```
ipx output-rip-delay delay  
no ipx output-rip-delay [delay]
```

### Syntax Description

*delay* Delay, in milliseconds, between packets in a multiple-packet RIP update. With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms. Novell recommends a delay of 55 ms.

### Default

With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms (that is, no additional delay between routing update packets).

### Command Mode

Interface configuration

### Usage Guidelines

The interpacket delay is the delay between the individual packets sent in a multiple-packet routing update. The **ipx output-rip-delay** command sets the interpacket delay for a single interface.

The system uses the interpacket delay specified by the **ipx output-rip-delay** command for periodic and triggered routing updates when no delay is set for triggered routing updates. When you set a delay for triggered routing updates, the system uses the delay specified by the **ipx output-rip-delay** command for only the periodic routing updates sent on the interface.

To set a delay for triggered routing updates, see the **ipx triggered-rip-delay** or **ipx default-triggered-rip-delay** commands.

You can also set a default RIP interpacket delay for all interfaces. See the **ipx default-output-rip-delay** command for more information.

Novell recommends a delay of 55 ms for compatibility with older and slower IPX machines. These machines may lose RIP updates because they process packets more slowly than the router sends them. The delay imposed by this command forces the router to pace its output to the slower-processing needs of these IPX machines.

The default delay on a NetWare 3.11 server is about 100 ms.

This command is also useful on limited bandwidth point-to-point links or X.25 and Frame Relay multipoint interfaces.

### Example

The following example establishes a 55-ms interpacket delay on serial interface 0:

```
interface serial 0
ipx network 106A
ipx output-rip-delay 55
```

### Related Command

**ipx update-time**  
**ipx default-output-rip-delay**  
**ipx default-triggered-rip-delay**  
**ipx triggered-rip-delay**

## ipx output-sap-delay

To set the interpacket delay for Service Advertisement Protocol (SAP) updates sent on a single interface, use the **ipx output-sap-delay** interface configuration command. To return to the default delay value, use the **no** form of this command.

```
ipx output-sap-delay delay  
no ipx output-sap-delay [delay]
```

### Syntax Description

<i>delay</i>	Delay, in milliseconds, between packets in a multiple-packet SAP update. With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms. Novell recommends a delay of 55 ms.
--------------	---

### Default

With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms (that is, no additional delay between update packets).

### Command Mode

Interface configuration

### Usage Guidelines

The interpacket delay is the delay between the individual packets sent in a multiple-packet SAP update. The **ipx output-sap-delay** command sets the interpacket delay for a single interface.

The system uses the interpacket delay specified by the **ipx output-sap-delay** command for periodic and triggered SAP updates when no delay is set for triggered updates. When you set a delay for triggered updates, the system uses the delay specified by the **ipx output-sap-delay** command only for the periodic updates sent on the interface.

To set a delay for triggered updates, see the **ipx triggered-sap-delay** or **ipx default-triggered-sap-delay** commands.

You can also set a default SAP interpacket delay for all interfaces. See the **ipx default-output-sap-delay** command for more information.

Novell recommends a delay of 55 ms for compatibility with older and slower IPX servers. These servers may lose SAP updates because they process packets more slowly than the router sends them. The delay imposed by the **ipx output-sap-delay** command forces the router to pace its output to the slower-processing needs of these servers.

The default delay on a NetWare 3.11 server is about 100 ms.

This command is also useful on limited bandwidth point-to-point links or X.25 and Frame Relay multipoint interfaces.

**Example**

The following example establishes a 55-ms delay between packets in multiple-packet SAP updates on Ethernet interface 0:

```
interface ethernet 0
 ipx network 106A
 ipx output-sap-delay 55
```

**Related Command**

**ipx default-output-sap-delay**  
**ipx default-triggered-sap-delay**  
**ipx sap-interval**  
**ipx triggered-sap-delay**

## ipx output-sap-filter

To control which services are included in Service Advertisement Protocol (SAP) updates sent by the communication server, use the **ipx output-network-filter** interface configuration command. To remove the filter, use the **no** form of this command.

```
ipx output-sap-filter access-list-number  
no ipx output-sap-filter access-list-number
```

### Syntax Description

<i>access-list-number</i>	Number of the SAP access list. All outgoing service advertisements are filtered by the entries in this access list. The argument <i>access-list-number</i> is a decimal number from 1000 to 1099.
---------------------------	---

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

The communication server applies output SAP filters prior to sending SAP packets.

You can issue only one **ipx output-sap-filter** command on each interface.

When configuring SAP filters for NetWare 3.11 and later servers, use the server's internal network and node number (the node number is always 0000.0000.0001) as its address in the SAP **access-list** command. Do not use the *network.node* address of the particular interface board.

### Example

The following example denies service advertisements about server 0000.0000.0001 on network aa from being sent on network 4d (via Ethernet interface 1). All other services are advertised via this network. All services, including those from server aa.0000.0000.0001, are advertised via networks 3c and 2b.

```
access-list 1000 deny aa.0000.0000.0001  
access-list 1000 permit -1  
interface ethernet 0  
ipx net 3c  
interface ethernet 1  
ipx network 4d  
ipx output-sap-filter 1000  
interface serial 0  
ipx network 2b
```



**Related Commands**

**access list** (SAP filtering)

**ipx gns-round-robin**

**ipx input-sap-filter**

**ipx router-sap-filter**

## ipx pad-process-switched-packets

To control whether odd-length packets are padded so as to be sent as even-length packets on an interface, use the **ipx pad-process-switched-packets** interface configuration command. To disable padding, use the **no** form of this command.

```
ipx pad-process-switched-packets  
no ipx pad-process-switched-packets
```

### Syntax Description

This command has no arguments or keywords.

### Default

Enabled on Ethernet interfaces  
Disabled on Token Ring, FDDI, and serial interfaces.

### Command Mode

Interface configuration

### Usage Guidelines

Use this command only under the guidance of a customer engineer or other service representative.

The **ipx pad-process-switched-packets** command affects process-switched packets only, so you must disable fast switching before the **ipx pad-process-switched-packets** command has any effect.

Some IPX end hosts reject Ethernet packets that are not padded. Certain topologies can result in such packets being forwarded onto a remote Ethernet network. Under specific conditions, padding on intermediate media can be used as a temporary workaround for this problem.

### Related Command

**ipx route-cache**

## ipx route

To add a static route to the routing table, use the **ipx route** global configuration command. To remove a route from the routing table, use the **no** form of this command.

```
ipx route network network.node  
no ipx route
```

### Syntax Description

<i>network</i>	Network to which you want to establish a static route. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA.
<i>network.node</i>	Communication server to which to forward packets destined for the specified network. The argument <i>network</i> is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can enter just AA. The argument <i>node</i> is the node number of the target communication server. This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers ( <i>xxx.xxxx.xxx</i> ).

### Default

No static routes are predefined.

### Command Mode

Global configuration

### Usage Guidelines

The **ipx route** command forwards packets destined for the specified network (*network*) via the specified communication server (*network.node*), regardless of whether that communication server is sending dynamic routing information.

Be careful when assigning static routes. When links associated with static routes are lost, traffic may stop being forwarded, even though alternative paths might be available.

### Example

In the following example, the communication server at address 3abc.0000.0c00.1ac9 handles all traffic destined for network 5e:

```
ipx routing  
ipx route 5e 3abc.0000.0c00.1ac9
```

**Related Command**  
**show ipx route**

## ipx route-cache

To enable IPX fast switching, use the **ipx route-cache** interface configuration command. To disable fast switching, use the **no** form of this command.

```
ipx route-cache  
no ipx route-cache
```

### Syntax Description

This command has no arguments or keywords.

### Default

Fast switching enabled

### Command Mode

Interface configuration

### Usage Guidelines

Fast switching allows higher throughput by switching packets using a cache created by previous transit packets. On ciscoBus2 interface cards, fast switching is done between all encapsulation types. On other interface cards, fast switching is done in all cases *except* the following: transfer of packets with **sap** encapsulation from an Ethernet or a Token Ring network to a standard serial line.

You might want to disable fast switching in two situations. One is if you want to save memory on the interface cards: fast-switching caches require more memory than those used for standard switching. The second situation is to avoid congestion on interface cards when a high-bandwidth interface is writing large amounts of information to a low-bandwidth interface.

### Examples

The following example enables fast switching on an interface:

```
interface ethernet 0  
ipx route-cache
```

The following example disables fast switching on an interface:

```
interface ethernet 0  
no ipx route-cache
```

### Related Commands

```
clear ipx cache  
ipx source-network-update  
ipx watchdog-spoof  
show ipx cache
```

## ipx router

To specify the routing protocol to use, use the **ipx router** global configuration command.

```
ipx router {eigrp autonomous-system-number | rip}
```

### Syntax Description

<b>eigrp</b> <i>autonomous-system-number</i>	Specifies the IPX Enhanced IGRP routing protocol. The argument <i>autonomous-system-number</i> is the IPX Enhanced IGRP autonomous system number. It can be a decimal integer from 1 to 65535.
<b>rip</b>	Specifies the RIP routing protocol. It is on by default.

### Default

RIP

### Command Mode

Global configuration

### Usage Guidelines

You can configure multiple IPX Enhanced IGRP processes on a router. To do so, assign each a different autonomous system number.

You must explicitly disable RIP if you do not want to use this routing protocol.

### Example

The following example enables IPX Enhanced IGRP on the router:

```
ipx router eigrp 4
```

### Related Commands

**network**  
**redistribute**

## ipx router-filter

To control the communication servers from which packets are accepted, use the **ipx router-filter** interface configuration command. To remove the filter from the interface, use the **no** form of this command.

```
ipx router-filter access-list-number  
no ipx router-filter
```

### Syntax Description

<i>access-list-number</i>	Number of the access list. All incoming packets defined with either standard or extended access lists are filtered by the entries in this access list. For standard access lists, <i>access-list-number</i> is a decimal number from 800 to 899. For extended access lists, it is a decimal number from 900 to 999.
---------------------------	---

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

You can issue only one **ipx router-filter** command on each interface.

### Example

In the following example, access list 866 controls the communication servers from which packets are accepted. For Ethernet interface 0, only packets from the communication server at 3c.0000.00c0.047d are accepted. All other packets are implicitly denied.

```
access-list 866 permit 3c.0000.00c0.047d  
interface ethernet 0  
ipx router-filter 866
```

### Related Commands

**access-list** (standard)  
**access-list** (extended)  
**ipx input-network-filter**  
**ipx output-network-filter**

## ipx router-sap-filter

To filter Service Advertisement Protocol (SAP) messages received from a particular communication server, use the **ipx router-sap-filter** interface configuration command. To remove the filter, use the **no** form of this command.

```
ipx router-sap-filter access-list-number  
no ipx router-sap-filter access-list-number
```

### Syntax Description

<i>access-list-number</i>	Number of the access list. All incoming service advertisements are filtered by the entries in this access list. The argument <i>access-list-number</i> is a decimal number from 1000 to 1099.
---------------------------	---

### Default

No filters are predefined.

### Command Mode

Interface configuration

### Usage Guidelines

You can issue only one **ipx router-sap-filter** command on each interface.

### Example

In the following example, the communication server will receive service advertisements only from communication server aa.0207.0104.0874:

```
access-list 1000 permit aa.0207.0104.0874  
access-list 1000 deny -1  
interface ethernet 0  
ipx router-sap-filter 1000
```

### Related Commands

- access-list** (SAP filtering)
- ipx input-sap-filter**
- ipx output-sap-filter**
- ipx sap**
- show ipx interface**



## ipx routing

To enable IPX routing, use the **ipx routing** global configuration command. To disable IPX routing, use the **no** form of this command.

```
ipx routing [node]  
no ipx routing
```

### Syntax Description

*node* (Optional) Node number of the communication server. This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers (*xxxx.xxxx.xxxx*). It must not be a multicast address.

If you omit *node*, the communication server uses the hardware MAC address currently assigned to it as its node address. This is the MAC address of the first Ethernet, Token Ring, or FDDI interface card. If no satisfactory interfaces are present in the communication server (such as only serial interfaces), you must specify *node*.

### Default

Disabled

### Command Mode

Global configuration

### Usage Guidelines

The **ipx routing** command enables the IPX Routing Information Protocol (RIP) and Service Advertisement Protocol (SAP) services on the communication server.

If you omit the argument *node* and if the MAC address later changes, the IPX node address automatically changes to the new address. However, connectivity may be lost between the time that the MAC address changes and the time that the IPX clients and servers learn the communication server's new address.

If you plan to use DECnet and IPX routing concurrently on the same interface, you should enable DECnet communication server first, then enable IPX routing without specifying the optional MAC node number. If you enable IPX before enabling DECnet routing, routing for IPX will be disrupted.

### Example

The following example enables IPX routing:

```
ipx routing
```

### Related Command

**ipx network**

## ipx sap

To specify static Service Advertisement Protocol (SAP) entries, use the **ipx sap** global configuration command. To remove static SAP entries, use the **no** form of this command.

```
ipx sap service-type name network.node socket hop-count  
no ipx sap service-type name network.node socket hop-count
```

### Syntax Description

<i>service-type</i>	SAP service-type number. Table 19-3 earlier in this chapter lists some IPX SAP services.
<i>name</i>	Name of the server that provides the service.
<i>network.node</i>	Network number and node address of the server. The argument <i>network</i> is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA you can enter just AA. The argument <i>node</i> is the node number of the target Novell server. This is a 48-bit value represented by a dotted triplet of four-digit hexadecimal numbers ( <i>xxx.xxxx.xxx</i> ).
<i>socket</i>	Socket number for this service. Table 19-2 earlier in this chapter lists some IPX socket numbers.
<i>hop-count</i>	Number of hops to the server.

### Default

Disabled

### Command Mode

Global configuration

### Usage Guidelines

The **ipx sap** command allows you to add static entries into the SAP table. Each entry has a SAP service associated with it. Static SAP assignments always override any identical entries in the SAP table that are learned dynamically, regardless of hop count. The communication server will not announce a static SAP entry unless it has a route to that network.

**Example**

In the following example, the route to JOES\_SERVER is not yet learned, so the system displays an informational message. The JOES\_SERVER service will not be announced in the regular SAP updates until the communication server learns the route to it either by means of a RIP update from a neighbor or an **ipx sap** command.

```
ipx sap 107 MAILSERV 160.0000.0c01.2b72 8104 1
ipx sap 4 FILESERV 165.0000.0c01.3d1b 451 1
ipx sap 143 JOES_SERVER A1.0000.0c01.1234 8170 2
no route to A1, JOES_SERVER won't be announced until route is learned
```

**Related Commands**

**ipx input-sap-filter**  
**ipx output-sap-filter**  
**ipx router-sap-filter**  
**show ipx servers**

## ipx sap-incremental eigrp

To send SAP updates only when a change occurs in the SAP table, use the **ipx sap-incremental eigrp** interface configuration command. To send periodic SAP updates, use the **no** form of this command.

**ipx sap-incremental eigrp** *autonomous-system-number* [**rsup-only**]  
**no ipx sap-incremental eigrp** *autonomous-system-number* [**rsup-only**]

### Syntax Description

*autonomous-system-number* IPX Enhanced IGRP autonomous system number. It can be a decimal integer from 1 to 65535.

**rsup-only** (Optional) Indicates that the system uses Enhanced IGRP on this interface to carry reliable SAP update information only.

### Default

Enabled on serial interfaces  
Disabled on LAN media (Ethernet, Token Ring, FDDI)

### Command Mode

Interface configuration

### Usage Guidelines

With this functionality enabled, if an IPX Enhanced IGRP peer is found on the interface, SAP updates will be sent only when a change occurs in the SAP table. Periodic SAP updates are not sent. When no IPX Enhanced IGRP peer is present on the interface, periodic SAPs are always sent regardless of how this command is set.

If you configure the local communication server to send incremental SAP updates on an Ethernet, and if the local communication server has at least one IPX Enhanced IGRP neighbor and any servers, clients, or routers that do not have IPX Enhanced IGRP configured on the Ethernet interface, these devices will not receive complete SAP information from the local communication server.

If the incremental sending of SAP updates on an interface is configured and no IPX Enhanced IGRP peer is found, SAP updates will be sent periodically until a peer is found. Then, updates will be sent only when changes occur in the SAP table.

To reduce SAP traffic by sending partial SAP updates, specify the **rsup-only** keyword. SAP updates are then sent only when changes occur, and only changes are sent. This feature works with existing IPX RIP networks and IPX Enhanced IGRP networks.

### Example

The following example sends SAP updates on Ethernet interface 0 only when there is a change in the SAP table:

```
interface ethernet 0
 ipx sap-incremental eigrp 200
```

## ipx sap-interval

To configure less frequent Service Advertisement Protocol (SAP) updates over slow links, use the **ipx sap-interval** interface configuration command. To return to the default value, use the **no** form of this command.

```
ipx sap-interval minutes  
no ipx sap-interval
```

### Syntax Description

*minutes* Interval, in minutes, between SAP updates sent by the communication server. The default value is 1 minute. If *minutes* is 0, periodic updates are never sent.

### Default

1 minute

### Command Mode

Interface configuration

### Usage Guidelines

Setting the interval at which SAP updates are sent is most useful on limited-bandwidth, point-to-point links or on X.25 interfaces.

You should ensure that all IPX servers and communication servers on a given network have the same SAP interval. Otherwise, they may decide that a server is down when it is really up.

It is not possible to change the interval at which SAP updates are sent on most PC-based servers. This means that you should never change the interval for an Ethernet or Token Ring network that has servers on it.

Setting the interval to zero means that periodic SAP updates are never sent. It is recommended that you never do this. If you set the interval to zero, communication servers that are inaccessible for any reason when a server powers up or shuts down will miss that event, and will either fail to learn about new servers or fail to detect that the server shut down.

### Example

In the following example, SAP updates are sent (and expected) on serial interface 0 every 5 minutes:

```
interface serial 0  
ipx sap-interval 5
```

### Related Command

**ipx output-sap-delay**

## ipx sap-queue-maximum

To configure the maximum length of the queue of pending input SAP GNS requests and SAP query packets, use the **ipx sap-queue-maximum** global configuration command. To return to the default value, use the **no** form of this command.

```
ipx sap-queue-maximum number  
no ipx sap-queue-maximum
```

### Syntax Description

<i>number</i>	Maximum length of the queue of pending SAP requests. By default, there is no limit to the number of pending SAP requests that the communication server stores in this queue.
---------------	--

### Default

No maximum queue size

### Command Mode

Global configuration

### Usage Guidelines

The communication server maintains a list of SAP requests to process, including all pending Get Nearest Server (GNS) queries from clients attempting to reach servers. When the network is restarted, the communication server can be inundated with hundreds of requests for servers. Most of these can be repeated requests from the same clients. The **ipx sap-queue-maximum** command allows you to configure the maximum length allowed for the pending SAP requests queue. Packets received when the queue is full are dropped.

### Example

The following example sets the length of the queue of pending SAP requests to 20:

```
ipx sap-queue-maximum 20
```

## ipx source-network-update

To repair corrupted network numbers, use the **ipx source-network-update** interface configuration command. To disable this feature, use the **no** form of this command.

```
ipx source-network-update  
no ipx source-network-update
```

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

In some early implementations of IPX client software, it was possible for the client's network number to become corrupted. The **ipx source-network-update** command repairs this number by setting the source network field of any packet on the local network that has a hop count of zero.

You must disable fast switching with the **no ipx route-cache** command before using the **ipx source-network-update** command.

This command interferes with the proper working of OS/2 Requestors. Therefore, do not use this command in a network that has OS/2 Requestors.

Do not use the **ipx source-network-update** command on interfaces on which NetWare servers are using internal network numbers.



**Caution** The **ipx source-network-update** command interferes with the proper working of OS/2 Requestors. Do not use this command in a network that has OS/2 Requestors.

Also, do not use the **ipx source-network-update** command on interfaces on which NetWare (NetWare 3.1x or 4.0 or later) Servers are using internal network numbers.

### Example

In the following example, corrupted network numbers on serial interface 0 are repaired:

```
interface serial 0  
no ipx route-cache  
ipx source-network-update
```

### Related Command

**ipx route-cache**

## ipx split-horizon eigrp

To configure split horizon, use the **ipx split-horizon eigrp** interface configuration command. To disable split horizon, use the **no** form of this command.

```
ipx split-horizon eigrp autonomous-system-number  
no ipx split-horizon eigrp autonomous-system-number
```

### Syntax Description

*autonomous-system-number* IPX Enhanced IGRP autonomous system number. It can be a decimal integer from 1 to 65535.

### Default

Enabled

### Command Mode

Interface configuration

### Usage Guidelines

When split horizon is enabled, IPX Enhanced IGRP update and query packets are not sent for destinations that have next hops on this interface. This reduces the number of Enhanced IGRP packets on the network.

Split horizon blocks information about routes from being advertised by a router out any interface from which that information originated. This behavior usually optimizes communication among multiple routers, particularly when links are broken. However, with nonbroadcast networks, such as Frame Relay and SMDS, situations can arise for which this behavior is less than ideal. For these situations, you may wish to disable split horizon.

### Example

The following example disables split horizon on serial interface 0:

```
interface serial 0  
no ipx split-horizon eigrp 200
```



## ipx triggered-rip-delay

To set the interpacket delay for triggered Routing Information Protocol (RIP) updates sent on a single interface, use the **ipx triggered-rip-delay** interface configuration command. To return to the default delay, use the **no** form of this command.

```
ipx triggered-rip-delay delay  
no ipx triggered-rip-delay [delay]
```

### Syntax Description

*delay* Delay, in milliseconds, between packets in a multiple-packet RIP update. With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms. Novell recommends a delay of 55 ms.

### Default

With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms (that is, no additional delay between routing update packets).

### Command Mode

Interface configuration

### Usage Guidelines

The interpacket delay is the delay between the individual packets sent in a multiple-packet routing update. A triggered routing update is one that the system sends in response to a “trigger” event, such as a request packet, interface up/down, route up/down, or server up/down.

The **ipx triggered-rip-delay** command sets the interpacket delay for triggered routing updates sent on a single interface. The delay value set by this command overrides the delay value set by the **ipx output-rip-delay** or **ipx default-output-rip-delay** command for triggered routing updates sent on the interface.

If the delay value set by the **ipx output-rip-delay** or **ipx default-output-rip-delay** command is high, then we strongly recommend a low delay value for triggered routing updates so that updates triggered by special events are sent in a more timely manner than periodic routing updates.

Novell recommends a delay of 55 ms for compatibility with older and slower IPX machines. These machines may lose RIP updates because they process packets more slowly than the router sends them. The delay imposed by this command forces the router to pace its output to the slower-processing needs of these IPX machines.

The default delay on a NetWare 3.11 server is about 100 ms.

When you do not set the interpacket delay for triggered routing updates, the system uses the delay specified by the **ipx output-rip-delay** or **ipx default-output-rip-delay** command for both periodic and triggered routing updates.

When you use the **no** form of the **ipx triggered-rip-delay** command, the system uses the global default delay set by the **ipx default-triggered-rip-delay** command for triggered RIP updates, if it is set. If it is not set, the system uses the delay set by the **ipx output-rip-delay** or **ipx default-output-rip-delay** command for triggered RIP updates, if set. Otherwise, the system uses the initial default delay as described in the “Default” section.

This command is also useful on limited bandwidth point-to-point links or X.25 and Frame Relay multipoint interfaces.

**Example**

The following example sets an interpacket delay of 55 ms for triggered routing updates sent on interface FDDI 0:

```
int FDDI 0
ipx triggered-rip-delay 55
```

**Related Command**

- ipx default-output-rip-delay**
- ipx default-triggered-rip-delay**
- ipx output-rip-delay**

## ipx triggered-sap-delay

To set the interpacket delay for triggered Service Advertisement Protocol (SAP) updates sent on a single interface, use the **ipx triggered-sap-delay** interface configuration command. To return to the default delay, use the **no** form of this command.

```
ipx triggered-sap-delay delay
no ipx triggered-sap-delay [delay]
```

### Syntax Description

*delay* Delay, in milliseconds, between packets in a multiple-packet SAP update. With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms. Novell recommends a delay of 55 ms.

### Default

With Cisco IOS Release 10.0 and Release 10.2, the default delay is 0 ms (that is, no additional delay between update packets).

### Command Mode

Interface configuration

### Usage Guidelines

The interpacket delay is the delay between the individual packets sent in a multiple-packet SAP update. A triggered SAP update is one that the system sends in response to a “trigger” event, such as a request packet, interface up/down, route up/down, or server up/down.

The **ipx triggered-sap-delay** command sets the interpacket delay for triggered updates sent on a single interface. The delay value set by this command overrides the delay value set by the **ipx output-sap-delay** or **ipx default-output-sap-delay** command for triggered updates sent on the interface.

If the delay value set by the **ipx output-sap-delay** or **ipx default-output-sap-delay** command is high, then we strongly recommend a low delay value for triggered updates so that updates triggered by special events are sent in a more timely manner than periodic updates.

Novell recommends a delay of 55 ms for compatibility with older and slower IPX servers. These servers may lose SAP updates because they process packets more slowly than the router sends them. The delay imposed by this command forces the router to pace its output to the slower-processing needs of these IPX servers.

The default delay on a NetWare 3.11 server is about 100 ms.

When you do not set the interpacket delay for triggered updates, the system uses the delay specified by the **ipx output-sap-delay** or **ipx default-output-sap-delay** command for both periodic and triggered SAP updates.

When you use the **no** form of the **ipx triggered-sap-delay** command, the system uses the global default delay set by the **ipx default-triggered-sap-delay** command for triggered SAP updates, if it is set. If it is not set, the system uses the delay set by the **ipx output-sap-delay** or **ipx default-output-sap-delay** command for triggered SAP updates, if set. Otherwise, the system uses the initial default delay as described in the “Default” section.

This command is also useful on limited bandwidth point-to-point links or X.25 and Frame Relay multipoint interfaces.

### **Example**

The following example sets an interpacket delay of 55 ms for triggered SAP updates sent on interface FDDI 0:

```
int FDDI 0
ipx triggered-sap-delay 55
```

### **Related Commands**

**ipx default-output-sap-delay**  
**ipx default-triggered-sap-delay**  
**ipx output-sap-delay**  
**ipx network**

## ipx type-20-helpered

To forward IPX type 20 propagation packet broadcasts to specific network segments, use the **ipx type-20-helpered** interface configuration command. To disable this function, use the **no** form of this command.

```
ipx type-20-helpered
no ipx type-20-helpered
```

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

The **ipx type-20-helpered** command disables the input and output of type 20 propagation packets as done by the **ipx type-20-propagation** interface configuration command.

The **ipx type-20-propagation** command broadcasts type 20 packets to all nodes on the network and imposes a hop-count limit of eight routers for broadcasting these packets. These functions are in compliance with the Novell IPX router specification. In contrast, the **ipx type-20-helpered** command broadcasts type 20 packets to only those nodes indicated by the **ipx helper-address** interface configuration command and extends the hop-count limit to 16 routers.

Use of the **ipx type-20-helpered** command does not comply with the Novell IPX router specification.

### Example

The following example forwards IPX type 20 propagation packet broadcasts to specific network segments:

```
interface ethernet 0
ipx network aa
ipx type-20-helpered
ipx helper-address bb.ffff.ffff.ffff
```

### Related Commands

```
ipx helper-address
ipx type-20-propagation
```

## ipx type-20-input-checks

To restrict the acceptance of IPX type 20 propagation packet broadcasts, use the **ipx type-20-input-checks** global configuration command. To remove these restrictions, use the **no** form of this command.

```
ipx type-20-input-checks  
no type-20-input-checks
```

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Global configuration

### Usage Guidelines

By default, the communication server is configured to block type 20 propagation packets. When type 20 packet handling is enabled on multiple interfaces, you can use the **ipx type-20-input-checks** command to impose additional restrictions on the acceptance of type 20 packets. Specifically, the communication server will accept type 20 propagation packets only on the single network that is the primary route back to the source network. Similar packets received via other networks will be dropped. This behavior can be advantageous in redundant topologies, because it reduces unnecessary duplication of type 20 packets.

### Example

The following example imposes additional restrictions on incoming type 20 broadcasts:

```
ipx type-20-input-checks
```

### Related Commands

```
ipx type-20-output-checks  
ipx type-20-propagation
```

## ipx type-20-output-checks

To restrict the forwarding of IPX type 20 propagation packet broadcasts, use the **ipx type-20-output-checks** global configuration command. To remove these restrictions, use the **no** form of this command.

```
ipx type-20-output-checks  
no type-20-output-checks
```

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Global configuration

### Usage Guidelines

By default, the communication server is configured to block type 20 propagation packets. When type 20 packet handling is enabled on multiple interfaces, you can use the **ipx type-20-output-checks** command to impose additional restrictions on outgoing type 20 packets. Specifically, the communication server will forward these packets only to networks that are not routes back to the source network. (The communication server uses the current routing table to determine routes.) This behavior can be advantageous in redundant topologies, because it reduces unnecessary duplication of type 20 packets.

### Example

The following example imposes restrictions on outgoing type 20 broadcasts:

```
ipx type-20-output-checks
```

### Related Commands

```
ipx type-20-input-checks  
ipx type-20-propagation
```

## ipx type-20-propagation

To forward IPX type 20 propagation packet broadcasts to other network segments, use the **ipx type-20-propagation** interface configuration command. To disable both the reception and forwarding of type 20 broadcasts on an interface, use the **no** form of this command.

**ipx type-20-propagation**  
**no type-20-propagation**

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

Communication servers normally block all broadcast requests. To allow input and output of type 20 propagation packets on an interface, use the **ipx type-20-propagation** command. Note that type 20 packets are subject to loop detection and control as specified in the IPX router specification.

Additional input and output checks may be imposed by the **ipx type-20-input-checks** and **ipx type-20-output-checks** commands.

IPX type 20 propagation packet broadcasts are subject to any filtering defined by the **ipx helper-list** command.

### Example

The following example enables both the reception and forwarding of type 20 broadcasts on Ethernet interface 0:

```
interface ethernet 0
 ipx type-20-propagation
```

### Related Commands

**ipx helper-address**  
**ipx helper-list**  
**ipx type-20-input-checks**  
**ipx type-20-output-checks**



## ipx update-time

To adjust the IPX routing update timers, use the **ipx update-time** interface configuration command. To restore the default value, use the **no** form of this command.

**ipx update-time** *seconds*  
**no ipx update-time**

### Syntax Description

*seconds* Interval, in seconds, at which IPX routing updates are sent. The default is 60 seconds. The minimum interval is 10 seconds.

### Default

60 seconds

### Command Mode

Interface configuration

### Usage Guidelines

The **ipx update-time** command sets the routing update timer on a per-interface basis.

Communication servers exchange information about routes by sending broadcast messages when they are brought up and shut down, and periodically while they are running. The **ipx update-time** command lets you modify the periodic update interval. By default, this interval is 60 seconds (this default is defined by Novell).

You can set RIP timers only in a configuration in which all communication servers are our communication servers or in which the IPX communication servers allow configurable timers. The timers should be the same for all communication servers connected to the same cable segment.

The update value you choose affects the internal IPX timers as follows:

- IPX routes are marked invalid if no routing updates are heard within three times the value of the update interval ( $3 \times \textit{seconds}$ ) and are advertised with a metric of infinity.
- IPX routes are removed from the routing table if no routing updates are heard within four times the value of the update interval ( $4 \times \textit{seconds}$ ).
- If you define an update timer for more than one interface in a communication server, the granularity of the update timer is determined by the lowest value defined for one of the interfaces in the communication server. The communication server “wakes up” at this granularity interval and determines what updates need to be sent.

The concept of granularity is best explained by an example. (This example is illustrated in the “Example” section following.) If you have two interfaces in the communication server and you set the update timer on one to 20 seconds and the second to 30 seconds, the communication server wakes up every 20 seconds to try to send routing updates. So at time 0:00:20, the communication server sends an update out the first interface only, and at time 0:00:40 it sends updates out the first and second interfaces. The communication server does not wake up at 0:00:30 to see if it needs to send an update out the second interface. This means that routing updates are sent out the second interface at N:NN:40 and N:NN:00. That is, the interval alternates between 40 seconds and 20 seconds; it is never 30 seconds. The interval on the first interface is always 20 seconds.

### Example

The following example sets the update timers on two interfaces in the communication server. The update timer granularity would be 20 seconds because this is the lowest value specified.

```
interface serial 0
 ipx update-time 40
interface ethernet 0
 ipx update-time 20
```

### Related Command

**show ipx interface**

## ipx watchdog-spoof

To have the communication server respond to a server's watchdog packets on behalf of a remote client, use the **ipx watchdog-spoof** interface configuration command. To disable spoofing, use the **no** form of this command.

```
ipx watchdog-spoof  
no ipx watchdog-spoof
```

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

You can use the **ipx watchdog-spoof** command only on a serial interface on which dial-on-demand routing (DDR) has been enabled. Also, fast switching and autonomous switching must be disabled on the interface.

IPX watchdog packets are keepalive packets that are sent from servers to clients after a client session has been idle for approximately 5 minutes. On a DDR link, this would mean that a call would be made every 5 minutes, regardless of whether there were data packets to send. You can prevent these calls from being made by configuring the communication server to respond to the server's watchdog packets on a remote client's behalf. This is sometimes referred to as "spoofing the server."

### Example

The following example enables spoofing on serial interface 0:

```
interface serial 0  
ipx watchdog-spoof  
no ipx route-cache
```

### Related Command

**ipx route-cache**

## netbios access-list

To define an IPX NetBIOS access list filter, use the **netbios access-list** global configuration command. To remove a filter, use the **no** form of this command.

```
netbios access-list host name {deny | permit} string
no netbios access-list host name {deny | permit} string
```

```
netbios access-list bytes name {deny | permit} offset byte-pattern
no netbios access-list bytes name {deny | permit} offset byte-pattern
```

### Syntax Description

<b>host</b>	Indicates that the following argument is the name of a NetBIOS access filter previously defined with one or more <b>netbios access-list host</b> commands.
<b>bytes</b>	Indicates that the following argument is the name of a NetBIOS access filter previously defined with one or more <b>netbios access-list bytes</b> commands.
<i>name</i>	Name of the access list being defined. The name can be an alphanumeric string.
<b>deny</b>	Denies access if the conditions are matched.
<b>permit</b>	Permits access if the conditions are matched.
<i>string</i>	Character string that identifies one or more NetBIOS host names. It can be up to 14 characters long. The argument <i>string</i> can include the following wildcard characters: <ul style="list-style-type: none"> <li>• *—Match one or more characters. You can use this wildcard character only at the end of a string.</li> <li>• ?—Match any single character.</li> </ul>
<i>offset</i>	Decimal number that indicates the number of bytes into the packet at which the byte comparison should begin. An offset of 0 indicates the beginning of the NetBIOS packet header, which is at the end of the IPX header.
<i>byte-pattern</i>	Hexadecimal pattern that represents the byte pattern to match. It can be up to 16 bytes (32 digits) long and must be an even number of digits. The argument <i>byte-pattern</i> can include the following wildcard character: <ul style="list-style-type: none"> <li>• **—Match any digits for that byte.</li> </ul>

### Default

No filters are predefined.

### Command Mode

Global configuration

## Usage Guidelines

Keep the following points in mind when configuring IPX NetBIOS access control:

- Host (node) names are case sensitive.
- Host and byte access lists can have the same names. They are independent of each other.
- When filtering by node name for IPX NetBIOS, the names in the access lists are compared with the destination name field for IPX NetBIOS “find name” requests.
- When filtering by byte offset, note that these access filters can have a significant impact on the packets’ transmission rate across the bridge because each packet must be examined. You should use these access lists only when absolutely necessary.
- If a node name is not found in an access list, the default action is to deny access.

These filters apply only to IPX NetBIOS packets. They have no effect on LLC2 NetBIOS packets.

To delete an IPX NetBIOS access list, specify the minimum number of keywords and arguments needed to delete the proper list. For example, to delete the entire list, use the following command:

```
no netbios access-list {host | bytes} name
```

To delete a single entry from the list, use the following command:

```
no netbios access-list host name {permit | deny} string
```

## Examples

The following example defines the IPX NetBIOS access list *engineering*:

```
netbios access-list host engineering permit eng-ws1 eng-ws2 eng-ws3
```

The following example removes a single entry from the *engineering* access list:

```
netbios access-list host engineering deny eng-ws3
```

The following example removes the entire *engineering* NetBIOS access list:

```
no netbios access-list host engineering
```

## Related Commands

**ipx netbios input-access filter**

**ipx netbios output-access filter**

**show ipx interface**

## ping (user)

To check host reachability and network connectivity, use the **ping** user EXEC command.

```
ping ipx {host | address}
```

### Syntax Description

<b>ipx</b>	Specifies the IPX protocol.
<i>host</i>	Host name of system to ping.
<i>address</i>	Address of system to ping.

### Command Mode

EXEC

### Usage Guidelines

The user-level **ping** (packet internet proper function) command provides a basic ping facility for users who do not have system privileges. This command is equivalent to the nonverbose form of the privileged **ping** command. It sends five 100-byte ping packets.

You cannot ping a communication server from itself.

If the system cannot map an address for a host name, it will return an “%Unrecognized host or address” error message.

To abort a **ping** session, type the escape sequence. By default, this is Ctrl-^ X. You enter this by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go, and then pressing the X key.

Table 19-4 describes the test characters displayed in **ping** responses.

**Table 19-4 Ping Test Characters**

Character	Meaning
!	Each exclamation point indicates receipt of a reply.
.	Each period indicates the network server timed out while waiting for a reply.
U	A destination unreachable error PDU was received.
C	A congestion experienced packet was received.
I	User interrupted the test.
?	Unknown packet type.
&	Packet lifetime exceeded.

### Sample Display

The following sample display shows input to and output from the user **ping** command:

```
cs> ping ipx 211.0000.0c01.f4cf
Type escape sequence to abort.
Sending 5, 100-byte Novell Echoes to 211.0000.0c01.f4cf, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

**Related Command**

**ping** (privileged)

## ping (privileged)

To check host reachability and network connectivity, use the **ping** privileged EXEC command.

```
ping [ipx] [host | address]
```

### Syntax Description

<b>ipx</b>	(Optional) Specifies the IPX protocol
<i>host</i>	(Optional) Host name of system to ping
<i>address</i>	(Optional) Address of system to ping

### Command Mode

Privileged EXEC

### Usage Guidelines

The privileged **ping** (packet internet proper function) command provides a complete **ping** facility for users who have system privileges.

You cannot ping a communication server from itself.

Novell IPX devices that support the echo function defined in version 1.0 of the NLSP specification will respond to this command if you answer **y** to the prompt `Novell Standard Echo` that is displayed when you use the ping command. If you answer **n** to this prompt, Novell IPX devices will not respond.

To abort a **ping** session, type the escape sequence. By default, this is Ctrl-^ X. You enter this by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go, and then pressing the X key.

Table 19-5 describes the test characters displayed in **ping** responses.

**Table 19-5 Ping Test Characters**

Character	Meaning
!	Each exclamation point indicates the receipt of a reply from the target address.
.	Each period indicates the network server timed out while waiting for a reply from the target address.
U	A destination unreachable error PDU was received.
C	A congestion experienced packet was received.
I	User interrupted the test.
?	Unknown packet type.
&	Packet lifetime exceeded.



## Sample Display

The following sample display shows input to and output from the **ping** command:

```
cs# ping
Protocol [ip]: ipx
Target Novell Address: 211.0000.0c01.f4cf
Repeat Count [5]:
Datagram Size [100]:
Timeout in seconds [2]:
Verbose [n]:
Novell Standard Echo [n]:
Type escape sequence to abort.
Sending 5 100-byte Novell echoes to 211.0000.0c01.f4cf, timeout is 2 seconds.
!!!!
Success rate is 100%, round trip min/avg/max = 1/2/4 ms.
```

## Related Command

**ping** (user)

## show ipx accounting

To display the active accounting or checkpointed database, use the **show ipx accounting EXEC** command.

**show ipx accounting [checkpoint]**

### Syntax Description

**checkpoint** (Optional) Displays entries in the checkpointed database.

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show ipx accounting** command:

```

cs# show ipx accounting
Source          Destination          Packets      Bytes
0000C003.0000.0c05.6030 0000C003.0260.8c9b.4e33      72          2880
0000C001.0260.8c8d.da75 0000C003.0260.8c9b.4e33      14           624
0000C003.0260.8c9b.4e33 0000C001.0260.8c8d.da75      62          3110
0000C001.0260.8c8d.e7c6 0000C003.0260.8c9b.4e33      20          1470
0000C003.0260.8c9b.4e33 0000C001.0260.8c8d.e7c6      20          1470

Accounting data age is      6
    
```

Table 19-6 describes the fields shown in the display.

**Table 19-6 Show IPX Accounting Field Descriptions**

Field	Description
Source	Source address of the packet.
Destination	Destination address of the packet.
Packets	Number of packets transmitted from the source address to the destination address.
Bytes	Number of bytes transmitted from the source address to the destination address.
Accounting data age is ...	Time since the accounting database has been cleared. It can be in one of the following formats: <i>mm</i> , <i>hh:mm</i> , <i>dd:hh</i> , and <i>xw yd</i> , where <i>m</i> is minutes, <i>h</i> is hours, <i>d</i> is days, and <i>w</i> is weeks.

### Related Commands

- clear ipx accounting**
- ipx accounting**
- ipx accounting-list**
- ipx accounting-threshold**
- ipx accounting-transits**

## show ipx cache

To display the contents of the IPX fast-switching cache, use the **show ipx cache** EXEC command.

**show ipx cache**

### Syntax Description

This command has no arguments or keywords.

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show ipx cache** command:

```
cs# show ipx cache

Novell routing cache version is 9
Destination      Interface      MAC Header
*1006A          Ethernet 0    00000C0062E600000C003EB0064
*14BB           Ethernet 1    00000C003E2A00000C003EB0064
```

Table 19-7 describes the fields shown in the display.

**Table 19-7 Show IPX Cache Field Descriptions**

Field	Description
Novell routing cache version is ...	Number identifying the version of the fast-switching cache table. It increments each time the table changes.
Destination	Destination network for this packet. Valid entries are marked by an asterisk (*).
Interface	Communication server interface through which this packet is transmitted.
MAC Header	Contents of this packet's MAC header.

### Related Commands

**clear ipx cache**

**ipx route-cache**

## show ipx eigrp neighbors

To display the neighbors discovered by Enhanced IGRP, use the **show ipx eigrp neighbors EXEC** command.

```
show ipx eigrp neighbors [servers] [autonomous-system-number | type number]
```

### Syntax Description

<b>servers</b>	(Optional) Displays the server list advertised by each neighbor. This is displayed only if the <b>ipx sap incremental</b> command is enabled on the interface on which the neighbor resides.
<i>autonomous-system-number</i>	(Optional) Autonomous system number. It can be a decimal integer from 1 to 65535.
<i>type</i>	(Optional) Interface type.
<i>number</i>	(Optional) Interface number.

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show ipx eigrp neighbors** command:

```
cs# show ipx eigrp neighbors
IPX EIGRP Neighbors for process 200
Address                Interface    Holdtime  Uptime    Q      Seq  SRTT  RTO
                   (secs)    (h:m:s)  Count    Num   (ms)  (ms)
160.0000.0c00.8ea9    Ethernet0    10       2:08:29  0      18   4     20
```

Table 19-8 explains the fields in the output.

**Table 19-8 Show IPX EIGRP Neighbors Field Descriptions**

Field	Description
process 200	Autonomous system number specified in the <b>ipx router</b> configuration command.
H	Handle. An arbitrary and unique number inside this communication server that identifies the neighbor.
Address	IPX address of the Enhanced IGRP peer.
Interface	Interface on which the communication server is receiving hello packets from the peer.
Holdtime	Length of time, in seconds, that the communication server will wait to hear from the peer before declaring it down. If the peer is using the default hold time, this number will be less than 15. If the peer configures a nondefault hold time, it will be reflected here.
Uptime	Elapsed time, in hours, minutes, and seconds, since the local communication server first heard from this neighbor.

---

<b>Field</b>	<b>Description</b>
Q Cnt	Number of IPX Enhanced IGRP packets (Update, Query, and Reply) that the communication server is waiting to send.
Seq Num	Sequence number of the last Update, Query, or Reply packet that was received from this neighbor.
SRTT	Smooth round-trip time. This is the number of milliseconds it takes for an IPX Enhanced IGRP packet to be sent to this neighbor and for the local communication server to receive an acknowledgment of that packet.
RTO	Retransmission timeout, in milliseconds. This is the amount of time the communication server waits before retransmitting a packet from the retransmission queue to a neighbor.

---

## show ipx eigrp topology

To display the IPX Enhanced IGRP topology table, use the **show ipx eigrp topology EXEC** command.

**show ipx eigrp topology** [*network-number*]

### Syntax Description

*network-number* (Optional) IPX network number whose topology table entry to display

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show ipx eigrp topology** command:

```

cs# show ipx eigrp topology
IPX EIGRP Topology Table for process 109
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - Reply status
P 42, 1 successors, FD is 0
    via 160.0000.0c00.8ea9 (345088/319488), Ethernet0
P 160, 1 successor via Connected, Ethernet
    via 160.0000.0c00.8ea9 (307200/281600), Ethernet0
P 165, 1 successors, FD is 307200
    via Redistributed (287744/0)
    via 160.0000.0c00.8ea9 (313344/287744), Ethernet0
P 164, 1 successors, flags: U, FD is 200
    via 160.0000.0c00.8ea9 (307200/281600), Ethernet1
    via 160.0000.0c01.2b71 (332800/307200), Ethernet1
P A112, 1 successors, FD is 0
    via Connected, Ethernet2
    via 160.0000.0c00.8ea9 (332800/307200), Ethernet0
P AAABBB, 1 successors, FD is 10003
    via Redistributed (287744/0),
    via 160.0000.0c00.8ea9 (313344/287744), Ethernet0
A A112, 0 successors, 1 replies, state: 0, FD is 0
    via 160.0000.0c01.2b71 (307200/281600), Ethernet1
    via 160.0000.0c00.8ea9 (332800/307200) , r, Ethernet1
    
```

Table 19-9 explains the fields in the output.

**Table 19-9 Show IPX EIGRP Topology Field Descriptions**

Field	Description
Codes	State of this topology table entry. Passive and Active refer to the Enhanced IGRP state with respect to this destination; Update, Query, and Reply refer to the type of packet that is being sent.
P – Passive	No Enhanced IGRP computations are being performed for this destination.
A – Active	Enhanced IGRP computations are being performed for this destination.
U – Update	Indicates that an update packet was sent to this destination.
Q – Query	Indicates that a query packet was sent to this destination.

Field	Description
R – Reply	Indicates that a reply packet was sent to this destination.
r – Reply status	Flag that is set after the router has sent a query and is waiting for a reply.
42, 160, and so on	Destination IPX network number.
successors	Number of successors. This number corresponds to the number of next hops in the IPX routing table.
FD	Feasible distance. This value is used in the feasibility condition check. If the neighbor's reported distance (the metric after the slash) is less than the feasible distance, the feasibility condition is met and that path is a feasible successor. Once the router determines it has a feasible successor, it does not have to send a query for that destination.
replies	Number of replies that are still outstanding (have not been received) with respect to this destination. This information appears only when the destination is in Active state.
state	Exact Enhanced IGRP state that this destination is in. It can be the number 0, 1, 2, or 3. This information appears only when the destination is Active.
via	IPX address of the peer who told the router about this destination. The first <i>n</i> of these entries, where <i>n</i> is the number of successors, are the current successors. The remaining entries on the list are feasible successors.
(345088/319488)	The first number is the Enhanced IGRP metric that represents the cost to the destination. The second number is the Enhanced IGRP metric that this peer advertised.
Ethernet0	Interface from which this information was learned.

The following is sample output from the **show ipx eigrp topology** command when you specify an IPX network number:

```

cs# show ipx eigrp topology 160
IPX-EIGRP topology entry for 160
State is Passive, Query origin flag is 1, 1 Successor(s)
Routing Descriptor Blocks:
  Next hop is Connected (Ethernet0), from 0.0000.0000.0000
  Composite metric is (0/0), Send flag is 0x0, Route is Internal
  Vector metric:
    Minimum bandwidth is 10000 Kbit
    Total delay is 1000000 nanoseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 0
  Next hop is 164.0000.0c00.8ea9 (Ethernet1), from 164.0000.0c00.8ea9
  Composite metric is (307200/281600), Send flag is 0x0, Route is External
  This is an ignored route
  Vector metric:
    Minimum bandwidth is 10000 Kbit
    Total delay is 2000000 nanoseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
  External data:
    Originating router is 0000.0c00.8ea9
    External protocol is RIP, metric is 1, delay 2
    Administrator tag is 0 (0x00000000)
    Flag is 0x00000000

```

Table 19-10 explains the fields in the output.

**Table 19-10 Show IPX EIGRP Topology Field Descriptions for a Specified Network**

Field	Description
160	IPX network number of the destination.
State is ...	State of this entry. It can be either Passive or Active. Passive means that no Enhanced IGRP computations are being performed for this destination, and Active means that they are being performed.
Query origin flag	Exact Enhanced IGRP state that this destination is in. It can be the number 0, 1, 2, or 3. This information appears only when the destination is Active.
Successors	Number of successors. This number corresponds to the number of next hops in the IPX routing table.
Next hop is ...	Indicates how this destination was learned. It can be one of the following: <ul style="list-style-type: none"> <li>• Connected—The destination is on a network directly connected to this router.</li> <li>• Redistributed—The destination was learned via RIP or another Enhanced IGRP process.</li> <li>• IPX host address—The destination was learned from that peer via this Enhanced IGRP process.</li> </ul>
Ethernet0	Interface from which this information was learned.
from	Peer from whom the information was learned. For connected and redistributed routers, this is 0.0000.0000.0000. For information learned via Enhanced IGRP, this is the peer's address. Currently, for information learned via Enhanced IGRP, the peer's IPX address always matches the address in the "Next hop is" field.
Composite metric is	Enhanced IGRP composite metric. The first number is this communication server's metric to the destination, and the second is the peer's metric to the destination.
Send flag	Numeric representation of the "flags" field described in Table 19-8. It is 0 when nothing is being sent, 1 when an Update is being sent, 3 when a Query is being sent, and 4 when a Reply is being sent. Currently, 2 is not used.
Route is ...	Type of communication server. It can be either internal or external. Internal routes are those that originated in an Enhanced IGRP autonomous system, and external are routes that did not. Routes learned via RIP are always external.
This is an ignored route	Indicates that this path is being ignored because of filtering.
Vector metric:	This section describes the components of the Enhanced IGRP metric.
Minimum bandwidth	Minimum bandwidth of the network used to reach the next hop.
Total delay	Delay time to reach the next hop.
Reliability	Reliability value used to reach the next hop.
Load	Load value used to reach the next hop.
Minimum MTU	Minimum MTU size of the network used to reach the next hop.
Hop count	Number of hops to the next hop.
External data	This section describes the original protocol from which this route was redistributed. It appears only for external routes.
Originating router	Network address of the communication server that first distributed this route into Enhanced IGRP.



Field	Description
External protocol..metric..delay	External protocol from which this route was learned. The metric will match the external hop count displayed by the <b>show ipx route</b> command for this destination. The delay is the external delay.
Administrator tag	Not currently used.
Flag	Not currently used.

**Related Command****show ipx route**

## show ipx interface

To display the status of the IPX interfaces configured in the communication server and the parameters configured on each interface, use the **show ipx interface EXEC** command.

**show ipx interface** [*type number*]

### Syntax Description

<i>type</i>	(Optional) Interface type. It can be one of the following types: asynchronous, dialer, Ethernet (IEEE 802.3), loopback, null, serial, token ring, or tunnel.
<i>number</i>	(Optional) Interface number.

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show ipx interface** command:

```
cs# show ipx interface ethernet 1
Ethernet1 is up, line protocol is up
  IPX address is C03.0000.0c05.6030, NOVELL-ETHER [up] line-up, RIPPQ: 0, SAPPQ : 0
  Delay of this Novell network, in ticks is 1
  IPXWAN processing not enabled on this interface.
  IPX SAP update interval is 1 minute(s)
  IPX type 20 propagation packet forwarding is disabled
  Outgoing access list is not set
  IPX Helper access list is not set
  SAP Input filter list is not set
  SAP Output filter list is not set
  SAP Router filter list is not set
  SAP GNS output filter list is not set
  Input filter list is not set
  Output filter list is not set
  Router filter list is not set
  Netbios Input host access list is not set
  Netbios Input bytes access list is not set
  Netbios Output host access list is not set
  Netbios Output bytes access list is not set
  Update time is 60 seconds
  IPX accounting is enabled
  IPX fast switching is configured (enabled)
  IPX SSE switching is disabled
```

Table 19-11 describes the fields shown in the display.

Table 19-11 Show IPX Interface Field Descriptions

Field	Description
Ethernet1 is ..., line protocol is ...	Type of interface and whether it is currently active and inserted into the network (up) or inactive and not inserted (down).
IPX address is ...	Network and node address of the local communication server interface, followed by the type of encapsulation configured on the interface and the interface's status. Refer to the <b>ipx network</b> command for a list of possible values.
NOVELL-ETHER	Type of encapsulation being used on the interface, if any.
[up] line-up	Indicates whether IPX routing is enabled or disabled on the interface. "line-up" indicates that IPX routing has been enabled with the <b>ipx routing</b> command. "line-down" indicates that it is not enabled. The word in square brackets provides more detail about the status of IPX routing when it is in the process of being enabled or disabled.
RIPPQ:	Number of packets in the RIP queue.
SAPPQ:	Number of packets in the SAP queue.
Secondary address is ...	Address of a secondary network configured on this interface, if any, followed by the type of encapsulation configured on the interface and the interface's status. Refer to the <b>ipx network</b> command for a list of possible values. This line is displayed only if you have configured a secondary address with the <b>ipx network</b> command.
Delay of this Novell network, in ticks, ...	Value of the ticks field (configured with the <b>ipx delay</b> command).
IPXWAN processing...	Indicates whether IPXWAN processing has been enabled on this interface with the <b>ipx ipxwan</b> command.
IPX SAP update interval	Indicates the frequency of outgoing SAP updates (configured with the <b>ipx sap-interval</b> command).
IPX type 20 propagation packet forwarding...	Indicates whether forwarding of IPX type 20 propagation packets (used by NetBIOS) is enabled or disabled on this interface, as configured with the <b>ipx type-20-propagation</b> command.
Outgoing access list	Indicates whether an access list has been enabled with the <b>ipx access-group</b> command.
IPX Helper access list	Number of the broadcast helper list applied to the interface with the <b>ipx helper-list</b> command.
SAP Input filter list	Number of the input SAP filter applied to the interface with the <b>ipx input-sap-filter</b> command.
SAP Output filter list	Number of the output SAP filter applied to the interface with the <b>ipx output-sap-filter</b> command.
SAP Router filter list	Number of the communication server SAP filter applied to the interface with the <b>ipx router-sap-filter</b> command.
SAP GNS output filter	Number of the Get Nearest Server (GNS) response filter applied to the interface with the <b>ipx output-gns-filter</b> command.
Input filter	Number of the input filter applied to the interface with the <b>ipx input-network-filter</b> command.
Output filter	Number of the output filter applied to the interface with the <b>ipx output-network-filter</b> command.
Router filter	Number of the communication server entry filter applied to the interface with the <b>ipx router-filter</b> command.

Field	Description
Netbios Input host access list	Name of the IPX NetBIOS input host filter applied to the interface with the <b>ipx netbios input-access-filter host</b> command.
Netbios Input bytes access list	Name of the IPX NetBIOS input bytes filter applied to the interface with the <b>ipx netbios input-access-filter bytes</b> command.
Netbios Output host access list	Name of the IPX NetBIOS output host filter applied to the interface with the <b>ipx netbios output-access-filter host</b> command.
Netbios Output bytes access list	Name of the IPX NetBIOS output bytes filter applied to the interface with the <b>ipx netbios output-access-filter bytes</b> command.
Update time	How often the communication server sends RIP updates, as configured with the <b>ipx update-time</b> command.
Watchdog spoofing ...	Indicates whether watchdog spoofing is enabled or disabled for this interface, as configured with the <b>ipx watchdog-spoof</b> command. This information is displayed only on serial interfaces.
IPX accounting	Indicates whether IPX accounting has been enabled with the <b>ipx accounting</b> command.
IPX Fast switching IPX Autonomous switching	Indicates whether IPX fast switching is enabled (default) or disabled for this interface, as configured with <b>ipx route-cache</b> command. (If IPX autonomous switching is enabled, it is configured with the <b>ipx route-cache dbus</b> command.)
IPX SSE switching	Indicates whether IPX SSE switching is enabled for this interface, as configured with the <b>ipx route-cache sse</b> command.

### Related Commands

**access-list** (standard)  
**access-list** (extended)  
**access-list** (SAP)  
**ipx accounting**  
**ipx delay**  
**ipx encapsulation**  
**ipx helper-list**  
**ipx input-network-filter**  
**ipx input-sap-filter**  
**ipx ipxwan**  
**ipx netbios input-access-filter**  
**ipx netbios output-access-filter**  
**ipx network**  
**ipx output-gns-filter**  
**ipx output-network-filter**  
**ipx output-rip-delay**  
**ipx output-sap-filter**  
**ipx route-cache**  
**ipx router filter**  
**ipx router-sap-filter**  
**ipx routing**  
**ipx sap-interval**  
**ipx update-time**  
**ipx watchdog-spoof**  
**netbios access-list**

## show ipx route

To display the contents of the IPX routing table, use the **show ipx route** EXEC command.

```
show ipx route [network]
```

### Syntax Description

*network* (Optional) Number of the network whose routing table entry you want to display. This is an eight-digit hexadecimal number that uniquely identifies a network cable segment. It can be a number in the range 1 to FFFFFFFE. A network number of 0 matches the local network. A network number of -1 matches all networks. You do not need to specify leading zeros in the network number. For example, for the network number 000000AA, you can just enter AA.

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show ipx route** command:

```
cs> show ipx route
Codes: C - Connected primary network, c - Connected secondary network
       R - RIP, E - EIGRP, S - static, W - IPXWAN connected
8 Total IPX routes

No parallel paths allowed          Novell routing algorithm variant in use

E  Net 1 [307200/0] via 2.0000.0c05.84b6, age 0:01:25,
    1 uses, Ethernet2
C  Net 2 (NOVELL-ETHER), is directly connected, 30 uses, Ethernet2
E  Net 3 [2195456/0] via 2.0000.0c05.84b6, age 0:01:25,
    1 uses, Ethernet2
E  Net 11 [281600/0] via 2.0000.0c05.84b6, age 0:01:25,
    1 uses, Ethernet2
c  Net 2000 (SAP), is directly connected, 3 uses, Ethernet2
R  Net 22 [1/1] via 2.0000.0c05.84b6, 13 sec, 1 uses, Ethernet2
C  Net 55 (NOVELL-ETHER), is directly connected, 30 uses, Ethernet3
S  Net 70 via 55.0011.0022.0033, 1 uses, Ethernet3
```

Table 19-12 describes the fields shown in the display.

**Table 19-12 Show IPX Route Field Descriptions**

Field	Description
Codes	Codes defining how the route was learned.
C	Directly connected primary network.
c	Directly connected secondary network
R	Route learned from a RIP update.
E	Route learned from an Enhanced IGRP update.
S	Statically defined route via the <b>ipx route</b> command.

Field	Description
W	Directly connected route determined via IPXWAN.
8 Total IPX routes	Numbers of routes in the IPX routing table.
No parallel paths allowed	Maximum number of parallel paths for which the communication server has been configured with the <b>ipx maximum-paths</b> command.
Novell routing algorithm variant in use	Indicates whether the communication server is using the IPX-compliant routing algorithms (default).
Net 1	Network to which the route goes.
[3/2]	Delay/Metric. Delay is the number of IBM clock ticks (each tick is 1/18 seconds) reported to the destination network. Metric is the number of hops reported to the same network. Delay is used as the primary routing metric, and the metric (hop count) is used as a tie breaker.
via <i>network.node</i>	Address of a communication server that is the next hop to the remote network.
age	Amount of time, in hours, minutes, and seconds, that has elapsed since information about this network was last received.
uses	Number of times this network has been looked up in the route table. This field is incremented when a packet is process-switched, even if the packet is eventually filtered and not sent. As such, this field represents a fair estimate of the number of times a route gets used.
Ethernet0	Interface through which packets to the remote network will be sent.
(NOVELL-ETHER) (HDLC) (SAP) (SNAP)	Encapsulation (frame) type. This is shown only for directly connected networks.
is directly connected	Indicates that the network is directly connected to the communication server.

### Related Commands

**clear ipx route**  
**ipx maximum-paths**  
**ipx route**

## show ipx servers

To list the IPX servers discovered through SAP advertisements, use the **show ipx servers EXEC** command.

```
show ipx servers [sorted [name | net | type]]
```

### Syntax Description

<b>sorted</b>	(Optional) Sorts the display of IPX servers according to the keyword that follows.
<b>name</b>	(Optional) Displays the IPX servers alphabetically by server name.
<b>net</b>	(Optional) Displays the IPX servers numerically by network number.
<b>type</b>	(Optional) Displays the IPX servers numerically by SAP service type. This is the default.

### Default

**type**

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show ipx servers** command:

```
cs> show ipx servers
Codes: R - RIP, E - EIGRP, C - connected, S - static, 5 Total IPX routes

No parallel paths allowed          Novell routing algorithm variant in use

E Net 42 [345088/2] via 160.0000.0c00.8ea9, age 2d21 1 uses, Ethernet0
C Net FFFF01 (NOVELL-ETHER), is directly connected, 33 uses, Ethernet3
R Net E04 [120/2] via 165.0000.0c01.d879, 17 sec, 1 uses, Ethernet0
R Net E02 [120/2] via 165.0000.0c01.d879, 17 sec, 2 uses, Ethernet0
R Net E03 [120/2] via 165.0000.0c01.d879, 17 sec, 57 uses, Ethernet0
C Net CC00C1 (HDLC), is directly connected, 0 uses, serial3, (down)
```

Table 19-13 describes the fields shown in the display.

**Table 19-13 Show IPX Server Field Descriptions**

Field	Description
Codes	Codes defining how the route was learned.
R	Route learned from a RIP update.
E	Route learned via Enhanced IGRP.
C	Directly connected network.

Field	Description
S	Statically defined route via the <b>ipx route</b> command.
No parallel paths allowed	Maximum number of parallel paths for which the router has been configured with the <b>ipx maximum-paths</b> command.
Novell routing algorithm variant in use	Indicates whether the router is using the IPX-compliant routing algorithms (default).
sec	Number of seconds that have elapsed since information about this network was last received.
uses	Fair estimate of the number of times a route gets used. It actually indicates the number of times the route has been selected for use prior to operations such as access list filtering.
Ethernet0 (NOVELL-ETHER) (HDLC)	Possible interface through which updates to the remote network will be sent. Encapsulation type. This is shown only for directly connected networks.
is directly connected	Indicates that the network is directly connected to the router.
Net E04 [345088/2]	Network to which the route goes. The first number is the Enhanced IGRP metric, and the second number is the external hop count (for routes learned from Enhanced IGRP). The external hop count is the hop count that was advertised when this destination first entered the Enhanced IGRP autonomous system. If the external hop count is 0, this is an internal Enhanced IGRP route. When this destination is announced via RIP by this router, the router will add one to this value and use the result as the RIP hop count. The external hop count is not incremented when the destination is advertised via Enhanced IGRP.
[120/2]	Ticks/metric (for routes learned from RIP). Ticks are the number of IBM clock ticks (1/18th second). When <b>ipx algorithms</b> is enabled, hop count is used as a tie-breaker between routes if the tick count is the same. The metric is the IPX metric used in making routing decisions.
age	(For routes learned from Enhanced IGRP only.) Length of time since this route was first learned. The time is represented in the format <i>hh:mm:ss</i> until 24 hours have elapsed, after which it is represented <i>dd:hh</i> until one week has elapsed. After one week, the age is represented <i>wade</i> .
via <i>network.node</i>	Address of a router that is the next hop to the remote network.

### Related Command

**ipx sap**



## show ipx traffic

To display information about the number and type of IPX packets transmitted and received by the communication server, use the **show ipx traffic** user EXEC command.

### show ipx traffic

### Syntax Description

This command has no arguments or keywords.

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show ipx traffic** command:

```

cs> show ipx traffic
Rcvd: 644 total, 1705 format errors, 0 checksum errors, 0 bad hop count,
      0 packets pitched, 644 local destination, 0 multicast
Bcast: 589 received, 324 sent
Sent: 380 generated, 0 forwarded
      0 encapsulation failed, 4 no route
SAP: 1 SAP requests, 1 SAP replies
      61 SAP advertisements received, 120 sent
      0 SAP flash updates sent, 0 SAP poison sent
      0 SAP format errors
RIP: 0 RIP format errors
Echo: Rcvd 55 requests, 0 replies
      Sent 0 requests, 55 replies
      0 unknown, 0 SAPs throttled, freed NDB len 0
Watchdog:
      0 packets received, 0 replies spoofed
Queue lengths: IPX input: 0, SAP 0, RIP 0, GNS 0
               Total length for SAP throttling purposes: 0/(no preset limit)
EIGRP: Total received 0, sent 0
        Updates received 0, sent 0
        Queries received 0, sent 0
        Replies received 0, sent 0
        SAPs received 0, sent 0

```

Table 19-14 describes the fields that might possibly be shown in the display.

**Table 19-14 Show IPX Traffic Field Descriptions**

Field	Description
Rcvd:	Description of the packets the router has received.
644 total	Total number of packets the router has received.
1705 format errors	Number of bad packets discarded (for example, packets with a corrupted header).
0 checksum errors	Number of packets containing a checksum error. This should always be 0, because IPX does not use a checksum.
0 bad hop count	Number of packets discarded because their hop count exceeded 16 (that is, the packets timed out).
0 packets pitched	Number of times the router received its own broadcast packet.

Field	Description
644 local destination	Number of packets sent to the local broadcast address or specifically to the router.
0 multicast	Number of packets received that were addressed to multiple destinations.
Bcast:	Description of the broadcast packets the router has received and sent.
589 received	Number of broadcast packets received.
324 sent	Number of broadcast packets sent. It includes broadcast packets the router is either forwarding or has generated.
Sent:	Description of those packets that the router generated and then sent, and also those the router has received and then routed to other destinations.
380 generated	Number of packets the router transmitted that it generated itself.
0 forwarded	Number of packets the router transmitted that it forwarded from other sources.
0 encapsulation failed	Number of packets the router was unable to encapsulate.
4 no route	Number of times the router could not locate a route to the destination in the routing table.
SAP:	Description of the SAP packets the router has sent and received.
1 SAP requests	Number of SAP requests the router has received.
1 SAP replies	Number of SAP replies the router has sent in response to SAP requests.
61 SAP advertisements received	Number of SAP advertisements the router has received from another router.
120 sent	Number of SAP advertisements the router has generated and then sent.
0 SAP flash updates sent	Number of SAP advertisements the router has generated and then sent as a result of a change in its routing table.
0 SAP poison sent	Number of times the router has generated an update indicating that a service is no longer reachable.
0 SAP format errors	Number of SAP advertisements that were incorrectly formatted.
RIP:	Description of the RIP packets the router has sent and received.
0 RIP format errors	Number of RIP packets that were incorrectly formatted.
freed NDB length	Number of Network Descriptor Blocks (NDBs) that have been removed from the network but still need to be removed from the router's routing table.
Watchdog:	Description of the watchdog packets the router has handled.
0 packets received	Number of watchdog packets the router has received from IPX servers on the local network.
0 replies spoofed	Number of times the router has responded to a watchdog packet on behalf of the remote client.
Echo:	Description of the ping replies and requests the router has sent and received.
Rcvd 55 request 0 replies	Number of ping requests and replies received by the router.
Sent 0 requests, 55 replies	Number of ping requests and replies sent by the router.
0 unknown	Number of incomprehensible ping packets received by the router.
0 SAPs throttled	Number of ping packets discarded because they exceeded buffer capacity.
Queue lengths	Description of outgoing packets currently in buffers that are waiting to be processed.
IPX input	Number of incoming packets waiting to be processed.

---

Field	Description
SAP	Number of outgoing SAP packets waiting to be processed.
RIP	Number of outgoing RIP packets waiting to be processed.
GNS	Number of outgoing GNS packets waiting to be processed.
Total length for SAP throttling purposes	Maximum number of outgoing SAP packets allowed in the buffer. Any packets received beyond this number are discarded.
EIGRP:	Description of the Enhanced IGRP packets the router has sent and received.
Updates	Number of Enhanced IGRP updates the router has sent and received.
Queries	Number of Enhanced IGRP queries the router has sent and received.
Replies	Number of Enhanced IGRP replies the router has sent and received.
SAPs	Number of SAP packets the router has sent to and received from Enhanced IGRP neighbors.
unknown counter	Number of packets the router was unable to forward, for example, because of a misconfigured helper address or because no route was available.

