

IBM i
7.4

*Networking
Getting started with IBM i
communications*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 41.](#)

This document may contain references to Licensed Internal Code. Licensed Internal Code is Machine Code and is licensed to you under the terms of the IBM License Agreement for Machine Code.

© **Copyright International Business Machines Corporation 1998, 2018.**

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

Contents

Getting started with IBM i communications.....	1
PDF file for Getting started with IBM i communications.....	1
Networking concepts.....	1
Advanced Peer-to-Peer Networking.....	1
Advanced Program-to-Program Communication.....	2
Dependent logical unit requester.....	3
High-Performance Routing.....	3
Systems Network Architecture.....	5
TCP/IP.....	5
Configuring your IBM i server for communications.....	5
Creating a line description.....	5
Optimizing communications performance.....	6
Improving wide area network performance.....	6
Adjusting WAN protocols for optimum communications performance.....	6
Adjusting the WAN line speed for optimum communications performance.....	6
WAN configuration considerations for IOAs.....	7
Improving local area network performance.....	7
Adjusting LANs for optimum communications performance.....	7
Adjusting LAN lines for optimum communications performance.....	8
Improving data path performance.....	8
Considerations for subsystem configuration for error recovery performance.....	8
Communications performance considerations for interactive jobs.....	9
Communications performance considerations for batch jobs.....	11
Considerations for mixing interactive and batch jobs on a WAN line.....	11
Configuring Enterprise Extender communications for the IBM i system.....	11
Migrating from AnyNet to Enterprise Extender.....	12
Migrating a system that can start AnyNet connections.....	12
Migrating a system that cannot start AnyNet connections.....	12
Subsystems.....	13
Communications applications.....	13
User-written APPC applications.....	13
Distributed data management.....	14
Application programming interface performance considerations.....	14
Performance considerations for intersystem communications function.....	15
Performance considerations for Common Programming Interface Communications.....	15
Communicating with a host system.....	16
Matching IBM i system parameters for a host system.....	16
Matching IBM i system line description parameters for a host system.....	16
Matching IBM i system controller description parameters for a host system.....	17
Matching IBM i system device description parameters for a host system.....	18
Matching IBM i system mode and class-of-service description parameters for a host system...	19
Configuring dependent logical unit requester.....	20
Configuring the host controller description.....	20
Configuring the device descriptions.....	21
Example: Enterprise Extender Configuration between IBM i and Host (VTAM).....	22
Communicating with a remote IBM i server.....	24
Matching IBM i system line description parameters for a remote IBM i system.....	24
Matching IBM i system controller description parameters for a remote IBM i system.....	25
Matching IBM i system device description parameters for a remote IBM i system.....	26
Examples: Connecting one IBM i server to another IBM i server.....	28
Example: IBM i server to IBM i server using Enterprise Extender (HPRIP).....	28

Common networking standards.....	29
Local area network standards.....	29
Ethernet networks.....	29
Wide area network standards.....	29
Asynchronous communications.....	29
Binary synchronous communications.....	30
Troubleshooting communications problems.....	30
Displaying message queues to solve communication problems.....	30
Displaying the Product Activity Log to solve communication problems.....	31
Displaying the Print Error Log to solve communication problems.....	31
Job logs and communication problems.....	31
Using communications trace to solve communication problems.....	32
Communications trace and communication problems.....	33
Trace CPI Communications (TRCCPIC) command.....	34
Using the system problem log to solve communication problems.....	35
Using the status information to solve communication problems.....	35
Using reason codes to solve communication problems.....	36
Considerations for system tuning during error recovery.....	38
Using error messages to aid in error recovery.....	39
Reference information.....	39
Example details: IBM i server to IBM i server connection using Enterprise Extender (HPRIP).....	39
Notices.....	41
Programming interface information.....	42
Trademarks.....	42
Terms and conditions.....	42

Getting started with IBM i communications

The IBM® i server can use many different networking technologies. Supported protocols include TCP/IP, APPC, APPN, HPR, asynchronous, and binary synchronous communications.

IBM i communications configuration is done by either manually or automatically creating a set of configuration objects that represent the local and remote systems that are to communicate. The types of objects that are created for a communications configuration depend on the type of communications being configured.

PDF file for Getting started with IBM i communications

You can view and print a PDF file of this information.


To view or download the PDF version, select [Getting started with IBM i communications](#).

Saving PDF files

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

1. Right-click the PDF link in your browser.
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**.

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](http://www.adobe.com/products/acrobat/readstep.html) (www.adobe.com/products/acrobat/readstep.html) .

Networking concepts

If you want more information about networking concepts, review these topics.

Advanced Peer-to-Peer Networking

Advanced Peer-to-Peer Networking (APPN) is one type of data communications support that is provided by the IBM i system. APPN support routes data between two or more advanced program-to-program systems in a network. The systems do not need to be directly connected, in the same network, or in adjacent networks.

The APPC/APPN support handles all of the SNA protocol requirements when your system is communicating with a remote system that uses the logical unit (LU) session type 6.2 and node type 2.1 architectures. The remote system can be any of the following systems:

- IBM i system
- System/36
- System/38
- IBM personal computer
- Displaywriter
- Series/1
- 5520 Administrative System
- RISC System/600 (Reduced Instruction Set Computer)

- DPPX/370 (Distribute Processing Programming Executive)
- One of the following host systems:
 - System/370
 - System/390®
 - 30XX processor
 - 43XX processor
 - 9370 system
 - Another system that supports the appropriate level of architecture

The IBM i APPN support is an enhancement to the SNA Node Type 2.1 architecture that supplies networking functions. These enhancements are easy-to-use, are dynamic; and give control of the network to the peer systems that make up the network. APPN provides you with the following advanced functions:

- Distributed directory services
- Dynamic route selection that is based on user-specified values
- Intermediate session routing
- Routing of data by using transmission priorities.

With the exception of intermediate session routing, HPR builds on and uses these APPN functions.

Related concepts

High-Performance Routing

High-Performance Routing (HPR) is an addition to Advanced Peer-to-Peer Networking (APPN). HPR enhances APPN data routing performance and reliability, especially when higher-speed lower-error links are used.

Advanced Program-to-Program Communication

Advanced Program-to-Program Communication (APPC) is a data communications support that allows programs on an IBM i server to communicate with programs on other systems that have compatible communications support.

APPC on the IBM i system provides an application programming interface to the Systems Network Architecture (SNA) logical unit (LU) type 6.2 and node type 2.1 architectures.

The APPC support handles all of the SNA protocol requirements when your system is communicating with a remote system that uses the LU type 6.2 and node type 2.1 architectures. You can connect your system to any other system that supports the APPC program interface. APPC application programs can also communicate over lines using TCP/IP.

The IBM i APPC support handles the protocol needed for communicating between an application program that runs on your IBM i system, and an application that runs on a remote system. The protocol consists of a set of verbs that are common to the local and remote systems in a network. However, the way in which each system provides a program interface to the verbs may differ.

The IBM i system provides the following program interfaces:

- The intersystem communications function (ICF) file interface. In ICF, the LU 6.2 verbs are carried out by using data description specifications (DDS) keywords and system-supplied formats.
- The Common Programming Interface (CPI) Communications call interface. Using CPI Communication calls carries the LU 6.2 verbs.
- The CICS® file interface. In CICS/400 support, the LU 6.2 verbs are carried out by using EXEC CICS commands.
- The sockets application programming interface (API). For the sockets API, the LU 6.2 verbs are carried out by using the socket functions.

The APPC support also handles networking functions, and allows peer systems in a network to start and end sessions without a controlling host system.

The IBM i Advanced Peer-to-Peer Networking (APPN) support is an enhancement to the node type 2.1 architecture. APPN provides additional networking functions such as searching distributed directories, dynamically selecting routes, routing of intermediate sessions, creating and starting remote locations, and routing data by using transmission priorities.

Built on APPN, High-Performance Routing (HPR) is an enhancement to APPN that enables improved availability and persistence during network outages.

Dependent logical unit requester

Dependent logical unit requester (DLUR) allows dependent secondary logical units (LU 0, 1, 2, and 3) an entry point into the APPN network. DLUR support gives the appearance of having an adjacent connection to Virtual Telecommunications Access Method (VTAM®), but allows traversing the APPN network through intermediate nodes.

DLUR supports the following controllers, displays, and printers:

- Host devices, including 3270 emulation (*EML), remote job entry (*RJE), and program-to-program communications (*PGM)
- SNA Passthrough upstream devices
- DHCF display devices
- NRF display and printer devices
- SNUF devices (DSNX)

The normal SSCP-PU and SSCP-LU flows for dependent LUs are encapsulated in a control point server (CP-SVR) pipe. This pipe consists of two LU 6.2 sessions:

- Send
- Receive

At the primary end of the pipe is a dependent logical unit server (DLUS). At the secondary end of the pipe is a DLUR. The DLUS and DLUR support the activation and deactivation of dependent physical units (PUs) and logical units (LUs) in the APPN network. The pipe consists of a pair of LU 6.2 conversations where two APPC applications (DLUR and DLUS) exchange dependent SNA SSCP flows. The flows are encapsulated in a general data stream (GDS) variable and sent in LU 6.2 logical records. The pair of conversations that are used to transmit encapsulated SNA is called the CP-SVR pipe.

Related tasks

[Configuring dependent logical unit requester](#)

Dependent logical unit requester (DLUR) allows dependent secondary logical units (LU 0, 1, 2, and 3) an entry point into the APPN network. DLUR support gives the appearance of having an adjacent connection to Virtual Telecommunications Access Method (VTAM), but allows traversing the APPN network through intermediate nodes.

Related reference

[Configuring the device descriptions](#)

You can create the device description using the Create Device Description (CRTDEV DSP) command.

High-Performance Routing

High-Performance Routing (HPR) is an addition to Advanced Peer-to-Peer Networking (APPN). HPR enhances APPN data routing performance and reliability, especially when higher-speed lower-error links are used.

To support high-speed communications facilities, certain changes to the APPN architecture are required. These changes allow switching in intermediate nodes to be done at a lower layer and enable faster switching than in base APPN support. HPR changes the existing APPN intermediate session routing by using automatic network routing (ANR), which minimizes the storage and processing requirements in intermediate nodes. Each outbound packet has a predetermined path through the network so that intermediate routing nodes do not need to store information about HPR sessions that flow through them.

Intermediate routing nodes in HPR route the data based on information that is contained within the packet.

Enterprise Extender (SNA over IP networks using HPR)

Enterprise Extender is a networking architecture that allows Systems Network Architecture (SNA) applications to run over IP networks using High Performance Routing (HPR). This is the preferred way to run SNA applications over IP networks with communications input/output adapters (IOAs), such as Gigabit Ethernet.

Enterprise Extender uses the following HPR option sets: 1401, 1402, 2006, and 2009. These option sets, as well as 1400, are described as follows.

The HPR function can operate under a base architecture, or can operate under the base architecture plus options. The RTP Tower (Rapid Transport Protocol) option has some performance capabilities that are not available under the base option. Here is a more thorough explanation of what architecture option is appropriate for you.

- **HPR Base option** (option set 1400): Its primary function is to provide automatic network routing (ANR). Products that only use this function can participate as intermediate nodes in one or more RTP connections. This type of implementation cannot be an endpoint of an RTP connection. An addition to the base option is HPR link-level error recovery. A system that supports high-speed links does not always require link-level error recovery. It is optional because when link-level error recovery is eliminated, faster communications might occur when high-quality data transmission is used.
- **RTP Tower option** (option set 1401): Implementations that support this option can act as an endpoint and are able to transport logical-unit to logical-unit session (LU-LU session) traffic across HPR networks by using RTP connections. An RTP connection can be made only between two systems that support RTP. That is, there can be a mix of systems only in a path of a given RTP connection through the network (systems that only support the HPR Base option and systems that support the HPR Tower option). However, at least two endpoints in the path must support the HPR Tower option; otherwise, APPN is used.

Note: An implementation that has the RTP Tower option also supports the base option. These systems can run as intermediate systems in the path.

- **Control Flows over RTP Tower option** (option set 1402): This option causes control-point to control-point sessions (CP-CP sessions) and route setup messages to flow over special RTP connections. CP-CP sessions are established between adjacent node pairs and are used to broadcast topology flows to the entire network so that every node has the topology for the entire network stored in its topology database. *Route setup messages* are request messages and reply messages that are used to obtain information about a route over which an RTP connection is established. The route setup request is sent by the origin node to the destination node over the exact route that is to be used. It stops at each intermediate node along the way to gather information associated with the forward path. The route setup reply is returned by the destination node after receiving the route setup request. The reply follows the same path as the request (in the reverse direction) and stops at each intermediate node along the way to gather information about the reverse path. When the origin node receives the reply, it uses the information to establish a new RTP connection or reroute an existing one.
- **Logical Data Link Control (LDLC) Support option** (option set 2006): LDLC is a Logical Link Control (LLC) type defined to be used with HPR networks in conjunction with the Control Flows over RTP Tower option (option set 1402) over reliable links that do not require link-level error recovery. LDLC is only used for Enterprise Extender links.
- **Native IP Data Link Control (DLC) option** (option set 2009): Native IP is a DLC option used with option sets 1400, 1401, 1402, and 2006 to allow you to take advantage of APPN and HPR functions, such as class of service (COS), and flow and congestion control that is based on adaptive rate in the IP environment. This option set contains the support for Enterprise Extender links.

Related concepts

Advanced Peer-to-Peer Networking

Advanced Peer-to-Peer Networking (APPN) is one type of data communications support that is provided by the IBM i system. APPN support routes data between two or more advanced program-to-program

systems in a network. The systems do not need to be directly connected, in the same network, or in adjacent networks.

Systems Network Architecture

In IBM networks, *Systems Network Architecture* (SNA) is the description of the logical structure, formats, protocols, and operational sequences for transmitting information through and controlling the configuration and operation of networks.

APPC, APPN, and HPR are some examples of the protocols included within SNA. They can be used to connect the IBM i server with other IBM systems or non-IBM systems, to connect remote controllers, and to maintain a high level of security on your system.

TCP/IP

TCP/IP is a set of network protocols that enables computers to share resources and to exchange information across a network.

TCP/IP allows hosts to communicate with each other regardless of the host or user's physical location, the operating system, or the network medium. TCP/IP operates in many different network environments, including the Internet and corporate intranets.

Related concepts

[TCP/IP setup](#)

Configuring your IBM i server for communications

Depending on the type of hardware, you need to complete these tasks to configure your IBM i server for communications.

Creating a line description

Line descriptions describe the physical line connection and the data link protocol to be used between the IBM i server and the network.

To create line descriptions, follow these steps:

1. Type the appropriate command from the following list on the IBM i system command line and press F4.

The command you should type depends on the type of line you are creating.

- Create Line Description (Ethernet) (CRTLINETH)
- Create Line Description (Asynchronous Communications) (CRTLINASC)
- Create Line Description (Binary Synchronous Communications) (CRTLINBSC)
- Create Line Description (Point to Point Protocol Communications) (CRTLINPPP)

2. Use the online help information to choose the correct parameter values.
3. Press Enter.

The line description is created.

Related concepts

[Matching IBM i system line description parameters for a host system](#)

You must match the line description parameters of the host system with IBM i system values. These IBM i system values are described in this table.

[Matching IBM i system line description parameters for a remote IBM i system](#)

You must coordinate line description parameters between the local and remote IBM i systems. This table shows those prompts and parameters that must be coordinated when you specify line descriptions for the local and remote IBM i systems.

Optimizing communications performance

Many factors can affect the performance of IBM i application programs. To achieve the best performance with your particular communications environment, review these topics.

Improving wide area network performance

To optimize the IBM i system performance for wide area networks, perform these tasks.

- Reduce the total number of frames by using larger frames.
- To take advantage of these large frame sizes, change the MAXFRAME parameter on the line description (LIND) to reflect the maximum value.
- Configure a WAN line as full-duplex to provide you with a higher throughput for applications that can take advantage of this mode. This can also provide higher throughput for multiple users.

The data rate for a given protocol might increase as the frame size increases. Under these circumstances, the central processing unit (CPU) does not do as much processing. Fewer and larger frames also make more efficient use of the communications line (higher effective data rate) because of fewer overhead bytes and line turn-arounds.

Adjusting WAN protocols for optimum communications performance

Wide area network (WAN) protocols affect the communications performance on the IBM i server.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Adjusting the WAN line speed for optimum communications performance

In many cases, the communications line is the largest contributor to the overall response time in the wide area network (WAN). Therefore, you must carefully plan and manage its performance. In general, having the appropriate line speed is the key consideration for gaining the best performance.

To adjust the line speed for your wide area network, perform these tasks:

- Check the difference in performance between half-duplex utilization and full-duplex utilization on the line description.
- For interactive environments, monitor line utilization and resulting interactive response times to determine overall line utilization.
- For large transfer environments, or for environments in which only a small number of users are sharing a line, increase line use to allow for acceptable response times.
- The CPU usage for fractional T1 support and other high-speed WAN connections is similar to any other line that runs the same type of work. As the speed of a line increases from a traditional low speed to a high-speed or full T1/E1/J1 speed, performance characteristics might change as follows:
 - With interactive transactions, performance might be slightly faster.
 - With a large transfer, performance might be significantly faster.
 - With a single job, performance might be too serialized to use the entire bandwidth.
 - With high throughput, performance is more sensitive to frame size.
 - With high throughput, performance is more sensitive to application efficiency.

Additional considerations for adjusting the wide area network line speed are the following:

- A common misconception about the line speed of each attached communications line is that the central processing unit (CPU) resource is used in a uniform fashion. Since the CPU resource might not be used

in a uniform fashion, exact statements cannot be made about the number of lines that any IBM i server model can support.

- Most communications applications use a lot of CPU resource (to process data, to support disk input and output) and communications line resource (to send and receive data or display I/O). The amount of line resource that is used is proportional to the total number of bytes that are sent or received on the line. Some additional CPU resource is used to process the communications software to support the individual sends (puts or writes) and receives (gets or reads). Communications input/output processor resource is also used to support the line activity.
- When a single job is running disk operations or doing non-overlapped CPU processing, the communications link is idle. If several sessions transfer data concurrently, then the jobs are more interleaved and make better use of the communications link.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[Performance Tools for IBM i PDF](#)

WAN configuration considerations for IOAs

When you configure a communications controller, you must consider both subsystem storage and aggregate line speed.

Subsystem storage is the amount of storage available on the communications controller. Aggregate line speed is the sum of the speeds of the individual lines that are attached to the communications controller.

The following information can help you understand network configuration considerations for input/output adapters (IOAs).

- You can attach multiple IOAs to an IBM i system. The maximum number of IOAs that can be attached is determined by the IBM i server model.
- Even though an IOA can support certain configurations, a given IBM i server model might not have enough system resource (for example, CPU processing capacity) to support the work load over the lines.
- The use of larger frames generally improves large transfer performance in terms of capacity for the communications IOA and in terms of system response time. The amount of time that the IOA spends processing a larger frame is only slightly more than the amount needed to process a smaller frame. If you use larger frames to transfer a single system message or block of data, the total number of frames required to complete the transfer is decreased.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[Performance Tools for IBM i PDF](#)

Improving local area network performance

To optimize the IBM i system performance for local area networks (LANs), perform these tasks.

Adjusting LANs for optimum communications performance

Improvements to the input/output adapters (IOAs) of the local area network (LAN) include increased central processing unit (CPU) time, IOA capacity, and offload of certain communications protocols to the IOA.

These improvements allow TCP/IP to offload to the IOA the cost of segmenting data and calculating checksums of the data to verify integrity.

The following information can help you understand the configuration considerations for local area networks.

- When several sessions use a line or LAN concurrently, the aggregate data rate might be higher than when only one session is used.
- To achieve good performance in a multi-user interactive LAN environment, you should manage the number of active users so that LAN media use does not exceed 50%. Operating at a higher utilization rate might decrease response time because of excess queueing time for the line. In a large transfer environment in which a small number of users contend for the line, a higher line use might still offer acceptable performance.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Adjusting LAN lines for optimum communications performance

You can configure these parameters on the line description (LIND) to improve system performance.

MAXFRAME on the line description (LIND)

Maximizing the frame size in a LAN environment supplies the best performance for large transfers. A large frame size does not negatively affect performance for small transfers. Configure both the IBM i system and the other link station for large frames. Otherwise, of the two maximum frame size values, the smaller is used when you transfer data. Bridges might also limit the maximum frame size.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Improving data path performance

To assess the performance of your data path, you might want to review these topics.

Considerations for subsystem configuration for error recovery performance

The configuration of subsystems has little effect on normal data path operations. However, multiple subsystems can provide multiple processes to do cleanup and recovery when error conditions occur, which can result in improved performance.

Each piece of work that runs on the IBM i system is called a job. Each job is a single, identifiable sequence of processing actions that represents a single use of the system. The basic types of jobs performed are interactive jobs, batch jobs, spooling jobs, autostart jobs, and prestart jobs.

Jobs that run in subsystems do all the work that is performed on the IBM i server. As the number of users on the system increases, you must consider how the communications and interactive subsystems should be configured.

- Consider limiting the number of devices that are serviced by a single subsystem. Between 200 and 300 devices for each subsystem are recommended. Use the following recommendations to divide these users:
 - The number of users in any given subsystem
 - The connectivity used to access the system
 - The type of work the users do
 - The geographic location of the users
- Create additional communications and interactive subsystems to split the work into multiple subsystems.
- The work that is performed in the QCMN subsystem is for connecting and disconnecting from the system. Error recovery considerations are important in the configuration of the communications subsystem.
- To prevent a subsystem from allocating a device, ensure that there are no workstation or type entries for the devices that you do not want to be allocated.
- Only use the AT(*ENTER) option if you must allow jobs to transfer into that subsystem.

- For each subsystem you have defined, you need to identify which users will run in which subsystems. Use the Add Work Station Entry (ADDWSE) command and the Remove Work Station Entry (RMVWSE) command. You can set up workstation entries that identify which devices that subsystem should allocate, as well as which devices a subsystem should not allocate.

Note: You can use the ADDWSE commands while the subsystem is active. However, subsystems do not reallocate device locks dynamically. Eventually, it might be necessary to end and restart the subsystems to have the device locks allocated to the chosen subsystem.

To specify the devices a communications subsystem **should** allocate:

```
ADDCMNE SBSD(libname/sbsname) DEV(devname*) MODE(modename)
```

To specify the devices a communications subsystem **should not** allocate:

```
ADDCMNE SBSD(libname/sbsname) DEV(devname*) MODE(modename) MAXACT(0)
```

Note: Database and file servers run only in QSERVER when running over APPC. Do not attempt to allocate sessions running over the QSERVER mode description. These servers can run over TCP/IP and only then can you run them in subsystems other than QSERVER.

See the following example for a way of configuring your communications subsystem.

Example: Communications subsystem configuration

1. Create a duplicate of QCMN:

```
CRTDUPOBJ OBJ(QCMN) FROMLIB(QSYS) OBJTYPE(*SBSD) TOLIB(MYLIB) NEWOBJ(MYCMN)
```

2. Set up the communication entries:

```
ADDCMNE SBSD(MYLIB/MYCMN) DEV(PC*)
ADDCMNE SBSD(MYLIB/MYCMN) DEV(PC*) MODE(QSERVER) MAXACT(0)
ADDCMNE SBSD(QSYS/QCMN) DEV(PC*) MODE(QPCSUPP) MAXACT(0)
```

3. Finally, you can update your system startup program to start your new subsystems automatically.

Related concepts

[Considerations for system tuning during error recovery](#)

The overall performance tuning by the system can play a significant role during error recovery. For example, you might need to change the machine pool if it is too small to avoid excessive error recovery time.

Communications performance considerations for interactive jobs

An interactive job uses a keyboard and a character-type display. If a job needs the user to type on the keyboard and the system displays character results, that job is probably interactive.

Interactive in this sense means that the job and the user depend on each other to get the work done.

To optimize communications performance for interactive jobs, consider the following points:

- Attach workstations through communications. This requires more CPU overhead than 5250 local workstations.
- Keep the line utilization below 30 percent for best performance when interactive users jobs are attached. This maintains predictable and consistent response times. Exceeding 50 to 60 percent line utilization will typically cause unacceptable response times.

If your system has interactive users jobs that are connected in many different ways, consider configuring your interactive subsystems to separate the users. 5250 display station pass-through, or Telnet are some examples of these types of connections that must be separated. When you configure interactive subsystems, identify how you want the interactive users jobs to be separated and create the appropriate subsystem descriptions.

During error recovery, when many users risk losing their sessions at one time, an interactive subsystem can be very busy performing device recovery. This device recovery can adversely affect the work of other users in the subsystem who would otherwise be unaffected by the failure. Therefore, you might need to change how the interactive subsystems are configured. However, multiple subsystems can provide multiple processes to do cleanup and recovery when error conditions occur. This can result in improved performance.

Use the following command to configure an interactive subsystem to allocate devices that begin with *devname** and present a sign-on display on those display devices:

```
ADDWSE SBSDB(libname/sbsname) WRKSTNDEV(devname*) AT(*SIGNON)
```

Use the following command to configure an interactive subsystem so that the device name *devname** is not allocated and a sign-on display does not appear.

```
ADDWSE SBSDB(libname/sbsname) WRKSTNDEV(devname*) AT(*ENTER)
```

If you specify AT(*ENTER) on the Add Work Station Entry (ADDWSE) command, you can use the Transfer Job (TFRJOB) command to allocate the interactive jobs that are associated with the work stations in this subsystem. If you do not need to allocate the interactive jobs that are associated with the work stations in this subsystem, you do not have to add the workstation entries with AT(*ENTER).

To specify the devices an interactive subsystem **should** allocate when the subsystem is started:

```
ADDWSE SBSDB(libname/sbsname) WRKSTN(devname*) AT(*SIGNON)
```

To specify the devices an interactive subsystem **should not** allocate when the subsystem is started:

```
ADDWSE SBSDB(libname/sbsname) WRKSTN(devname*) AT(*ENTER)
```

See the following example for a way of configuring your interactive subsystem.

Example: Interactive subsystem configuration

1. Create a subsystem description:

```
CRTSBSDB SBSDB(MYLIB/MYINTER) POOLS((1 *BASE) (2 *INTERACT))
```

2. Create a class

```
CRTCLS CLS(MYLIB/MYCLASS) RUNPTY(20)
```

3. add routing entries to your subsystem:

```
ADDRTGE SBSDB(MYLIB/MYINTER) SEQNBR(10) CMPVAL(QCMDI) PGM(QSYS/QCMD) POOLID(2)  
ADDRTGE SBSDB(MYLIB/MYINTER) SEQNBR(9999) CMPVAL(*ANY) PGM(QSYS/QCMD) POOLID(2)
```

4. Create a job queue, and add the job queue entry to your new subsystem:

```
CRTJOBQ JOBQ(MYLIB/MYJOBQ)  
ADDJOBQE SBSDB(MYLIB/MYINTER) JOBQ(MYLIB/MYJOBQ) MAXACT(200)
```

5. Set up the workstation name entries. Remove all the *ALL workstation type entries first, and then add the appropriate workstation name entries:

```
RMVWSE SBSDB(QSYS/QINTER) WRKSTNTYPE(*ALL)  
ADDWSE SBSDB(QSYS/QINTER) WRKSTN(QPADEV*)  
ADDWSE SBSDB(MYLIB/MYINTER) WRKSTN(PC*)
```

6. Finally, you can update your system startup program to start your new subsystems automatically.

Communications performance considerations for batch jobs

Batch jobs are predefined groups of processing actions that are submitted to the system to be performed with little or no interaction between the user and the system. You can tune batch jobs for optimized performance.

To optimize batch jobs for communications, consider the following points:

- Break the application into pieces and have multiple batch threads (jobs) operate concurrently.
- Reduce the number of open and close operations and input and output operations.
- Try to limit the number of communications input and output operations by doing fewer (and perhaps larger) application sends and receives when communications lines are used.
- If you have a considerable amount of main storage available, consider using the Set Object Access (SETOBJACC) command. This command preloads the complete database file, database index, or program into the assigned main storage pool if sufficient storage is available. The objective is to improve performance by eliminating disk-read/write operations.
- Block the data in the application. Try to place the application on the same system as the frequently accessed data.

Related reference

[Communications Management PDF](#)

Considerations for mixing interactive and batch jobs on a WAN line

When interactive jobs and large transfers are running on a communications line concurrently, you might need to change configuration parameters to work with interactive and batch jobs.

To keep interactive performance acceptable when mixing interactive and batch jobs on a wide area network (WAN) line, consider the following items:

- Use Advanced Peer-to-Peer Networking (APPN) transmission priority to prioritize interactive jobs over large transfers. This is the preferred method to transfer batch and interactive jobs.
- Change the request/response unit size to a lower value for large transfers. This parameter setting optimizes response time at the expense of large transfer performance.
- Reduce the pacing values for large transfers to slow the transfers down, which allows the interactive jobs more opportunities to get on the line.

Note: The overall central processing unit time increases for the large transfer.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Configuring Enterprise Extender communications for the IBM i system

Enterprise Extender is a networking architecture that allows Systems Network Architecture (SNA) applications to run over IP networks using High Performance Routing (HPR).

Enterprise Extender is the only way to run SNA applications over IP networks with communications input/output adapters (IOAs), such as Gigabit Ethernet.

Setting up the Enterprise Extender environment

Note: Before you configure HPR to run over IP networks, you must enable HPR transport support by specifying ALWHPRTWR(*yes) on the CHGNETA command.

To configure HPR to run over IP networks, follow these steps:

1. Configure your TCP/IP network.
2. Create an APPC controller description and specify the link type *HPRIP.

Related tasks

[Create an APPC controller description](#)

Configuring Enterprise Extender communications for the IBM i system

Enterprise Extender is a networking architecture that allows Systems Network Architecture (SNA) applications to run over IP networks using High Performance Routing (HPR).

Migrating from AnyNet to Enterprise Extender

Enterprise Extender is to be used in place of AnyNet®. To do the conversion, you must migrate the existing AnyNet configurations to the HPRIP controllers.

Note: Starting in IBM i 7.1, AnyNet is no longer supported. Users of AnyNet are encouraged to migrate to Enterprise Extender as a replacement.

To configure Enterprise Extender support to run over Point-to-Point Protocol (PPP) lines and perform switched line disconnect, refer to [Configure Enterprise Extender to perform switched line disconnect](#).

Consider the following factors when you do migration:

- While AnyNet could handle several remote links with only one controller, Enterprise Extender needs one HPRIP controller for each remote node that has a direct link. A new controller description must be created for each remote node that is being migrated from AnyNet.
- AnyNet provided low-entry networking (LEN) functions, and not full APPN support. HPRIP controllers provide full APPN functions. Remote definitions on the configuration lists need to be deleted.
- TCP host definition entries are no longer required for Enterprise Extender because the mapping is done in the controller definition.

To perform a migration from AnyNet to Enterprise Extender, see the following topics:

Migrating a system that can start AnyNet connections

Before you migrate the system to Enterprise Extender (HPRIP), make sure that the network configuration is as follows.

- One APPC controller of type *ANYNW, with a remote control point value of TCPIP.
- One entry on the QAPPNRMT configuration list for each remote node that defines the remote node and sets the control point name as TCPIP.
- One entry on the TCP host table that has the suffix SNA.IBM.COM, and that has the remote SNA host name and the remote network ID with the same suffix.

To migrate to Enterprise Extender (HPRIP), follow these steps:

1. For each entry on the host table that has an SNA.IBM.COM suffix, create one APPC controller of the link type *HPRIP and the IP address of the remote host using the Create Controller Description (APPC) (CRTCTLAPPC) command.
2. Delete the entry for the remote system in the QAPPNRMT configuration list using either the Remove Configuration List Entries (RMVCFGLE) or Work with Configuration Lists (WRKCFGL) (WRKCFGL CFGL(*APPNRMT)) command.

You can either keep or delete the entry on the host table.

Related reference

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\) command](#)

[Remove Cfg List Entries \(RMVCFGLE\) command](#)

[Work with Configuration Lists \(WRKCFGL\) command](#)

Migrating a system that cannot start AnyNet connections

Before you migrate the system to Enterprise Extender (HPRIP), make sure that the network configuration is as follows.

- One APPC controller of type *ANYNW

To migrate to Enterprise Extender (HPRIP), manually create an HPRIP controller for each remote node that needs to access the system.

Subsystems

A subsystem is a single, predefined operating environment through which the system coordinates work flow and resource usage.

IBM i can contain several independent operating subsystems. The run-time characteristics of a subsystem are defined in the subsystem description.

IBM supplies several subsystem descriptions that can be used with or without modification:

QINTER

Used for interactive jobs

QBATCH

Used for batch jobs

QBASE

Used for both interactive and communications batch jobs

QCMN

Used for communications batch jobs

QSERVER

Used for file server system

QSYSWRK

Used for general system work

QUSRWRK

Used to run TCP/IP server jobs that do work on behalf of a specific user.

You can also define a new subsystem using the Create Subsystem Description (CRTSBSD) command.

Related concepts

[Work management](#)

Communications applications

Communications applications that are used in an Advanced Program-to-Program Communication (APPC) environment can also be used in an Advanced Peer-to-Peer Networking (APPN) and High-Performance Routing (HPR) environment.

Only the method used to transport data is changed. APPC delivers the data from applications higher in the SNA layers down to APPN for transportation through the network.

When you encounter problems that indicate that the route to the remote location cannot be found, you can attempt to make the connection again using the Start Pass-Through (STRPASTHR) command.

Related concepts

[Solving remote communication problems using STRPASTHR](#)

User-written APPC applications

Advanced Peer-to-Peer Networking (APPN) performs many functions in a communications environment. Therefore, you must consider timeout parameters in APPC programs that use intersystem communications function (ICF). In particular, you might need to increase the WAITFILE parameter for these applications so that they do not time out when waiting for APPN functions to be performed.

The APPN function is not apparent to APPC programs. APPN takes advantage of the following routing functions:

- Nonadjacent nodes appear adjacent and so APPC programs can communicate directly to programs in nonadjacent nodes (without any APPC programs on the intermediate nodes).
- Performance is improved for APPC programs with session endpoints that are not physically adjacent in the network.

- APPC programs can communicate directly to programs in nodes in an adjacent APPN network through network nodes.

Distributed data management

Distributed data management (DDM) is a function of the operating system that allows an application program or a user on one system to use database files stored on remote systems. The system must be connected to a communications network and the remote systems must also use DDM.

With DDM on the IBM i server, application programs or users can perform the following tasks:

- Access data files that reside on remote systems (target systems). The remote systems can also access data files on the local IBM i system.
- An application can add, change, and delete data records in a file that exist on a target system.
- Create, delete, or rename files on a remote system.
- Copy a file from one system to another.

When DDM is in use, neither the application program nor the program user needs to know if the file that is needed exists locally or on a remote system. Remote and local file processing are essentially handled the same way.

Related concepts

[Distributed database programming](#)

Application programming interface performance considerations

An application programming interface (API) is an interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

To achieve better performance with your IBM i server, you need to consider the APIs available on the IBM i server.

To optimize Advanced Program-to-Program Communication (APPC) performance, consider the following points:

- Using larger record sizes for a large transfer provides a higher application data rate and decreases CPU time. With the larger record size, the CPU has less processing to do because fewer application read and write operations are needed to transfer the same amount of data.
- If a value of *CALC is selected for a maximum Systems Network Architecture (SNA) request/response unit (RU), the system selects an efficient size that is compatible with the frame size. The frame size is on the line description that you choose. Changing the RU size to a value other than *CALC might negate this performance feature.
- Compression with APPC must be used with caution and only for slower-speed wide area network (WAN) environments. It is suggested that compression might be used with speeds 19.2 kbps and slower.
- If you are doing tasks that include repetitive, small put operations, you can achieve better performance by using intersystem communications function (ICF) or Common Programming Interface (CPI) Communications.
- Set an appropriate value for the time that APPC waits for an acknowledgement from the remote system after sending the DETACH signal to end the transaction. You can specify the wait time by using QACRETRY and QACINTERVL data areas in QGPL library.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[CICS for i Series Administration and Operations Guide PDF](#)

Related information

[Maximum frame size \(MAXFRAME\) parameter](#)

Maximum length request/response unit size (MAXLENRU) parameter

Wait time (QACRETRY and QACINTERVL) data areas

Performance considerations for intersystem communications function

You can use the intersystem communications function (ICF) to write application programs to communicate with Advanced Program-to-Program Communication (APPC). ICF also provides program-to-device communications between the IBM i system and hardware devices.

You must determine which system is to send data first before you write the program. ICF data management handles the communication functions and the data for your program. In particular, ICF should be used to do tasks that include repetitive, small inputs.

To optimize ICF performance, consider the following factors:

- Eliminate unused record formats.
- Use separate record formats instead of multipurpose record formats with option indicators.
- Code to use the same record format for repeated operations.
- Set the maximum program devices equal to 1.
- Use a nonshared file.
- Use a separate indicator area.
- The use of the ICF keywords force data and confirm should be minimized.
- Use the Request to Send keyword only when necessary.
- Use the Invite Only keyword when soliciting input from multiple devices, otherwise use the Read keyword instead.
- If using the Invite keyword to solicit from multiple program devices, follow it with a Read-from-invited operation, not a Read operation.

To create device descriptions to set up your system for ICF, follow these steps:

1. Type the appropriate Create Device Description commands on the IBM i system command line and press F4.
2. Use the online help information to choose the parameter values.
3. Press Enter.

The device description is created.

Related reference

[ICF Programming PDF](#)

Performance considerations for Common Programming Interface Communications

You can use Common Programming Interface for Communications (CPI-C) to write application programs to communicate with Advanced Program-to-Program Communication (APPC).

The interface uses the System Network Architecture (SNA) LU (logical unit) 6.2 architecture to perform the following tasks:

- Establish a conversation
- Send and receive data
- Exchange control information
- End a conversation
- Notify a partner program of errors.

Intersystem communications feature (ICF) and CPI-C programs have similar performances for small data transfers.

To optimize CPI-C application programs, consider the following factors:

- Minimize the use of flush and confirm.
- Receive a compile record and parse it in your buffer.
- Do not use multiple receive calls to receive a single record.
- Use Request-to-Send only when necessary.

To add or change communications entries to get the system set up for CPI-C, do the following:

1. Type appropriate command on the IBM i system command line and press F4.
 - Add Communications Entry (ADDCMNE)
 - Remove Communications Entry (RMVCMNE)
 - Change Communications Entry (CHGCMNE)
2. Use the online help information to change, add, or remove parameter values.
3. Press Enter.

The communications entries are added, changed or removed.

Related reference

[CICS/400 Administration and Operations Guide PDF](#)

Communicating with a host system

You can configure the IBM i system to communicate with a host system by matching IBM i system parameters. Another option for IBM i system users is dependent logical unit requester (DLUR).

DLUR allows dependent secondary logical units (LU 0, 1, 2, and 3) an entry point into the APPN network. DLUR support gives the appearance of having an adjacent connection to Virtual Telecommunications Access Method (VTAM), but allows traversing the APPN network through intermediate nodes.

Matching IBM i system parameters for a host system

You can configure the IBM i system to communicate with a host system. This configuration requires the coordination of parameters and values.

The tables in the following topics only list those configuration prompts and parameters that require coordination between the IBM i system and the host system. In addition, some of the parameters that are listed might not apply to your particular configuration.

For information about configuring host systems, see the manuals *VTAM Installation and Resource Definition*, SC23-0111, and *Network Control Program Resource Definition Reference*, SC30-3254.

Matching IBM i system line description parameters for a host system

You must match the line description parameters of the host system with IBM i system values. These IBM i system values are described in this table.

For information about configuring host systems, see the manuals *VTAM Installation and Resource Definition*, SC23-0111, and *Network Control Program Resource Definition Reference*, SC30-3254.

You can specify some host system parameters on multiple definition statements, such as the GROUP, LINE, PU, and LU definition statements. The following table lists only the lowest-level definition statement that is used by the host system.

IBM i prompt	IBM i parameter	Host definition statement	Host parameter
Connection type	CNN	GROUP	DIAL If the IBM i line description CNN parameter is *SWTPP or *SHM, DIAL=YES must be specified for the host system; if CNN is *MP or *NONSWTPP, DIAL=NO must be specified. If CNN(*MP) is specified, the SERVICE macroinstruction must be used to specify the sequence in which stations are served.
Line speed	LINESPEED	LINE	SPEED Line speeds specified for each system must match.
Maximum frame size	MAXFRAME	PU	MAXDATA Values specified for each system must match.
NRZI data encoding	NRZI	LINE	NRZI Values specified for each system must match.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related tasks

[Creating a line description](#)

Line descriptions describe the physical line connection and the data link protocol to be used between the IBM i server and the network.

Matching IBM i system controller description parameters for a host system

You must match the controller description parameters of the host system with IBM i system values. These IBM i system values are described in this table.

For information about configuring host systems, see the manuals *VTAM Installation and Resource Definition*, SC23-0111, and *Network Control Program Resource Definition Reference*, SC30-3254.

You can specify some host system parameters on multiple definition statements, such as the GROUP, LINE, PU, and LU definition statements. The following table lists only the lowest-level definition statement that is used by the host system.

IBM i prompt	IBM i parameter	Host definition statement	Host parameter
Destination service access point	DSAP	PORT	SAPADDR For 9370/LAN only, the IBM i controller description DSAP must match the host SAPADDR parameter. The SAPADDR parameter is a decimal value (4-252); the IBM i value is specified as a 2-digit hexadecimal number.

IBM i prompt	IBM i parameter	Host definition statement	Host parameter
Local exchange identifier	LCLEXCHID	PU	IDBLK, IDNUM For parallel connections only. Required if the IBM i system specifies RMTCPNAME(*ANY), SWITCHED(*YES), and LINKTYPE is *SDLC. The LCLEXCHID specified must match the values specified in the switched major node definition PU macro instruction.
Maximum frame size	MAXFRAME	GROUP	MAXDATA Values specified for each system must match.
Remote control point name	RMTCPNAME	VTAMLST	SSCPNAME Required only if APPN(*YES). IBM i controller description value must match SSCPNAME specified in the Virtual Telecommunications Access Method (VTAM) start options list (ATCSTRyy).
Remote network identifier	RMTNETID	VTAMLST	NETID Required only if APPN(*YES). IBM i controller description value must match NETID specified in the VTAM start options list (ATCSTRyy).
Source service access point	SSAP	PU	SAPADDR For 9370/LAN only, the IBM i controller description DSAP must match the host SAPADDR parameter. The SAPADDR parameter is a decimal value (4-252); the IBM i value is specified as a 2-digit hexadecimal number.
SSCP identifier	SSCPID	VTAMLST	SSCPID Required if APPN(*YES) or if RMTCPNAME is not specified. IBM i controller description value must match SSCPID specified in the VTAM start options list (ATCSTRyy). The SSCPID parameter is a decimal value (0-65535); the IBM i value is specified as a 12-digit hexadecimal number, of which the first 2 digits are 05.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Matching IBM i system device description parameters for a host system

You must match the device description parameters of the host system with IBM i system values. These IBM i system values are described in this table.

For information about configuring host systems, see the manuals *VTAM Installation and Resource Definition*, SC23-0111, and *Network Control Program Resource Definition Reference*, SC30-3254.

You can specify some host system parameters on multiple definition statements, such as the GROUP, LINE, PU, and LU definition statements. The following table lists only the lowest-level definition statement that is used by the host system.

IBM i prompt	IBM i parameter	Host definition statement	Host parameter
Local location name	LCLLOCNAME	DFHTCT	NETNAME IBM i LCLLOCNAME value must match CICS/VS terminal control table NETNAME parameter and the label used on the LU definition statement.
Local location address	LOCADR	LU	LOCADDR Values specified for each system must match. The LOCADDR parameter is a decimal value (0-255); the IBM i value is specified as a 2-digit hexadecimal number.
Location password	LOCPWD	DFHTCT	BINDPWD Values specified for each system must match.
Dependent location name	DEPLOCNAME	LU	LU This parameter is only used for dependent logical unit requester (DLUR) support. This value is optional. If specified, it must match LUNAME received on ACTLUREQUEST.
Mode description name	MODE	MODEENT	LOGMODE IBM i mode description name must be defined in the host logon mode table using the LOGMODE parameter on the MODEENT macro instruction. The mode name must also be included in the CICS/VS terminal control table (DFHTCT) MODENAM parameter.
Remote location name	RMTLOCNAME	LU	LOGAPPL Values specified for each system must match.
Remote network identifier	RMTNETID	BUILD	NETID Values specified for each system must match.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Matching IBM i system mode and class-of-service description parameters for a host system

You must match the mode and class-of-service description parameters of the host system with IBM i system values. These IBM i system values are described in this table.

For information about configuring host systems, see the manuals *VTAM Installation and Resource Definition*, SC23-0111, and *Network Control Program Resource Definition Reference*, SC30-3254.

You can specify some host system parameters on multiple definition statements, such as the GROUP, LINE, PU, and LU definition statements. The following table lists only the lowest-level definition statement that is used by the host system.

IBM i prompt	IBM i parameter	Host definition statement	Host parameter
Mode description name	MODD	MODEENT	LOGMODE IBM i mode description name specified on the IBM i CRTMODD command (MODD parameter) must be defined in the host logon mode table using the LOGMODE parameter on the MODEENT macro instruction. The mode name must also be included in the CICS/VS terminal control table (DFHTCT) MODENAM parameter.
Class-of-service description name	COSD	MODEENT	COS IBM i class-of-service description name specified on the IBM i Create Class-of-Service Description (CRTCOSD) command (COSD parameter) and CRTMODD command (COS parameter) must be defined in the host logon mode table using the COS parameter on the MODEENT macro instruction. The class-of-service description must also be defined in the VTAM class-of-service table.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Configuring dependent logical unit requester

Dependent logical unit requester (DLUR) allows dependent secondary logical units (LU 0, 1, 2, and 3) an entry point into the APPN network. DLUR support gives the appearance of having an adjacent connection to Virtual Telecommunications Access Method (VTAM), but allows traversing the APPN network through intermediate nodes.

Note: DLUR uses logmode CPSVRMGR. This is created internally as part of the APPN and DLUR support. If CPSVRMGR exists as a user-defined logmode on any of the systems in your network, it must be deleted. Use the Work with Mode Descriptions (WRKMODD) command and specify the option to delete CPSVRMGR.

To configure the IBM i server to communicate with DLUR, follow these steps:

1. Configure a host controller description.
2. Configure device descriptions.
3. Verify that an APPN connection into the network exists (host or APPC controller with *YES specified for the APPN parameter).

Related concepts

Dependent logical unit requester

Dependent logical unit requester (DLUR) allows dependent secondary logical units (LU 0, 1, 2, and 3) an entry point into the APPN network. DLUR support gives the appearance of having an adjacent connection to Virtual Telecommunications Access Method (VTAM[®]), but allows traversing the APPN network through intermediate nodes.

Configuring the host controller description

You can create the host controller description using the Create Controller Description (SNA Host) (CRTCTLHOST) command.

If you have already created a controller description for such functions as 3270 emulation or NRF, you must change the link type to *DLUR by following these steps:

1. Retrieve the configuration description for the dependent logical unit requester (DLUR) controller description using the Retrieve Configuration Source (RTVCFGSRC) command.
2. Edit the member to change the link type to *DLUR.
3. Convert the source to a CL program.
4. Create the CL program using the Create CL Program (CRTCLPGM) command.
5. Delete the configuration using the Delete Controller Description (DLTCTLD) command.
6. Call the CL program to create the new configuration.

Some of the parameters on the Create Controller Description (SNA Host) (CRTCTLHOST) command are explained as follows:

Local exchange identifier

Matches the ID block and ID number parameters from the PU definition on Virtual Telecommunications Access Method (VTAM).

Dependent PU name

Matches the name of the PU specified on the PU definition on VTAM.

Note: If the local exchange identifier and the dependent PU name are specified, both must match the definitions on VTAM. If both parameter values do not match, the ACTPU will be rejected.

If the *DIAL value is specified for the INLCNN parameter, the primary dependent logical unit server (DLUS) name (PRIDLUS), and either the local exchange identifier (LCLEXCHID), or the dependent PU name (DEPPUNAME) must be specified.

Control point name and network identifier for the primary DLUS name

Matches the SSCP name and NETID parameters on the VTAM startup options.

Related reference

[Configuring the device descriptions](#)

You can create the device description using the Create Device Description (CRTDEV DSP) command.

Configuring the device descriptions

You can create the device description using the Create Device Description (CRTDEV DSP) command.

Here is an explanation of the DEPLOYNAME field on the Create Device Description (CRTDEV DSP) display:

Dependent location name

Matches the LU name on the LU definition on Virtual Telecommunications Access Method (VTAM).

Note: This must match the VTAM LU name with the corresponding local location address (LOCADDR) on VTAM.

Related concepts

[Dependent logical unit requester](#)

Dependent logical unit requester (DLUR) allows dependent secondary logical units (LU 0, 1, 2, and 3) an entry point into the APPN network. DLUR support gives the appearance of having an adjacent connection to Virtual Telecommunications Access Method (VTAM®), but allows traversing the APPN network through intermediate nodes.

Related tasks

[Configuring the host controller description](#)

You can create the host controller description using the Create Controller Description (SNA Host) (CRTCTLHOST) command.

Example: Enterprise Extender Configuration between IBM i and Host (VTAM)

These CL commands are used to define the Enterprise Extender configuration for an IBM i to Host (Virtual Telecommunications Access Method (VTAM)) by using Dependent Logical Unit Requester/Server (DLUR/DLUS).

DLUR/DLUS allows for dependent logical units (LU) types 0, 1, 2, and 3 and an entry point into an APPN network and ultimately a means to connect over Enterprise Extender. The dependent LU data is encapsulated into an independent session.

The following requirements are necessary to configure this environment:

- DLUR/DLUS configuration
- Enterprise Extender (EE) HPR over IP configuration

This configuration combines the DLUR and EE configurations so the following controller descriptions are related to the entire configuration.

- The controller to which the dependent devices are attached is the Host DLUR controller with **LINKYPE(*DLUR)**.
- The controller that is configured for Enterprise Extender is the APPC (HPR over IP) controller.
- Enterprise Extender uses Virtual APPN, so the APPC device is attached to the virtual APPN controller.

The example shows the CL commands for configuration on the IBM i. This example shows one dependent LU used for 3270 emulation and does not show all possible LU configurations.

Note: By using the following code examples, you agree to the terms of the [Code license and disclaimer information](#).

Note: When an APPC device is configured for an address other than 00, it is implied that it is a dependent device. EE must be APPN capable. If it is APPN capable, that means that any configuration must support APPN, hence an APPC device under an EE/HPRIP controller must be independent. The exception is a Host controller. It supports dependent device under it, so an APPC device with an address other than 00 can be specified. However, when a Host controller is set up for EE, it becomes a DLUR. By definition, a Host DLUR controller cannot support an APPC device.

Note: When EE is configured across a network that traverses firewalls, ensure UDP ports 12000 - 12004 are open to successfully establish a connection.

The sample IBM i CL commands use the following Host system configuration.

VTAM Startup Parameters:

```
1 NETID = APPN
2 SSCPNAME = MFRAME
```

Host HPR Configuration:

```
NAME = HPRIP, TYPE = XCA MAJOR NODE
STATUS= ACTIV, DESIRED STATE= ACTIV
MEDIUM = HPRIP
TCP/IP JOB NAME = NETTCPIP
3 LOCAL IP ADDRESS 199.199.199.10
VNNAME = IBMLVRN.INTERNAL VNGROUP = EEGROUP
RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
HPRIP AC/R 21 NO 90750000000000000000000017100808080
I/O TRACE = OFF, BUFFER TRACE = OFF
VTAMTOPO = REPORT, NODE REPORTED - YES
LINES:
EEL000 ACTIV
EEL001 ACTIV
EEL002 ACTIV
EEL003 ACTIV
EEL004 ACTIV
EEL005 ACTIV
```

DLUR Node Configuration:

```
4 DLRMFRAME PU ADDR=01,
5 IDBLK=056,
5 IDNUM=66400,
ISTATUS=ACTIVE,
MAXPATH=1
6 DLR3270 LU LOCADDR=4, 7
MODETAB=LOGMOD38
LOGAPPL=MYSIGNON
```

VTAM Switched Major Node:

```
SWASFE3 VBUILD TYPE=SWNET, REQUIRED PARAMETER
MAXNO=8, MAX OF 8 DIAL NUMBERS MAY BE DEFINED
MAXGRP=5 MAX OF 5 LINE GROUPS MAY BE DEFINED

SWFE3A PU ADDR=C1, 8 BIT STATION UNIQUE ADDRESS
DISCNT=NO, ACCEPT DISCONTACT FROM PU (DEFAULT)
8 CPNAME=IBMI, IDBLK=056, IDNUM=66400
MAXDATA=1994, IBM i MAXDATA FOR TOKEN-RING
MAXPATH=8, MAX OF 8 PATHS FOR CALL OUT
1 NETID=APPN, IBM i LOCAL NETID VALUE
DWACTION=YES, ATTEMPT CONNECTION AT ACTIVATION
MAXOUT=7, 7 PIUS TO DEVICE BEFORE RESPONSE
PACING=7, NCP SENDS 7 TO LU AND MARKS FIRST
PASSLIM=7, 7 CONTIGUOUS PIUS MAX AT ONE TIME
PUTYPE=2, VALID FOR TYPE 2 OR 2.1 SUPPORT
ISTATUS=ACTIVE, ACTIVATE WITH MAJOR NODE
MODETAB=LOGMOD38, VTAM LOGN MODE TABLE ON S/370
SSCPFM=USSSCS, VTAM TO STRIP/ADD MEDIA CONTROL CHAR
VPACING=7, VTAM SENDS 7 TO NCP AND MARKS FIRST

7 SWFE3A04 LU LOCADDR=04, CORRESPONDS TO IBM i LOCADDR 04
ISTATUS=ACTIVE
```

Create a TCP interface on the line description ETHLINE that has a route to the host system's IP address **3**.

```
ADDTCPIFC INTNETADR('199.199.199.5') LIND(ETHLINE)
SUBNETMASK('255.255.255.128')
```

Change the network attributes and set the local network ID **1** and local control point name **8** to the same values as the host system.

```
CHGNETA LCLNETID(APPN) LCLCPNAME(IBMI)
ALWVRTAPPN(*NO) ALWHPRTWR(*YES)
```

Create the HPR over IP controller description, HOSTCTL **9**, on the IBM i with the local IP address, remote host network identifier **1**, remote host control point **2**, and remote host IP address **3**.

```
CRTCTLAPPC CTLD(HOSTCTL) LINKTYPE(*HPRIP) ONLINE(*NO)
RMTNETID(APPN) RMTCPNAME(MFRAME)
```

```
RMTINTNETA('199.199.199.10')
LCLINTNETA('199.199.199.5')
```

Create the DLUR host controller description using the controller description, HOSTCTL **9**, with the host's exchange identifier **5**, control point name **2**, network identifier **1**, and dependent PU name **4**.

```
CRTCTLHOST CTLD(HOSTCTL) LINKTYPE(*DLUR)
LCLEXCHID(05666400) PRIDLUS(MFRAME APPN)
DEPPUNAME(LRMFRAME)
```

Create the device description for the host with the local location address **7**, remote location **6**, and attached controller **9**.

```
CRTDEVHOST DEVD(DEVHOSTD) LOCADR(04) RMTLOCNAME(DLR3270)
CTL(HOSTCTL) APPTYPE(*EML) EMLDEV(3278)
```

Related reference

- [Add TCP/IP Interface \(ADDTCPIFC\)](#)
- [Change Network Attributes \(CHGNETA\)](#)
- [Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Related information

- [Create Ctl Desc \(SNA Host\) \(CRTCTLHOST\)](#)
- [Create Device Desc \(SNA Host\) \(CRTDEVHOST\)](#)

Communicating with a remote IBM i server

Using Advanced Program-to-Program Communication (APPC), you can configure the IBM i server to communicate with another IBM i server. This configuration requires the coordination of configuration parameters and values.

The tables in the following topics only list those configuration prompts and parameters that require coordination on both the local and remote IBM i servers. In addition, some of the parameters that are listed might not apply to your particular configuration.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Matching IBM i system line description parameters for a remote IBM i system

You must coordinate line description parameters between the local and remote IBM i systems. This table shows those prompts and parameters that must be coordinated when you specify line descriptions for the local and remote IBM i systems.

To configure a local IBM i server to a remote IBM i server:

- See [“Examples: Connecting one IBM i server to another IBM i server”](#) on page 28 for an example of connecting one IBM i server to another IBM i server.
- Use the following table for the line descriptions.

IBM i system prompt	IBM i parameter	Remote IBM i parameter	Notes
Data bits per character	BITSCHAR	BITSCHAR	Values specified for each system must match.
Duplex	DUPLEX	DUPLEX	Depending on the type of communications used, the values specified for the DUPLEX parameters may need to be coordinated.

IBM i system prompt	IBM i parameter	Remote IBM i parameter	Notes
Ethernet standard	ETHSTD	ETHSTD	Values specified for each system must be coordinated. Both systems must specify the same standard (*ETHV2 or *IEEE8023) or at least one system must specify *ALL.
Line speed	LINESPEED	LINESPEED	For asynchronous lines, the line speeds specified for each system must match.
Local network address	NETADR	CNNNBR	For switched virtual circuits (SVCs), the NETADR parameter on the local system line description must match the CNNNBR parameter on the <i>controller description</i> for the remote system.
NRZI data encoding	NRZI	NRZI	Values specified for each system must match (*YES or *NO).
Data link role	ROLE	ROLE	The value specified for the local system line description ROLE parameter should match the controller description ROLE parameter specified at the remote system.
Number of stop bits	STOPBITS	STOPBITS	Values specified for each system must match.
Switched connection type	SWTCNN	SWTCNN	Values specified for each system must be compatible. (*DIAL or *ANS must not be specified for <i>both</i> systems.)

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related concepts

Examples: [Connecting one IBM i server to another IBM i server](#)

When you connect a local IBM i server to a remote IBM i server, you must coordinate the configuration parameters.

Related tasks

[Creating a line description](#)

Line descriptions describe the physical line connection and the data link protocol to be used between the IBM i server and the network.

Matching IBM i system controller description parameters for a remote IBM i system

You must coordinate controller description parameters between the local and remote IBM i systems. This table shows those prompts and parameters that must be coordinated when you specify controller descriptions for the local and remote IBM i systems.

To configure a local IBM i server to a remote IBM i server:

- See [“Examples: Connecting one IBM i server to another IBM i server”](#) on page 28 for an example of connecting one IBM i server to another IBM i server.
- Use the following table for the controller descriptions.

IBM i prompt	IBM i parameter	Remote IBM i parameter	Notes
Destination service access point	DSAP	SSAP	DSAP specified for the local IBM i server must match the SSAP specified in the remote IBM i controller description.

IBM i prompt	IBM i parameter	Remote IBM i parameter	Notes
Exchange identifier	EXCHID	EXCHID	If used, the local IBM i controller description EXCHID must match the remote IBM i line description EXCHID. The first three digits of the exchange identifier, known as the block number, is 056 for the IBM i line. You can use the WRKLIND command to determine this value.
Initial connection	INLCNN	INLCNN	Values specified for each system must be coordinated; INLCNN(*ANS) must not be specified for both systems.
Remote control point name	RMTCPNAME	LCLCPNAME	RMTCPNAME specified on the local IBM i system controller description must match the local control point name specified in the network attributes of the remote IBM i system.
Remote network identifier	RMTNETID	LCLNETID	RMTNETID specified on the local IBM i server controller description must match the local network ID specified in the network attributes of the remote IBM i server.
Data link role	ROLE	ROLE	The value specified for the local IBM i controller description ROLE parameter must match the remote IBM i line description ROLE value.
Source service access point	SSAP	DSAP	SSAP specified for the local IBM i system must match the DSAP specified in the remote IBM i controller description.

Note: For asynchronous controllers (CRTCTLASC command), if the remote system controller description specifies RMTVIFY(*YES), the local system controller description must specify a local identifier (LCLID parameter) and local location name (LCLLOCNAME parameter). The remote system must also create a configuration list with the LCLID and LCLLOCNAME values from the local system controller description.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related concepts

Examples: Connecting one IBM i server to another IBM i server

When you connect a local IBM i server to a remote IBM i server, you must coordinate the configuration parameters.

Matching IBM i system device description parameters for a remote IBM i system

You must coordinate device description parameters between the local and remote IBM i systems. This table shows those prompts and parameters that must be coordinated when you specify device descriptions for the local and remote IBM i systems.

To configure a local IBM i server to a remote IBM i server:

- See [“Examples: Connecting one IBM i server to another IBM i server”](#) on page 28 for an example of connecting one IBM i server to another IBM i server.
- Use the following table for the device description.

IBM i prompt	IBM i parameter	Remote IBM i parameter	Notes
	LCLLOCNAME	RMTLOCNAME	<p>For systems not using APPN (APPN(*NO) specified for the controller and device descriptions), this value must match the value specified by the RMTLOCNAME parameter on the remote system device description.</p> <p>APPC device descriptions are automatically created as needed by the IBM i server APPN support, when the following is specified on the controller description:</p> <ul style="list-style-type: none"> • APPN(*YES) • AUTOCRTDEV(*ALL)
Location Password	LOCPWD	LOCPWD	<p>This parameter must match on both the local and remote APPC device.</p> <p>Note: If you want a value other than *NONE for APPN devices, this value must be configured on the QAPPNRMT configuration list.</p>
Mode	MODE	MODE	<p>For systems not using APPN (APPN(*NO) specified for the controller and device descriptions), this value must match the value specified by the MODE parameter on the remote device description.</p> <p>For systems using APPN (APPN(*YES) specified for the controller and device descriptions), the specified mode description must exist on the remote system. The mode description name need not be specified in the remote device description.</p>
Remote location name	RMTLOCNAME	LCLLOCNAME	<p>For systems not using APPN (APPN(*NO) specified for the controller and device descriptions), this value must match the value specified by the LCLLOCNAME parameter on the remote device description.</p> <p>APPC device descriptions are automatically created as needed by IBM i APPN support if APPN(*YES) is specified for the controller description.</p>
	RMTNETID	LCLNETID	<p>RMTNETID specified on the local IBM i server device description must match the local network ID specified in the network attributes of the remote IBM i server.</p>
Single session	SNGSSN	SNGSSN	<p>For Element 1 (single-session device description), this parameter must match on both the local and remote APPC device. For Element 2 (number of single-session conversations), this parameter does not need to match the remote device.</p> <p>Note: If you want a value other than *NO for APPN devices, this value must be configured on the QAPPNRMT configuration list.</p>

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related concepts

[Examples: Connecting one IBM i server to another IBM i server](#)

When you connect a local IBM i server to a remote IBM i server, you must coordinate the configuration parameters.

Examples: Connecting one IBM i server to another IBM i server

When you connect a local IBM i server to a remote IBM i server, you must coordinate the configuration parameters.

Related concepts

Matching IBM i system line description parameters for a remote IBM i system

You must coordinate line description parameters between the local and remote IBM i systems. This table shows those prompts and parameters that must be coordinated when you specify line descriptions for the local and remote IBM i systems.

Matching IBM i system controller description parameters for a remote IBM i system

You must coordinate controller description parameters between the local and remote IBM i systems. This table shows those prompts and parameters that must be coordinated when you specify controller descriptions for the local and remote IBM i systems.

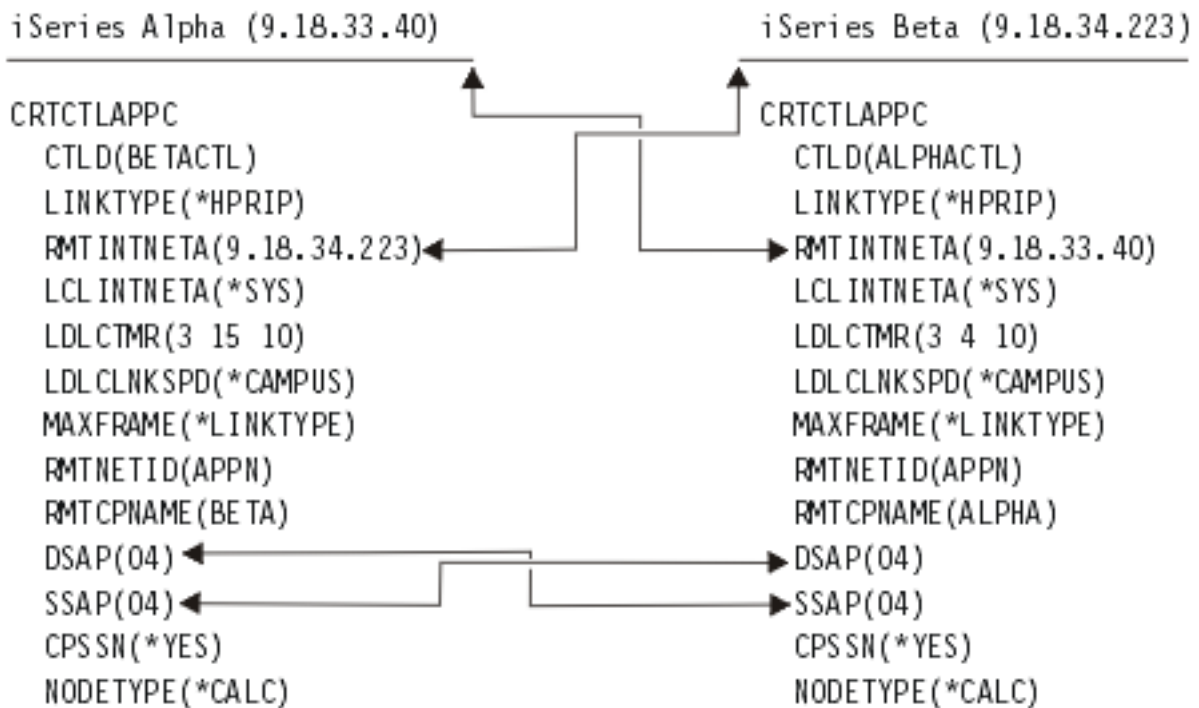
Matching IBM i system device description parameters for a remote IBM i system

You must coordinate device description parameters between the local and remote IBM i systems. This table shows those prompts and parameters that must be coordinated when you specify device descriptions for the local and remote IBM i systems.

Example: IBM i server to IBM i server using Enterprise Extender (HPRIP)

This figure shows the matching parameters between two IBM i servers that are connected by using Enterprise Extender.

Read a description of this figure at [“Example details: IBM i server to IBM i server connection using Enterprise Extender \(HPRIP\)”](#) on page 39.



RV4T219-0

Figure 1. IBM i server to IBM i server using Enterprise Extender (HPRIP)

Common networking standards

Local area network standards and wide area network standards are the two types of common networking standards that are supported by the IBM i system.

Local area network standards

A *local area network* (LAN) is a communications system that allows the interconnection and sharing of resources between independent devices within a moderately sized geographic area.

Ethernet networks

Ethernet is one type of local area network (LAN) topology that the IBM i operating system supports. IBM i Ethernet provides support for the Ethernet Version 2 and the IEEE 802.3 standards.

Half-duplex Ethernet

Generally, multiple stations in an Ethernet network show a single data path. Therefore, only one station may transmit data at a time. This is called half-duplex Ethernet. The station may transmit only or receive only, but not both simultaneously.

Full-duplex Ethernet

Full-duplex Ethernet enables stations to simultaneously send and receive data on the network, eliminating collisions. This is accomplished through the use of a full-duplex LAN switch. Ethernet switching splits a large Ethernet into smaller segments. Full-duplex Ethernet requires the following:

- Twisted-pair cable transmission medium
- Ethernet network interface cards
- A full-duplex LAN switch

Full-duplex 10 Mbps Ethernet has simultaneous 10 Mbps receiving and 10 Mbps sending paths.

Fast Ethernet

Fast Ethernet standard (IEEE 802.3U) increases Ethernet by operating speeds from 10 Mbps to 100, half or full duplex. The IBM i Ethernet adapters support 100BASE-TX network devices that use category 5 shielded and unshielded twisted-pair (STP, UTP) cable.

Related concepts

[Ethernet](#)

Wide area network standards

A *wide area network* (WAN) is a data communications network that serves an area of hundreds or thousands of miles. Public and private packet-switching networks and national telephone networks are examples of WANs.

Asynchronous communications

IBM i *asynchronous communications* support allows an IBM i application program to exchange data with a remote system or device using an asynchronous (start-stop) line.

Asynchronous communications support includes file transfer support (also used with other communications types) and interactive terminal facility (ITF). Asynchronous communications support provides program-to-program and program-to-device communications between systems that use asynchronous (start-stop) lines.

With asynchronous communications support, you can send data to and receive data from a remote program or device using either an asynchronous (start-stop) line. Your application program must provide the data stream that is required by the remote device. IBM i application programs can be written in the Integrated Language Environment® (ILE) C/400*, ILE COBOL/400*, ILE FORTRAN/400*, or ILE RPG/400* programming languages.

Related reference

[Asynchronous Communications Programming PDF](#)

Binary synchronous communications

Binary synchronous communications (BSC) is a data communications line protocol that uses a standard set of transmission control characters and control character sequences to send binary-coded data over a communications line.

Binary synchronous communications equivalence link (BSCCEL) support is the intersystem communications function (ICF) support on the IBM i system. BSCCEL support provides binary synchronous communications with a remote system or device. BSCCEL also supplies online and batch communications between application programs on different BSC systems. IBM i application programs can be written in the Integrated Language Environment® (ILE) C/400®, ILE COBOL/400, ILE FORTRAN/400, or ILE RPG/400® programming languages.

Related reference

[BSC Equivalence Link Programming PDF](#)

Troubleshooting communications problems

If you have a problem with the communications connectivity, you can use a set of tools that the IBM i system provides to analyze the problem.

The following topics provide some common tools for communications problem analysis. You can use these tools to identify and solve communication problems.

Displaying message queues to solve communication problems

Message queues receive messages that relate to communications failures. A message lists possible causes of the problem and additional information that depends on the problem and the suggested problem analysis tool.

To display message queues, follow these steps:

1. On the IBM i system command line, type `DSPMSG MSGQ(XXXX)`, in which `XXXX` is:
 - The message queue identified by the `QCFGMSGQ` system value
 - The default value is `QSYSOPR`
 - Or, message queue if the system value has been changed
 - For lines, controllers, and devices which support the `MSGQ` parameter, the message queue is specified in the configuration object
 - For display devices, the message queue that matches the device name
2. Press the Enter key.
3. In the Display Message display, read the messages pertaining to communications problems that are displayed in the message queue.

The object name in the message directs you to the communications objects in error.
4. For messages in the queue with an `*` in the farthest left position, press `F14` to perform additional tests.

This calls the Work with Problems tool.

Related concepts

[Using the system problem log to solve communication problems](#)

Error conditions that are communications related can make entries in the system problem log. You can access the log to see the lists of problems that are detected by the system or by the user.

[Communications trace and communication problems](#)

You might sometimes need to obtain an error log printout or communications trace data that your IBM service representative can review. For the line trace, someone familiar with the protocol used on the line might need to review the files.

[Message queues](#)

Related reference

[Job logs and communication problems](#)

A variety of job logs might contain information that can help you determine the cause of a communications problem. Many of these logs contain messages that can help you understand what the system has done concerning your communications functions.

Displaying the Product Activity Log to solve communication problems

The Product Activity Log provides you with important information for solving communications problems.

To display or print the Product Activity Log, follow these steps:

1. On any IBM i command line, enter STRSST (Start System Service Tools), enter your service tools user ID and password, and press Enter.
2. From the System Service Tools menu, select option 1 to display or print the Product Activity Log.

Related concepts

[History logs](#)

[Communications trace and communication problems](#)

You might sometimes need to obtain an error log printout or communications trace data that your IBM service representative can review. For the line trace, someone familiar with the protocol used on the line might need to review the files.

Related reference

[Communications Management PDF](#)

Displaying the Print Error Log to solve communication problems

The Print Error Log provides you with important information for solving communications problems.

To view the Print Error Log, follow these steps:

1. On any IBM i command line, enter PRTERLOG (Print Error Log), and press Enter.

The command places a formatted printer file of the machine error log in a spooled printer file that is named QPCSMPT or in a specified output file.

2. Find and read these error logs.

A variety of job logs might contain information that helps you determine the cause of a communications problem. For a detailed description of these job logs, see the following topic.

Related reference

[Communications Management PDF](#)

Job logs and communication problems

A variety of job logs might contain information that can help you determine the cause of a communications problem. Many of these logs contain messages that can help you understand what the system has done concerning your communications functions.

The following are some of the most useful jobs to review when you have a communications problem:

QSYSARB

System arbiter. This job log is for devices, and communications in general. It also contains ONLINE at IPL messages.

QSYSCOMM1

Communications and input/output system job. This job log is for problem logging and for local area network (LAN) manager messages. It also contains ONLINE at IPL messages for network servers and their lines.

QCMNARB01 through QCMNARB99

Communications arbiter. These job logs contain information for communications startup, take-down, and error recovery.

QLUS

Logical unit services.

QLUR

Logical unit (LU) 6.2 resynchronization job. This job log is for two-phase commit synchronization processing.

QPASVRP

Target 5250 display station pass-through primary server job. This job log is for target pass-through communications functions.

QPASVRS

Target 5250 display station pass-through secondary server job. These contain more detailed messages for target pass-through communication functions

Subsystem jobs (QINTER and QCMN)

Interactive subsystem and communication subsystem. These job logs are for subsystem jobs.

Related tasks

Displaying message queues to solve communication problems

Message queues receive messages that relate to communications failures. A message lists possible causes of the problem and additional information that depends on the problem and the suggested problem analysis tool.

Related reference

[Remote Work Station Support PDF](#)

Using communications trace to solve communication problems

Sometimes, program debugging tasks are easier if you can trace the data that is sent and received on the communications line or within the network server.

To perform a communications trace, you must have IBM *SERVICE special authority or be authorized to the Service Trace function of the IBM i operating system through IBM Navigator for i.

The following commands can be used to perform a communications trace.

Start Service Tools (STRSST)

The STRSST command takes you to a menu of tools to obtain error log information and communications trace information. For a detailed description of system service tools, see the following topic of *Communications trace and communication problems*.

Start Communications Trace (STRCMNTRC)

The STRCMNTRC command starts a communications trace for the specified line, network interface description, or network server description. The communications trace continues until one of the following occurs:

- The system runs the End Communications Trace (ENDCMNTRC) command
- A physical line problem causes the trace to end
- The Communications Trace function of the STRSST command ends the trace
- The *STOPTRC parameter is specified, and the buffer becomes full

End Communications Trace (ENDCMNTRC)

The ENDCMNTRC command ends the trace currently running on the specified line, network interface description, or network server description. The ENDCMNTRC command saves the communications trace buffer and the associated System Licensed Internal Code (SLIC) data.

Print Communications Trace (PRTCMNTRC)

The PRTCMNTRC command writes the communications trace data for the specified line, network interface description, or network server description to a spooled file or a database file. The system can print trace data multiple times in either form, and parameters on the command allow for dividing and formatting of the data.

Delete Communications Trace (DLTCMNTRC)

The DLTCMNTRC command deletes the communications trace buffer and associated SLIC data for the specified line, network interface description, or network server description. The communications trace can be deleted once the trace has ended.

Check Communications Trace (CHKCMNTRC)

The CHKCMNTRC command returns the communications trace status for a specific line, network interface description, or network server description. The CHKCMNTRC command returns status for all of the traces of a specific type that exist on the system. The system returns the status through a message.

Trace CPI Communications (TRCCPIC)

You can start to trace Common Programming Interface (CPI) communications either before running a job or after a job is active to find out where the error might have occurred. The TRCCPIC command captures information about CPI-Communications calls that is processed by your program.

For more information on how to access System Service Tools, see the *Backup and Recovery* book.

Related reference

[Security reference](#)

[Recovering your system](#)

Related information

[Performing a communications trace](#)

Communications trace and communication problems

You might sometimes need to obtain an error log printout or communications trace data that your IBM service representative can review. For the line trace, someone familiar with the protocol used on the line might need to review the files.

To perform a communications trace, you must have IBM *SERVICE special authority or be authorized to the Service Trace function of the IBM i operating system through IBM Navigator for i.

Use the communications trace function in the following situations:

- Message information or other problem analysis is not sufficient to identify a problem
- Communications support personnel suspects a protocol error
- Whether the system sends and receives valid data needs to be verified.

You can trace multiple lines from each workstation by using the communications trace option. The system traces a maximum of two lines on the same communications controller subsystem at the same time. Only one trace can exist for the same configuration object at the same time. The system supports all line speeds and protocols.

For more information about these tests, contact your IBM service representative.

Related tasks

[Displaying message queues to solve communication problems](#)

Message queues receive messages that relate to communications failures. A message lists possible causes of the problem and additional information that depends on the problem and the suggested problem analysis tool.

[Displaying the Product Activity Log to solve communication problems](#)

The Product Activity Log provides you with important information for solving communications problems.

Related reference

[Security reference](#)

Trace CPI Communications (TRCCPIC) command

You can start to trace Common Programming Interface (CPI) communications either before running a job or after a job is active to find out where the error might have occurred. The Trace CPI Communications (TRCCPIC) command captures information about CPI-Communications calls that are processed by your program.

The system collects trace information in a current job or in a job that is serviced by the Start Service Job (STRSRVJOB) command. (For a CPI Communication program, you can trace a job that is started as a result of a received program start request.) You can issue the TRCCPIC command in one of the following ways:

- Using the System Menu
- Typing TRCCPIC *ON on a command line
- Adding the TRCCPIC command to a control language (CL) or a REstructured eXtended eXecutor (REXX) program
- Typing TRCCPIC on the command line and pressing F4 (Prompt)

If you enter TRCCPIC on the command line and press F4, an initial prompt is displayed for the Trace Option Setting. If you specify *ON and press enter, the Trace CPI Communications display opens.

You can use this display to set the following parameters:

Trace option setting

Specifies whether the collection of trace information is to be started, stopped, or ended.

***ON**

Starts Trace CPI Communications. This is the default value for the command.

***OFF**

Stops Trace CPI Communications. The current information is written to the spooled printer file or to the database file, and the trace table. The trace information is then deleted.

***END**

Ends Trace CPI Communications. The trace table and all trace information are destroyed.

Maximum storage to use

Specifies the maximum amount of storage to use for the trace information collected. The prompt appears only if you have selected *ON for the *Trace option setting* prompt.

200 K

The number of bytes (1 K equals 1024 bytes) of storage. This is the default value.

1-16000 K

The valid range for the maximum number of bytes used for storing collected trace information.

Trace full

Specifies whether new trace records replace old trace records or whether the trace is stopped when the maximum storage that you specified has been reached. This prompt appears only if you have selected *ON for the *Trace option setting* prompt.

***WRAP**

When the trace storage area is full, new trace information is written over the old trace information, starting at the beginning of the storage area. This is the default value.

***STOPTRC**

No new trace information is saved when the trace storage area is full. You must reissue the TRCCPIC command, specifying (*OFF) for the SET parameter, to retrieve the output of the trace information collected in the trace storage area.

User data length

Specifies the maximum length of user data to be saved for each trace record in the storage area. This prompt affects only the tracing of user data on the Send_Data and Receive calls. This parameter does not affect the tracing of log data on Set_Log_Data, Send_Error, or Deallocate calls. This prompt appears only if you specified *ON on the *Trace option setting* prompt.

128

The number of bytes for the user data length. This is the default value.

0-4096

The valid range of bytes for the user data length.

Trace CPI Communications continues to collect trace records until you stop the trace or until the trace storage area becomes full. The amount of trace storage depends on the value that is specified on the *Trace full* prompt. If the trace storage area becomes full and the collection of trace records stops, you must enter the TRCCPIC command again to create output. The output that is created by the TRCCPIC command is directed either to the spooled printer file, QSYSPRT, or to a database output file that you specify. If the output file that you specify already exists, it must have the same attributes as the system-supplied file, QACMOTRC.

You can stop a trace procedure in one of the following ways: If you type TRCCPIC on the command line and press F4, an initial prompt is displayed for the *Trace Option Setting*.

- Using the System Menu
- Typing TRCCPIC *OFF on the command line
- Adding the TRCCPIC command to a CL or a REXX program
- Typing TRCCPIC on the command line and pressing F4 (Prompt)

If you specify *OFF, you are prompted for the OUTPUT parameter.

Using the system problem log to solve communication problems

Error conditions that are communications related can make entries in the system problem log. You can access the log to see the lists of problems that are detected by the system or by the user.

To access the system problem log, type WRKPRB on any IBM i system command line, and press F4.

Tip: You can select a subset of the problems that are listed in the problem log by selecting the problem status. A problem that is listed in the log has one of the following statuses:

- OPENED: The problem was identified; problem analysis has not been run.
- READY: The system has run problem analysis; the problem is ready to be prepared.
- PREPARED: The system added information that relates to the problem.
- SENT: The problem was sent to the service support location.

You can also sort the WRKPRB display by the date when the problem was entered into the log.

Note: Use the WRKPRB command for the menu options, additional problem analysis, or documenting problem records.

Related tasks

Displaying message queues to solve communication problems

Message queues receive messages that relate to communications failures. A message lists possible causes of the problem and additional information that depends on the problem and the suggested problem analysis tool.

Using the status information to solve communication problems

You can diagnose the communications problem by checking the communications status. The status information for network servers, network interfaces, lines, controllers, or devices might represent the symptom of the problem.

To check and change the communication configuration on the system, follow these steps:

1. Enter the Work with Configuration Status (WRKCFGSTS) command on any IBM i command line.
2. Press F4. The Work with Configuration Status display opens.
3. Specify the configuration type for the CFGTYPE parameter.
4. Specify the configuration description for the CFGD parameter.

Note: You can sort the list that is produced by the WRKCFGSTS command by the status of the objects using the STATUS parameter. For example, if you want to see only the failed objects, specify STATUS(*FAILED).

Using reason codes to solve communication problems

When the local system rejects an incoming program start request, a message is sent to the message queue of the system operator. You can use the message information to determine why the program start request was rejected.

See the following table for an explanation of the reason codes for failed program start requests.

Table 1. Reason codes for rejected program start requests

Reason code	Reason description
401	Program start request received to a device that is not allocated to an active subsystem.
402	Requested device is currently being held by a Hold Communications Device (HLDCMNDEV) command.
403	User profile is not accessible.
404	Job description is not accessible.
405	Output queue is not accessible.
406	Maximum number of jobs defined by subsystem description are already active.
407	Maximum number of jobs defined by communications entry are already active.
408	Maximum number of jobs defined by routing entry are already active.
409	Library on library list is exclusively in use by another job.
410	Group profile cannot be accessed.
411	Insufficient storage in machine pool to start job.
412	System value not accessible.
413	QSERVER not started
501	Job description was not found.
502	Output queue was not found.
503	Class was not found.
504	Library on initial library list was not found.
505	Job description or job description library is damaged.
506	Library on library list is destroyed.
507	Duplicate libraries were found on library list.
508	Storage-pool defined size is zero.
602	Transaction program-name value is reserved but not supported.
604	Matching routing entry was not found.
605	Program was not found.
704	Password is not valid.
705	User is not authorized to device.
706	User is not authorized to subsystem description.

Table 1. Reason codes for rejected program start requests (continued)

Reason code	Reason description
707	User is not authorized to job description.
708	User is not authorized to output queue.
709	User is not authorized to program.
710	User is not authorized to class.
711	User is not authorized to library on library list.
712	User is not authorized to group profile.
713	User ID is not valid.
714	Default user profile is not valid.
715	Neither password nor user ID was provided, and no default user profile was specified in the communications entry.
718	No user ID.
722	A user ID was received but a password was not sent.
723	No password was associated with the user ID.
725	User ID does not follow naming convention.
726	User profile is disabled.
730	Password has expired.
801	Program initialization parameters are present but not allowed.
802	Program initialization parameter exceeds 2000 bytes.
803	Subsystem is ending.
804	Prestart job is inactive or is ending.
805	WAIT(NO) was specified on the prestart job entry and no prestart job was available.
806	The maximum number of prestart jobs that can be active on a prestart job entry was exceeded.
807	Prestart job ended when a program start request was being received.
901	Program initialization parameters are not valid.
902	Number of parameters for program not valid.
903	Program initialization parameters required but not present.
1001	System logic error. Function check or unexpected return code encountered.
1002	System logic error. Function check or unexpected return code encountered while receiving program initialization parameters.
1501	Character in procedure name not valid.
1502	Procedure not found.
1503	System/36 environment library not found.
1504	Library QSSP not found.
1505	File QS36PRC not found in library QSSP.
1506	Procedure or library name is greater than 8 characters.

Table 1. Reason codes for rejected program start requests (continued)

Reason code	Reason description
1507	Current library not found.
1508	Not authorized to current library.
1509	Not authorized to QS36PRC in current library.
1510	Not authorized to procedure in current library.
1511	Not authorized to System/36 environment library.
1512	Not authorized to file QS36PRC in System/36 environment library.
1513	Not authorized to procedure in System/36 environment library.
1514	Not authorized in library QSSP.
1515	Not authorized to file QS36PRC in QSSP.
1516	Not authorized to procedure in QS36PRC in QSSP.
1517	Unexpected return code from System/36 environment support.
1518	Problem phase program not found in QSSP.
1519	Not authorized to problem phase program in QSSP.
1520	Maximum number of target programs started (100 per System/36 environment).
2501	System logic error. Function check or unexpected return code encountered while processing a program start request.
2502	Temporarily unable to allocate needed resources for a program start request.
2503	No subsystem accepting program start requests for this device.

Considerations for system tuning during error recovery

The overall performance tuning by the system can play a significant role during error recovery. For example, you might need to change the machine pool if it is too small to avoid excessive error recovery time.

- Performance Adjustment

You can use the QPFRADJ system value to specify when to automatically adjust memory pools and activity levels. The system can then automatically adjust the performance of the system based on this system value. The QPFRADJ system value is set to 2 (periodically after restart) when the system is shipped. Automatic adjustment might be a desirable feature, particularly when unexpected loads hit the system. Automatic adjustment can help the system perform better through these peak loads.

- Subsystem considerations

You should consider dividing communications users (whether they are remote workstation users or APPC communication users) into multiple subsystems. If communications fail, all users in a single subsystem might be affected by the communications recovery that is performed on their systems.

Related concepts

[Considerations for subsystem configuration for error recovery performance](#)

The configuration of subsystems has little effect on normal data path operations. However, multiple subsystems can provide multiple processes to do cleanup and recovery when error conditions occur, which can result in improved performance.

Related reference

[Performance system values: Automatically adjust memory pools and activity levels](#)

Using error messages to aid in error recovery

When problems occur in communications, you can look for error messages and additional information in many places to help resolve the problems.

See the following topics for the most common places to look for error information.

- Messages queues, see [“Displaying message queues to solve communication problems” on page 30](#)
- Job logs, see [“Job logs and communication problems” on page 31](#)
- Other logs, see [“Displaying the Product Activity Log to solve communication problems” on page 31](#) and [“Displaying the Print Error Log to solve communication problems” on page 31](#)
- Start Service tools, see [“Communications trace and communication problems” on page 33](#)
- Communications trace, see [“Using communications trace to solve communication problems” on page 32](#)

Reference information

These topics provide details for the example topics in this topic collection.

Example details: IBM i server to IBM i server connection using Enterprise Extender (HPRIP)

This table describes the relationships between the local IBM i system values and the remote IBM i values when Enterprise Extender (HPRIP) is used. The values shown and described here are example values.

Figure 1 on page 28 shows the matching parameters between the two IBM i servers using Enterprise Extender.

Local IBM i system (ALPHA) parameter name and value	IBM i parameter value description	Remote IBM i (BETA) system values
Controller description: CTLD = BETACTL	The value for this local IBM i parameter matches the name for the remote IBM i name, and vice versa.	Controller description: CTLD = ALPHACTL
Controller description: LINKTYPE = *HPRIP	The values for IBM i parameter LINKTYPE must be *HPRIP on both systems to use Enterprise Extender.	Controller description: LINKTYPE = *HPRIP
Controller description: RMTINTNETA = 9.18.34.223	The value for this local IBM i parameter matches the value for the remote IBM i IP address, and vice versa.	Controller description: RMTINTNETA = 9.18.33.40
Controller description: RMTCPNAME = BETA	The value for this local IBM i parameter matches the value for the remote IBM i CP name, and vice versa.	Controller description: RMTCPNAME = ALPHA
Controller description: RMTNETID = ALPHA	The value for this local IBM i parameter matches the value for the remote IBM i network ID if they are in the same network.	Controller description: RMTNETID = ALPHA

Table 2. Local IBM i system value relationships to remote IBM i system values (continued)

Local IBM i system (ALPHA) parameter name and value	IBM i parameter value description	Remote IBM i (BETA) system values
Controller description: DSAP = 04	The value for this local IBM i parameter matches the value for the remote IBM i parameter SSAP.	Controller description: DSAP = 04
Controller description: SSAP = 04	The value for this local IBM i parameter matches the value for the remote IBM i parameter DSAP.	Controller description: SSAP = 04

Note: The following fields are added for Enterprise Extender:

- RMTINTNETA : Remote internet address. An IPv4 internet address in the format of xxx.xxx.xxx.xxx..
- LCLINTNETA: Local internet address. An IPv4 internet address in the format of xxx.xxx.xxx.xxx. If the value of *SYS is not selected, the output traffic for this controller will go through the defined IP address. Otherwise, the output traffic will use the TCP/IP routing tables to determine the best interface to send the data.
- LDLCTIMR: LDLC timers. These will control the timers required by the LDLC protocol to retry a send, the number of retries, and how often a test frame will be sent to the remote. A test frame will only be sent to the remote system if data hasn't been received from the remote system during that time period.
- LDLCLNKSPD: This gives the APPN value of link speed. Possible values are: *CAMPUS (4 Mbps), *WAN (56 kbps), 10 Mbps, 4 Mbps, 16 Mbps, 100 Mbps, *MAX (greater than 100Mbps), and *MIN (less than 1200 bps). APPN uses this value to calculate the weight of this particular link when calculating the path of any request.
- LDLCTMSGRP: The rest of the APPN values are used to calculate the link weight: cost per connection, cost per byte, link security, and propagation delay. All of these values are used by APPN to calculate the best route to a remote host. These values only appear when the LDLCLNKSPD parameter has a value other than the default.

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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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
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 document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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.

COPYRIGHT LICENSE:

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

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

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

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

Programming interface information

This Getting started with IBM i communications publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i.

Trademarks

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

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Other product and service names might be trademarks of IBM or other companies.

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.



Product Number: 5770-SS1