System i

Networking
TCP/IP troubleshooting

Version 5 Release 4

IBM

System i

Networking
TCP/IP troubleshooting

Version 5 Release 4

Note

Before using this information and the product it supports, read the information in “Notices,” on page 83.

Fifth Edition (February 2006)

This edition applies to version 5, release 4, modification 0 of IBM i5/OS (5722–SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

© Copyright International Business Machines Corporation 1997, 2006. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 1997, 2006

TCP/IP troubleshooting

The TCP/IP troubleshooting topic collection provides tools and techniques to help you solve problems with TCP/IP connectivity.

This topic is a centralized resource for finding answers to TCP/IP problems. You might have a general connectivity problem that is quickly identified or a more localized problem that requires in-depth consideration. Troubleshooting tools are provided in this topic to help you solve the problem.

Note: By using the code examples, you agree to the terms of the Code license and disclaimer information.

What's new for V5R4

This topic highlights changes to the TCP/IP troubleshooting topic for V5R4.

What's new for TCP/IP troubleshooting

Communication Trace Analyzer

The Communications Trace Analyzer allows you to analyze an i5/OS® Communications Trace, using either the Start Communications Trace (STRCMNTRC) command or the Trace Connection (TRCCNN) command. You can use this tool to troubleshoot various performance, connection, or security problems.

Note: The Communication Trace Analyzer can only be installed on systems running V5R2 or later.

How to see what's new or changed

To help you see where technical changes have been made, this information uses:

- The » image to mark where new or changed information begins.
- The « image to mark where new or changed information ends.

To find other information about what's new or changed this release, see the Memo to users.

Printable PDF

Use this to view and print a PDF of this information.

To view or download the PDF version of this document, select TCP/IP troubleshooting (about 1046 KB).

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF in your browser (right-click the link above).
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

© Copyright IBM Corp. 1997, 2006

Downloading Adobe Reader

| You need Adobe Reader installed on your system to view or print these PDFs. You can download a free
| copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Troubleshooting tools and techniques

The i5/OS operating system offers several tools and techniques for troubleshooting TCP/IP on your system and the network.

Use these tools and techniques to determine the most effective way to troubleshoot your TCP/IP problem.

Tools to verify your network structure

Using these tools, you can check basic network functions. For example, you can check the status of interfaces, routes, and connections, and you can determine whether the IP packets are reaching their destination.

Netstat

Netstat is a tool for managing and monitoring the status of your system's interfaces, routes, and connections, and it is useful for troubleshooting TCP/IP problems. You can use Netstat whether you are using IPv4 or IPv6 connectivity on the network.

To access Netstat, select a character-based interface or iSeries™ Navigator.

Related concepts

Internet Protocol version 6

Related tasks

"Starting interfaces" on page 73
Start the appropriate interfaces to ensure your network communication.

Using Netstat from a character-based interface:

You can use Netstat from a character-based interface to help you troubleshoot problems with TCP/IP connectivity.

From the character-based interface, use the Work with Network Status menu to work with the network status functions. You must have TCP/IP started on the system to use the menu options.

To start TCP/IP, type `STRTCP` on the command line, and press Enter.

To display the Work with Network Status menu, type `NETSTAT` or `WRKTCPSTS` on the command line, and press Enter.

Select one of these network components to begin troubleshooting.

Using Netstat from a character-based interface: Interfaces:

You might want to verify that the appropriate IPv4 or IPv6 interfaces are configured on your system and that they are active.

IPv4 interfaces

To display information about the IPv4 interfaces on your system, complete these steps:

1. Type `NETSTAT` or `WRKTCPSTS` on the command line to display the Work with Network Status menu, and then select option 1 on this menu.
2. You should have at least two active interfaces. Verify that these interfaces are active:
   - Loopback (127.0.0.1).
   - i5/OS IP address interface. This is the interface on your local system.
3. If these interfaces are not active, select option 9 (Start) to start the interfaces.

You might want to check the status of other interfaces. For example, if you are trying to ping interfaces on other hosts on the network, you should verify that those interfaces are active.

IPv6 interfaces

To display information about the IPv6 interfaces on your system, complete these steps:

1. Type `NETSTAT` or `WRKTCPSTS` on the command line to display the Work with Network Status menu, and then select option 4 on this menu.
2. You should have at least one active interface. Verify that this interface is active:
   - Loopback (::1)
3. If this interface is not active, select option 9 (Start) to start the interface.

You might want to check the status of other interfaces. For example, if you are trying to ping interfaces on other hosts on the network, you should verify that those interfaces are active.

Using Netstat from a character-based interface: Routes:

If you are trying to ping an interface address and do not receive a reply, you should verify that your routes are configured and available.

Your system needs routes to send packets to other systems or hosts. The route determines the path that a packet takes to its destination. To communicate between a local and remote network, whether you are using IPv4 or IPv6 connectivity, you should have at least these two types of routes configured on the system:

- A direct route (*DIRECT) allows packets to travel between interfaces on the local network. It is automatically configured and activated by the system for each interface.
- A default route (*DFTROUTE) allows packets to travel to hosts that are not directly connected to your network. It provides a path for the packets to take. A default route identifies a specific node as a next hop to which the packets travel and then continue their trip to their final destination on a different network. The packets take the default route whenever there is no other (more specific) route matching the destination IP address.

Keep in mind that routes are unidirectional. Just because a packet from a client can get to your system does not mean that your system can send a packet to the client.

Verify that the appropriate IPv4 or IPv6 routes are configured on your system.

IPv4 routes

To display information about the IPv4 routes on your system, follow these steps:

1. Type `NETSTAT` or `WRKTCPSTS` on the command line to display the Work with Network Status menu, and then select option 2 on this menu.
2. Select option 5 (Display details) for details about a specific route.
3. If you do not have a default route configured, you should configure it now. To configure a default route, follow these steps:
   a. At the command line, type `CFGTCP` to access the Configure TCP/IP menu.

b. Select option 2 (Work with TCP/IP Routes).
c. Select option 1 (Add) to go to the Add TCP/IP Route (ADDTCPRTE) display.
d. For the *Route destination* prompt, specify `*DFTROUTE`.
e. For the *Subnet mask* prompt, specify `*NONE`.
f. For the *Next hop* prompt, specify the appropriate IP address.

As an alternative, you can configure a default route using the **New IPv4 Route** wizard in iSeries Navigator. See the routes information for iSeries Navigator for more information.

IPv6 routes

To display information about the IPv6 routes on your system, follow these steps:

1. Type `NETSTAT` or `WRKTCPSTS` on the command line to display the Work with Network Status menu, and then select option 5 on the menu.
2. Select option 5 (Display details) for details about a specific route.

For IPv6, the Internet Protocol automatically configures default routes for each interface on the system. However, if you prefer, you can use the **New IPv6 Route** wizard in iSeries Navigator to create new routes yourself. See the routes information for iSeries Navigator for more information.

Related concepts

"Using Netstat from iSeries Navigator: Routes" on page 6
If you are trying to ping an interface address and do not receive a reply, you should verify that your routes are configured and available.

Using Netstat from a character-based interface: Connections:

You need to verify the status of your IPv4 and IPv6 connections.

For both IPv4 and IPv6 connections, you should verify the following information:

- You should have at least one passive listening connection for each of the servers you need to use. A passive listening connection indicates that the connection is ready for work. Passive listening connections are indicated by an asterisk in the Remote Address and Remote Port columns. See the server table for a list of all the servers and their associated jobs and subsystems.
- The passive listening connections should not be ended. If they have been ended, then remote systems are unable to use the servers represented by the connections.
- You can verify the status for jobs associated with a connection. This allows you to work with a job that might be impacting the connection.

IPv4 connection status

To display information about the status of your IPv4 connections, follow these steps:

1. Type `NETSTAT` or `WRKTCPSTS` on the command line to display the Work with Network Status menu, and then select option 3 on this menu.
2. If you need to end and restart the passive listening connection, you should do it by ending and restarting the server. At the command line, type `ENDTCPSVR *myserver` (where *myserver* is the server you want to end) and `STRTCPSVR *myserver`. If you are ending and restarting a host server, type `ENDHOSTSVR *myserver` (where *myserver* is the server you want to end) and `STRHOSTSVR *myserver`. See the server table to find out how to start and end various servers.

IPv6 connection status

To display information about the status of your IPv6 connections, follow these steps:

1. Type `NETSTAT` or `WRKTCPSTS` on the command line to display the Work with Network Status menu, and then select option 6 on this menu.
2. If you need to end and restart the passive listening connection, you should do it by ending and restarting the server. At the command line, type `ENDTCPSVR *myserver` (where *myserver* is the server you want to end) and `STRTCPSVR *myserver`. See the server table to find out how to start and end various servers.

   Related reference

   "Server table" on page 35
   You can use this server table as a reference to find out how servers, server jobs, job descriptions, and subsystems are mapped to one another.

Using Netstat from iSeries Navigator:

You can use the network status functions (known as Netstat in the character-based interface) in iSeries Navigator to troubleshoot TCP/IP problems.

iSeries Navigator is a graphical user interface that provides dialog boxes and wizards to configure and manage TCP/IP. To use the network status functions in iSeries Navigator, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration**.
2. Expand **IPv4** to access the status of interfaces, routes, and connections for your IPv4 connectivity, or expand **IPv6** to access the status of interfaces, routes, connections, and the neighbor cache for your IPv6 connectivity.
3. Expand **Lines** to view a list of the physical lines that are used for TCP/IP.

Select one of these network components to begin troubleshooting.

Using Netstat from iSeries Navigator: Interfaces:

You might want to verify that the appropriate IPv4 or IPv6 interfaces are configured on your system and that they are active.

IPv4 interfaces

To display information about the IPv4 interfaces on your system, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv4**.
2. Select **Interfaces**.
3. You should have at least two active interfaces. Verify that these interfaces are active:
   - Loopback (127.0.0.1).
   - i5/OS IP address interface. This is the interface on your local system.
4. If these interfaces are not active, right-click the IP address of the interface you want to start, and select **Start**.

You might want to check the status of other interfaces. For example, if you are trying to ping interfaces on other hosts on the network, you should verify that those interfaces are active.

IPv6 interfaces

To display information about the IPv6 interfaces on your system, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv6**.
2. Select **Interfaces**. You should have at least one active interface. Verify that this interface is active:
   - Loopback (::1)

3. If this interface is not active, right-click the IP address of the interface you want to start, and select **Start**.

You might want to check the status of other interfaces. For example, if you are trying to ping interfaces on other hosts on the network, you should verify that those interfaces are active.

Using Netstat from iSeries Navigator: Routes:

If you are trying to ping an interface address and do not receive a reply, you should verify that your routes are configured and available.

Your system needs routes to send packets to other systems or hosts. The route determines the path that a packet takes to its destination. To communicate between a local and remote network, whether you are using IPv4 or IPv6 connectivity, you should have at least these two types of routes configured on the system:

- A direct route (*DIRECT) allows packets to travel between interfaces on the local network. It is automatically configured and activated by the system for each interface.
- A default route (*DFTROUTE) allows packets to travel to hosts that are not directly connected to your network. It provides a path for the packets to take. A default route identifies a specific node as a next hop to which the packets travel and then continue their trip to their final destination on a different network. The packets take the default route whenever there is no other (more specific) route matching the destination IP address.

Keep in mind that routes are unidirectional. Just because a packet from a client can get to your system does not mean that your system can send a packet to the client.

IPv4 routes

To display information about the IPv4 routes on your system, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv4**.
2. Select **Routes**.
3. Right-click the IP address in the Remote Network column of the route you want to see, and select **Properties**.
4. If you do not have a default route configured, you should configure it now. To configure an IPv4 default route, follow these steps:
   a. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv4**.
   b. Right-click **Routes** and select **New Route**.
   c. Follow the wizard's instructions to create a new default route.

IPv6 routes

To display information about the IPv6 routes on your system, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv6**.
2. Select **Routes**.
3. Right-click the IP address in the Destination Address column of the route you want to see, and select **Properties**.
4. For IPv6, the Internet Protocol automatically configures default routes for each interface on the system. However, if you prefer, you can use the **New IPv6 Route** wizard in iSeries Navigator to create new IPv6 routes yourself. To configure an IPv6 default route, follow these steps:
   a. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv6**.
   b. Right-click **Routes** and select **New Route**.
   c. Follow the wizard's instructions to create a new default route.

Related concepts

"Using Netstat from a character-based interface: Routes" on page 3
If you are trying to ping an interface address and do not receive a reply, you should verify that your routes are configured and available.

Using Netstat from iSeries Navigator: Connections:

You need to verify the status of your IPv4 and IPv6 connections.

For both IPv4 and IPv6 connections, you should verify the following information:

- You should have at least one passive listening connection for each of the servers you need to use. A passive listening connection indicates that the connection is ready for work. Passive listening connections are indicated by an asterisk in the Remote Address and Remote Port columns. See the server table for a list of all the servers and their associated jobs and subsystems.
- The passive listening connections should not be ended. If they have been ended, then remote systems are unable to use the servers represented by the connections.

IPv4 connection status

To display information about the status of your IPv4 connections, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv4**.
2. Select **Connections**.
3. If you need to end and restart the passive listening connection, then you should do it by ending and restarting the server. To end and restart a server, follow these steps:
   a. In iSeries Navigator, expand ***your system*** → **Network** → **Servers**.
   b. Select **TCP/IP** for TCP/IP servers or select **iSeries Access** for host servers, right-click the server you want to end and restart, and select **End**.
   c. Right-click the server you want to restart, and select **Start**.

IPv6 connection status

To display information about the status of your IPv6 connections, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv6**.
2. Select **Connections**.
3. If you need to end and restart the passive listening connection, then you should do it by ending and restarting the server. To end and restart a server, follow these steps:
   a. In iSeries Navigator, expand ***your system*** → **Network** → **Servers**.
   b. Select **TCP/IP**, right-click the server you want to end and restart, and select **End**.
   c. Right-click the server you want to restart, and select **Start**. IBM provides a limited number of TCP/IP applications that support IPv6.

Related reference

"Server table" on page 35
You can use this server table as a reference to find out how servers, server jobs, job descriptions, and subsystems are mapped to one another.

Ping

You can use the Packet Internet Groper (Ping) function to test IP-level connectivity between two TCP/IP-capable interfaces or systems.

The Ping function sends out a special IP packet to a specified host. If the destination host receives this special packet, it replies to you with a message which indicates to you that you can communicate with that host. You can use the Ping function in two different ways:

- Use the Ping function to test your local TCP/IP configuration. For example, after setting up TCP/IP on the system for the first time, you can use Ping to test the TCP/IP configuration.
- Use the Ping function to test your ability to communicate with other hosts on the local or remote networks.

Note: You can use the Ping function for both IPv4 and IPv6 connectivity.

Related concepts

Internet Protocol version 6

Using Ping from a character-based interface:

You can use the PING command from the character-based interface to test your TCP/IP connectivity.

For example, if you want to test whether your data is traveling from your system to an interface with IP address 10.5.5.1 and host name FIRSTHOST, type `PING '10.5.5.1'` or `PING` *`firsthost`* at the command line.

You can specify either the IP address or host name of the remote node you want to reach. Ping appends the local domain to a host name if a domain name is not specified or if a period (.) does not appear at the end of the specified host name.

A successful Ping operation means your packets are reaching the 10.5.5.1 interface. An unsuccessful Ping operation indicates there is a problem with the connectivity between your system and interface 10.5.5.1.

Pinging the loopback interface on your system:

To verify that your TCP/IP software is installed, started, and working properly, ping the loopback interface.

You can perform the test without being connected to a physical line or network.

i5/OS reserves the IP address 127.0.0.1, the host name LOOPBACK, and the line description value of *LOOPBACK for verifying the software. Similarly for IPv6, i5/OS reserves the IP address ::1 and the line description *LOOPBACK for this purpose. The IPv6 loopback interface does not have a corresponding host name because the local host tables do not currently support IPv6 addresses. However, you can use a Domain Name System (DNS) to store the IPv6 host name instead of using the local host table.

To ping the loopback interface on your system to troubleshoot the problem, follow these steps:

1. At the command line, type these commands:
   - For IPv4: `PING '127.0.0.1'` or `PING LOOPBACK`
   - For IPv6: `PING '::1'`

   See PING parameters to fine-tune the PING command to get the most accurate results. Prompt on the PING command by selecting F4 for complete details on the PING parameters.

2. Failures might indicate the following problems.

Problem	Recovery
The local host table does not have an entry for the IPv4 LOOPBACK host name and IP address of 127.0.0.1.	You need to add the entry to the host table. This is only relevant for IPv4 because host tables do not currently support IPv6. To verify the host table entries, follow these steps: 1. At the command line, type `CFGTCP` (Configure TCP/IP). 2. Select option 10 (Work with TCP/IP Host Table Entries). 3. Verify that the host table contains an entry for the LOOPBACK host name and IP address 127.0.0.1.
The loopback interface is not active.	To activate the loopback interface: 1. At the command line, type `NETSTAT`. 2. Select option 1 (Work with TCP/IP interface status) for IPv4 interfaces, or select option 4 (Work with IPv6 interface status) for IPv6 interfaces. 3. Scroll down to find the loopback interface (127.0.0.1 or ::1), and select option 9 (Start) from the Work with TCP/IP interface status menu.
TCP/IP has not been started.	To start TCP/IP, type `STRTCP` (Start TCP/IP) at the command line.

Related reference

"Common error messages" on page 14
When you use the PING command to verify the connection to another host in the network, TCP/IP can give you an error message. Use this table to identify common error messages and to determine what you should do to solve the problems.

"PING parameters" on page 15
With PING command parameters, you can adjust the way the PING command performs its test of connectivity.

Pinging your own system:

To test whether packets can reach interfaces on your local area network (LAN), ping the local interface.

For IPv4, this is the IP address of a manually configured interface. For IPv6, this is the IP address of an automatically configured interface or a manually configured interface. It is also useful to ping an interface that is beyond your local system but attached to the LAN.

To ping your own system to troubleshoot the problem, follow these steps:

1. At the command line, type these commands:
   - For IPv4: `PING 'nnn.nnn.nnn.nnn'` or `PING` *`hostname`*
   - For IPv6: `PING 'x:x:x:x:x:x:x:x'` or `PING` *`hostname`*

   PING parameters can be used to tune the PING command to get the most accurate results. Prompt on the PING command by selecting F4 for complete details on the PING parameters.
2. Failures might indicate the following problems.

Problem	Recovery
The TCP/IP stack has not been activated on your system.	At the command line, type `STRTCP` to start the stack.

Problem	Recovery
The local host table does not have an entry for the IPv4 host name and IP address.	You need to add the entry to the host table. This is only relevant for IPv4 because host tables do not currently support IPv6. To verify the host table entries, follow these steps: 1. At the command line, type `CFGTCP` (Configure TCP/IP). 2. Select option 10 (Work with TCP/IP Host Table Entries). 3. Verify that the host table contains an entry for the host name and IP address.
Your line description or local interface has not been properly configured.	The line should be varied on, and the interface should be started.
If you are using IPv6, the IPv6 stack has not been activated on your system.	You can start IPv6 by specifying *YES for the STRIP6 parameter on the STRTCP (Start TCP/IP) command. If TCP/IP has already been started, you will need to end and restart TCP/IP. At the command line, type ENDTCP (End TCP/IP) to end TCP/IP. To restart TCP/IP and the IPv6 stack, type STRTCP STRIP6(*YES) at the command line. **Note:** Through ending TCP/IP, you end all Telnet sessions and all TCP/IP servers that are running.

Related reference

"PING parameters" on page 15
With PING command parameters, you can adjust the way the PING command performs its test of connectivity.

Pinging the interface on a network not directly attached to your local network:

Ping a remote interface to test whether packets can leave your network and reach a remote system. Ping a remote Domain Name System (DNS) to make sure your system can resolve domain names.

1. At the command line, type these commands:
   - For IPv4: `PING 'nnn.nnn.nnn.nnn'` or `PING` *hostname*
   - For IPv6: `PING 'x:x:x:x:x:x:x:x'` or `PING` *hostname*

   See PING parameters to fine-tune the PING command to get the most accurate results. Prompt on the PING command by selecting F4 for complete details on the PING parameters.
2. Failures might indicate the following problems:
   - TCP/IP has not been started. To start TCP/IP, type `STRTCP` (Start TCP/IP) at the command line.
   - The remote system is not available.
   - A frame size problem. The frame size on the line description should be greater than or equal to the maximum transmission unit (MTU) of the interface.
   - A network, router, next hop, or bridge problem.
   - The default route is not configured on your system.
   - The remote system or intermediate firewall has ICMP Echo requests or replies disabled.
   - If you have multiple IP addresses and subnets, make sure that IP datagram forwarding is set to *YES.
   - If the interface you are trying to reach is configured to an Ethernet adapter, you might need to change the Ethernet standard in the Ethernet line description. Specify either the correct Ethernet standard or *ALL.

- A DNS or host name table problem. For example, if the Ping works for the interface's IP address but not the host or domain name, you need to check your host table or DNS entries.

Related reference

"PING parameters" on page 15
With PING command parameters, you can adjust the way the PING command performs its test of connectivity.

"Common error messages" on page 14
When you use the PING command to verify the connection to another host in the network, TCP/IP can give you an error message. Use this table to identify common error messages and to determine what you should do to solve the problems.

Using Ping from iSeries Navigator:

You can use Ping from iSeries Navigator to test your TCP/IP connectivity.

iSeries Navigator is a graphical user interface that provides dialog boxes and wizards to configure and manage TCP/IP.

To test TCP/IP connectivity by using the Ping utility in iSeries Navigator, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network**.
2. Right-click **TCP/IP Configuration**, and select **Utilities** → **Ping**.
3. Specify the IP address or host name. If you are testing by using the host name, you must select the Protocol for host names.
4. Click **Ping Now** to send the Ping. View the responses to your Ping in the list of results.

Pinging the loopback interface on your system:

To verify that your TCP/IP software is installed and working properly, ping the loopback interface.

You can perform the test without being connected to a physical line or network.

i5/OS reserves the IP address 127.0.0.1, the host name LOOPBACK, and the line description value of *LOOPBACK for verifying the software. Similarly for IPv6, i5/OS reserves the IP address ::1 and the line description *LOOPBACK for this purpose. The IPv6 loopback interface does not have a corresponding host name because the local host tables do not currently support IPv6 addresses. However, you can use a Domain Name System (DNS) to store the IPv6 host name instead of using the local host table.

To ping the loopback interface on your system to troubleshoot the problem, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network**.
2. Right-click **TCP/IP Configuration**, and select **Utilities** → **Ping**.
3. Specify the loopback interface IP address or host name. If you are testing by using the host name, you must select the Protocol for host names.
4. Click **Ping Now** to send the Ping. View the responses to your Ping in the list of results.

5. Failures might indicate these problems:

Problem	Recovery
The local host table does not have an entry for the LOOPBACK host name and IP address of 127.0.0.1.	Add the entry to the host table. This is only relevant for IPv4 because host tables do not currently support IPv6. To verify the host table entries, follow these steps: 1. In iSeries Navigator, expand ***your system*** → **Network**. 2. Right-click **TCP/IP Configuration** and select **Host Table**. 3. Verify that the host table contains an entry for the LOOPBACK host name and IP address 127.0.0.1.
The loopback interface is not active.	To activate the loopback interface: • For IPv4: 1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv4** → **Interfaces**. 2. In the right pane, right-click the loopback interface (127.0.0.1) and select **Start**. • For IPv6: 1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv6** → **Interfaces**. 2. In the right pane, right-click the loopback interface (::1) and select **Start**.
TCP/IP has not been started.	Start TCP/IP.

Related reference

"Common error messages" on page 14
When you use the PING command to verify the connection to another host in the network, TCP/IP can give you an error message. Use this table to identify common error messages and to determine what you should do to solve the problems.

Pinging your own system:

To test whether packets can reach interfaces on your local area network (LAN), ping the local interface.

For IPv4, this is the IP address of a manually configured interface. For IPv6, this is the IP address of an automatically or a manually configured interface.

To ping your system to troubleshoot the problem, follow these steps:
1. In iSeries Navigator, expand ***your system*** → **Network**.
2. Right-click **TCP/IP Configuration**, and select **Utilities** → **Ping**.
3. Specify the IP address or host name for one of the interfaces on your LAN. If you are testing by using the host name, you must select the Protocol for host names.
4. Click **Ping Now** to send the Ping. View the responses to your Ping in the list of results.
5. Failures might indicate the following problems:

Problem	Recovery
The TCP/IP stack has not been activated on your system.	At the command line, type STRTCP to start the stack.

Problem	Recovery
The local host table does not have an entry for the host name and IP address.	You need to add the entry to the host table. This is only relevant for IPv4 because host tables do not currently support IPv6. To verify the host table entries, follow these steps: 1. In iSeries Navigator, expand ***your system*** → **Network**. 2. Right-click **TCP/IP Configuration** and select **Host Table**. 3. Verify that the host table contains an entry for the host name and IP address.
Your line description or local interface has not been properly configured.	The line should be varied on, and the interface should be started.
If you are using IPv6, the IPv6 stack has not been activated on your system.	You can start IPv6 by specifying *YES for the STRIP6 parameter on the STRTCP (Start TCP/IP) command. If TCP/IP has already been started, you will need to end and restart TCP/IP. At the command line, type ENDTCP (End TCP/IP) to end TCP/IP. To restart TCP/IP and the IPv6 stack, type STRTCP STRIP6(*YES) at the command line. **Note:** Through ending TCP/IP, you end all Telnet sessions and all TCP/IP servers that are running.
If you are trying to ping an IPv6 address, the interface's lifetime might have expired.	Check the status of the interface. If the lifetime has expired, the interface will not be active.

Related reference

"Common error messages" on page 14
When you use the PING command to verify the connection to another host in the network, TCP/IP can give you an error message. Use this table to identify common error messages and to determine what you should do to solve the problems.

Pinging the interface on a network not directly attached to your local network:

To test whether packets can leave your network and reach a remote system, ping the remote interface.

Ping a remote Domain Name System (DNS) to make sure your system can resolve domain names.

To ping the interface to troubleshoot the problem, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network**.
2. Right-click **TCP/IP Configuration**, and select **Utilities** → **Ping**.
3. Specify the IP address or host name of a remote interface. If you are testing by using the host name, you must select the Protocol for host names.
4. Click **Ping Now** to send the Ping. View the responses to your Ping in the list of results.
5. Failures might indicate the following problems:
   - TCP/IP has not been started.
   - The remote system is not available.
   - A frame size problem. The frame size on the line description should be greater than or equal to the maximum transmission unit (MTU) of the interface.
   - A network, router, next hop, or bridge problem.
   - The default route is not configured on your system.
   - The remote system or intermediate firewall has ICMP Echo Requests or Replies disabled.
   - If you have multiple IP addresses and subnets, make sure that IP datagram forwarding is set to *YES.

- If the interface you are trying to reach is configured to an Ethernet adapter, you might need to change the Ethernet standard in the Ethernet line description. Specify either the correct Ethernet standard or *ALL.
- A DNS or host name table problem. For example, if the Ping works for the interface's IP address but not the host or domain name, you need to check your host table or DNS entries.

Related reference

"Common error messages"
When you use the PING command to verify the connection to another host in the network, TCP/IP can give you an error message. Use this table to identify common error messages and to determine what you should do to solve the problems.

Common error messages:

When you use the PING command to verify the connection to another host in the network, TCP/IP can give you an error message. Use this table to identify common error messages and to determine what you should do to solve the problems.

Error message	What you should do
Message ID TCP2670 `Not able to complete request. TCP/IP services are not available`	TCP/IP has not been started yet or has not completed starting. Use the NETSTAT command to see if TCP/IP is active.
Message ID TCP3423 `No TCP/IP service available`	• TCP/IP has not been started yet or has not completed starting. Use the NETSTAT command to see if TCP/IP is active. • All jobs might not be started in the QSYSWRK subsystem. Use the Work with Active Jobs (WRKACTJOB) command to verify that the QSYSWRK subsystem and related jobs are active. Specifically, the QTCPIP job must be active. If they are not active, look in the job log or system default output queue for any messages. **Note:** If you are using TCP/IP when the operating system is in restricted state, the QTCPIP job is not active.
Message ID TCP3409 `Not able to establish connection with remote host system`	Check your configured interfaces, their related line descriptions, and the TCP/IP routes.
Message ID TCP3213 `Cannot reach remote system`	TCP/IP cannot find a route to the requested destination. Check NETSTAT option 2 and verify that a *DFTROUTE or equivalent network route has been configured and is active.
Message ID TCP3206 `Remote host did not respond to VFYTCPCNN within 10 seconds for connection verification 1.`	• Your configuration is probably correct, but you do not get an answer back from the remote system. Ensure that the remote host is able to reach your system. Call the remote system operator and ask them to verify the connection to your system. • Check the host tables or remote name server (if you are using a name server) for both systems, and the TCP/IP interfaces and routes. The remote name server might not be able to serve you for some reason. • If you are using an Ethernet line, make sure you specified the correct Ethernet standard or *ALL.
Message ID TCP3202 `VFYTCPCNN: Unknown host xxxxxx` where xxxxxx is the host name.	The host name could not be resolved to an IP address, either using the host table or a name server. Check the local host table or the remote name servers (if you are using a name server) for the remote host's entry. Verify that you can reach the remote name server by issuing a Ping to the remote name server.

Related tasks

Configuring TCP/IP when the operating system is in restricted state

PING parameters:

With PING command parameters, you can adjust the way the PING command performs its test of connectivity.

The PING command includes various parameters, such as packet length and wait time for a response. The default wait time of 1 second allows the remote system enough time to respond in most networks. However, if the remote system is far away or if the network is busy, increasing the wait time parameter can improve the results.

It is recommended that the parameter values be left at their default values. Be aware that if you do change them, a combination of large packet length and short wait time might not give the network enough time to transmit and receive the response, and time-outs can occur. If the network is not given enough time to transmit and receive the response, it can appear that you do not have connectivity to a system when, in fact, you do.

Related tasks

"Pinging your own system" on page 9
To test whether packets can reach interfaces on your local area network (LAN), ping the local interface.

"Pinging the interface on a network not directly attached to your local network" on page 10
Ping a remote interface to test whether packets can leave your network and reach a remote system. Ping a remote Domain Name System (DNS) to make sure your system can resolve domain names.

Trace route

The trace route function allows you to trace the route of IP packets to a user-specified destination system so you can locate the connectivity problem.

The route can involve many different systems along the way. Each system along the route is referred to as a hop. You can trace all hops along the route or specify the starting and ending hops to be traced.

Trace route displays a list of routers between your local network and the destination node. Examine the list of routers that the trace encounters to locate the problem on the network. For example, if the trace stops at a particular router, the problem might lie with that router or somewhere on the network after that point.

Use trace route for both IPv4 and IPv6 connectivity.

Related concepts

Internet Protocol version 6

Using trace route from a character-based interface:

To use the trace route function from the character-based interface, you can specify the destination system by system name or IP address. Either a valid IPv4 or IPv6 address is accepted.

Specify any of these examples at the command line:

- `TRACEROUTE` *`SYSNAME`*
- `TRACEROUTE '10.1.1.1'`
- `TRACEROUTE '2001:DB8::1'`

Using trace route from iSeries Navigator:

You can use trace route from iSeries Navigator to troubleshoot your connectivity problem.

To use trace route from iSeries Navigator, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network**.
2. Right-click **TCP/IP Configuration**, and select **Utilities** → **Trace Route**.
3. Specify the IP address or host name. If you are testing by using the host name, you must select the protocol for host names.
4. Click **Trace** to send the trace. View the list of routes that the trace discovered.

Tools for tracing data and jobs

You can use various trace tools to solve problems with your TCP/IP connectivity.

Communications trace

You can use communications trace to determine whether your data is being transmitted correctly across the network.

Use the communications trace function to troubleshoot TCP/IP. Communications trace is a service function that allows the data that flows over a communications line, such as a local area network (LAN) or wide area network (WAN), to be captured for analysis. The communications trace traces only the packets received by or sent from i5/OS. It does not trace other packets flowing on the network. After the data has been traced, the raw data can be dumped into a stream file, or it can be formatted and placed in a spooled file to be displayed or printed.

If your system is multihomed, data might be sent on one interface and received on a different interface. In this situation, you should trace two communications lines to see packets that are sent and received.

Communications trace can be used for troubleshooting both IPv4 and IPv6 communications.

Use communications trace in these situations:

- Your problem analysis procedures do not give enough information about the problem.
- You suspect that a protocol violation is the problem.
- You suspect that line noise is the problem.
- You want to know if your application is transmitting information correctly across the network.
- You want to know if you have performance problems with network congestion or data throughput.

To use the CL commands to perform a communications trace, you must have *SERVICE special authority, or you must be authorized to the service trace function of i5/OS through iSeries Navigator. See the chapter on user profiles in iSeries Security Reference for more information about this type of authority.

The trace connection function is an alternative method of getting a trace that is similar to a communications trace. See Trace connection for more information.

To use the communications trace function, perform these tasks.

Related concepts

"Trace connection" on page 24
You can trace encrypted data to locate the source of the problem by using trace connection. Trace connection is especially useful for connections, such as virtual Ethernet and OptiConnect, that do not support the general communications trace function.

Related tasks

"Job trace" on page 26
Use the job trace tool to trace data in any job to help identify your problem.

Related reference

iSeries Security Reference PDF

Planning a communications trace:

You need to prepare for the communications trace before using it to determine whether your data is being transmitted correctly across the network.

Before starting to work with a communications trace, perform these tasks:

1. Obtain the name of the line description associated with the TCP/IP interface with which you have the problem or which is used by the application or network with which you have a problem. Use NETSTAT *IFC to determine the name of the line description associated with the interface.
2. Ensure that the line is varied on and that the TCP/IP interface associated with the line has been started so that TCP/IP data can be sent and received over the interface and the line. Use NETSTAT *IFC to verify that the interface is active.

Performing a communications trace:

You can use CL commands in the character-based interface to perform a communications trace. If you want to start a new trace on the same line, you must first delete the existing communications trace.

Starting a communications trace:

This action starts a communications trace for the specified line or network interface description.

Note: A communications trace can no longer be used to trace data on a network server description (*NWS). Use the communications trace function to trace data on either a specific line (*LIN) or a network interface description (*NWI).

If your system is multihomed, data might be sent on one interface and received on a different interface. In this situation, you should trace two communications lines to see packets that are sent and received.

To start a communications trace, follow these steps:

1. Optional: To collect very large traces, you need to set the value for maximum storage size on the system. This value represents the amount of storage, in megabytes, that the communications trace function can allocate to contain the trace data from all traces run. This can only be done through the System Service Tools (SST) menu. To specify the value for maximum storage size, follow these steps:
   a. At the command line, type `STRSST` (Start System Service Tools).
   b. Type your Service Tools user ID and password.
   c. Select option 1 (Start a Service Tool).
   d. Select option 3 (Work with communications trace).
   e. Press F10 (Change size).
   f. For the *New maximum storage size* prompt, specify a sufficient amount of storage for the traces you collect, and press Enter.
   g. Press F3 (Exit) to exit System Service Tools.
2. At the command line, type `STRCMNTRC`.
3. For the *Configuration object* prompt, specify the name of the line, such as `TRNLINE`.
4. For the *Type* prompt, specify the type of resource, either `*LIN` or `*NWI`.
5. For the *Buffer size* prompt, specify a sufficient amount of storage for the anticipated volume of data. For most protocols, 8 MB is sufficient storage. For a 10/100 Ethernet connection, 16 MB through 1 GB is sufficient. If you are uncertain, specify `16 MB` for the maximum amount of storage allowed for the protocol.

6. For the *Communications trace options* prompt, specify `*RMTIPADR` if you want to limit the data collected to a trace of one remote interface. Otherwise, use the default value.
7. For the *Remote IP address* prompt, specify the IP address associated with the remote interface to which the trace data will be collected.

The communications trace continues until one of the following situation occurs:

- The ENDCMNTRC command is run.
- A physical line problem causes the trace to end.
- The *Trace full* prompt specifies `*STOPTRC` and the buffer becomes full.

Ending a communications trace:

To format and display the trace, you must first end the trace. This action ends the trace but saves the data in the communications trace buffer.

To end a communications trace, follow these steps:

1. At the command line, type `ENDCMNTRC`.
2. For the *Configuration object* prompt, specify the same line you specified when you started the trace, such as `TRNLINE`.
3. For the *Type* prompt, specify the type of resource, either `*LIN` or `*NWI`.

Dumping a communications trace:

Dumping the data to a stream file offers several advantages. Consider these advantages when deciding whether to use this function.

- You can run new traces without losing data from the existing trace.
- You can run an initial program load (IPL) on the system and still keep the raw trace data in the stream file.
- You can format trace data multiple times, even after you run an IPL or delete the prior trace buffer. If you do not dump the raw data to a stream file and you delete the trace or run an IPL on the system, you will not be able to format the trace again.
- You can use a custom formatter to analyze the trace data.

Note: If you are using Internet Protocol version 6 (IPv6), you must dump the trace data into a stream file by following these steps. However, if you are using IPv4, this is an optional part of the communications trace process.

To dump a communications trace, follow these steps:

1. Create a directory, such as `mydir`. See the CRTDIR (Create Directory) command description in the Control Language (CL) topic, to create a directory.
2. At the command line, type `DMPCMNTRC`.
3. For the *Configuration object* prompt, specify the same line you specified when you started the trace, such as `TRNLINE`.
4. For the *Type* prompt, specify the type of resource, either `*LIN` or `*NWI`.
5. For the *To stream file* prompt, specify the path name, such as `/mydir/mytraces/trace1`.

   Related reference

   Create Directory (CRTDIR) command

Printing a communications trace:

For IPv4, you can print a communications trace from the raw data you collected, or you can print from a stream file in which you previously dumped the raw data. For IPv6, you can only print from a stream file.

This action writes the communications trace data for the specified line or network interface description to a spooled file or an output file.

Note: If you are using Enterprise Extender to run System Network Architecture (SNA) applications over Internet Protocol (IP) networks using High Performance Routing (HPR), specify the following additional parameters for the PRTCMNTRC command in the following procedures.

- For the Format SNA data only prompt, type `*Yes`.
- For the Format HPR over IP prompt, type `*Yes`.
- For the Format LDLC over IP prompt, type `*Yes`.

Printing from raw data collected

If you collected the raw data without dumping it, follow these steps to print the data:

1. At the command line, type `PRTCMNTRC`.
2. For the *Configuration object* prompt, specify the same line you specified when you started the trace, such as `TRNLINE`, and press Enter.
3. For the *Type* prompt, specify the type of resource, either `*LIN` or `*NWI`.
4. For the *Character code* prompt, specify either `*EBCDIC` or `*ASCII`. You should print the data twice, once specifying `*EBCDIC` and then specifying `*ASCII`.
5. For the *Format TCP/IP data* prompt, type `*YES`, and press Enter twice.
6. Perform steps 1 through 5 again, but specify the other character code.

Printing from a stream file

If you dumped the data to a stream file, follow these steps to print the data:

1. At the command line, type `PRTCMNTRC`.
2. For the *From stream file* prompt, specify the path name, such as `/mydir/mytraces/trace1`, and press Enter.
3. For the *Character code* prompt, specify `*EBCDIC` or `*ASCII`. You should print the data twice, once specifying `*EBCDIC` and then specifying `*ASCII`.
4. Perform steps 1 through 3 again, but specify the other character code.

Viewing the contents of a communications trace:

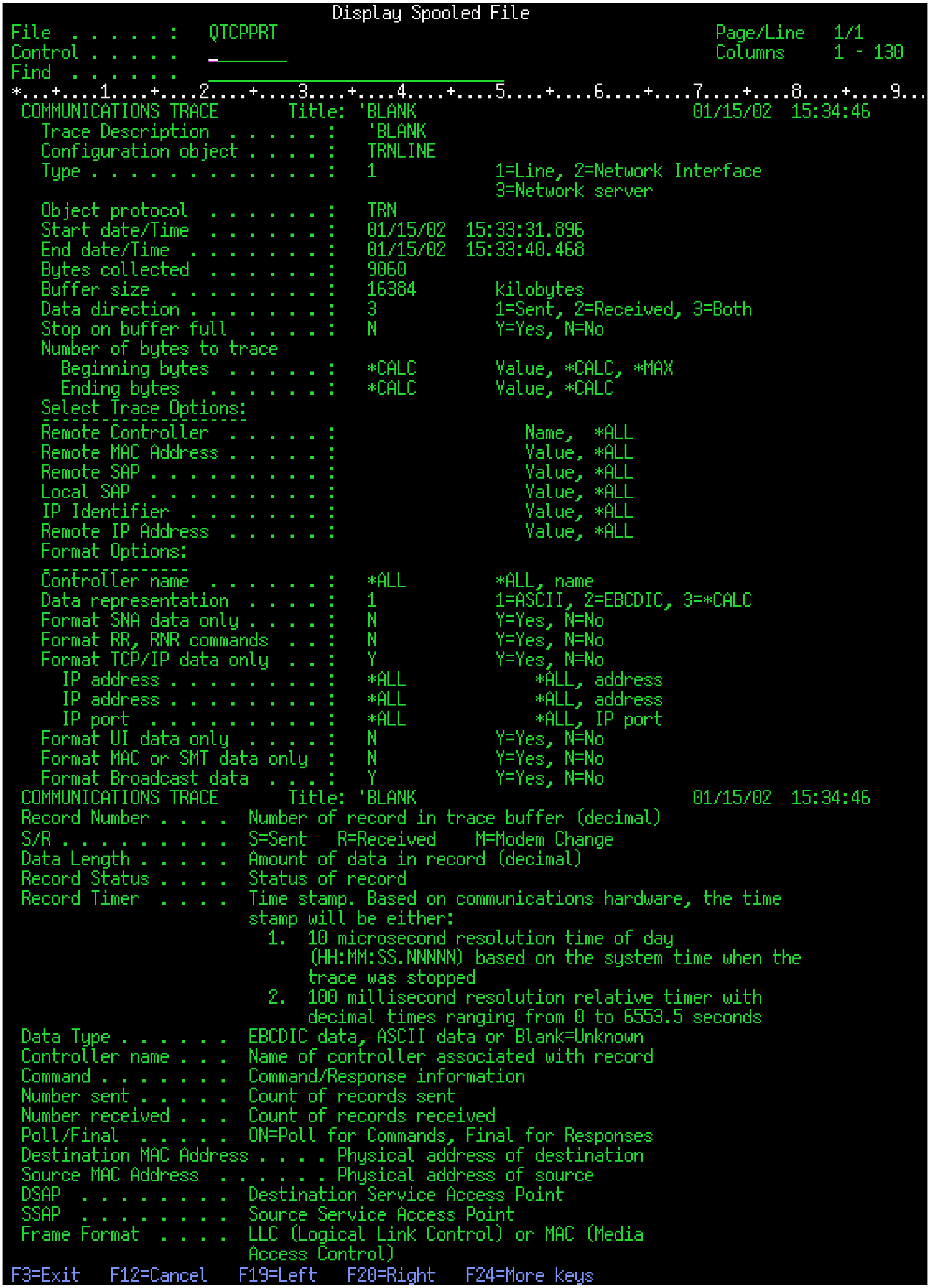
To view the contents of a communications trace, follow these steps.

1. At the command line, specify `WRKSPLF`.
2. On the **Work with Spooled File** dialog, press F11 (View 2) to view the date and time of the spooled file with which you want to work. If `More...` appears on the display and you need to continue searching for the spooled file, page forward or backward through the list of files; otherwise, continue with the next step.
3. Specify 5 in the Opt column next to the spooled file you want to display. The last files contain the most recent communications traces.
4. Verify that this is a communications trace for the line traced and that the times that the trace started and ended are correct.

Reading a communications trace:

The communications trace displays several types of information.

The first part of the communications trace summarizes the prompts that you specified when you started the trace, such as the name of the *Configuration object*. Page down to find a list of items, such as *Record Number* and *S/R*, with associated definitions. These items represent titles that are later used to identify sections of the communications trace data. It might be useful to refer back to this list as you read the trace data. This figure shows the preliminary information in a communications trace.



```
                                    Display Spooled File
File  . . . . . :   QTCPPRT                                                   Page/Line   1/1
Control . . . . .   _______                                                   Columns     1 - 130
Find  . . . . . .   ______________________________
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9...
 COMMUNICATIONS TRACE       Title: 'BLANK                                   01/15/02  15:34:46
   Trace Description  . . . . . :   'BLANK
   Configuration object . . . . :   TRNLINE
   Type . . . . . . . . . . . . :   1            1=Line, 2=Network Interface
                                                 3=Network server
   Object protocol  . . . . . . :   TRN
   Start date/Time  . . . . . . :   01/15/02  15:33:31.896
   End date/Time  . . . . . . . :   01/15/02  15:33:40.468
   Bytes collected  . . . . . . :   9060
   Buffer size  . . . . . . . . :   16384        kilobytes
   Data direction . . . . . . . :   3            1=Sent, 2=Received, 3=Both
   Stop on buffer full  . . . . :   N            Y=Yes, N=No
   Number of bytes to trace
     Beginning bytes  . . . . . :   *CALC        Value, *CALC, *MAX
     Ending bytes   . . . . . . :   *CALC        Value, *CALC
   Select Trace Options:
   ---------------------
   Remote Controller  . . . . . :                   Name,  *ALL
   Remote MAC Address . . . . . :                   Value, *ALL
   Remote SAP . . . . . . . . . :                   Value, *ALL
   Local SAP  . . . . . . . . . :                   Value, *ALL
   IP Identifier  . . . . . . . :                   Value, *ALL
   Remote IP Address  . . . . . :                   Value, *ALL
   Format Options:
   ---------------
   Controller name  . . . . . . :   *ALL         *ALL, name
   Data representation  . . . . :   1            1=ASCII, 2=EBCDIC, 3=*CALC
   Format SNA data only . . . . :   N            Y=Yes, N=No
   Format RR, RNR commands  . . :   N            Y=Yes, N=No
   Format TCP/IP data only  . . :   Y            Y=Yes, N=No
     IP address . . . . . . . . :   *ALL             *ALL, address
     IP address . . . . . . . . :   *ALL             *ALL, address
     IP port  . . . . . . . . . :   *ALL             *ALL, IP port
   Format UI data only  . . . . :   N            Y=Yes, N=No
   Format MAC or SMT data only  :   N            Y=Yes, N=No
   Format Broadcast data  . . . :   Y            Y=Yes, N=No
 COMMUNICATIONS TRACE       Title: 'BLANK                                   01/15/02  15:34:46
 Record Number  . . . .  Number of record in trace buffer (decimal)
 S/R  . . . . . . . . .  S=Sent    R=Received    M=Modem Change
 Data Length  . . . . .  Amount of data in record (decimal)
 Record Status  . . . .  Status of record
 Record Timer   . . . .  Time stamp. Based on communications hardware, the time
                         stamp will be either:
                           1.  10 microsecond resolution time of day
                               (HH:MM:SS.NNNNN) based on the system time when the
                               trace was stopped
                           2.  100 millisecond resolution relative timer with
                               decimal times ranging from 0 to 6553.5 seconds
 Data Type  . . . . . .  EBCDIC data, ASCII data or Blank=Unknown
 Controller name  . . .  Name of controller associated with record
 Command  . . . . . . .  Command/Response information
 Number sent  . . . . .  Count of records sent
 Number received  . . .  Count of records received
 Poll/Final   . . . . .  ON=Poll for Commands, Final for Responses
 Destination MAC Address . . . .  Physical address of destination
 Source MAC Address  . . . . . .  Physical address of source
 DSAP  . . . . . . . . . Destination Service Access Point
 SSAP  . . . . . . . . . Source Service Access Point
 Frame Format  . . . .   LLC (Logical Link Control) or MAC (Media
                         Access Control)
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
```

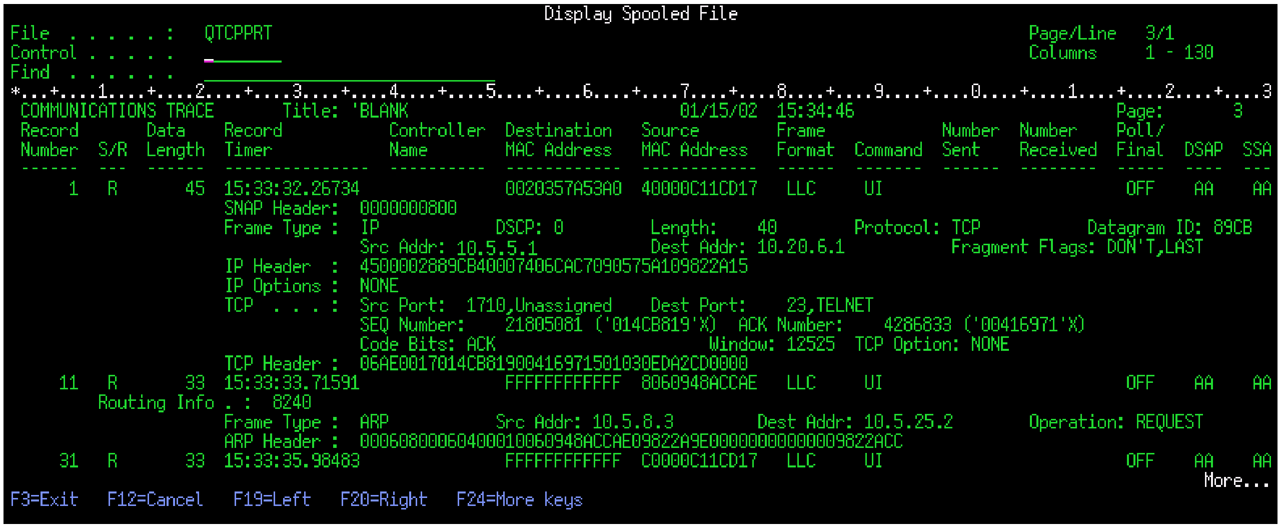
After reading the preliminary information, page down to the actual TCP/IP data in the communications trace. A row of titles, starting with *Record Number*, identifies each section of the data records. Each record number represents a frame, and it includes information that can help you debug the problem that you are having with TCP/IP on this system or in the associated network.

If you find an asterisk (*) after a record number, for example, `31*`, be aware that the asterisk represents missing trace data. This missing trace data appears when communications trace records are dropped. Communications trace data is collected by the input/output processor (IOP). If the communications line is very busy, the IOP prioritizes all the network traffic and gives a higher priority to the data path input/output than to the communications trace information. Under these circumstances, the IOP might drop some of the communications trace records. This can indicate that the IOP is not capable of handling the excessive speeds or traffic on the network.

If your communications trace is missing data, consider these options:

- Acknowledge that your communications line is busy and that frames will be missing from your communications trace.
- Investigate the traffic on the communications line to determine if there is traffic that can be moved to another line or TCP/IP interface.

This figure shows the TCP/IP data portion of the communications trace.



```
                                                       Display Spooled File
File  . . . . . :   QTCPPRT                                                                              Page/Line   3/1
Control . . . . .   ________                                                                             Columns     1 - 130
Find  . . . . . .   ______________________________
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0....+....1....+....2....+....3
 COMMUNICATIONS TRACE       Title: 'BLANK                            01/15/02  15:34:46                           Page:       3
 Record       Data    Record           Controller  Destination   Source        Frame            Number  Number    Poll/
 Number  S/R  Length  Timer            Name        MAC Address   MAC Address   Format  Command  Sent    Received  Final  DSAP  SSA
 ------  ---  ------  ---------------  ----------  ------------  ------------  ------  -------  ------  --------  -----  ----  ---
      1   R       45  15:33:32.26734               0020357A53A0  40000C11CD17   LLC     UI                         OFF    AA    AA
                      SNAP Header:  0000000800
                      Frame Type :  IP            DSCP: 0         Length:    40        Protocol: TCP           Datagram ID: 89CB
                                    Src Addr: 10.5.5.1            Dest Addr: 10.20.6.1           Fragment Flags: DON'T,LAST
                      IP Header  :  4500002889CB40007406CAC7090575A109822A15
                      IP Options :  NONE
                      TCP  . . . :  Src Port:  1710,Unassigned    Dest Port:    23,TELNET
                                    SEQ Number:    21805081 ('014CB819'X)  ACK Number:    4286833 ('00416971'X)
                                    Code Bits: ACK                      Window: 12525  TCP Option: NONE
                      TCP Header :  06AE0017014CB81900416971501030EDA2CD0000
     11   R       33  15:33:33.71591               FFFFFFFFFFFF  8060948ACCAE   LLC     UI                         OFF    AA    AA
         Routing Info . :  8240
                      Frame Type :  ARP           Src Addr: 10.5.8.3         Dest Addr: 10.5.25.2        Operation: REQUEST
                      ARP Header :  00060800060400010060948ACCAE09822A9E00000000000009822ACC
     31   R       33  15:33:35.98483               FFFFFFFFFFFF  C0000C11CD17   LLC     UI                         OFF    AA    AA
                                                                                                                           More...
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
```

Deleting a communications trace:

You must delete a communications trace before starting a new trace on the same line. The communications trace can be deleted after the trace has ended. This action deletes the communications trace buffer for the specified line or network interface description.

To delete a communications trace, follow these steps:

1. At the command line, type `DLTCMNTRC`.
2. For the *Configuration object* prompt, specify the name of the line, such as `TRNLINE`.
3. For the *Type* prompt, specify the type of resource, either `*LIN` or `*NWI`.

Tools for analyzing a communication trace:

The Communications Trace Analyzer is designed to analyze a communications trace using either the Start Communications Trace (STRCMNTRC) command or the Trace Connection (TRCCNN) command for various performance, connection, or security problems.

The Communications Trace Analyzer helps determine the type of communication problem that you encounter. It asks questions about the problem, the location of the trace, and analyzes the trace to show you where potential problems might exist and validates that they are, in fact, problems. For each problem it discovers, it provides a detailed explanation and offers resolution suggestions.

It also shows you the frames within the trace that provide the evidence for each problem. You can also use the analyzer to browse the trace by individual port pair conversations or other levels, either viewing summaries of each frame or the actual frames as they appear in the trace.

Note: The Communication Trace Analyzer can only be installed on systems running Operating System/400® (OS/400®) V5R2, i5/OS V5R3, or later.

Installing the Communication Trace Analyzer:

To install the Communication Trace Analyzer, complete these steps.

1. In iSeries Navigator, right-click **My Connections** → **Install Options** → **Install Plug-ins**.
2. Select the system from which you want to install the Communication Trace Analyzer.
3. Enter a valid user name and password for the system on which you want to install Communication Trace Analyzer.
4. From the list of plug-ins, select **Communication Trace Analyzer**.
5. Click **Next**.
6. Click **Finish**.

Starting the Communications Trace Analyzer:

To start the Communication Trace Analyzer, complete these steps.

1. In iSeries Navigator, select the system that has the Communication Trace Analyzer installed.
2. Right-click **Configuration and Service** .
3. Select **Tools** → **Communications Trace Analyzer**.

Additional communications trace functions:

The Check Communications Trace (CHKCMNTRC) command and Check Communications Trace (QSCCHKCT) API provide additional communications trace functions.

With the CHKCMNTRC command and the QSCCHKCT API, you can check the status of existing communications traces and programmatically check the storage space currently allocated for traces.

Checking a communications trace:

You might want to find out if communications traces currently exist on your system. Use the Check Communications Trace (CHKCMNTRC) command to return the communications trace status for a specific line or network interface description, or for all of the traces of a specific type that exist on the system. The status is returned to you in a message.

To check the status of a communications trace, follow these steps:

1. At the command line, type `CHKCMNTRC`.
2. For the *Configuration object* prompt, specify the name of the line, such as `TRNLINE`, or specify `*ALL` if you want to check the status of all traces for a specific type.

3. For the *Type* prompt, specify the type of resource, either `*LIN` or `*NWI`.

Programmatically checking storage space:

To programmatically check the maximum space allocated for traces and the sizes, in bytes, of all traces in active or stopped status on the system, use the Check Communication Trace (QSCCHKCT) API.

Related concepts

Application programming interfaces

Trace connection

You can trace encrypted data to locate the source of the problem by using trace connection. Trace connection is especially useful for connections, such as virtual Ethernet and OptiConnect, that do not support the general communications trace function.

The Trace Connection (TRCCNN) command is a service function that provides output similar to the general communications trace. The TRCCNN SET (*ON) TRCTYPE(*IP) SIZE(128000) traces data at the Licensed Internal Code TCP/IP layer.

Trace connection is useful for situations in which the general communications trace is not available or not effective. For example:

- You have TCP applications that use Secure Sockets Layer (SSL) or you use IP security. In either case, the data that flows over the communications line is encrypted. Therefore, the general communications trace might not be helpful if you need to see the data. Trace connection traces the data before encryption and after decryption and therefore, can be used when the general communications trace is not effective.
- You are using TCP/IP over a connection that does not support the general communications trace function, such as Loopback, OptiConnect, or Twinaxial. In this situation, you can use the trace connection as an alternative method for generating a trace.

To use the CL commands to perform a trace connection, you must have *SERVICE special authority, or be authorized to the Service Trace function of i5/OS through iSeries Navigator. See the chapter on user profiles in iSeries Security Reference for more information about this type of authority.

Related concepts

"Trace connection"
You can trace encrypted data to locate the source of the problem by using trace connection. Trace connection is especially useful for connections, such as virtual Ethernet and OptiConnect, that do not support the general communications trace function.

"Trace TCP/IP application" on page 25
Use the Trace TCP/IP Application (TRCTCPAPP) command to trace data that pertains to specific TCP/IP application servers.

Related tasks

"Job trace" on page 26
Use the job trace tool to trace data in any job to help identify your problem.

"Communications trace" on page 16
You can use communications trace to determine whether your data is being transmitted correctly across the network.

Related reference

iSeries Security Reference PDF

Trace Connection (TRCCNN) command

Trace TCP/IP application

Use the Trace TCP/IP Application (TRCTCPAPP) command to trace data that pertains to specific TCP/IP application servers.

This function is typically used at the request of your service provider. For troubleshooting information related to some of these specific application servers, see Troubleshooting problems related to specific applications.

TRCTCPAPP is supported by these applications:

- Certificate Services server
- Directory Services server
- Distributed data management (DDM with DRDA®) running over TCP/IP
- File Transfer Protocol (FTP)
- Host servers
  - Central server
  - Database server
  - Data queue server
  - Network print server
  - Remote command server
  - Server mapper
  - Signon server
- HTTP server (Apache)
- Layer Two Tunneling Protocol (L2TP)
- Packet rules
- Point-to-Point Protocol (PPP)
- Quality of Service (QoS)
- Simple Mail Transfer Protocol (SMTP) client and server
- Simple Network Time Protocol (SNTP) client and server
- Telnet
- Virtual private network (VPN) server
- Virtual terminal APIs

To use the CL commands to perform this type of trace, you must have *SERVICE special authority, or be authorized to the service trace function of i5/OS through iSeries Navigator. See the chapter on user profiles in *iSeries Security Reference* for more information about this type of authority.

Related concepts

"Trace connection" on page 24
You can trace encrypted data to locate the source of the problem by using trace connection. Trace connection is especially useful for connections, such as virtual Ethernet and OptiConnect, that do not support the general communications trace function.

Related tasks

"Job trace" on page 26
Use the job trace tool to trace data in any job to help identify your problem.

Related reference

iSeries Security Reference PDF

Trace TCP/IP Application (TRCTCPAPP) command

Job trace

Use the job trace tool to trace data in any job to help identify your problem.

Job trace is a problem analysis tool that allows you to look at what any application is doing. Use job trace as a first step in locating a problem with an application. You can turn on job trace in any job and see the call and return flows of that application. Job trace records the raw data and then stores it in a set of database files.

The job trace is performed using a series of CL commands, such as STRTRC (Start Trace), ENDTRC (End Trace), and PRTTRC (Print Trace). Starting a job trace consumes relatively few system resources. However, ending the job trace and printing the job trace require more time and processing resources. If you have a limited amount of interactive capability available on your system, you might want to submit the ENDTRC and PRTTRC to batch.

Note that if the application code is created with OPTIMIZE(40), the optimization disables call and instruction tracing. Although you can specify LICOPT (CallTracingAtHighOpt) to enable job call tracing, the optimization might still disable some calls. Therefore, job trace might not be effective when using OPTIMIZE(40).

Use job trace in these situations:

- You want to debug any job on your system. See the server table to understand the correlation between the servers and the applications and jobs they represent.
- You want to troubleshoot your sockets application.
- You are developing an application for i5/OS and encounter a problem. By tracing the application, you can identify the problem.

To use the CL commands to perform a job trace, you must have *SERVICE special authority, or be authorized to the service trace function of i5/OS through iSeries Navigator. See the chapter on user profiles in iSeries Security Reference for more information about this type of authority.

Use the following job trace instructions as a guide. The example demonstrates how to use job trace to troubleshoot a sockets application. Sockets adds information to the job trace output when errors are returned on the sockets APIs. You might need to specify different parameters, depending on the type of application you are troubleshooting. Note that communications trace is also useful when troubleshooting sockets applications.

Related concepts

“Trace connection” on page 24
You can trace encrypted data to locate the source of the problem by using trace connection. Trace connection is especially useful for connections, such as virtual Ethernet and OptiConnect, that do not support the general communications trace function.

“Trace TCP/IP application” on page 25
Use the Trace TCP/IP Application (TRCTCPAPP) command to trace data that pertains to specific TCP/IP application servers.

Related tasks

“Communications trace” on page 16
You can use communications trace to determine whether your data is being transmitted correctly across the network.

Related reference

iSeries Security Reference PDF

“Server table” on page 35
You can use this server table as a reference to find out how servers, server jobs, job descriptions, and subsystems are mapped to one another.

Starting a job trace:

This action starts a job trace for one or more jobs. You can start any number of trace sessions, but active trace session identifiers must be unique across the system.

Note: If you have not identified the job that needs to be traced, use the server table as a reference in identifying jobs and their corresponding servers.

To start a job trace, follow these steps:

1. At the command line, type `STRTRC` (Start Trace), and press F4 (Prompt).
2. For the *Session ID* prompt, specify a meaningful session identifier, such as *`mytrace`*. You will use this session identifier later, to specify the trace you want to end or print.
3. For the *Jobs* parameter, you need to specify values for these three prompts. Remember that you cannot specify the value *ALL for all three of these prompts. At least one of the prompts must contain a value other than *ALL.
   - For the *Jobs, Job Name* prompt, choose one of these options:
     - To trace only the job that issued the Start Trace (STRTRC) command, type `*`.
     - To trace a specific job, specify the name of the job you want to trace, such as *`job`*. You can specify up to ten jobs.
     - To trace a set of jobs that all begin with the same string, specify the job name in a manner such that it is not a specific job, such as *`job*`*. This traces all jobs that begin with the prefix JOB. See "Multiple generic traces" on page 29 to find out different ways to format a generic job trace.
     - To trace all the jobs, type `*ALL`. However, tracing all jobs is not recommended.
   - For the *Jobs, User* prompt, specify the name of the user of the job, such as `USER`. Other valid values include USER* and *ALL. However, tracing all users is not recommended.
   - For the *Jobs, Number* prompt, type `*ALL` or the job number. If you type `*ALL`, the *Job Name* prompt specification is considered a generic job name.
4. For the *Thread ID to include* prompt, type `*ALL` unless you want to trace a specific thread.
5. For the *Maximum storage to use* prompt, specify a value that you think will be large enough to collect the trace information you need. The amount of storage used for the trace buffer depends on how long the trace runs and how busy the job being traced is. The default value is 10000 KB (10 MB).
6. For the *Trace full* prompt, type `*WRAP` or `*STOPTRC`, depending on what you want to happen when the trace buffer gets full. If you want to collect trace information until the problem occurs, type `*WRAP`; the older trace information is overlaid with newer trace information when the buffer is full. If you do not want the trace information to be overlaid, type `*STOPTRC`.
7. For the *Trace type* prompt, type `*ALL` to store all of the job trace data.
8. For the *Trace type: Component* prompt, type `*SOCKETS`.
9. For the *Trace type: Trace level*, specify `*VERBOSE`.
10. For the *Trace filter* prompt, type `*NONE`. If you want to use a filter to collect specific information in the trace, specify the name of the trace filter, such as `tracefiltername`. If you have not already created a trace filter, do so by using the Add Trace Filter (ADDTRCFTR) command. The trace filter applies to the *FLOW trace only.
11. Press Enter. You should receive the message `STRTRC session ID MYTRACE successfully started.` If you meet problem with configuration or starting or ending servers, you can specify *TCPIPCFG, checking whether your configuration is right.

    Related reference

    "Server table" on page 35

    You can use this server table as a reference to find out how servers, server jobs, job descriptions, and subsystems are mapped to one another.

"Multiple generic traces" on page 29
The generic job trace allows you to trace jobs in several different ways. You can designate specific criteria to get precise results from the trace.

Re-creating the problem:

Re-create the problem by repeating the series of actions you previously took.

Ending a job trace:

This action ends the trace and stores the collected trace records in a set of database files. The stored trace records stay in the database files until you run the Delete Trace Data (DLTTRC) command.

To end a job trace, follow these steps:

1. At the command line, type `ENDTRC`, and press F4 (Prompt).
2. For the *Session ID* prompt, specify the name of the trace you want to end, such as *`mytrace`*.
3. For the *Data option* prompt, type `*LIB` to store the trace data in database files so it can be printed later.
4. For the *Data library* prompt, specify the name of the library in which the trace data will be stored, such as *`lib`*. The library must exist before running the ENDTRC command. If you do not specify a specific library, the default library QGPL is used.
5. Press Enter. You should receive the message `ENDTRC session ID MYTRACE successfully saved into library LIB`.

Note: The ENDTRC (End Trace) process can use a substantial amount of processing time and resources. If you have a limited amount of interactive capability available on your system, you might want to submit the ENDTRC to batch.

Printing a job trace:

This action formats and writes the stored trace records to a spooled output file or to a database output file.

To print a job trace, follow these steps:

1. At the command line, type `PRTTRC` (Print Trace), and press F4 (Prompt).
2. For the *Data member* prompt, type *`mytrace`*.
3. For the *Data library* prompt, type *`lib`*. This is the same library you specified under the ENDTRC command, and press Enter.
4. Programmatically process the trace information that was collected with the help of outfile support. This is most useful if you want to develop your own custom trace output formatter. The outfile parameter is used with the PRTTRC command.

Note: The PRTTRC (Print Trace) command can use a substantial amount of processing time and resources. If you have a limited amount of interactive capability available on your system, you might want to submit the PRTTRC to batch.

Related reference
Print Trace Data (PRTTRC) command

Deleting a job trace:

This action deletes the trace records that were stored in the database files as a result of the End Trace (ENDTRC) command.

To delete a job trace, follow these steps:

1. At the command line, type `DLTTRC` (Delete Trace Data), and press F4 (Prompt).
2. For the *Data member* prompt, type *`mytrace`*.
3. For the *Data library* prompt, type *`lib`*. This is the same library you specified under the ENDTRC command.
4. Press Enter. You should receive the message `Removing data member name MYTRACE from database files.`

Advanced job trace functions:

Job trace offers some advanced functions that enhance the results of the job trace.

Multiple generic traces:

The generic job trace allows you to trace jobs in several different ways. You can designate specific criteria to get precise results from the trace.

The generic job trace allows you to:

- Start an unlimited number of job traces. This allows you to trace more than one job at a time. This capability has additional considerations if you are using the TRCTYPE prompt for tracing additional components. See Trace type information is cumulative for more information.
- Start more than one trace session that has the generic job specification.

These examples show several different ways to specify a generic job name for your trace. These are all valid formats. Note that in all cases, the job number is *ALL:

- Generic job name, full user name: STRTRC SSNID(TEST) JOB((*ALL/USER/JOB*))
- Full job name, generic user name: STRTRC SSNID(TEST) JOB((*ALL/USER*/JOB))
- Full job name, full user name: STRTRC SSNID(TEST) JOB((*ALL/USER/JOB))
- Generic job name, generic user name: STRTRC SSNID(TEST) JOB((*ALL/USER*/JOB*))

  Related tasks

  "Starting a job trace" on page 27
  This action starts a job trace for one or more jobs. You can start any number of trace sessions, but active trace session identifiers must be unique across the system.

Trace type information is cumulative:

You can simultaneously run multiple traces on the same job and view cumulative output. In addition, you can view the results of all the traces within the output of each trace.

If you are tracing the same job in more than one trace session, and you are using the Trace type prompt, the Trace type component selections will accumulate and the results of all trace types will be included in the output of all of the trace sessions.

For example, you and your colleague both need to troubleshoot problems on the same Web server jobs. You start a job trace using the Start trace (STRTRC) command with these parameters: JOBTRCTYPE(*ALL) and TRCTYPE(*HTTP). Some time later, your colleague starts a trace using the STRTRC command with these parameters: JOBTRCTYPE(*ALL), and TRCTYPE(*SOCKETS).

Both traces contain the call and return flow for the period of time each trace is active. However, the additional TRCTYPE data that is collected is cumulative; that is, as new traces are started the trace types accumulate and the requested trace type information is collected until all traces are ended.

When your trace starts, it is collecting only the *HTTP trace type information. When your colleague's trace starts, your output and your colleague's output both contain the same type of information; they contain both the *HTTP trace type information and the *SOCKETS trace type information. Even if you end your trace shortly after your colleague begins a trace, your colleague's trace continues to collect both *HTTP and *SOCKETS trace type information until that trace ends.

Advanced trace function: Watch support

Watch support enhances the trace functions in i5/OS by automatically monitoring and ending traces when certain predetermined criteria are met. This prevents the loss of valuable trace data and reduces the amount of time you need to spend monitoring traces.

For example, when you start a trace on a busy system, it is possible for large amounts of trace data to be collected very quickly so that the trace buffer wraps, overlaying previous trace data. By the time you can manually determine the problem has occurred and stop the trace, the previous trace data needed to solve the problem has been overlaid. The result is lost trace data. The watch function solves this problem by allowing you to set certain watch criteria using the watch parameters. When a failure occurs, there is often a message or a Licensed Internal Code log that is generated at the time of the failure. You can specify which messages or Licensed Internal Code logs should be monitored during the trace collection, and when they occur the system automatically ends the trace.

Scenarios: Using watch support with traces:

You can enhance the i5/OS trace functions, such as communications trace and job trace, by using watch support.

Scenario: Using watch support with a communications trace:

Suppose that Telnet sessions are dropping intermittently on the system, but nothing else seems to be wrong. When the sessions drop, message TCP2617 is sent to the QSYS/QSYSOPR message queue. To solve the problem, you need to perform a communications trace using watch support.

With watch support, the trace is automatically stopped when the TCP2617 message is sent to QSYSOPR. This allows you to capture only the data that you need to analyze the problem and prevents the trace from running longer than necessary.

To perform the communications trace using watch support, follow these steps:

1. Start the communications trace:
   a. At the command line, type `STRCMNTRC` and press F4.
   b. For the *Configuration object* prompt, specify the name of the line, such as TRNLINE.
   c. For the *Type* prompt, specify the type of resource, such as *LIN.
   d. For the *Watch for message, Message identifier* prompt, type `TCP2617`.
   e. For the *Watched message queue, Message queue* prompt, type `*SYSOPR`. This ensures that the communications trace stops running when the TCP2617 message is sent to the QSYSOPR message queue.
   f. For the *Length of time to watch* prompt, type `2880`. The value 2880 indicates that the communications trace runs for a maximum of two days (2880 minutes) if the message does not occur. When two days elapse, the trace ends. If you do not want the trace to end if the message does not occur during the specified time, specify *NOMAX for this parameter.
2. Verify that the watch support started:
   a. At the command line, type `WRKWCH` and press F4.
   b. For the *Watch* prompt, type `*TRCCMD`. You should see QSCCMNxxxx session listed under Trace type. Note that CMN in the middle of the session identifier indicates that the watch session was started by the STRCMNTRC command. xxxx indicates a unique identifier for the watch session.
3. Verify that the watch support is running:
   - At the command line, type `WRKACTJOB SBS(QUSRWRK)`.

- You should see the watch job QSCCMNxxxx running in the QUSRWRK subsystem. The job is typically in DEQW status if the watched message has not been sent.

4. After the TCP2617 message is sent to the QSYS/QSYSOPR message queue, you should verify that the trace has ended:
   - At the command line, type `DSPMSG MSGQ(*SYSOPR)`.
   - You should see the CPI3999 message which indicates that the QSCCMNxxxx watch session associated with STRCMNTRC command was ended because of reason code 02. Reason code 02 indicates that `Watch for event criteria met because of message id TCP2617 found in QSYS/QSYSOPR`.
   - You can also verify that the watch session has ended using WRKWCH command as indicated in step 2.
5. Format the trace output using the Print Communications Trace (PRTCMNTRC) command to analyze the collected trace data. You might see that information is sent to the remote system but a response is not sent back. This indicates that the problem lies outside the local system.

Scenario: Using watch support with a job trace:

Assume that you write a sockets server application that occasionally fails. When the application fails, the TCP3B04 socket API error is sent to the job log. To solve the problem, you need to perform a job trace using watch support.

With watch support, the trace is automatically stopped when the TCP3B04 error is sent to the job log. This allows you to capture only the data that you need to analyze the problem and prevents the trace from running longer than necessary.

To perform the job trace using watch support, follow these steps:

1. Start the job trace:
   a. At the command line, type `STRTRC` and press F4.
   b. For the *Session ID* prompt, specify a meaningful session identifier, such as *mytrace*.
   c. For the *Jobs* parameter, specify these values:
      - For the *Jobs, Job Name* prompt, type the job name, such as `SOCKETAPP`.
      - For the *Jobs, User* prompt, specify the user ID, such as *user*.
      - For the *Jobs, Number* prompt, type `*ALL`.
   d. For the *Watch for message, Message identifier* prompt, type `TCP3B04`.
   e. For the *Watched message queue, Message queue* prompt, type `*JOBLOG`. This ensures that the job trace stops running when the TCP3B04 message is sent to the job log.
   f. For the *Watched job* parameter, specify these values:
      - For the *Jobs, Job Name* prompt, type `SOCKETAPP`.
      - For the *Jobs, User* prompt, specify the user ID, such as *user*.
      - For the *Jobs, Number* prompt, type `*ALL`.
2. Verify that the watch support started:
   a. At the command line, type `WRKWCH` and press F4.
   b. For the *Watch* prompt, type `*TRCCMD`. You should see QSCSTTxxxx session listed under Trace type. Note that STT in the middle of the session identifier indicates that the watch session was started by the STRTRC command. xxxx indicates a unique identifier for the watch session.
3. Verify that the watch support is running:
   - At the command line, type `WRKACTJOB SBS(QUSRWRK)`.
   - You should see the watch job QSCSTTxxxx running in the QUSRWRK subsystem. The job is usually in DEQW status if the watched message has not been sent.

4. After the TCP3B04 message is sent to the jobnumber/user/SOCKETAPP job log, you should verify that the trace has ended:
   - At the command line, type `DSPMSG MSGQ(*SYSOPR)`.
   - You should see the CPI3999 message which indicates that the QSCSTTxxxx watch session associated with STRTRC command was ended because of reason code 02. Reason code 02 indicates that `Watch for event criteria met because of message id TCP3B04 found in` *`jobnumber/user`*`/SOCKETAPP`.
   - You can also verify that the watch session has ended using WRKWCH command as indicated in step 2.
5. Format the trace output using the Print Trace (PRTTRC) command to analyze the data you collected.

Watch parameters:

Here are the parameters that you can use to specify watch criteria for watch support.

The trace commands have the following parameters to specify the watch criteria. See the trace commands help information for the descriptions of each parameter:

- Watch for message (WCHMSG)
- Watched message queue (WCHMSGQ)
- Watched job (WCHJOB)
- Watch for LIC log entry (WCHLICLOG)
- Length of time to watch (WCHTIMO)
- Trace program (TRCPGM)
- Time interval (TRCPGMITV)

Using watch exit programs:

You can specify exit programs in the trace program parameter to extend the capability of the watch function.

Click these links to learn how you can implement exit programs.

Example: Watch exit program:

This is sample code for a watch trace exit program. It is written in control language (CL).

Use this exit program as a starting point to help you create your own watch trace program. You can modify the code to allow the program to perform additional function. Using the watch exit program example, you can learn how to extend the capability of the watch function.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 81.

```
/*********************************************************/
/* THIS IS A SAMPLE CODE FOR WATCH FOR TRACE EVENT       */
/* FACILITY                                              */
/*                                                       */
/* FUNCTION: WHEN THE TRACE OPTION SETTING PARAMETER     */
/* INDICATES THAT A MESSAGE ID MATCHED WITH THE ONE BEING*/
/* WATCHED, THIS PROGRAM WILL PRINT THE HISTORY LOG AND  */
/* STOP THE TRACE COMMAND EXECUTION. OTHERWISE, THIS     */
/* WILL INDICATE TO CONTINUE WITH THE EXECUTION.         */
/*                                                       */
/* NOTE: MYLIB/MYOBJECT IS A DATA AREA THAT IS           */
/*   CONTINUOUSLY CHANGING DURING THE PROCESS. THE USER  */
/*   WANTS TO DUMP IT PERIODICALLY TO CHECK HOW ITS      */
/*   CONTENT IS CHANGING AND WHAT IS THE FINAL VALUE     */
/*   WHEN THE WATCHED MESSAGE OCCURS. THIS DATA AREA     */
/*   WILL BE DUMPED AT THE BEGINNING (*ON), WHEN THE     */
/*   INTERVAL TIME ELAPSES (*INTVAL), AND WHEN THE       */
/*   WATCHED MESSAGE OCCURS (*MSGID)                     */
/*                                                       */
/* THE FOLLOWING IS AN EXAMPLE OF THE WATCH FOR TRACE    */
/* EVENTS PARAMETERS, AS THEY WOULD BE SPECIFIED FOR A   */
/* TRACE COMMAND ISSUING THE CURRENT SAMPLE CODE:        */
/*                                                       */
/* WCHMSG((CPF0001)) TRCPGM(MYLIB/WCHEXTP) TRCPGMITV(30) */
/*********************************************************/
PGM PARM(&TRCOPTSET &RESERVED &OUTPUT &COMPDATA)
          DCL        VAR(&TRCOPTSET) TYPE(*CHAR) LEN(10) /* +
                       Reason why the program was called */
          DCL        VAR(&RESERVED) TYPE(*CHAR) LEN(10) /* This +
                       parameter is only used of TRCTCPAPP +
                       command and it is not relevant for Watch +
                       for Trace Event Facility */
          DCL        VAR(&OUTPUT) TYPE(*CHAR) LEN(10) /* +
                       Indicates if watch facility should stop +
                       or continue running */
          DCL        VAR(&COMPDATA) TYPE(*CHAR) LEN(92) /* Not +
                       needed for this sample */
/*********************************************************/
/*              BEGIN OF PROGRAM PROCESSING              */
/*********************************************************/
   IF         COND(&TRCOPTSET *EQ '*ON       ') THEN(DO) +
                       /* If the program was called at the +
                       beginning of the processing.          */
            /*  This section is usually used to set up +
                the environment before the trace starts */
          DMPOBJ     OBJ(MYLIB/MYOBJECT) OBJTYPE(*DTAARA) /* Dump +
                       Object for problem determination */
          CHGVAR     VAR(&OUTPUT) VALUE('*CONTINUE ') /* Let the +
                       trace to continue running */
   ENDDO      /* End if *ON */
   ELSE       CMD(IF COND(&TRCOPTSET *EQ '*MSGID    ') +
                       THEN(DO)) /* If the message id matched */
          DSPLOG     LOG(QHST) OUTPUT(*PRTSECLVL) /* Print the +
                       History Log */
          DMPOBJ     OBJ(MYLIB/MYOBJECT) OBJTYPE(*DTAARA) /* Dump +
                       object for problem determination */
          CHGVAR     VAR(&OUTPUT) VALUE('*STOP     ') /* +
                       Indicates Watch Facility to Stop */
   ENDDO      /* End if *MSGID */
   ELSE       CMD(IF COND(&TRCOPTSET *EQ '*INTVAL   ') +
                       THEN(DO)) /* If the exit program was +
                       called because the interval +
                       elapsed                                  */
            /*  This section is usually used to perform +
                tasks periodically. Like dumping objects, +
                checking conditions and optionally end +
                the watch facility      */
          DMPOBJ     OBJ(MYLIB/MYOBJECT) OBJTYPE(*DTAARA) /* Dump +
                       object for problem determination */
          CHGVAR     VAR(&OUTPUT) VALUE('*CONTINUE ') /* Let the +
                       trace and the watch facility to continue +
                       running */
   ENDDO      /* End if *INTVAL */
   ELSE       CMD(CHGVAR VAR(&OUTPUT) VALUE('*CONTINUE ')) +
                       /* Otherwise, watch facility will +
                       continue running */
ENDPGM
```

Related reference

"Example: Modifying the exit program" on page 34
Using the example exit program as a starting point, you can modify the code to allow the program to perform additional function.

Example: Modifying the exit program:

Using the example exit program as a starting point, you can modify the code to allow the program to perform additional function.

The following table provides suggestions for ways you can extend the capability of the watch function on your system by performing different actions based on the Trace option setting parameter in the exit program. Refer to each Trace option setting parameter value and the corresponding sample functions that can be performed.

Trace option setting parameter value	Sample functions that can be performed
*ON	• To set up the environment right before the trace starts. For example: – Start a process – Run commands – Change some special values • To register the status of the system right before the trace starts. For example: – Retrieve system values – Dump a job – Dump key objects for problem analysis • To verify that everything is ready for the trace and the watch function to start. For example: – Check certain system values – Check for the existence of key objects. If the exit program detects that something is not ready, the value *STOP for the Output parameter could be specified to prevent the trace command and the watch function from starting.
*MSGID or *LICLOG	• To register the final status of the system right after the event being watched for occurs. For example: – Retrieve system values – Dump a job – Dump key objects for problem analysis • To set the environment back to the initial status. For example: – End a process – Run commands – Change special values
*COMPDATA	The exit program can determine whether the trace and the watch function should stop or continue running. This is determined by returning *STOP or *CONTINUE for the Output parameter.
*INTVAL	• To perform activities periodically. For example, dump key objects for problem analysis. • To check conditions periodically. For example, check for the existence of key objects. The exit program can determine whether the trace and the watch function should stop or continue running. This is determined by returning *STOP or *CONTINUE for the Output parameter.

Trace option setting parameter value	Sample functions that can be performed
*WCHTIMO	To set the environment back to the initial status. For example: • End a process • Run commands • Change some special values

Related reference

"Example: Watch exit program" on page 32
This is sample code for a watch trace exit program. It is written in control language (CL).

Traces that use watch support:

You can use watch support with these trace functions.

- Communications trace
- Job trace
- Licensed Internal Code trace
- Trace connection
- Trace TCP/IP application

Troubleshooting tips

These troubleshooting tips enable you to solve basic TCP/IP problems.

You can find out how to check logs and verify that your interfaces and other network components are active.

If you are experiencing problems related to TCP/IP, you should check for error messages in the QTCP message queue located in library QUSRSYS. Many errors relating to TCP/IP functions get logged in this message queue. To display the QTCP messages, enter `DSPMSG QUSRSYS/QTCP` on the command line.

Server table

You can use this server table as a reference to find out how servers, server jobs, job descriptions, and subsystems are mapped to one another.

Use this table to locate a variety of information related to specific servers.

The first column provides the following information:

Server name:
The server name identifies the server. In most cases, this is the name of the server as it appears in iSeries Navigator.

To start:
The method that is used to start the server. Some servers are started by using CL commands, such as STRTCPSVR *DHCP. Other servers start when certain subsystems or jobs are started.

To end:
The method that is used to end the server. Some servers are ended by using CL commands, such as ENDTCPSVR *DHCP. Other servers end when certain subsystems are ended.

Product:
The name of the licensed product under which this server is shipped.

Server type:

The server type is a 30 byte character string that uniquely identifies the server to the system. All IBM-supplied servers have their server type begin with QIBM_. The server type is set by the server job using the Change Job API.

The subsequent columns provide the following information:

Job description:

The name and library of the job description that is used by this server's job to perform the work for this server. For example, QTCP/QTGSTELN indicates library QTCP and job description QTGSTELN.

Subsystem:

The name of the subsystem where this particular server runs.

Job name:

The name of the job(s) that are active for this server.

Shipped default value for *Autostart servers* **parameter:**

i5/OS is shipped to you with certain default values specified for the *Autostart servers* parameter for many of the servers. When the value is set to *YES, the server will start automatically when TCP/IP is started. When the value is set to *NO, the server will not start automatically when TCP/IP is started. If the server does not support the Autostart servers function, then no value is indicated for this parameter.

Note: To view or change the *Autostart servers* parameter, follow these steps:

- From the character based interface:

  Type `CHG`*`xxx`*`A` at the i5/OS command line, where *xxx* is the name of the server. For example, `CHGFTPA` to work with the attributes of the FTP server. The *Autostart servers* parameter appears at the top of the list of parameters.
- From iSeries Navigator:

  In iSeries Navigator, the equivalent of the *Autostart servers* parameter is indicated as one of the properties of a server, **Start when TCP/IP is started**.

  1. In iSeries Navigator, expand ***your system*** → **Network** → **Servers**.
  2. Click **TCP/IP**, **iSeries Access**, **DNS**, or **User-Defined**, depending on the type of server you want to view.
  3. In the right pane, right-click the server you want to view, such as FTP.
  4. On the **General** page, verify whether **Start when TCP/IP is started** is selected.

Default port:

The port from which the server job listens for client requests. Some of the ports indicate a service name within parentheses. This service name refers to the name that is defined in the Service Table Entries.

Note: To view the Service Table Entries display, type `WRKSRVTBLE` at the i5/OS command line.

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

ASFTomcat Basic Servlet and JSP Engine for the Apache Web Server
To Start:
`STRTCPSVR *ASFTOMCAT`
To End:
`ENDTCPSVR *ASFTOMCAT`
Product: 5722–DG1 *BASE option
Server Type: QIBM_ASFTOMCAT_*xxxxx* (where *xxxxx* is the name of the server instance)
Server Description: Is a standalone Web application servlet container. Through a socket connection, Web servers can use the various Web applications that an ASFTomcat server can provide.
	QHTTPSVR/QZTC	QSYSWRK	Name of the instance (user defined)	*NO	8009

Block I/O Daemon
To Start:
`STRNFSSVR *BIO`
To End:
`ENDNFSSVR *BIO`
Product: 5722–SS1
Server Type: QIBM_NFS_BIOD
Server Description: The Network File System client might use the Block I/O daemon to handle bulk I/O traffic.
	QSYS/QP0LBIOD	QSYSWRK	QNFSBIOD*	*NO	No port is used

BootP DHCP Relay Agent
To Start:
`STRTCPSVR *DHCP`
To End:
`ENDTCPSVR *DHCP`
Product: 5722–SS1
Server Type: QIBM_BOOTP_DHCP_RA
Server Description: Forwards Bootstrap Protocol (BootP) and Dynamic Host Configuration Protocols (DHCP) packets from the local system to one or more different DHCP servers.
	QSYS/QTODDJDS	QSYSWRK	QTODDHCPR	*NO	67 (dhcps)
					942

BootP Server
To Start:
`STRTCPSVR *BOOTP`
To End:
`ENDTCPSVR *BOOTP`
Product: 5722–SS1
Server Type: QIBM_BOOTP
Server Description: Provides a dynamic method for associating workstations with servers, or for assigning workstation IP addresses and initial program load (IPL) sources.
	QSYS/QTODBTPJ	QSYSWRK	QTBOOTP	*NO	67 (bootps)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
CCServer Agent **To Start:** STRMGDSYS **To End:** ENDMGDSYS **Product:** 5722–MG1 **Server Type:** QIBM_CCSERVER **Server Description:** Handles the distribution of integrated file system objects that are sent to the change control server.	QSYS/QSYSWRK	QSYSWRK	QCQNCMPS	Not applicable	No port is used
Central Server **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QSYS/QZSCSRVS), where *subsystem name* is QUSRWRK or the user-configured subsystem **To End:** Ends when the subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_CENTRAL	QSYS/QZBSJOBD	QUSRWRK or configurable	QZSCSRVS	*YES	No port is used
Central Server Daemon **To Start:** STRHOSTSVR *CENTRAL **To End:** ENDHOSTSVR *CENTRAL **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_CENTRAL	QSYS/QZBSJOBD	QSYSWRK	QZSCSRVSD	*YES	8470 (as-central) 9470 (as-central-s)
CIM Object Manager **To Start:** STRTCPSVR *CIMOM **To End:** ENDTCPSVR *CIMOM **Product:** 5722–SS1 **Server Type:** QIBM_CIMOM	QSYS/QYCMJOBD	QSYSWRK	QYCMCIMOM	N/A	5988 (wbem-http)
Cluster Resource Services **To Start:** Starts via QSYSWRK subsystem autostart entry **To End:** Ends when the QSYSWRK subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_CLUSTER_RESOURCE_SERVICES **Server Description:** Provides the set of services necessary to support a clustered environment. A cluster is a collection of one or more systems that work together to provide a single, unified computing capability.	QSYS/QCSTSRCD	QSYSWRK	QCSTSRCD	N/A	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
Cluster Resource Services **To Start:** Starts when the daemon QCSTCTSRCD job starts **To End:** Ends when the daemon QCSTCTSRCD job ends **Product:** 5722–SS1 **Server Type:** QIBM_CLUSTER_RESOURCE_SERVICES	QSYS/QCSTSRCD	QSYSWRK	QCSTCTRMCD QCSTCTCASD	N/A	657
Cluster Resource Services **To Start:** Starts when the daemon QCSTCTRMCD job starts **To End:** Ends when the daemon QCSTCTRMCD job ends **Product:** 5722–SS1 **Server Type:** QIBM_CLUSTER_RESOURCE_SERVICES	QSYS/QCSTSRCD	QSYSWRK	QSVRMSERMD QCSTHRMD QYUSCMCRMD QYUSALRMD	N/A	No port is used
Cluster Resource Services **To Start:** APIs: Start Cluster Node, Create Cluster Resource Group, Create Cluster, or Add Cluster Node Entry CL commands: STRCLUNOD, CRTCRG, CRTCLU, or ADDCLUNODE **To End:** End Cluster Node API or ENDCLUNOD CL command **Product:** 5722–SS1 **Server Type:** QIBM_CLUSTER_RESOURCE_SERVICES	QGPL/QDFTJOBD	QSYSWRK	QCSTCTL QCSTCRGM CRG-name	N/A	No port is used
Cluster Resource Services **To Start:** APIs: Start Cluster Node, Create Cluster Resource Group, Create Cluster, or Add Cluster Node Entry CL commands: STRCLUNOD, CRTCRG, CRTCLU, or ADDCLUNODE **To End:** End Cluster Node API or ENDCLUNOD CL command **Product:** 5722–SS1 **Server Type:** QIBM_CLUSTER_RESOURCE_SERVICES	QSYS/QCSTSRCD	QSYSWRK	QCSTCRGRM QCSTSAM QCSTCTCFRM	N/A	No port is used

Clustered Hash Table (CHT) Server
To Start:
`STRCHTSVR`
To End:
`ENDCHTSVR`
Product: 5722–SS1
Server Type: QIBM_CHT
Server Description: Enables applications to store and retrieve data that must be highly available across the cluster.
	QGPL/QDFTJOBD	QSYSWRK	Clustered Hash table (CHT) name	N/A	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
Collection Services Server **To Start:** Starts automatically when an application uses the QPMWKCOL function. **To End:** Ends when there are no application requests for data collection. **Product:** 5722–SS1 **Server Type:** QIBM_COLLECTION_SERVICES **Server Description:** Is a set of jobs that perform the system functions for collection services and real-time performance data collection.	QGPL/ QCOLJOBD	QSYSWRK	QPMASERV	N/A	No port is used
Collection Services Server **To Start:** Submitted by QYPSPFRCOL if configured (QYPSCSCA API or CHGPRFCOL CMD) **To End:** Ends automatically when data collection (QYPSPFRCOL) ends or current collection is cycled (restarted). **Product:** 5722–SS1 **Server Type:** QIBM_COLLECTION_SERVICES	QSYS/QYPSJOBD	QSYSWRK	CRTPFRDT	N/A	No port is used
Collection Services Server **To Start:** Starts by QPMASERV job **To End:** Ends when QPMASERV ends. **Product:** 5722–SS1 **Server Type:** QIBM_COLLECTION_SERVICES	QGPL/QCOLJOBD	QSYSWRK	QPMACLCT	N/A	No port is used
Collection Services Server **To Start:** QYPSSTRC API, GUI, or STRPRFCOL command. Can also be started by application requests for data. **To End:** QYPSENDC API, GUI, or ENDPFRCOL and if there are no active application data requests. **Product:** 5722–SS1 **Server Type:** QIBM_COLLECTION_SERVICES	QSYS/QYPSJOBD	QSYSWRK	QYPSPFRCOL	N/A	No port is used
Collection Services Server **To Start:** Starts when the QYPSPFRCOL job starts if user category is configured and collection is enabled **To End:** Ends automatically when data collection (QYPSPFRCOL job) ends or current collection is cycled (restarted). **Product:** 5722–SS1 **Server Type:** QIBM_COLLECTION_SERVICES	QGPL/QPMUSRCAT	QSYSWRK (default but depends on category owner JOBD)	Category name	N/A	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
Commerce Payments **To Start:** Product-specific commands **To End:** Product-specific commands **Product:** 5733-PYS **Server Type:** Not applicable	Subsystem of installed version of WebSphere®	QSYSWRK	User-specified name of the instance	N/A	Configurable
Connect FlowManager **To Start:** Starts using Connect Web Admin interface **To End:** Ends using Connect Web Admin interface **Product:** 5733–CO2 **Server Type:** QIBM_CONNECT_FM **Server Description:** This server takes extensible markup language (XML) request messages from the Connect Delivery Gateway. It routes those messages to a series of applications that process the request message and generates a response message.	Same as user profile	QCONNECT	QBEFMNTR QBEFSRVR	N/A	No ports are used

Content Manager for iSeries
To Start:
`STRTCPSVR`
To End:
`ENDTCPSVR`
Product: 5722–VI1 *BASE and 5722–VI1 Option 1
Server Type: None
	User-defined	QSERVER or user-defined	User-defined	*NO	User-defined

Controlled End TCP/IP Processing
To Start:
`STRTCP`
To End:
`ENDTCP`
Product: 5722–SS1
Server Type:
QIBM_TOC_ENDTCP_CONTROLED
Server Description: Ends the TCP/IP job in a controlled manner.
	QSYS/QTOCTCPIP	QSYSWRK	QTCPEND	N/A	No port is used

Customer Information Control System (CICS®) TCP/IP Server
To Start:
`STRCICS`
To End:
`ENDCICS`
Product: 5722–DFH
Server Type: QIBM_CICS
Server Description: Provides CICS support over TCP/IP.
	Specified in CICS's control region user profile	CICS's control region subsystem	AEGWPWKR and AEGWPSSN	N/A	1435 (ibm-cics)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
Database Server **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QSYS/QZDASOINIT), where *subsystem name* is QUSRWRK or the user-configured subsystem **To End:** Ends when the subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_DATABASE	QGPL/QDFTSVR	QUSRWRK or configurable	QZDASOINIT	*YES	No port is used
Database Server Daemon **To Start:** STRHOSTSVR *DATABASE (Requires QSERVER up) **To End:** ENDHOSTSVR *DATABASE **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_DATABASE	QSYS/QZBSJOBD	QSERVER	QZDASRVSD	*YES	8471 as-database 8478 as-transfer 9471 as-database-s
Database SSL Server **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QSYS/QZDASSINIT), where *subsystem name* is QUSRWRK or the user-configured subsystem **To End:** Ends when the subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_DATABASE	QGPL/QDFTSVR	QUSRWRK or configurable	QZDASSINIT	*YES	No port is used

Datalink File Manager
To Start:
`STRTCPSVR *DLFM`
To End:
`ENDTCPSVR *DLFM`
Product: 5722–SS1
Server Type: QIBM_DLFM
Server Description: Allows database files to contain references to objects that are not traditionally stored inside a database file. Those objects can be video clips or pictures, and are stored in the integrated file system. The references can link to the objects on the same system or on other systems.
	QGPL/QDFTJOBD	QSYSWRK	QZDFMCOD
			QZDFMCPD
			QZDFMDGD
			QZDFMGCD
			QZDFMRTD
			QZDFMSVR
			QZDFMUPD
			QZDFMCHD (A child server job that receives and processes DLFM requests as needed. Multiple instances of the QZDFMCHD job can run simultaneously.)
				*NO	20001 (dlfm)

Data Queue Server **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QSYS/QZHQSSRV), where *subsystem name* is QUSRWRK or the user-configured subsystem **To End:** Ends when the subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_DTAQ	QSYS/QZBSJOBD	QUSRWRK or configurable	QZHQSSRV	*YES	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

Data Queue Server Daemon
To Start:
`STRHOSTSVR *DTAQ`
To End:
`ENDHOSTSVR *DTAQ`
Product: 5722–SS1
Server Type:
QIBM_OS400_QZBS_SVR_DTAQ
	QSYS/QZBSJOBD	QSYSWRK	QZHQSRVD	*YES	8472
					(as-dtaq)
					9472
					(as-dtaq-s)

DB2® Text Extender Administration Server **To Start:** SBMJOB invoked by dessrvsp stored procedure **To End:** Ends automatically when task is complete. To interrupt abnormally, use ENDJOB. **Product:** 5722–DE1 Option 1 **Server Type:** QIBM_TEXT_EXTENDER_ADMIN **Server Description:** Controls all Text Extender administration user commands.	QGPL/QDFTJOBD	QSYSWRK	DESSRVBG	N/A	No port is used
DB2 Text Extender Daemon **To Start:** SBMJOB invoked by CALL PGM(QDB2TX/TXSTART) **To End:** CALL PGM(QDB2TX/TXSTOP) **Product:** 5722–DE1 Option 1 **Server Type:** QIBM_TEXT_EXTENDER_DAEMON **Server Description:** Controls the scheduling for automatic updates of the Text Extender.	QGPL/QDFTJOBD	QSYSWRK	DESDEM	N/A	No port is used
DB2 Text Extender Update Index Server **To Start:** SBMJOB invoked by desdem program **To End:** Ends automatically when task is complete. To interrupt abnormally, use ENDJOB. **Product:** 5722–DE1 Option 1 **Server Type:** QIBM_TEXT_EXTENDER_UPDATE **Server Description:** Maintains the Text Extender log tables and triggers index updates by scheduling documents that are referenced in the log tables.	QGPL/QDFTJOBD	QSYSWRK	DESXCTL	N/A	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

Dynamic Host Configuration Protocol (DHCP) server
To Start:
`STRTCPSVR *DHCP`
To End:
`ENDTCPSVR *DHCP`
Product: 5722–SS1
Server Type: QIBM_DHCP
Server Description: Passes configuration information to hosts on a TCP/IP network. DHCP enables client systems to get network configuration information, including an IP address, from a central DHCP server.
	QSYS/QTODDJDS	QSYSWRK	QTODDHCPS	*NO	67 (dhcps)
					942

Domain Name System (DNS) server **To Start:** `STRTCPSVR *DNS` **To End:** `STRTCPSVR *DNS` **Product:** 5722–SS1 Option 31 **Server Type:** QIBM_DNS **Server Description:** Maintains a database of domain (host) names and their corresponding IP addresses. It defines a mapping between the host name and the IP addresses in a centralized location. Systems within the TCP/IP network can use the lookup function of the DNS server to locate the IP for that system.	QDNS/QTOBJOBD	QSYSWRK	QTOBDNS (BIND 4) QTOBD*xxxxx* (BIND 8, *xxxxx* chosen by customer)	*NO	53 (domain)

Domino® server
To Start:
`STRTCPSVR *DOMINO`
or
`STRDOMSVR`
To End:
`ENDTCPSVR *DOMINO`
or
`ENDDOMSVR`
Product:
Domino 6.0.*x*: 5733–LD6
Domino 6.5.*x*: 5733–L65 or later
Server Type: QIBM_DOMINO
Server Description: Runs on multiple hardware and operating system platforms. Lotus® Domino includes e-mail, groupware, workflow, calendar and scheduling, and Web server functions.
	Same as subsystem	Notes® subsystem or configurable	Job names vary	*NO	Configurable (typically 1352)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
DRDA DDM Server TCP/IP **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QGPL/QRWTSRVR), where *subsystem name* is QUSRWRK or the user-configured subsystem **To End:** Ends when the subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QRW_SVR_DDM_DRDA **Server Description:** Allows a TCP/IP user on a remote client system to use SQL or native file I/O (DDM) to access the database on the i5/OS operating system. The DDM server allows applications or users to access remote databases.	QGPL/QDFTSVR	QUSRWRK or configurable	QRWTSRVR	*YES	No port is used
DRDA DDM Server TCP/IP Listener **To Start:** STRTCPSVR *DDM **To End:** ENDTCPSVR *DDM **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QRW_SVR_DDM_DRDA	jobd in QUSER profile (defaults to QGPL/QDFTJOBD)	QSYSWRK	QRWTLSTN	*YES	446 (drda) 447 (ddm) 448 (ddm-ssl)
Extended Dynamic Remote SQL **To Start:** STRTCPSVR *EDRSQL **To End:** ENDTCPSVR *EDRSQL **Product:** 5722–SS1 **Server Type:** QIBM_EDRSQL **Server Description:** Performs extended dynamic SQL and related functions on either a remote or local system. For more information, see APIs by category	QSYS/QXDAJOBD	QSYSWRK	QXDAEDRSQL	*NO	4402 (as-edrsql)
E-Z Setup Servers **To Start:** Starts via QSYSWRK subsystem autostart entry **To End:** Ends when QSYSWRK subsystem is ended **Product:** 5722–SS1 **Server Type:** QIBM_ALTCOMM	QSYS/QNEOJOBD	QSYSWRK	QNEOSOEM	N/A	No port is used
File Server Daemon and Server **To Start:** STRHOSTSVR *FILE (Requires QSERVER up) **To End:** ENDHOSTSVR *FILE **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_FILE	QSYS/QZBSJOBD	QSERVER	QPWFSERVSD	*YES	8473 (as-file) 8477 (as-netdrive) 9473 (as-file-s)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
File Server S2 **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QSYS/QPWFSERVS2), where *subsystem name* is QSERVER or the user-configured subsystem **To End:** ENDSBS QSERVER (or user configured subsystem) **Product:** 5722–SS1 **Server Type:** QIBM_NETDRIVE	QGPL/QDFTSVR	QSERVER or configurable	QPWFSERVS2	*YES	No port is used
File Server SO **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QSYS/QPWFSERVSO), where *subsystem name* is QSERVER or the user-configured subsystem **To End:** `ENDSBS QSERVER` (or user configured subsystem) **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_FILE	QGPL/QDFTSVR	QSERVER or configurable	QPWFSERVSO	*YES	No port is used
File Server SSL Server **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QSYS/QPWFSERVSS), where *subsystem name* is QSERVER or the user-configured subsystem **To End:** `ENDSBS QSERVER` (or user configured subsystem) **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_FILE	QGPL/QDFTSVR	QSERVER or configurable	QPWFSERVSS	*YES	No port is used

File Transfer Protocol (FTP) server
To Start:
`STRTCPSVR *FTP`
To End:
`ENDTCPSVR *FTP`
Product: 5722–TC1
Server Type: QIBM_FTP
Server Description: Transfers data between local and remote hosts. FTP consists of a client, from which FTP requests are issued, and the server, where client requests are processed.
	QUSRSYS/QTMFTPS	QSYSWRK or configurable	QTFTP*	*YES	21 (ftp-control)
					990 (ftps-control)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

Graphical Debug Server (Hub)
To Start:
`STRTCPSVR *DBG`
To End:
`ENDTCPSVR *DBG`
Product: 5722–SS1
Server Type: QIBM_DEBUG_SERVER
Server Description: Debugs i5/OS programs using a graphical debug user interface. The graphical debug user interface runs on your desktop and communicates with the Debug server using TCP/IP.
	QGPL/QDFTJOBD	QSYSWRK	QTESDBGHUB	*NO	4026 (as-debug)

Graphical Debug Server **To Start:** Started by the QTESDBGHUB server in the previous entry and is attached to a user interface **To End:** Ends when the user interface quits **Product:** 5722–SS1 **Server Type:** QIBM_DEBUG_SERVER	Job description that is indicated in the debug user profile	QUSRWRK	QTESDBGSVR	*NO	No port is used
Help Server **To Start:** Starts with the STRTCPSVR command, or the scripts in the Qshell environment (/QIBM/ProdData/OS400/Eclipse/EclipseStart), or from the HTTP Admin server interface. **To End:** Ends with the ENDTCPSVR command, or the scripts in the Qshell environment (/QIBM/ProdData/OS400/Eclipse/EclipseStop), or from the HTTP Admin server interface. **Product:** SS03 **Server Type:** HTTP/web application **Description:** Is an Eclipse-based Information Center that is used to deliver Help documentation.	QGPL/QDFTSVR	QSYSWRK	QIBMHELP	Yes	4111

HTTP Server
To Start:
`STRTCPSVR *HTTP`
To End:
`ENDTCPSVR *HTTP`
Product: 5722–DG1 *BASE option
Server Type: QIBM_HTTP_*xxxxx* (where *xxxxx* is the name of the server instance)
Server Description: Allows you to server multimedia objects, such as hypertext markup language (HTML) documents, to World Wide Web browser clients with your system.
	QHTTPSVR/
	QZHBHTTP
	QHTTPSVR/
	QZHBHTTP
		QHTTPSVR	Name of the instance (for example, ADMIN)	*NO	80 (www-http)
					2001 (as-admin-http)
					2010 (as-admin-https)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

IBM® Director
To Start:
Qshell script
`/qibm/userdata/ director/bin/twgstart`
To End:
Qshell script
`/qibm/userdata/ director/bin/twgend`
Product: 5722-DR1
Server Type:
QIBM_DIRECTOR_AGENT
QIBM_DIRECTOR
Server Description: Provides basic functions, such as discovery of the managed systems, storage of configuration and management data, an inventory database, event listening, security and authentication, management console support, and administrative tasks.
	QCPMGTDIR/
	QCPMGTDIR
	QCPMGTDIR
	QCPMGTDIR
		QSYSWRK	QCPMGTAGT
			QCPMGTSVR
				N/A	14247
					14248

IBM Directory Server
To Start:
`STRTCPSVR *DIRSRV`
To End:
`ENDTCPSVR *DIRSRV`
Product: 5722–SS1
Server Type: QIBM_DIRSRV_SERVER
Server Description: Is a Lightweight Directory Access Protocol (LDAP) server. The Directory server allows LDAP-enabled applications, such as mail applications that look up e-mail addresses, to store and retrieve information using the LDAP.
	QSYS/QDIRSRV	QSYSWRK	QDIRSRV	*YES	389 (ldap)
					636 (ldaps)

InfoPrint Server Font Downloader
To Start:
`STRFNTDWN`
To End:
`ENDFNTDWN`
Product: 5722–IP1
Server Type:
QIBM_IPS_FONTDOWNLOADER
Server Description: Listens on a TCP/IP port for Infoprint® Manager Double-Byte Character Set (DBCS) Font Downloader connections. After connecting, new or refreshed PostScript fonts can be sent to the system for use with Infoprint server. The font downloader job receives and installs these fonts.
	QGPL/QDFTJOBD	QUSRWRK	QXTFRNTDWN	N/A	8251

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
InfoPrint Server for iSeries Transform Job **To Start:** Spawned by transform manager **To End:** Ended by transform manager **Product:** 5722-IP1 **Server Type:** QIBM_IPS_TRANSFORM_JOB **Server Description:** Converts Adobe PDF 1.3 and PS Level 3 data streams to IBM Advanced Function Presentation™ (AFP™) data stream. This transform is indirectly called through the Image Print Transform function of i5/OS.	QGPL/QDFTJOBD	QUSRWRK	QADBDAEMON QXIODAEMON	N/A	No port is used
InfoPrint Server for iSeries Transform Manager **To Start:** STRTFMMGR **To End:** ENDTFMMGR **Product:** 5722-IP1 **Server Type:** QIBM_IPS_TRANSFORM_MGR **Server Description:** Manages heavyweight data stream transform jobs for InfoPrint Server/400-provided transforms.	QGPL/QDFTJOBD	QUSRWRK	QXTRTFMMGR	N/A	No port is used
Internet Daemon (INETD) Super Server **To Start:** STRTCPSVR *INETD **To End:** ENDTCPSVR *INETD **Product:** 5722–SS1 **Server Type:** QIBM_INETD **Server Description:** Listens for client requests for many different programs. Using INETD saves system resources by not requiring processes to be started and listing on ports for services that are not used often. When a client request is received, INETD generates a process to run the configured program to handle the request.	QSYS/QTOINETD	QSYSWRK	QTOGINTD	*NO	13 (daytime) 37 (time)
Internet PTF Delivery Server **To Start:** Starts on demand by iPTF process **To End:** Ends by the iPTF process **Product:** 5722–SS1 **Server Type:** QIBM_PTF **Server Description:** Allows you to order and download PTFs using the Internet.	Varies, based on user profile starting server	QSYSWRK	QESISRV	N/A	Dynamically assigned
iSeries Access for Web PDF Server **To Start:** Started by the iSeries Access for Web printer servlet support when any user needs to transform a spooled file to PDF using InfoPrint Server support. **To End:** Ends when you end the QIWAPDFSRV job. **Product:** iSeries Access for Web (5722–XH2) **Server Type:** QIBM_IWA_PDF_SVR	jobd in QUSER profile (defaults to QGPL/QDFTJOBD)	QSYSWRK	QIWAPDFSRV QJVACMDSRVA	N/A	8490 (as-iwapdfsrv)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

iSeries NetServer™ Daemon
To Start:
`STRTCPSVR *NETSVR`
To End:
`ENDTCPSVR *NETSVR`
Product: 5722–SS1
Server Type: QIBM_NETSERVER
	QSYS/QZLSSERVER	QSERVER	QZLSSERVER	*YES	137 TCP (netbios-ns)
					137 UDP (netbios-ns)
					138 UDP (netbios-dgm)
					139 TCP (netbios-ssn)
					445 TCP (cifs)

iSeries NetServer **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue STRPJ SBS(*subsystem name*) PGM(QSYS/QZLSFILE), where *subsystem name* is QSERVER or the user-configured subsystem **To End:** ENDSBS QSERVER (or user configured subsystem) **Product:** 5722–SS1 **Server Type:** QIBM_NETSERVER **Server Description:** Enables Microsoft® Windows® and Linux® Samba clients to access shared directory paths and shared output queues on the system. Clients on a network use the file and print-sharing functions for their operating systems.	QGPL/QDFTSVR	QSERVER or configurable	QZLSFILE	*YES	QNo port is used
Job Log Server **To Start:** Starts when QSYSWRK subsystem is started, or with the STRLOGSVR command. **To End:** Ends when the QSYSWRK subsystem ends, or with the ENDLOGSVR command. **Product:** 5722–SS1 **Server Type:** QIBM_JOBLOG_SERVER **Description:** Generates spooled job logs in the background.	QSYS/QJOBLOGSVR QSYS/QJOBLOGAJ	QSYSWRK	QJOBLOGSVR	N/A	No port is used
LDAP Publishing Agent **To Start:** Starts when QSYSWRK subsystem is started **To End:** Ends when QSYSWRK subsystem is ended **Product:** 5722–SS1 **Server Type:** QIBM_DIRSRV_PUB_AGENT **Server Description:** Publishes or stores information in a Directory Services (LDAP) server. Multiple jobs of this type can be running on a given system, each publishing a different type of information.	QSYS/QGLDPUBA	QSYSWRK	QGLDPUBA	N/A	No port is used
LDAP Publishing Engine **To Start:** Starts when QSYSWRK subsystem is started **To End:** Ends when QSYSWRK subsystem is ended **Product:** 5722–SS1 **Server Type:** QIBM_DIRSRV_PUB_ENGINE **Server Description:** Asynchronously processes LDAP publishing requests made with the QgldPubDirObj API.	QSYS/QGLDPUBE	QSYSWRK	QGLDPUBE	N/A	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
Licensed Internal Code 3494 TCP/IP Tape Server **To Start:** Started by the Licensed Internal Code when a 3494 tape library is varied on. **To End:** Ended by the Licensed Internal Code when the last 3494 tape library is varied off. **Product:** 5722–999 **Server Type:** QIBM_TASK_TCPIPTAPE	None	None	None	N/A	3494 (ibm3494)

Line Printer Daemon (LPD)
To Start:
`STRTCPSVR *LPD`
To End:
`ENDTCPSVR *LPD`
Product: 5722–TC1
Server Type: QIBM_LPD
Server Description: Receives files sent by the Line Printer Request (LPR). You can use the LPD server to receive spooled files from another system, or you can use the LPD server to receive print output from another system.
	QTCP/QTMPLPD	QSYSWRK	QTLPD*	*NO	515 (lpd)

Managed System Agent
To Start:
`STRMGDSYS`
To End:
`ENDMGDSYS`
Product: 5722-MG1
Server Type: QIBM_MANAGED_SYSTEM
Server Description: Monitors scheduled jobs and the control language (CL) input streams that are run as a result of activities received from the central site system.
	QSYS/QSYSWRK	QSYSWRK	QCQEPMON	N/A	No port is used

Management Central Agent **To Start:** Started by main Management Central Server as needed **To End:** Not applicable **Product:** 5722–SS1 **Server Type:** QIBM_MGMTCENTRAL_AGENT **Server Description:** Performs work for the Management Central server.	QSYS/QYPSJOBD	QSYSWRK	QYPSAPI QYPSPTF QYPSRMTCMD QYPSGETINV QYPSPRC QYPSUSRADM QYPSBDTSVR	*YES	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

Management Central Server
To Start:
`STRTCPSVR *MGTC`
To End:
`ENDTCPSVR *MGTC`
Product: 5722–SS1
Server Type: QIBM_MGMTCENTRAL
Server Description: Manages multiple systems from a single system in the TCP/IP network. You use this central system to manage the other systems (called endpoint systems) in your network. Once you add endpoint systems to your network, you only need to do your system administration tasks once. Your central system initiates your tasks and stores all Management Central data.
	QSYS/QYPSJOBD	QSYSWRK	QYPSJSRV	*YES	5544 (as-mgtcrlj)
					5555 (as-mgtctrl)
					5566 (as-mgtctrl-ss)
					5577 (as-mgtctrl-cs)

Mount Server
To Start:
`STRNFSSVR *MNT`
To End:
`ENDNFSSVR *MNT`
Product: 5722–SS1
Server Type: QIBM_NFS_MNTD
Server Description: Is a Remote Procedure Call (RPC) registered Network File System (NFS) service that handles mount and unmount requests for NFS clients.
	QSYS/QP0LMNTD	QSYSWRK	QNFSMNTD	*NO	No port is used

MQ Series Server
To Start:
`STRMQMLSR`
To End:
`ENDMQMLSR`
Product: 5724-B41
Server Type: QIBM_MQSERIES
Server Description: Provides the infrastructure for mission-critical communication between applications, either within an organization or business to business.
	QMQM/QMQMJOBD	QSYSWRK	RUNMQLSR	N/A	1414

Network Lock Manager
To Start:
`STRNFSSVR *NLM`
To End:
`ENDNFSSVR *NLM`
Product: 5722–SS1
Server Type: QIBM_NFS_NLMD
Server Description: Is a RPC-registered NFS service that provides byte-range locking for NFS files.
	QSYS/QP0LLCKD	QSYSWRK	QNFSNLMD	*NO	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
Network Print Server **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue `STRPJ SBS(`*`subsystem name`*`) PGM(QSYS/QNPSERVS)`, where *subsystem name* is QUSRWRK or the user-configured subsystem **To End:** Ends when the subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_NETPRT	QSYS/QZBSJOBD	QUSRWRK or configurable	QNPSERVS	*YES	No port is used

Network Print Server Daemon
To Start:
`STRHOSTSVR *NETPRT`
To End:
`ENDHOSTSVR *NETPRT`
Product: 5722–SS1
Server Type:
QIBM_OS400_QZBS_SVR_NETPRT
	QSYS/QZBSJOBD	QSYSWRK	QNPSERVD	*YES	8474 (as-netprt)
					8479 (as-vrtprint)
					9474 (as-netprt-s)

Network Station® Login Daemon
To Start:
`CALL QYTCV2/QYTCUSVR ('STRTCPSVR ')`
To End:
`CALL QYTCV2/ QYTCUSVR ('ENDTCPSVR ')`
Product: 5648–C07
Server Type: QIBM_NSLOGIN
Server Description: Allows IBM Network Stations and other applications that use remote authentication protocol (RAP) to authenticate on the i5/OS operating system.
	QYTCV2/ QYTCNSLD
	QYTCV2/ QYTCNSLD
		QSYSWRK	QYTCNSLD	*NO	256

Network Status Monitor (NSM)
To Start:
`STRNFSSVR *NSM`
To End:
`ENDNFSSVR *NSM`
Product: 5722–SS1
Server Type: QIBM_NFS_NSMD
Server Description: Provides applications with information about the status of network hosts. The Network Lock Manager (NLM) daemon uses the NSM to track network hosts that have locks.
	QSYS/QP0LSTATD	QSYSWRK	QNFSNSMD	*NO	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

NFS Server
To Start:
`STRNFSSVR *SVR`
To End:
`ENDNFSSVR *SVR`
Product: 5722–SS1
Server Type: QIBM_NFS_NFSD
Server Description: Stores files on a system and allows clients in the network to access and use the single set of files. NFS is typically used to share files among UNIX-type systems.
	QSYS/QP0LNFSD	QSYSWRK	QNFSNFSD*	*NO	2049

OnDemand Daemon
To Start:
`STRTCPSVR *ONDMD`
To End:
`ENDTCPSVR *ONDMD`
Product: 5722–RD1 Option 5
Server Type: QIBM_ON_DEMAND
	QRDARS/ QRDARS400
	QRDARS/ QRDARS400
		QSYSWRK	QRLGMGR	*YES	1445

OnDemand Common Server
To Start:
`STRTCPSVR *ONDMD`
To End:
`ENDTCPSVR *ONDMD`
Product: 5722–RD1 Option 10
Server Type: QIBM_ON_DEMAND
	QRDARS/QOND400	QSYSWRK	Instance name	*YES	1450

OnDemand Server
To Start:
`STRTCPSVR *ONDMD`
To End:
`ENDTCPSVR *ONDMD`
Product: 5722–RD1 Option 5
Server Type: QIBM_ON_DEMAND
Server Description: Lets you store large amounts of historical data on a disk, high-capacity optical volumes, or tape. It also provides quick access to stored data by online retrieval.
	QRDARS/ QRDARS400
	QRDARS/ QRDARS400
		QSYSWRK	QRLGSRV	*YES	1445

Open List Server **To Start:** Dynamically starts when needed **To End:** Ends when no longer needed **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QGYE_SVR **Server Description:** Handles the asynchronous building of lists by Open List APIs.	Varies	Variable (usually same as QZRCSRVS job)	QGYSERVER	N/A	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
Portable Applications Solutions Environment (PASE) syslog **To Start:** Starts by running /usr/sbin/syslogd in i5/OS PASE **To End:** ENDJOB CL command or the kill utility in i5/OS PASE **Product:** 5722-SS1 Option 33 **Server Type:** Not applicable	Varies (User can choose)	Varies (User can choose)	PGM-syslogdAlso user-defined	N/A	UDP 514 (syslog)

Post Office Protocol (POP)
To Start:
`STRTCPSVR *POP`
To End:
`ENDTCPSVR *POP`
Product: 5722–TC1
Server Type: QIBM_POP
Server Description: Allows the system to store e-mail for clients who use POP for their e-mail. E-mail is stored on the server until clients request it, at which time the mail is forwarded to the client and deleted from the server.
	QTCP/QTMMTPS	QSYSWRK	QTPOP*	*NO	110 (pop3)

QoS Policy Agent
To Start:
`STRTCPSVR *QOS`
To End:
`ENDTCPSVR *QOS`
Product: 5722–SS1
Server Type: QIBM_QOS
Server Description: Provides Network Quality of Service functions for the system. These services include: Differentiated Services that allow a user to specify special handling for TCP/IP connections and Integrated Services that allow applications using the RSVP internet protocol to request special handling for TCP/IP connections.
	QSYS/QTOQJOBDR	QSYSWRK	QTOQSRVR	*NO	No port is used

QoS RSVP Agent
To Start:
`STRTCPSVR *QOS`
To End:
`ENDTCPSVR *QOS`
Product: 5722–SS1
Server Type: QIBM_QOS
	QSYS/QTOQJOBDR	QSYSWRK	QTOQRAGENT	*NO	1698

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

QuickPlace® Server
To Start:
`STRTCPSVR *LQP`
or
`STRLQPSVR`
To End:
`ENDTCPSVR *LQP`
or
`ENDLQPSVR`
Product: 5733-LQP
Server Type: QIBM_QUICKPLACE
Server Description: Allows non-technical professionals to create, tailor, and administer an electronic shared workspace to support a project or initiative. With browser access to an intranet or the Internet, authorized team members can access the workspace to communicate, share ideas, maintain a project calendar, and organize team information.
	Same as subsystem	QPLACE00 or Notes subsystem	Configurable	*NO	Same as Domino HTTP task (typically 80)

Remote Command Agent
To Start:
`STRMGDSYS`
To End:
`ENDMGDSYS`
Product: 5722–MG1
Server Type: QIBM_REMOTE_COMMAND
Server Description: Accepts the remote commands from central site systems. From any location in your network, you can send commands to run on distributed systems that have Managed System Services installed.
	QSVMSS/QVARRCV	QSYSWRK	QVARRCV	N/A	No port is used

Remote Command Server **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue `STRPJ SBS(`*`subsystem name`*`) PGM(QSYS/QZRCSRVS)`, where *subsystem name* is QUSRWRK or the user-configured subsystem **To End:** Ends when the subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_RMTCMD	QSYS/QZBSJOBD	QUSRWRK or configurable	QZRCSRVS	*YES	No port is used

Remote Command Server Daemon
To Start:
`STRHOSTSVR *RMTCMD`
To End:
`ENDHOSTSVR *RMTCMD`
Product: 5722–SS1
Server Type:
QIBM_OS400_QZBS_SVR_RMTCMD
	QSYS/QZBSJOBD	QSYSWRK	QZRCSRVSD	*YES	8475 (as-rmtcmd)
					9475 (as-rmtcmd-s)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
Remote Execution (RExec) **To Start:** STRTCPSVR *REXEC **To End:** ENDTCPSVR *REXEC **Product:** 5722–TC1 **Server Type:** QIBM_REXEC Server Description: Allows a client user to send system commands to a remote system for processing. When RExec receives a client request, it first validates the user profile and password and then runs the requested command. The results are returned to the client.	QTCP/QTMXRXCS	QSYSWRK	QTRXC*	*NO	512 (exec)
RouteD **To Start:** STRTCPSVR *ROUTED **To End:** ENDTCPSVR *ROUTED **Product:** 5722–SS1 **Server Type:** QIBM_ROUTED **Server Description:** Provides dynamic routing. Dynamic routing deals with the ability to determine how to route traffic based upon a changing network topology.	QSYS/QTOROUTED	QSYSWRK	QTRTD*	*NO	UDP 520 (routed)
Remote Procedure Call (RPC) **To Start:** STRNFSSVR *RPC **To End:** ENDNFSSVR *RPC **Product:** 5722–SS1 **Server Type:** QIBM_NFS_RPCD **Server Description:** Runs Network File System daemons and other commands.	QSYS/QP0LRPCD	QSYSWRK	QNFSRPCD	*NO	111 (sunrpc)
Secure Shell daemon (SSHD) **To Start:** Starts by running /usr/sbin/sshd in i5/OS PASE. **To End:** Ends by using the ENDJOB command or the kill utility in i5/OS PASE. **Product:** 5733–SC1 **Server Type:** Not applicable. **Description:** Accepts incoming secure shell protocol (SSH) connections. SSH verifies the authenticity of the client and the server. All of the data is encrypted as it travels on the network.	Varies	Varies	PGM-sshd or user-defined name	Not applicable	22 (ssh)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

Server Port Mapper
To Start:
`STRHOSTSVR *SVRMAP`
To End:
`ENDHOSTSVR *SVRMAP`
Product: 5722–SS1
Server Type:
QIBM_OS400_QZBS_SVR_SVRMAP
Server Description: Allows the client to find the port of the particular service. The client sends in a request with the service name, and the port mapper looks up the service in the service table and returns the port number to the client.
	QSYS/QZBSJOBD	QSYSWRK	QZSOSMAPD	*YES	449 (as-svrmap)

Service Agent Hardware Problem Reporting **To Start:** Started by autostart job, or STRSRVAGT command **To End:** ENDSRVAGT command **Product:** 5722–SS1 **Server Type:** QIBM_SERVICE_AGENT_PRB	QSYS/QS9SRVAGT	QSYSWRK	QS9PRBMON QS9PALMON	N/A	No port is used
Service Agent Inventory Transmission **To Start:** Submitted from QYPSSRV **To End:** `ENDJOB` **Product:** 5722–SS1 **Server Type:** QIBM_SERVICE_AGENT_INV	QSYS/QSJINV	QSYSWRK	QYIVRIPS	N/A	No port is used

Signon Server Daemon
To Start:
`STRHOSTSVR *SIGNON`
To End:
`ENDHOSTSVR *SIGNON`
Product: 5722–SS1
Server Type: QIBM_OS400_QZBS_SVR_ SIGNON
	QSYS/QZBSJOBD	QSYSWRK	QZSOSGND	*YES	8476 (as-signon)
					9476 (as-signon-s)

Signon Server **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue `STRPJ SBS(`*`subsystem name`*`) PGM(QSYS/QZSOSIGN)`, where *subsystem name* is QUSRWRK or the user-configured subsystem **To End:** Ends when the subsystem ends **Product:** 5722–SS1 **Server Type:** QIBM_OS400_QZBS_SVR_ SIGNON	QSYS/QZBSJOBD	QUSRWRK or configurable	QZSOSIGN	*YES	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

Simple Network Time Protocol Service
To Start:
`STRTCPSVR *NTP`
To End:
`ENDTCPSVR *NTP`
Product: 5722–SS1
Server Type: QIBM_NTP
Server Description: Provides time synchronization services to other systems.
	QSYS/QTOTNTP	QSYSWRK	QTOTNTP	*NO	123 (ntp)

Simple Mail Transfer Protocol (SMTP) Bridge Client
To Start:
`STRTCPSVR *SMTP`
To End:
`ENDTCPSVR *SMTP`
Product: 5722–TC1
Server Type: QIBM_SMTP_BR_CLIENT
Server Description: Converts Systems Network Architecture distribution services (SNADS) outbound mail to simple SMTP mail for an SMTP client to send.
	QUSRSYS/ QTMSMTPS	QSYSWRK or configurable	QTSMTPBRCL	*YES	No port is used

SMTP Bridge Server
To Start:
`STRTCPSVR *SMTP`
To End:
`ENDTCPSVR *SMTP`
Product: 5722–TC1
Server Type: QIBM_SMTP_BR_SERVER
Server Description: Creates an mail server framework (MSF) message and stream file from the inbound mail received from the SMTP server.
	QUSRSYS/ QTMSMTPS	QSYSWRK or configurable	QTSMTPBRSR	*YES	No port is used

SMTP Client Daemon
To Start:
`STRTCPSVR *SMTP`
To End:
`ENDTCPSVR *SMTP`
Product: 5722–TC1
Server Type: QIBM_SMTP_CLIENT
	QUSRSYS/ QTMSMTPS	QSYSWRK or configurable	QTSMTPCLTD	*YES	No port is used

SMTP Client **To Start:** Starts when the client daemon job QTSMTPCLTD starts the client prestart jobs **To End:** Ends when the client daemon job QTSMTPCLTD ends the client prestart jobs **Product:** 5722–TC1 **Server Type:** QIBM_SMTP_CLIENT **Server Description:** Allows end-to-end delivery of mail from one mail server to another. A direct connection exists between the SMTP sender and the destination SMTP receiver. The client keeps the mail at the sender until it transmits and copies it.	QUSRSYS/ QTMSMTPS	QSYSWRK or configurable	QTMSCLCLTP	*YES	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

SMTP Mail Scheduler
To Start:
`STRTCPSVR *SMTP`
when configured
To End:
`ENDTCPSVR *SMTP`
Product: 5722–TC1
Server Type: QIBM_SMTP_MAIL_SCHED
Server Description: Sets the time intervals that you want the system to connect to your Internet service provider (ISP) and send your e-mail.
	QUSRSYS/ QTMSMTPS	QSYSWRK or configurable	QTSMTPSCH	*YES	No port is used

SMTP Server Daemon
To Start:
`STRTCPSVR *SMTP`
To End:
`ENDTCPSVR *SMTP`
Product: 5722–TC1
Server Type: QIBM_SMTP_SERVER
	QUSRSYS/ QTMSMTPS	QSYSWRK or configurable	QTSMTPSRVD	*YES	25 (smtp)

SMTP Server **To Start:** Starts when the server daemon job QTSMTPSRVD starts the server prestart jobs **To End:** Ends when the server daemon job QTSMTPSRVD ends the server prestart jobs **Product:** 5722–TC1 **Server Type:** QIBM_SMTP_SERVER **Server Description:** Allows end-to-end delivery of mail from one mail server to another. A direct connection exists between the SMTP sender and the destination SMTP receiver. The client keeps the mail at the sender until it transmits and copies it.	QUSRSYS/ QTMSMTPS	QSYSWRK or configurable	QTSMTPSRVP	*YES	No port is used

Simple Network Management Protocol (SNMP) Agent
To Start:
`STRTCPSVR *SNMP`
(Cannot be started from iSeries Navigator)
To End:
`ENDTCPSVR *SNMP`
(Cannot be ended from iSeries Navigator)
Product: 5722–SS1
Server Type: QIBM_SNMP
Server Description: Supports the exchange of network management messages and information among hosts.
	QSYS/QSYSWRK	QSYSWRK	QSNMPSA	*NO	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

SNMP Agent
To Start:
`STRTCPSVR *SNMP`
(Cannot be started from iSeries Navigator)
To End:
`ENDTCPSVR *SNMP`
(Cannot be ended from iSeries Navigator)
Product: 5722–SS1
Server Type: QIBM_SNMP
	QSYS/QTMSNMP	QSYSWRK	QTMSNMPRCV	*NO	161 (snmp)

SNMP Agent
To Start:
`STRTCPSVR *SNMP`
(Cannot be started from iSeries Navigator)
To End:
`ENDTCPSVR *SNMP`
(Cannot be ended from iSeries Navigator)
Product: 5722–SS1
Server Type: QIBM_SNMP
	QSYS/QTMSNMP	QSYSWRK	QTMSNMP	*NO	No port is used

SNMP Trap Manager
To Start:
`STRTRPMGR`
(Cannot be started from iSeries Navigator)
To End:
`ENDTRPMGR`
(Cannot be ended from iSeries Navigator)
Product: 5722–SS1
Server Type: QIBM_SNMP
	QSYS/QTMSNMP	QSYSWRK	QTRPMGR	*NO	No port is used

SNMP Trap Manager
To Start:
`STRTRPMGR`
(Cannot be started from iSeries Navigator)
To End:
`ENDTRPMGR`
(Cannot be ended from iSeries Navigator)
Product: 5722–SS1
Server Type: QIBM_SNMP
	QSYS/QTMSNMP	QSYSWRK	QTRPRCV	*NO	162 (snmp-trap)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
SQL **To Start:** Automatically started at first use of a function that needs the server, such as server mode SQL. **To End:** `ENDPJ SBS(QSYSWRK) PGM(QSQSRVR)` **Product:** 5722-ST1 **Server Type:** QIBM_SQL **Server Description:** Processes SQL statements from an application that is running SQL in server mode. In server mode, each SQL connection is processed by a separate job.	QGPL/QDFTSVR	QSYSWRK	QSQSRVR	*NO	No port is used

System Manager
To Start:
`STRSYSMGR`
To End:
`ENDSYSMGR`
Product: 5722-SM1
Server Type: QIBM_SYSTEM_MANAGER
Server Description: Receives PTF requests, service requests, and test requests from service requesters.
	QSMU/QNSECS	QSYSWRK	QECS	N/A	No port is used

TCP/IP Event Monitor
To Start:
`STRTCP`
To End:
`ENDTCP`
Product: 5722–SS1
Server Type: QIBM_TOC_TCPMONITOR
Server Description: Runs whenever TCP/IP is running and provides an internal mechanism to communicate data and events among TCP/IP services and processes.
	QSYS/QTOCTCPIP	QSYSWRK	QTCPMONITR	N/A	No port is used

TCP/IP Interface Daemon
To Start:
`STRTCP`
To End:
`ENDTCP`
Product: 5722–SS1
Server Type: QIBM_TOC_QTCPIP
Server Description: Starts or ends TCP/IP interfaces. When starting a TCP/IP interface, this daemon also attempts to vary on the line, controller, and device used by that TCP/IP interface.
	QSYS/QTOCTCPIP	QSYSWRK	QTCPIP	N/A	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
TCP/IP L2TP Server **To Start:** The L2TP server job starts automatically when the first L2TP connection profile is started using the STRTCPPTP command or iSeries Navigator. **To End:** The L2TP server job ends automatically when the last L2TP connection profile is ended using the ENDTCPPTP command or iSeries Navigator. **Product:** 5722–SS1 **Server Type:** QIBM_TOCPPP_L2TP **Server Description:** Manages Layer Two Tunneling Protocol (L2TP) connections.	QSYS/QTOCPPJOBD	QSYSWRK	QTPPPL2TP	N/A	1701
TCP/IP PPP Server **To Start:** The PPP server job starts automatically when the first PPP connection profile is started using the STRTCPPTP command or iSeries Navigator. **To End:** The PPP server job ends automatically when the last PPP connection profile is ended using the ENDTCPPTP command or iSeries Navigator. **Product:** 5722–SS1 **Server Type:** QIBM_TOCPPP_CTL **Server Description:** Manages Point-to-Point Protocol (PPP) connections.	QSYS/QTOCPPJOBD	QSYSWRK	QTPPPCTL	N/A	No port is used
TCP/IP SLIP Server **To Start:** One SLIP server job is started for each SLIP connection profile that is started using the STRTCPPTP command or iSeries Navigator. **To End:** Each SLIP server job is ended when its associated SLIP connection profile is ended using the ENDTCPPTP command or iSeries Navigator. **Product:** 5722–SS1 **Server Type:** QIBM_TOCSLIP_SSN **Server Description:** Manages Serial Line Internet Protocol (SLIP) connections.	QSYS/QTOCPPJOBD	QSYSWRK	QTPPDIAL*xx* for SLIP dial connections, where *xx* is a number. QTPPANS*xxx* for SLIP answer connections, where *xxx* is a number.	N/A	No port is used
TELNET Device Manager **To Start:** `STRTCPSVR *TELNET` when QAUTOVRT system value is greater than 0 **To End:** Not applicable **Product:** 5722–TC1 **Server Type:** QIBM_TELNET_DEVMGR **Server Description:** Manages device descriptions when clients start and end Telnet sessions. The Telnet Device Manager stores the client's IP address and port in the device description.	QTCP/QTGSTELN	QSYSWRK	QTVDEVICE	*YES	No port is used

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

TELNET Server
To Start:
`STRTCPSVR *TELNET`
when QAUTOVRT system value is greater than 0
To End:
`ENDTCPSVR *TELNET`
Product: 5722–TC1
Server Type: QIBM_TELNET_SERVER
Server Description: Signs on to an interactive job on the system from another system in a TCP/IP network with a Telnet client.
	QTCP/QTGSTELN	QSYSWRK	QTVTELNET	*YES	23 (telnet)
					992(telnet-ssl)

Text Search Engine Background Process **To Start:** SBMJOB invoked by Update Index Server program DESXCTL **To End:** Ends automatically when task is complete. To interrupt abnormally, use ENDJOB. **Product:** 5722–DE1 Option 3 **Server Type:** QIBM_TEXT_SEARCH_ BGPROC **Server Description:** Updates or reorganizes a text search index.	QGPL/QDFTJOBD	QSYSWRK	IMOSMBCK	N/A	No port is used

Text Search Engine Daemon
To Start:
`CALL PGM(QDB2TX/TXSTART)`
To End:
`CALL PGM(QDB2TX/TXSTOP)`
Product: 5722–DE1 Option 3
Server Type: QIBM_TEXT_SEARCH_ DAEMON
Server Description: Controls access and processing tasks for indexes belonging to the text search engine instance.
	QGPL/QDFTJOBD	QSYSWRK	IMOSMDEM	N/A	No port is used

Transfer Function Server TCP/IP **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue `STRPJ SBS(QSERVER) PGM(QIWS/QTFPJTCP)` **To End:** Ends when the subsystem ends **Product:** 5722–SS1 Option 12 **Server Type:** QIBM_XFER_FUNCTION **Server Description:** Transfers data between the i5/OS operating system and a personal computer.	QGPL/QDFTJOBD	QSERVER	QTFPJTCP	*YES	No port is used

Triggered Cache Manager (TCM)
To Start:
`STRTCPSVR *TCM`
To End:
`ENDTCPSVR *TCM`
Product: 5722–DG1 Option 1
Server Type: QIBM_TCMN*x* (where x is a unique number for each server)
Server Description: Provides applications with a universal cache interface. TCM can keep multiple caches synchronized with current data.
	QTCM/QZHT	QSYSWRK	User Defined	*NO	7049

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port

Trivial FTP
To Start:
`STRTCPSVR *TFTP`
To End:
`ENDTCPSVR *TFTP`
Product: 5722–SS1
Server Type: QIBM_TFTP
Server Description: Provides basic file transfer functions with no user authentication.
	QSYS/QTODTFTP	QSYSWRK	QTTFT*	*NO	UDP 69 (tftp)

Virtual Print Server TCP/IP **To Start:** 1) Starts when the subsystem starts 2) If subsystem is active and the jobs are not active, issue `STRPJ SBS(QSYSWRK) PGM(QIWS/QIWVPPJT)` **To End:** Ends when the subsystem ends **Product:** 5722–SS1 Option 12 **Server Type:** QIBM_VRT_PRINT **Server Description:** Prints data from PC application programs on printers connected to the system. You can use a printer that is attached to the host system as though the printer were directly attached to your personal computer.	QGPL/QDFTJOBD	QSYSWRK	QIWVPPJT	*YES	No port is used

Virtual Private Networking (VPN) Connection Manager
To Start:
`STRTCPSVR *VPN`
To End:
`ENDTCPSVR *VPN`
Product: 5722–SS1
Server Type: QIBM_VPN
Server Description: Performs Internet Key Exchange (IKE) protocol processing and manages all VPN connections. VPN allows you to securely extend your private intranet over a public network, such as the Internet.
	QSYS/QTOVMAN	QSYSWRK	QTOVMAN	*NO	No port is used

VPN Key Manager
To Start:
`STRTCPSVR *VPN`
To End:
`ENDTCPSVR *VPN`
Product: 5722–SS1
Server Type: QIBM_VPN
	QSYS/QTOKMAN	QSYSWRK	QTOKVPNIKE	*NO	No port is used

WebFacing Server
To Start:
`STRTCPSVR *WEBFACING`
To End:
`ENDTCPSVR *WEBFACING`
Product: 5722–SS1
Server Type: QIBM_WEBFACING
Server Description: Gives a Web-based application access to application data from interactive programs running on the system.
	QSYS/QSYSJOBD	QSYSWRK	QQFWFSVR	*NO	4004 (as-WebFacing)

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
WebSphere Application Server V4 Advanced Edition, Administration Server **To Start:** Either when SBS starts (default Administration server) or explicit command **To End:** Through WAS UI **Product:** 5733–WA4 **Server Type:** QIBM_WSA_ADMIN **Server Description:** Allows a WebSphere user to connect a WebSphere Administrative Console to the system to administer the WebSphere configuration.	QEJBADV4/ QEJBJOBD QEJBADV4/ QEJBJOBD	QEJBADV4	Configurable (default of QEJBADMIN)	N/A	900 9000
WebSphere Application Server V4 Advanced Edition, Application Server **To Start:** Through WAS UI or automatic at start of administration server **To End:** Through WAS UI **Product:** 5733–WA4 **Server Type:** QIBM_WSA_EJBSERVER **Server Description:** Allows you to implement and manage server-side Java™ components, enterprise beans, JavaSever Pages, and JSP files.	QEJBADV4/ QEJBJOBD QEJBADV4/ QEJBJOBD	QEJBADV4	Configurable (default of DEFAULT_SE)	N/A	9080
WebSphere Application Server V4 Advanced Single Server Edition, Application Server **To Start:** Either when SBS starts (default server) or explicit command **To End:** Through WAS UI **Product:** 5733–WS4 **Server Type:** QIBM_WSA_EJBSERVER	QEJBADV4/ QEJBJOBD QEJBADV4/ QEJBJOBD	QEJBAES4	Configurable (default of DEFAULT_SE)	N/A	900 9000 9080
WebSphere Application Server V5 Express **To Start:** Can be started via QShell script or Web ADMIN **To End:** Can be ended via QShell script or Web ADMIN **Product:** 5722–IWE Option 2 **Server Type:** QIBM_WSA_EJBSERVER	QASE5/QASE5	QASE5	Configurable instance name	N/A	Configurable
WebSphere Application Server V5, Application Server **To Start:** Either when SBS starts (default server) or explicit command **To End:** Explicit command **Product:** 5733–WS5 Option 2 **Server Type:** QIBM_WSA_EJBSERVER	QEJBAS5/ QEJBJOBD	QEJBAS5	Configurable (default of SERVER1)	N/A	9090 9043 2809 8880 9080 7873 5557 5558 5559 9501 9502 9503
WebSphere Application Server V5 Network Deployment Edition, Application Server **To Start:** Through WAS admin interfaces or automatic at start of node agent **To End:** Through WAS UI **Product:** 5733–WS5 Option 2, 5 **Server Type:** QIBM_WSA_EJBSERVER	QEJBAS5/ QEJBJOBD	QEJBAS5	Configurable (default of SERVER1)	N/A	9810 8880 9080 7873 9501 9502 9503

Server name	Job description	Subsystem	Job name	Shipped default value for autostart parameter	Default port
WebSphere Application Server V5 Network Deployment Edition, Deployment Manager **To Start:** Either when SBS starts (default server) or explicit command **To End:** Explicit command **Product:** 5733–WS5 Option 5 **Server Type:** QIBM_WSA_EJBSERVER	QEJBAS5/ QEJBNDJOBD QEJBAS5/ QEJBNDJOBD	QEJBASND5	Configurable (default of DMGR)	N/A	9090 9043 9809 8879 7989 9401 9402 9403 9100 7277
WebSphere Application Server V5 Network Deployment Edition, Node Agent **To Start:** Either when SBS starts (default instance) or explicit command **To End:** Through WAS UI or explicit command **Product:** 5733–WS5 Option 2, 5 **Server Type:** QIBM_WSA_EJBSERVER	QEJBAS5/ QEJBJOBD	QEJBAS5	NODEAGENT	N/A	Configurable

WebSphere Host On-Demand Service Manager
To Start:
`STRTCPSVR *HOD`
To End:
`ENDTCPSVR *HOD`
Product: 5733–A59
Server Type: QIBM_HOST_ONDEMAND
	Configurable (default is QGPL/QDFTJOBD)	QSYSWRK	QHODSVM	*NO	8999

Workload Management Server **To Start:** STRWLM (CHGWLMA must be run prior to first start of server) **To End:** ENDWLM **Product:** 5798–WLD **Server Type:** QIBM_WLM_SERVER	QWLMDE/QWLMDE	QSYSWRK	QWLMSVR	N/A	Configurable

Related concepts

"Using Netstat from a character-based interface: Connections" on page 4
You need to verify the status of your IPv4 and IPv6 connections.

"Using Netstat from iSeries Navigator: Connections" on page 7
You need to verify the status of your IPv4 and IPv6 connections.

Related tasks

"Job trace" on page 26
Use the job trace tool to trace data in any job to help identify your problem.

"Starting a job trace" on page 27
This action starts a job trace for one or more jobs. You can start any number of trace sessions, but active trace session identifiers must be unique across the system.

Checking jobs, job logs, and message logs

You can view jobs, job logs, and messages to identify problems and make adjustments to solve them.

If you are having problems with TCP/IP connectivity, you should take a look at the jobs that are running on your system. All work on your system is performed through jobs. Most jobs have associated job logs that record the jobs' activities. The job log contains information, such as when the job starts and ends, which commands are running, and error messages. Here are some ways to use jobs and job logs to help solve your TCP/IP problems.

Verifying that necessary jobs exist:

TCP/IP requires that certain basic jobs are running. You can verify that these basic jobs are running.

For normal usage, you need to have the QTCPIP job running in the QSYSWRK subsystem. The QTCPIP job controls starting and ending TCP/IP interfaces. However, you can run TCP/IP when the operating system is in the restricted state. In this case, the QTCPIP job is not active.

In addition, you should have at least one job for each of the servers you are attempting to use.

To verify the required jobs, select one of these interfaces.

Related tasks

Configuring TCP/IP when the operating system is in restricted state

Verifying jobs from a character-based interface:

You can use the character-based interface to verify jobs.

Verifying the QTCPIP job

To find the QTCPIP job using the character-based interface, follow these steps:

1. At the command line, type `WRKACTJOB SBS(QSYSWRK)` (Work with Active Jobs).
2. Press F7 (Find).
3. At **String** type `QTCPIP` to search for the job. When found, the QTCPIP job is displayed at the top of the subsystem/job list.

Verifying one job for each server

To verify that you have at least one job for each of the servers you are attempting to use, follow these steps:

1. At the command line, type WRKSBS (Work with Subsystems).
2. View the list of subsystems, and locate QSYSWRK.
3. Select option 8 (Work with subsystem jobs) in front of QSYSWRK, and press Enter.
4. View the list of jobs associated with QSYSWRK. Locate at least one job for each of the applications you are attempting to use and verify that each of the jobs is active.

In addition to verifying active jobs in the QSYSWRK subsystem, you should verify jobs in the QUSRWRK and QSERVER subsystems. If you have servers that run in their own subsystems, you should also check the jobs in those subsystems. See the server table to find the job name associated to the server you want to verify.

Verifying jobs from iSeries Navigator:

You can use iSeries Navigator to verify jobs.

Verifying the QTCPIP job

To find the QTCPIP job, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Work Management** → **Server Jobs**.
2. From the Edit menu, select **Find (Ctrl+F)**.
3. In the **Search for** field, type `Qtcpip`. All the job columns are searched for the job.
4. Click **Find**. iSeries Navigator will highlight your job once it is found.

Verifying one job for each server

To see if you have at least one job for each of the servers you are attempting to use, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Work Management** → **Subsystems** → **Active Subsystems**.
2. Click **Qsyswrk**.

   Note: QSYSWRK and the controlling subsystem are always started for you by the operating system. QUSRWRK and QSERVER are started by the IBM-supplied startup program, so unless you have changed the IBM-supplied startup program these subsystems should be started automatically for you. The server jobs might also be in QUSRWRK, QSERVER, or their own subsystem.
3. View the list of jobs in the **Job name** column in the right pane, and locate at least one job for each of the applications you are attempting to use.

See the server table to find the job name associated to the server you want to verify.

Checking the job logs for error messages and other indication of problems:

You can use job logs to help identify the source of your problem.

A *job log* is a record of the activities associated with a particular job, such as the time an interface started and processing delays or failures. Job logs help you identify the source of your problem.

To work with job logs, select one of these interfaces.

Checking job logs using the character-based interface:

You can use the character-based interface to check job logs.

To access the job log for an active job or server job, follow these steps:

1. At the command line, type `WRKACTJOB` (Work with Active Jobs).
2. Press F7 (Find) to locate the specific job. See the server table if you need help finding the job name associated to the server.
3. Select option 5 (Work with) in front of the job in the listing.
4. On the Work with Job Display, select option 10 (Display Job Log if Active or on Job Queue), and press Enter. View the messages displayed in the Job Log to help identify problems associated with this job.

Checking job logs using iSeries Navigator:

You can use iSeries Navigator to check job logs.

To access the job log for an active job or server job, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Work Management** → **Active Jobs** or **Server Jobs**. You can see a job log from any place within Work Management that you access jobs (for example, through the subsystem area or the memory pool area).
2. Right-click a job (for example, Qsyswrk) and select **Job Log**. View the messages displayed in the Job Log to help identify problems associated with this job.

   To view more details of a message, double-click a specific message ID. A Message Details dialog box appears. This dialog shows the details of the message as well as the message help. The detailed message gives you information to help solve a problem.

Changing the message logging level on job descriptions and active jobs:

If you are having problems with TCP/IP or the server jobs, you might need to change the message logging level text value on the job description or on the active job associated with your TCP/IP server.

You should change the message logging level text value from the default value *NOLIST to *SECLVL. The value *SECLVL causes a job log to be generated. It is helpful to review the job log for messages that can identify problems.

Note that changes to job descriptions do not affect currently running jobs. You must end and then restart the server for the change to take effect.

To change the message logging levels on job descriptions or on active jobs, select one of these interfaces.

Changing the message logging level from a character-based interface:

You can use the character-based interface to change the message logging level.

Changing the message logging level on a job description

To change the message logging level on the job description, follow these steps using the character-based interface:

1. At the command line, type `WRKJOBD` (Work with Job Descriptions), and press F4 (Prompt).
2. For the *Job description* prompt, specify the name of the job description, such as `MYJOBD`.
3. For the *Library* prompt, specify the library that contains the job description you want to change, and press Enter.
4. On the Work with Job Descriptions display, select option 2 (Change) in front of the job description you want to change, and press Enter.
5. On the Change Job Description display, page down to the **Message logging**.
6. For the *Message Logging* prompt, type `4` for the *Level* parameter, `00` for the *Severity* parameter, `*SECLVL` for the *Text* parameter, and press Enter.
7. You must end and then restart the server for the change to take effect. At the command line, type ENDTCPSVR *MYSERVER, where MYSERVER is the server you want to stop. Then, type STRTCPSVR *MYSERVER to restart the server. Be aware that if you only type ENDTCPSVR, the default *ALL will end all of the TCP servers. If you need to end and restart a server that is not started with the STRTCPSVR command, you need to specify different commands. See the server table for the appropriate commands to end and restart those servers.

Changing the message logging level on an active job

To change the message logging level of a server job that is currently active, follow these steps:

1. At the command line, type `CHGJOB`, and press F4 (Prompt).
2. For the *Job name* prompt, specify the name of the job you want to change, such as `MYJOB`, and press Enter. See the server table to find the job name associated to your server.
3. On the Change Job display, page down to **Message logging**.
4. For the *Message Logging* prompt, type `4` for the *Level* parameter, `00` for the *Severity* parameter, type `*SECLVL` for the Text parameter, and press Enter.

Changing the message logging level from iSeries Navigator:

You can use iSeries Navigator to change the message logging level.

Changing the message logging level on a job description

You must use the character-based interface to change the message logging level text value on a job description.

Changing the message logging level on an active job

To change the message logging level of a server that is currently active, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Work Management** → **Server Jobs**.
2. Right-click the job you want to change and select **Properties**.
3. Click the **Job Log** tab.
4. Select **Create printer output for job log if job ends normally**, select **Print message, cause, and recovery**, and click **OK**.

Other job considerations:

Considerations regarding the job log maximum size and the resulting job actions might help you solve the problem.

Job log maximum size

If you are having problems with storage consumption, you might need to change the job log maximum size on the server job. You should specify a relatively small size for the job log to avoid consuming excessive storage and, in some cases, excessive processing time. These types of resource consumption occur when the system produces job logs. For example, if a repetitive error occurs on a long running server job, your job log fills up with repetitive messages and increases your storage consumption level.

The value specified for the *Job message queue maximum size* (JOBMSGQMX) parameter for the job indicates the size of the job log. This value, along with the rest of the job's properties, is given to the job when the job is started. Some server jobs specify this value in the job description used by the job. Other server jobs specify this value by defaulting to the setting of the QJOBMSGQMX system value.

The recommended value for the *JOBMSGQMX* parameter is 8 MB. You cannot change the value for this parameter by using the Change Job (CHGJOB) command. However, you can change the value by accessing the parameter through the job description using the Change Job Description (CHGJOBD) command.

Job log full action

When the job log reaches its maximum capacity as determined by the *JOBMSGQMX* parameter, several different actions can occur depending on the value specified in the *Job message queue full action* (JOBMSGQFL) parameter for the job. In most cases, the job description indicates *WRAP as the default value. Many server jobs specify this value in the job description used by the jobs.

You should verify that *WRAP is specified for the JOBMSGQFL parameter by accessing the job description. This value ensures that the job log messages overlay one another when the job log reaches its maximum capacity. Be aware that other values, such as *NOWRAP, can cause the server job to end when the job log reaches its maximum capacity.

Checking for active filter rules

Your network communications might be failing because IP packet filters are stopping your incoming or outgoing data. You can find out if filter rules are limiting your communication.

Packet filter rules are designed to protect a network by filtering packets according to rules that the network administrator defines. Packet rules might have been created on either your system or the destination system, and the packet rules might filter data that is incoming or outgoing. Rules might have also been defined on one or more intermediate routers.

To find out whether you have active filter rules on your system, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **IP policies** → **Packet rules**. If the right pane is empty, then your system is not currently using packet rules. If the right pane contains a list of interfaces, then proceed with the next step.
2. Select the interface that you suspect contains the active filter rules.
3. View the list of active packet rules in the right pane. Click **Help** to find out how to edit and work with packet rules.

To remove active filters on the system, type `RMVTCPTBL *ALL` (Remove TCP/IP Table) at the command line. This command also causes virtual private networking (*VPN) tunnels to fail, so use this command with caution.

To find out whether filter rules are active on the destination system, call the network administrator at that location.

Related concepts

IP filtering and network address translation

Verifying system startup considerations for networking

You need to start subsystems, TCP/IP, interfaces, and servers in the correct order and know how to locate problems associated with startup.

Your network communications might be failing because the server and its associated subsystems and interfaces have not been started properly. You need to start the appropriate subsystems, servers, interfaces, and the TCP/IP stack in the correct order to ensure successful network communications. Follow this order when starting the subsystems, stack, interfaces, and servers.

Starting subsystems:

Before you start TCP/IP, start the appropriate subsystems.

The following subsystems should be started before you start TCP/IP:

- QSYSWRK
- QUSRWRK
- QSERVER

QSYSWRK and the controlling subsystem are always started for you by the operating system. QUSRWRK and QSERVER are started by the IBM-supplied startup program, so unless you have changed the IBM-supplied startup program these subsystems should be started automatically for you.

If you are using any subsystems other than the IBM-supplied subsystems, you might also need to start these subsystems before you start TCP/IP.

See the server table to understand how the servers map to the actual jobs and subsystems they represent.

Starting TCP/IP:

Before you can communicate over the network, TCP/IP must be started.

Note: The server automatically starts TCP/IP for you when you start the system. However, if you end TCP/IP due to problems so that you need to manually restart TCP/IP, read the following information.

Remember that your line descriptions, network server descriptions, and network interface descriptions should be configured to vary on with TCP/IP. This allows these configuration objects to start at the same time TCP/IP starts. See Varying on communication lines, controllers and devices for more information.

Starting TCP/IP using the character-based interface:

You can use the character-based interface to start TCP/IP.

To start TCP/IP, follow these steps:

1. At the command line, type `STRTCP`.
2. Verify that TCP/IP has started. If TCP/IP was already active when you entered STRTCP, you should receive the message `TCP/IP currently active`. If TCP/IP was not active and STRTCP started TCP/IP successfully, you should receive the message `STRTCP completed successfully`.

Starting TCP/IP using iSeries Navigator:

You can use iSeries Navigator to start TCP/IP.

Note: You can use iSeries Navigator to stop TCP/IP. However, if you stop TCP/IP, you are likely to lose your iSeries Navigator connection to the server because iSeries Navigator requires TCP/IP for its own connection. Therefore, in most situations you should use some form of console to start and stop TCP/IP so that you do not lose the very connection you are working with. In this case and depending on your hardware configuration, you can use a Twinaxial console, Operations Console, or the Hardware Management Console (HMC) to start and stop TCP/IP because these consoles do not require TCP/IP to be started in i5/OS.

To start TCP/IP, follow these steps:

1. In iSeries Navigator, select ***your system*** → **Network**.
2. Right-click **TCP/IP Configuration**, and select **Start**.
3. Verify that TCP/IP has started. You should receive the message `TCP/IP currently active`.

Starting interfaces:

Start the appropriate interfaces to ensure your network communication.

Your network communication might be failing because your interfaces have not been activated. Remember these tips to ensure your interfaces are functioning correctly.

- Verify that your interfaces are configured and activated by using Netstat. For those interfaces you always want active, you should specify AUTOSTART (*YES). They will automatically start when TCP/IP is started.
- If you are using profiles for remote access services, such as Point-to-Point Protocol (PPP) or Layer Two Tunneling Protocol (L2TP), you should verify that the profiles are active. To verify the status of the profiles, follow these steps:
  1. In iSeries Navigator, select ***your system*** → **Network** → **Remote Access Services**.
  2. Click **Originator Connection Profiles** or **Receiver Connection Profiles** depending on the type of profile you want to verify, and view the list of profiles in the right pane to verify the status. To start a profile, right-click the profile and select **Start**.

  If you want any of the remote access services profiles to automatically start when TCP/IP is started, you should specify AUTOSTART (*YES) for those profiles. It might be useful to set the profiles to automatically start with TCP/IP in these types of situations:

  - You want to have a constant dialup connection to the ISP.
  - You schedule an IPL at midnight and you want the profiles to automatically start during the IPL.

- Verify that the QTCPIP job is active. See Verifying that necessary jobs exist for these instructions. The QTCPIP job must be active before you can start or end your interfaces.
- Be aware that your line descriptions, network server descriptions, and network interface descriptions should be configured to vary on with TCP/IP. This allows these configuration objects to start at the same time TCP/IP starts. See Varying on communication lines, controllers, and devices for more information.

  Related tasks

  "Netstat" on page 2
  Netstat is a tool for managing and monitoring the status of your system's interfaces, routes, and connections, and it is useful for troubleshooting TCP/IP problems. You can use Netstat whether you are using IPv4 or IPv6 connectivity on the network.

Starting servers:

Start the appropriate servers to avoid trouble communicating over TCP/IP.

The system is shipped with several servers configured to automatically start when TCP/IP starts. However, you can configure additional servers to automatically start when TCP/IP starts, or you can manually start the individual servers at any time.

Remember that most subsystems required by your servers must be active before the server starts. However, some servers start their own subsystems. See the server table to understand how the servers map to the actual jobs and subsystems they represent.

Note: Servers that are required to run iSeries Navigator, such as the remote command server, signon server, server mapper, and database server, must be started from the character-based interface.

Starting servers from a character-based interface:

You can use a character-based interface to start the servers.

Configuring a server to start when TCP/IP starts

To configure a server to start when TCP/IP starts, follow these steps:

1. At the command line, type `CHG`*`xxx`*`A` where *xxx* is the name of the server. For example, `CHGFTPA` to work with the attributes of the FTP server.
2. For the *Autostart servers* prompt, type `*YES`. This will start the number of servers you indicate in the *Number of initial servers* prompt.
3. Specify either the `STRTCP` (Start TCP/IP) command or the `STRTCPSVR SERVER (*AUTOSTART)` command to automatically start the server.

Starting a server manually

This example shows how to start certain types of TCP servers. See the server table for a list of servers and the commands you can use to start them. To manually start a server, follow these steps:

1. At the command line, type `STRTCPSVR`, and press F4 (Prompt).
2. For the *Server application* prompt, specify the servers you want to start, and press Enter.

Starting servers from iSeries Navigator:

You can use iSeries Navigator to start the servers.

Configuring a server to start when TCP/IP starts

To configure a server to start when TCP/IP starts, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network**.
2. Right-click **TCP/IP Configuration** and select **Properties**.
3. On the **Servers to Start** page, select the servers you want to automatically start when TCP/IP starts.

Starting a server manually

To manually start a server, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **Servers**.
2. Click **TCP/IP**, **iSeries Access**, **DNS**, or **User-Defined**, depending on the type of server you want to start.
3. In the right pane, right-click the server you want to start, and select **Start**.

Some servers cannot be started from iSeries Navigator. Servers that are required to run iSeries Navigator, such as the remote command server, signon server, server mapper, and database server, must be started from the character-based interface.

Timing considerations:

Timing considerations during startup can affect network communications.

i5/OS has the capability to automatically start the necessary subsystems, the TCP/IP stack, lines, interfaces, and servers at the appropriate times during IPL. In most situations, your network communications will start smoothly using this automatic startup process.

However, depending on your unique hardware and software configuration you might have problems starting the network communication due to timing problems during IPL. Timing problems can occur for several different reasons. For example:

- The processing speed and the number of input-output processors (IOPs) can affect the startup of the network hardware resource. If your hardware resource is slow to start, it might not be ready when TCP/IP tries to start. Your network communications fail because the TCP/IP interfaces cannot be started.
- You might encounter timing problems if you have customized your server so that you are using subsystems other than the IBM-supplied subsystems. Many subsystems are typically started by the IPL startup program. However, if you are using customized subsystems that are not recognized by the IPL startup program, they will not be automatically started at IPL. Your network communications fail because these subsystems have not been started.

If these types of timing problems occur, you can automatically start the subsystems, the TCP/IP stack, interfaces, and servers in the correct order by creating a customized IPL startup program. You might need to put delays in the startup program to ensure that each step of the startup process is initiated at the appropriate time. For example, the subsystems should be started before the TCP/IP stack, and the interfaces should be started after the communications resources are available.

To change from using the default IPL startup program to using a customized startup program, follow these steps:

1. Create a customized startup program. Things to consider when creating a new startup program:

   Note: These steps are defined to ensure that all required resources are active before the next step.

   - Start the subsystems.
   - Allow delays after the subsystems start.

- Use the Retrieve Subsystem Information (QWDRSBSD) API to ensure that the subsystems are active. Although this API is not required, it can help you avoid timing problems.
- Start TCP/IP specifying STRSVR *NO, STRIFC *NO and STRPTPPRF(*NO).

  Note: You will start TCP/IP for both IPv4 and IPv6 by performing this step. If you do not want to start IPv6, specify STRIP6 (*NO) on the STRTCP command.
- Start the interfaces with STRTCPIFC *AUTOSTART. Remember that TCP/IP should vary on your communication lines, controllers, and devices.
- Allow delays to ensure the required interfaces are active.
- Start the TCP/IP point-to-point session jobs with STRTCPPTP *AUTOSTART.
- Start the servers with STRTCPSVR *AUTOSTART.
- Start any other servers that are not started with the STRTCPSVR command. Use STRHOSTSVR *ALL.

2. Test the customized startup program by calling the program. To properly test the program you need to end TCP/IP and the subsystems. However, be aware this can terminate the connections that other users are using. Plan accordingly to test when the system is dedicated to the test.
3. Change the QSTRUPPGM system value to point to your customized start-up program. It is not recommended to directly change QSTRUP.
4. Change the IPL attribute to no longer start TCP/IP automatically when the system is started. To change the IPL attribute, follow these steps:
   a. At the command line, type `CHGIPLA` (Change IPL Attributes), and press F4.
   b. For the *Start TCP/IP* prompt, type `*NO`. This prevents TCP/IP from starting at IPL, letting your startup program control the startup.

Varying on lines, controllers, and devices

Your line descriptions, network server descriptions, and network interface descriptions should be configured to vary on when TCP/IP starts. This allows these configuration objects to start at the same time TCP/IP starts.

To configure your configuration objects to vary on when TCP/IP starts, follow these steps:

1. At the command line, type `WRKLIND` for the line description, `WRKNWSD` for the network server description, or `WRKNWID` for the network interface description, depending on the type of configuration object you want to change.
2. Select option 2 (Change) in front of the object description you want to change, and press Enter.
3. For the *Online at IPL* prompt, type `*NO`, and press Enter.

Verifying the logical partition configuration

You might need to verify that the logical partition (LPAR) configuration is correct.

If you are having problems communicating among partitions over a virtual Ethernet, you should verify that your logical partitions are configured correctly. The partitions must be configured to be able to communicate with one another. If the partition configuration is wrong, then your TCP/IP configuration will not work even if you configured TCP/IP correctly.

To work with LPAR, you must have *SERVICE special authority. See the chapter on user profiles in iSeries Security Reference for more information about this type of authority.

Verifying the LPAR configuration from a character-based interface:

You can use a character-based interface to verify the LPAR configuration.

To verify the LPAR configuration, follow these steps:

1. At the command line, type STRSST (Start System Service Tools).
2. Type your Service Tools user ID and password.
3. Select option 5 (Work with system partitions).
4. Select option 3 (Work with partition configuration).
5. Press F10 (Work with Virtual Ethernet Configuration).
6. Verify that all the partitions on the virtual Ethernet are configured to communicate with one another.

Verifying the LPAR configuration from iSeries Navigator:

You can use iSeries Navigator to verify the LPAR configuration.

To verify the LPAR configuration, follow these steps:

1. In iSeries Navigator, expand the primary partition of the system **Configuration and Service** → **Logical Partitions**.
2. Type your Service Tools user ID and password, and click **OK**.
3. Right-click **Properties**, and select the **Virtual Ethernet** page.
4. Verify that all the partitions on the virtual Ethernet are configured to communicate with one another.

Troubleshooting IPv6–related problems

If you are using IPv6 for network connectivity, you can use several of the same troubleshooting tools as you do for IPv4. For instance, you can use trace route and Ping to test connections and routes for both types of networks. In addition, you can use the Netstat and communications trace functions for IPv6.

Remember these tips when troubleshooting problems that are specific to IPv6:

- Ensure that your Ethernet line is configured and active. To check the status of lines that are configured on the system, follow these steps:
  1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **Lines**.
  2. In the right pane, find the line that should be configured for IPv6 and check the Status column. If the line does not appear in the list, you must configure a line for IPv6 either by manually configuring interfaces on an existing line or by using IPv6's Stateless Address Autoconfiguration feature, or both.
- If your Ping to an IPv6 address was unsuccessful, verify the address state of both interfaces. Both interfaces should have an address state of Preferred. If either the target or source interface is not in the preferred state, then either choose other interfaces for the test or change the interfaces being used to the correct status and address state. To verify the address state of the source interface, follow these steps:
  1. In iSeries Navigator, expand ***your system*** → **Network** → **TCP/IP Configuration** → **IPv6** → **Interfaces**.
  2. In the right pane, right-click the IP address associated with the interface, select **Properties**, and select the **Options** page. This dialog allows you to view the preferred lifetime or valid lifetime for the interface. Repeat these steps to check the state of the target interface address.

Related tasks

"Netstat" on page 2
Netstat is a tool for managing and monitoring the status of your system's interfaces, routes, and connections, and it is useful for troubleshooting TCP/IP problems. You can use Netstat whether you are using IPv4 or IPv6 connectivity on the network.

"Ping" on page 7
You can use the Packet Internet Groper (Ping) function to test IP-level connectivity between two TCP/IP-capable interfaces or systems.

"Trace route" on page 15
The trace route function allows you to trace the route of IP packets to a user-specified destination system so you can locate the connectivity problem.

“Communications trace” on page 16
You can use communications trace to determine whether your data is being transmitted correctly across the network.

Configuring IPv6

Advanced troubleshooting tools

You can use these advanced problem solving techniques to solve complex problems. Most of these techniques require the collection of various debugging information.

The following advanced troubleshooting tools are typically used at the request of your service provider. However, you should familiarize yourself with these tools, then work with your service provider to maximize the benefits of these tools.

Note: If you are reporting your TCP/IP problem to your service provider, you might be asked to provide a copy of the configuration files used for TCP/IP processing or a copy of the integrated file system (IFS) files. Use their instructions for sending the files to your service provider.

To solve network problems by using the i5/OS performance tools, see the **Performance** topic.

Licensed Internal Code logs

Locate the Licensed Internal Code logs so that you can send the logs to your service provider for troubleshooting when requested.

This function is typically used at the request of your service provider.

To work with the Licensed Internal Code logs, you must have *SERVICE special authority. See the chapter on user profiles in iSeries Security Reference for more information about this type of authority.

To check the Licensed Internal Code logs, follow these steps:

1. At the command line, type `STRSST` (Start System Service Tools).
2. Type your Service Tools user ID and password.
3. Select option 1 (Start a service tool).
4. Select option 5 (Licensed Internal Code log).
5. Contact your service provider for assistance.

Trace Internal (TRCINT) command

To debug problems associated with the internal operation of the Licensed Internal Code, use the Trace Internal (TRCINT) command to collect data.

This function is typically used at the request of your service provider.

The Trace Internal (TRCINT) command is used to collect data about the internal operation of the Licensed Internal Code. Use TRCINT to debug a problem that you can re-create, but is not visible at the application level. For example, you can use TRCINT to debug Licensed Internal Code in the TCP/IP protocol stack and sockets.

To use the CL commands to perform the internal trace, you must have *SERVICE special authority, or be authorized to the service trace function of i5/OS through iSeries Navigator. See the chapter on user profiles in iSeries Security Reference for more information about this type of authority.

Related reference

Trace Internal (TRCINT) command

Product activity log

Locate the product activity log and work with your service provider to determine why your IP packets are being discarded.

This function is typically used at the request of your service provider.

To work with the product activity log, you must have *SERVICE special authority. See the chapter on user profiles in iSeries Security Reference for more information about this type of authority.

Use the product activity log to view error log data. Whenever a TCP/IP datagram is discarded because of a protocol error, the TCP/IP Licensed Internal Code creates an entry in the product activity log.

You can view entries for discarded datagrams that are outbound or inbound:

- Outbound datagrams - For outbound TCP/IP datagrams, an error is reported to the user and the outbound datagram is discarded. For example, you try to send a datagram over your X.25 connection, but the connection fails.
- Inbound datagrams - Inbound datagrams cause an entry in the product activity log to be created when both of these conditions are met:
  - The Log protocol errors TCP/IP attribute is set to *YES.
  - The datagram has failed one of the TCP/IP protocol validity tests specified in RFC 1122, causing the system to discard it. (**Silently discarded** means the following: Discard the received datagram without reporting an error to the originating host device.) Examples of such datagrams are those with checksums or destination addresses that are not valid.

When a datagram is discarded, the IP and TCP or UDP datagram headers are logged in the detailed data of the product activity log entry. The system reference code for these product activity log entries is 7004.

To display the Product Activity Log, follow these steps:

1. At the command line, start STRSST (Start System Service Tools), and press Enter.
2. Type your Service Tools user ID and password, and press Enter.
3. In the System Service Tools menu, select Option 1 (Start a Service Tool), and press Enter.
4. In the Start a Service Tool menu, select Option 1 (Product Activity Log), and press Enter.
5. Contact your service provider for assistance.

IOP dump

Dump an IOP when requested by your service provider.

This function is typically used at the request of your service provider.

Process dump

Dump a process when requested by your service provider.

These functions are typically used at the request of your service provider.

To use the CL commands to perform a dump, you must have authority to one of these IBM-supplied user profiles:

- QPGMR
- QSYSOPR
- QSRV
- QSRVBAS

See the chapter on user profiles in iSeries Security Reference for more information about these types of authority.

You might be asked by your service provider to perform one of the following types of dumps. Click each dump for step-by-step instructions:

Call stack dump:

To perform a call stack dump, follow these steps.

1. At the command line, type `DMPJOB`, and press F4 (Prompt).
2. For the *Program* prompt, type `*NONE`.
3. For the *Job structure areas* prompt, type `*NONE`.
4. For the *Objects referenced by address* prompt, type `*NO`.
5. For the *Job threads* prompt, type `*THDSTK`, and press Enter.

This particular set of values is used to get a dump of the call stacks for all threads in the process. This is most useful for multi-threaded jobs.

Full job dump:

To perform a full job dump, follow these steps.

1. At the command line, type `DMPJOB`, and press F4 (Prompt).
2. For the *Program to dump, Program* prompt, type `*ALL`.
3. For the *Job structure areas* prompt, type `*ALL`.
4. For the *Objects referenced by address* prompt, type `*YES`.
5. For the *Job threads* prompt, type `*YES`.
6. For the *Thread ID to include* prompt, type `*ALL`.

Troubleshooting problems related to specific applications

If you know that your problem lies within a specific application, use this information to troubleshoot the specific application.

If you have determined that your problem lies within a specific application that you are running on TCP/IP, then select the application for detailed troubleshooting information. If you do not find the application in the list, perform a search for the application you need. Then, use the troubleshooting information provided there.

The following information might help you understand troubleshooting problems related to specific applications.

Domain Name System

This topic provides a flow chart for problem analysis and guides you through debugging strategies for Domain Name System (DNS) problems.

File Transfer Protocol

This topic suggests solutions for File Transfer Protocol (FTP) problems and demonstrates the server job log as a troubleshooting tool.

Point-to-Point Protocol

This topic offers solutions to common Point-to-Point Protocol (PPP) connection problems.

Post Office Protocol

This topic helps troubleshoot the Post Office Protocol (POP) server and other e-mail applications.

Remote Execution
This topic provides a flow chart to help you identify the Remote Execution (REXEC) problem and to find potential solutions.

Simple Mail Transfer Protocol
This topic provides several methods for solving problems with Simple Mail Transfer Protocol (SMTP) and other e-mail applications.

Telnet This topic assists you with general Telnet problems as well as specific problems related to the emulation type and to the SSL server. In addition, find out what information is necessary for reporting the problem.

Virtual private networking
This topic guides you through several troubleshooting strategies for Virtual private networking (VPN) problems related to connection, configuration errors, filter rules, and more.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Corporation

© Copyright IBM Corp. 1997, 2006

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

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

COPYRIGHT LICENSE:

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

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

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This TCP/IP troubleshooting publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Presentation
AFP
CICS
DB2
Domino
DRDA
i5/OS
IBM
IBM (logo)
Infoprint
iSeries
Lotus
NetServer
Network Station
Notes
Operating System/400
OS/400
QuickPlace
System i
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Printed in USA