# Connect:Direct®
# for Windows

## System Guide

**Version 4.4**

*Connect:Direct for Windows System Guide*
**Version 4.4**

**First Edition**

(c) Copyright 1995-2007 Sterling Commerce, Inc. All rights reserved. Additional copyright information is located at the end of the release notes.

**STERLING COMMERCE SOFTWARE**

**\*\*\*TRADE SECRET NOTICE\*\*\***

# Contents

## Chapter 4  Stopping Connect:Direct for Windows                             47

## Chapter 5  Creating Processes                                             49

## Chapter 8  Managing a Connect:Direct Server                                                    109

## Chapter 9  Troubleshooting                                                                      117

## Appendix A  Worksheets                                                                          125

## Appendix B  Changing Connect:Direct for Windows Settings                                        133

# About Connect:Direct for Windows

Connect:Direct for Windows is an enterprise solution for secure, automated file delivery between business applications. Its session management and extended recover features help assure data delivery in support of 24 hours a day, 7 days a week business operations.

Connect:Direct for Windows links technologies, moves all types of information, and manages high-performance transfers by providing easy-to-use automation, checkpoint/restart, and error recovery. Connect:Direct offers choices in communications protocols, hardware platforms, and operating systems. It provides the flexibility to move information from mainframes to midrange systems to desktop and LAN-based workstations.

## Using Nodes to Transfer Data

Each data transfer involves a local and a remote node. The active Connect:Direct server is the local node and the remote node is the partner node. The two servers (local and remote) function together to perform the work. Either Connect:Direct node initiates the work. You can submit Processes from a remote node to run on your local Connect:Direct node.

When Connect:Direct establishes a session between the local and a remote node, one is the initiator of the session and the other serves as the partner and has a secondary function. The node that initiates the session has primary control, regardless of the direction of information flow. Connect:Direct must be installed on both the local node and the remote node.

The following figure illustrates the distinction between primary control and local/remote nodes. Process monitors and statistics reports running on the local Windows node show both Processes as

they affect the local system. You are able to determine whether the local node is the initiating node or the receiving node.

Process 1: A Connect:Direct for Windows user attaches to the server and submits a Process to run on the Windows node. The Process copies data from the Windows node to the z/OS node. The Windows node is the local node and has primary control because it initiates the activity.

Local Node                         Remote Node

User

Connect:Direct for Windows         Connect:Direct for z/OS

Primary Control                    Secondary Control
(PNODE)                            (SNODE)

Process 2: A Connect:Direct for z/OS user submits a Process to run on the z/OS node. The Process copies data from the Connect:Direct for Windows node to the z/OS node. The z/OS node has primary control because it initiates the activity and is called the PNODE.

Local Node                         Remote Node

Connect:Direct for Windows         Connect:Direct z/OS

User

Secondary Control                  Primary Control
(SNODE)                            (PNODE)

# Planning to Use Connect:Direct for Windows

Before you use Connect:Direct for Windows, plan how you will use it to perform your work. Complete the following tasks to use Connect:Direct for Windows:

✦ Configure Connect:Direct for Windows

✦ Define the work to perform

✦ Perform and manage the work that is performed

For most tasks, you have a choice on the tool to use to create the component. Connect:Direct Requester is one of the available tools. Its graphical interface makes it easy to configure Connect:Direct for Windows and create Connect:Direct Processes. Because you can use this tool to perform all functions necessary to use Connect:Direct for Windows, this document provides instructions on using Connect:Direct Requester to create and submit Connect:Direct Processes, view Process activity and output, and perform administrative tasks such as editing the network map, modifying initialization parameters, and defining user authorizations.

## Configuring Connect:Direct for Windows

Before you begin using Connect:Direct for Windows, first configure your environment. To prepare to use Connect:Direct to submit Processes, perform the following tasks:

✦ Configure the Local Node—Configure a local definition for each server you install at the local site, define a user ID for each user who will access the local node, define user authorities for each user ID and if remote users need to access the local network, define remote user proxies. Use Connect:Direct Requester to configure the local node.

When you copy data from a local domain to another domain on a network, the Domain user ID and password must be exactly the same on both domains and a trust between the two domains must exist.

✦ Configure the Network Map—The network map identifies the remote nodes that each local node can communicate with and the communications information needed to establish a connection. Use Connect:Direct Requester or the Configuration tool to configure the network map.

## Defining the Work to Perform

After you determine what tasks to perform, you define Processes and work lists. A Process provides instructions, called *statements*, for transferring files, running programs, submitting Processes on the remote node, and altering the sequence of Process step execution. It also controls the execution of the Process. Use Connect:Direct Requester for Windows to define Processes.

You create Processes and store them in a file for repeated use. You can customize a Process when you submit it by temporarily changing its controlling parameters (such as the remote node) and by using symbolic substitution variables. You can submit Processes to any Connect:Direct node. The local node and a remote node work together to execute the Process.

If you want to define the work to perform but you are not ready to submit the work for execution, define a work list. A work list is a document containing a sequential list of Process submit requests. It can contain the four types of ad hoc Processes that you can submit to the Connect:Direct server: Send/Receive File, Submit Process, Run Task, or Run Job. Using work lists can save you time and effort by automating routine or repetitive submission tasks. You can build a work list to periodically submit related work as a single work flow. Your work list can serve as a library of related or unrelated Processes.

## Performing the Work

Once you define the Processes necessary to perform the work, you are ready to begin transferring files. You make requests through a computer that accesses the Connect:Direct server across a communications network. Files that you want to transfer must reside on or be accessible to the server.

The following events occur when you begin the file transfer process:

✦ The commands you issue are passed to the appropriate API.

✦ The API verifies that they are syntactically correct and that they contain all the required information to execute the request. It then packages and ships the commands across the network to the server.

✦ Command processors on the server either execute the commands or pass them on to the main components of the server where the actual data transfer, statistics management, and other services take place.

✦ The Connect:Direct server packages the results of the command and routes them back across the network to the originating workstation. You can then use the management tools to display the results of the Process.

Use Browser User Interface, Connect:Direct Requester, or the Command Line Interface to control Processes.

## Managing the Work

As Processes are submitted, the Transmission Control Queue (TCQ) holds information about Connect:Direct Processes that are currently executing or scheduled to execute in the future. The TCQ contains four logical queues: Execution, Wait, Timer, and Hold.

Connect:Direct places the Process in the appropriate queue based on Process control parameters that affect scheduling. Connect:Direct selects Processes in a priority-based manner for execution as sessions are available.

Connect:Direct for Windows provides tools to allow you to manage Processes. These tools include:

✦ Process Monitor—Use this tool to view Processes in the Transmission Control Queue (TCQ), release held Processes, change the status of a Process, and delete a Process.

✦ Process Notification Utility—Use this utility to change the notification method you defined when you installed Connect:Direct for Windows to notify users of Process execution.

✦ Windows Event Logging—Use this utility to log informational, warning, and error messages that are generated.

✦ Message Lookup—If you need to troubleshoot the meaning of an error message, use this utility to view more explanation about an error message.

✦ SNMP—If you want to use Microsoft Windows SNMP to capture messages, you must identify which messages you want to include and determine if messages are trapped or logged to the event log.

✦ CRC checking—If you want to enable CRC checking, a CRC is generated for the entire buffer, including the header. The SNODE calculates the CRC value and compares it to the CRC value in the TCP header. If the values do not match, the Process is terminated. When Secure+ Option is enabled during a session, CRC checking is disabled.

✦ CLI—The command line interface provides commands to access queues and manage Processes. These commands enable you to control Process execution, view Process status and results, and affect the Connect:Direct server. Issue these commands through or in a native command text format through the Applications Programming Interface (API).

✦ Connect:Direct Browser User Interface—Connect:Direct Browser User Interface allows you to build, submit, and monitor Connect:Direct Processes from an Internet browser, such as Microsoft Internet Explorer.

You can also perform Connect:Direct system administration tasks, such as viewing and changing the network map or initialization parameters, from Connect:Direct Browser. The specific administration tasks that you can perform depend on the Connect:Direct platform that your browser is signed on to and your security level.

Connect:Direct Browser is distributed on CD-ROM with Connect:Direct for z/OS, Connect:Direct for Windows, Connect:Direct for UNIX, Connect:Direct for HP NonStop, and Sterling Control Center. It can also be downloaded from the Sterling Commerce Web site. Connect:Direct Browser is installed on a Web server and can be accessed by administrators and users through a URL. The following example shows the page used to graphically build a Process:
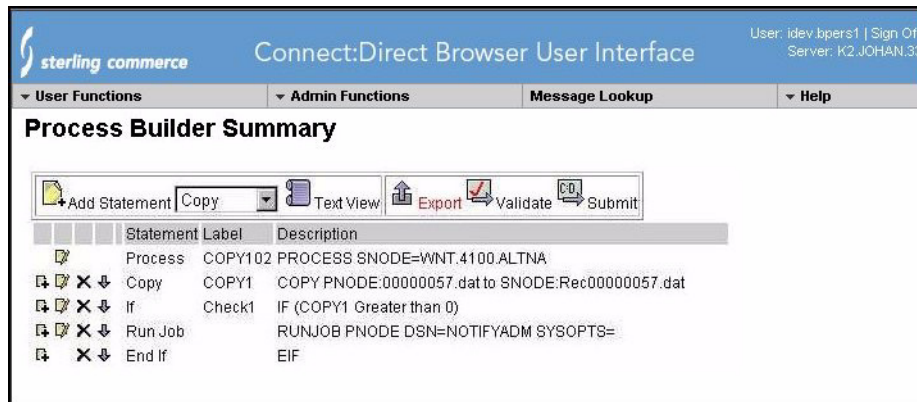


To learn more about Connect:Direct Browser, see the documentation on the Connect:Direct Browser CD-ROM or available online from the Sterling Commerce Documentation Library.

✦ Transmission Control Queue—Connect:Direct stores submitted Processes in the Transmission Control Queue (TCQ). The TCQ is divided into four logical queues: Execution, Wait, Timer, and Hold. Processes are run from the Execution queue. Connect:Direct places a Process in the appropriate queue based on Process statement parameters, such as the **hold**, **retain**, and **startt** parameters.

Connect:Direct runs Processes based on their priority and when the Process is placed in the Execution queue. Higher priority Processes are selected for execution ahead of Processes with a lower priority. You can access the queues and manage the Processes through Connect:Direct commands.

✦ File Agent—Connect:Direct File Agent is a feature of Connect:Direct which provides unattended file management. File Agent monitors *watched* directories to detect new files. When File Agent detects a new file, it either submits a default Process or evaluates the file using rules to override the default Process and to determine which Process to submit. You create rules to submit different Processes based on the following properties:

◆ Specific or partial file names

◆ File size

◆ System events

You create the Processes used by File Agent on Connect:Direct; you cannot create them using File Agent.

To achieve optimum performance, configure File Agent to communicate with the Connect:Direct node where it is installed. File Agent can be installed on UNIX, Windows, and z/OS operating systems. For information to help you plan how to implement File Agent, see the *Managing Files with Connect:Direct File Agent* chapter in the *Connect:Direct for Windows System Guide*. The Connect:Direct File Agent Help contains instructions for configuring File Agent.

✦ Sterling Control Center —Sterling Control Center is a centralized management system that provides operations personnel with continuous enterprise-wide business activity monitoring capabilities for Connect:Direct for z/OS, UNIX, Windows, HP NonStop, Select, and OS/400 (iSeries) servers, Connect:Enterprise for UNIX and z/OS servers, and Gentran Integration Suite (GIS) servers. Sterling Control Center enables you to:

   ◆ Manage multiple servers

      • Group individual servers into server groups for a single view of system-wide activity

      • View status and statistics on active or completed processing

      • Suspend, release, and handle Connect:Direct Processes on z/OS, UNIX, Windows, Select, and HP NonStop platforms

      • Stop Connect:Direct servers on z/OS, Windows, HP NonStop, OS/400, and UNIX platforms.

   ◆ Monitor service levels

      • View processing across Connect:Direct for z/OS, UNIX, Select, Windows, HP NonStop, and OS/400 (iSeries) servers, Connect:Enterprise for UNIX and z/OS servers, and GIS servers within your network, and retrieve information about active and completed processing
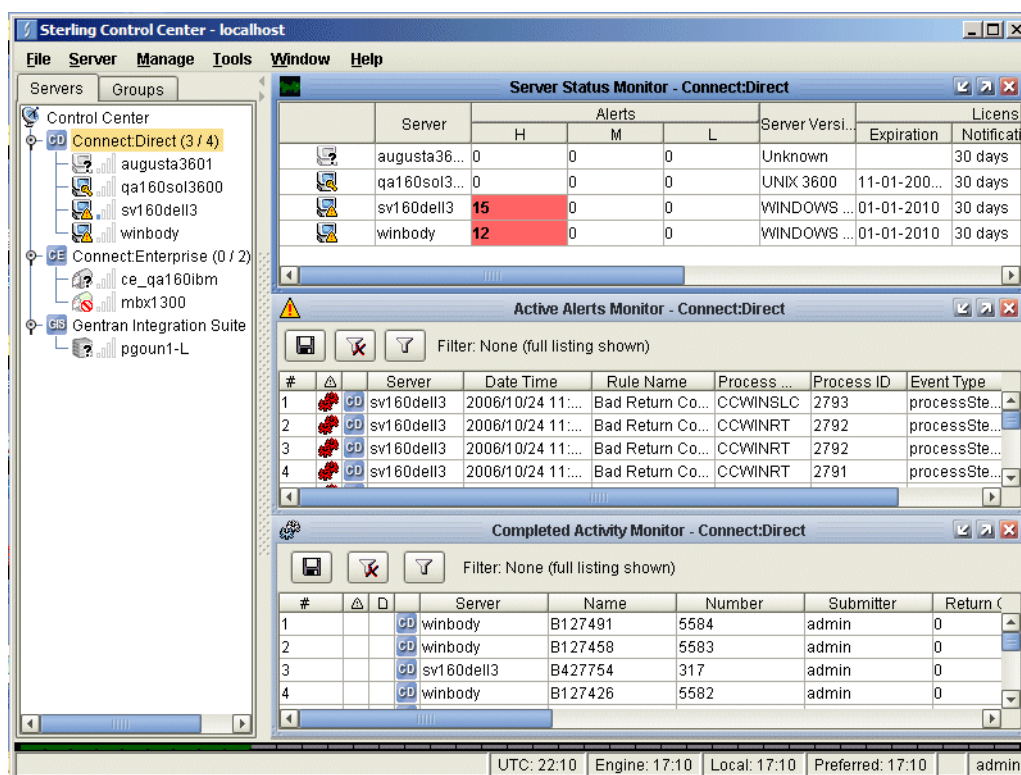


      • Receive notification of data delivery events that occur or do not occur as scheduled

      • Define rules based on processing criteria that can generate an alert, send an e-mail notification, generate a Simple Network Management Protocol (SNMP) trap to an Enterprise Systems Manager (ESM), run a system command, or issue a Connect:Direct server command

- Monitor for alerts, such as a server failure or late-starting Process
- Create service level criteria (SLCs) that define processing schedules, monitor Processes, files within Processes, and file transfers for compliance with these schedules, and generate alerts when the schedules are not met

◆ Analyze key operational metrics through reports to document and analyze processing activity

◆ Create customized reports based on criteria you define

◆ Validate the authenticity of a user logging on to Sterling Control Center, using one or more of four authentication methods, including password validation, host name identification, Windows domain, and TCP/IP address

◆ Identify additional Connect:Direct servers that may need to be monitored based on communications with a currently monitored server using the Guided Node Discovery feature

Sterling Control Center enhances operational productivity and improves the quality of service by:

◆ Ensuring that critical processing windows are met

◆ Reducing impact on downstream processing by verifying that expected processing occurs

◆ Providing proactive notification for at-risk business processes

◆ Consolidating information for throughput analysis, capacity planning, post-processing operational or security audits, and workload analysis

◆ Reducing the risk of errors associated with manual system administration, including eliminating individual server logon to view activity, and the need to separately configure each server for error and exception notifications

Sterling Control Center is available for purchase as a separate product. Contact your Sterling Commerce representative to learn more about Sterling Control Center.

# Restarting Processes

Several facilities are provided for Process recovery after a system malfunction. The purpose of Process recovery is to resume execution as quickly as possible and to minimize redundant data transmission after a system failure. The following Connect:Direct facilities are available to enable Process recovery:

✦ Process step restart—As a Process runs, the steps are recorded in the TCQ. If a Process is interrupted for any reason, the Process is held in the TCQ. When you release the Process to continue running, the Process automatically begins at the step where it halted.

✦ Automatic session retry—Two sets of connection retry parameters are defined in the remote node information record of the network map file: short-term and long-term. If you do not specify a value for these parameters in the remote node information record, default values are used from the local.node entry of the network map file. The short-term parameters allow immediate retry attempts. Long-term parameters are used after all short-term retries are attempted. Long-term attempts assume that the connection problem cannot be fixed quickly and retry attempts occur after a longer time period, thus saving the overhead of connection retry attempts.

✦ Checkpoint/restart—This feature is available with both the **copy** statement and the **run task** statement. To activate the checkpoint/restart feature in the **run task** statement, specify a **restart=yes** parameter in the initialization parameters file.

Connect:Direct records the file positioning checkpoint information in the checkpoint/restart file at specified intervals during a **copy** operation. If a **copy** step is interrupted for any reason, it can be restarted at the last checkpoint position. You can code the **restart** parameter on the **run task** step or in the **runtask.parms** record in the initialization parameters file. If you add the **restart** parameter on the **run task** step, it overrides the initialization parameter. When you add **restart=yes**, Connect:Direct runs the program again. When you code **restart=no**, the Process skips the **run task** step when the Process is restarted. Both the PNODE and the SNODE must support checkpoint/restart.

✦ Intelligent session restart—When an error occurs on a node connection and multiple Processes are scheduled for the node, this feature permits a session retry on only the first Process submitted. The session retry is attempted on the first Process while all other Processes are placed in the Timer queue. After a session is established for the first Process, the other Processes are released so that they can be executed.

If a connection is lost during Process execution, one Process is placed in the Timer queue in RE or retry status using the short and long-term connection retry wait intervals and retry counts. All other Processes that are executing on the node terminate and are placed in the Timer queue with RE status.

If a Process is submitted to a Connect:Direct node for which a session cannot be established, the Process is placed on the Timer queue. Any session retry attempt that is already underway will continue. If all attempts have already been exhausted, then nothing else is done.

If a connection to a remote Connect:Direct node is reestablished, the Process is restarted and establishes a connection with the remote node. The SMGR notifies the PMGR that a session has been established. Each Process in the TCQ Timer queue for this node is released, scheduled, and executed according to Process priority and session class.

If a connection cannot be reestablished with the remote node, all Processes for the remote node are placed in the Hold queue in HE (hold due to error) status.

✦ Short-term and long-term retry parameter definitions—Enables a Process, which was unable to execute because a file that it called was unavailable, to be retried at a later time. If a file allocation error occurs when a Process is executing on either the local or the remote node, and the file allocation error is identified as a condition to retry, the Process is placed in the Timer queue. The Process is then retried using the short-term and long-term retry parameter definitions.

# Connect:Direct for Windows Documentation

See *Connect:Direct for Windows Version 4.4.00 Release Notes* for a complete list of the product documentation.

## About This Guide

*Connect:Direct for Windows System Guide* is for programmers and network operations staff who install, configure, and maintain the Connect:Direct for Windows version 4.4 product.

This guide assumes knowledge of the Windows operating system, including its applications, network, and environment. If you are not familiar with the Windows operating system, refer to the Windows library of manuals.

## Task Overview

The following table directs you to the information required to perform the tasks documented in this guide:

| Task | For More Information, See |
|------|---------------------------|
| Understanding Connect:Direct for Windows | Chapter 1, *About Connect:Direct for Windows* |
| Modifying default Connect:Direct settings | Appendix B, *Changing Connect:Direct for Windows Settings*<br>Appendix C, *Specifying IP Addresses, Host Names, and Ports* |
| Starting and stopping Connect:Direct for Windows servers | Chapter 8, *Managing a Connect:Direct Server*<br>Chapter 4, *Stopping Connect:Direct for Windows* |
| Modifying server settings configured during installation | Chapter 8, *Managing a Connect:Direct Server*<br>Appendix C, *Specifying IP Addresses, Host Names, and Ports* |
| Configuring local node definitions | Chapter 2, *Configuring the Connect:Direct Local Node*<br>Appendix C, *Specifying IP Addresses, Host Names, and Ports*<br>Appendix F, *Editing Connection Settings* |
| Configuring and managing user IDs and authorities and remote user proxies for accessing local servers | Appendix A, *Worksheets*<br>Chapter 2, *Configuring the Connect:Direct Local Node* |
| Configuring definitions for remote nodes that communicate with the local node | Appendix A, *Worksheets*<br>Chapter 3, *Defining and Managing the Connect:Direct Network*<br>Appendix C, *Specifying IP Addresses, Host Names, and Ports* |
| Defining the work to perform using Processes | Chapter 5, *Creating Processes* |
| Configuring unattended file management | Chapter 6, *Managing Files with Connect:Direct File Agent* |
| Monitoring and managing Processes | Chapter 7, *Managing Processes* |
| Testing Connect:Direct | Appendix G, *Using Connect:Direct for Windows in Test Mode* |
| Using the Command Line Interface to submit Processes | Appendix D, *Submitting Processes Using the CLI* |
| Modifying translation tables | Appendix E, *Modifying Translation Tables* |
| Troubleshooting | Chapter 9, *Troubleshooting* |

# Configuring the Connect:Direct Local Node

Before you begin using Connect:Direct for Windows, you configure the local node. Although you can configure the local node using the Configuration Tool or Connect:Direct Requester, this chapter provides instructions on how to configure the local node using Connect:Direct Requester. Connect:Direct Requester is a graphical tool that enables you to define local nodes, user IDs, user authorities, and remote user proxies.

The Configuration Tool enables you to edit text files of user authorization and network map parameters as well as export these files; however, this tool requires that you understand all of the keywords and their parameters. You can more easily define user authorization parameters with Connect:Direct Requester. The Configuration Tool also enables you to view or print a configuration. Refer to *Printing and Viewing Node and Network Map Definitions* on page 43 for more information.

## About Configuring the Local Node

To configure the local node using Connect:Direct Requester:

✦ Add a node definition for each server defined at the local site. Complete the procedure *Adding or Modifying a Local Node Definition* on page 20.

✦ Add a user ID for each person who accesses a local server. Complete the procedure *Adding, Modifying, and Deleting a User ID* on page 20.

✦ Define the functions each local user can perform. Complete the procedure *Defining Local User Authorities* on page 22.

✦ If necessary, configure user proxies for remote user access to the local node. Perform the procedure *Defining Remote User Proxies* on page 27.

After you configure a local node and set the user authority for that node, you can modify the local node definition as necessary.

# Adding or Modifying a Local Node Definition

To begin using Connect:Direct for Windows, you configure a local node for each server in the local network. You identified a local node when you installed Connect:Direct. Depending on the configuration at your location, you can have more than one local node.

To add a local node definition for each server at your site:

1. From the **Node** menu of Connect:Direct Requester, select **Connection Settings** then **Insert Node**.

2. Type the node name in the **Name** field or select a node in the **Active Directory Nodes** pull-down box, if Active Directory is enabled.

3. If desired, select the default user ID to associate with the local node in the **Default User ID** field.

4. Select the operating system used on the node in the **Operating System** field. If any nodes are registered in Active Directory, select **Windows** to display nodes registered in the **Active Directory Nodes** field.

5. Enter the TCP/IP address in the **Address** field. For more information on specifying IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*.

6. Type the port number in the **Port** field.

7. If this is the default node, select the **Set as the default node** check box.

8. Click **OK** to save the changes to the node properties.

To modify a local node definition for each server at your site:

1. Double-click the node name in the **Nodes** window of Connect:Direct Requester.

2. Continue with step 2 through step 8 to add a local node definition.

# Adding, Modifying, and Deleting a User ID

For each user who has access to the local node, you must configure a user ID. Each user must already be defined in Windows. Select **Start > Programs > Administrative Tools (Common) > User Manager** to view a list of users defined in Windows.

When you copy data from a local domain to another domain on a network, the Domain user ID and password must be exactly the same on both domains and a trust between the two domains must exist.

Complete the following procedures to add, modify, or delete a Connect:Direct user ID.

## Adding a User ID

To add a new user ID:

1. From the **Node** menu of Connect:Direct Requester, select **Connection Settings** > **Edit Userids**.

2. Click **Insert**.

3. Type the user name in the **Name** field.

---

**Note:** You must type the user name exactly as the user name is defined in Windows.

---

4. To remember the password, click **Remember password**.

5. Type the password in the **Password** field.

6. Retype the password in the **Verify Password** field.

7. To set this user as the default user, click **Set as the default user**.

8. Click **OK**.

## Modifying a User ID

To modify the password for a current user:

1. From the **Node** menu of Connect:Direct Requester, select **Connection Settings** then **Edit Userids.**

2. Double-click the user ID to modify.

3. In the **Password** field, type the new password.

4. Retype the new password in the **Verify Password** field.

5. To remember the password, click **Remember password**.

6. To set this user as the default user, click **Set as the default user**.

7. Click **OK**.

To establish a user as the current user and log that user on to the Connect:Direct server:

1. From the **Node** menu of Connect:Direct Requester, select **Connection Settings** > **Edit Userids**.

2. Select the user ID and click **Set Current**.

The user that you set to current is automatically connected to the server.

## Deleting a User ID

To delete a user's access to the local node:

1. From the **Node** menu of Connect:Direct Requester, select **Connection Settings** > **Edit Userids**.

2. Select the user ID and click **Delete.**

3. Click **OK** to verify the deletion. You cannot undo a delete.

# Attaching to a Node

After you configure the local node and define the network users, you must attach to the local node. If a user ID is set as the default and has the option **Remember password** activated, the user is automatically attached to the server.

To attach to a node:

1.  From the **Node** menu of Connect:Direct Requester, select **Attach.**

2.  Type the user ID in the **Userid** field or select a user ID from the drop-down list.

3.  Type the password that corresponds to the user ID in the **Password** field.

4.  Click **OK**.

# Defining Local User Authorities

After you define a user ID for each user who has access to the local node, you can restrict the ability of each user to perform certain tasks by defining user authorities for each user ID. For example, you can permit a user to submit Processes but not to monitor or to delete Processes.

You can define two types of users: administrators and general users. Each user type has a set of default privileges. The default administrator and general user definitions allow the user to perform basic Connect:Direct tasks. You can use these templates to assign user authorities and restrict user privileges. The following table defines the default authorities of the administrator and the general user.

| Authority | Default Administrator | Default General User |
| --- | --- | --- |
| View Processes in the TCQ | yes | yes |
| Issue the **copy receive, copy send**, **run job**, and **run task** Process statements | yes | yes |
| Issue the **submit** Process statements | all | yes |
| Monitor, submit, change, and delete all Processes | yes | no |
| Monitor, submit, change, and delete your own Processes | yes | yes |
| Run programs | yes | yes |
| Access Process statistics | all | yes |
| Upload and download files from any directory | yes | yes |
| Upload and download files to or from specific directories | no | no |

| Authority | Default Administrator | Default General User |
|---|---|---|
| Run programs from any directory | yes | yes |
| Run programs from specific directories | yes | no |
| Update the network map | yes | no |
| Update the translation table | yes | yes |
| Update local user authorities | yes | no |
| Update remote user secure point-of-entry proxies | yes | no |
| Stop Connect:Direct for Windows | yes | no |
| Invoke the refresh initialization parameters option | yes | yes |
| Use the trace tool or issue **traceon** and **traceoff** commands | yes | no |
| Override execution priority, including Hold, Retain, and Plexclass status | all | yes |
| User type can override the CRC status | on | off |
| Override Process options such as file attributes and remote node ID | all | yes |

You can also define remote user proxies. A remote user proxy associates a remote user with a local user ID and gives the remote user authority to perform the same functions authorized to the proxy. This is useful if you need to give a remote user access to a server, but you do not want to define a user ID and user authorities for the remote user. This also provides the remote user access to the local node without the need to remember password information.

## Defining User Authority

To define user authority information for a local Connect:Direct user:

1.  From the **Admin** menu of Connect:Direct Requester, select **User Authorities**. The **User Authorities** dialog box is displayed.

2.  Choose one of the following types of users:

    a.  Click **New Admin** to create a new user authority with full privileges for Process controls and functions.

    b.  Click **New Genusr** to create a user authority with reduced privileges for Process controls and functions. The **Adding New User** dialog box is displayed.

3.  Type a name, from 1–50 alphanumeric characters, for the new user in the **Name** field. You can use spaces.

> **Note:** You can enter a user ID in UPN format such as user@csg.stercomm.com. The UPN format allows you to identify both the user name and the domain.

4. To save the new user authority with the default privileges, click **OK**, or to modify the default user privileges, continue with step 5.

5. To restrict the control functions or statements a user is authorized to perform, change the value of one or more of the fields on the **Main** tab to **No** to deny user authority for that privilege and click **OK**. The following table shows the name, definition, and valid values for each field.

| Field Name | Field Definition | Valid Values |
| --- | --- | --- |
| Submit | Grants authority to submit Processes to the TCQ using the Submit Process command.<br><br>Yes specifies that you can submit your own Processes. | Yes \| No |
| Monitor | Grants access to the Process Monitor function.<br><br>Yes specifies that you can monitor your own Processes; All specifies that you can monitor anyone's Processes. | Yes \| No \| All |
| Change | Grants authority to change Processes in the TCQ.<br><br>Yes specifies that you can change your own Processes; All specifies that you can change anyone's Processes. | Yes \| No \| All |
| Delete | Grants authority to delete Processes from the TCQ.<br><br>Yes specifies that you can delete your own Processes; All specifies that you can delete anyone's Processes. | Yes \| No \| All |
| Statistics | Grants authority to access Process statistics using the Select Statistics command.<br><br>Yes specifies that you can access statistics for your own Processes; All specifies that you can access statistics for anyone's Processes. | Yes \| No \| All |
| Copy Send | Grants authority to submit copy Process statements. | Yes \| No |
| Copy Receive | Grants authority to receive copy Process statements. | Yes \| No |
| Run Job | Grants authority to submit run job Process statements. | Yes \| No |
| Run Task | Grants authority to submit run task Process statements. | Yes \| No |
| Submit | Grants authority to submit Processes from within another Process. | Yes \| No |

6. To define directory restrictions, click the **Directories** tab.

If you do not specify a directory for a function, the user can perform the task from any directory, regardless of whether the request is submitted from the PNODE or the SNODE; however, the remote user proxy can override the directory specification. Directory Restrictions for the Upload and Download directory can be bypassed if restrictions are not also provided for the Process and Program directory paths. As a result, if the remote user is allowed to use the Run Task and Run Job features to execute commands from any directory, then they could execute operating system commands. These commands could be "copy" commands to copy files to any directory, bypassing the Upload and Download restrictions.

To prevent this, the Connect:Direct administrator should also set the Directory Restrictions for the Process and Program features. These should be set to a separate directory path from the Upload and Download directory path. Or, the Connect:Direct administrator could disable the Run Job and Run Task features for this user altogether. The Connect:Direct administrator must decide which programs are allowed to execute by placing them in the Process and Program directories.

7. To restrict a user's access to directories, specify the directory from which the user can perform a function, submit Processes, or run programs and click **OK**. Refer to the following table for the name and definition of the **Directory Restrictions** functions:

| Field Name | Field Definition |
|---|---|
| Upload | Specifies the directory that the user can copy files from and use as a source directory. |
| Download | Specifies the directory that the user can copy files to and use as a destination directory. |
| Process | Specifies the directory from which the user can submit a Process. |
| Program | Specifies the directory from which the user can run a program. |
| Redirect | Specifies the directory (or directories) to which redirected files can be written. Separate multiple directory entries with a semicolon (;). |

8. To define administrative privileges, click the **Admin** tab.

9. Change one or more of the administrative functions to a value of **Yes** to grant access to the function and click **OK**. Refer to the following table for the name, definition, and valid values for each of the **Administrative** functions:

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Netmap | Grants authority to update the network map. | Yes | No |
| Translation Table | Grants authority to update the translation tables. | Yes | No |
| User Authorities | Grants authority to update local user Connect:Direct functional authorities. | Yes | No |
| User Proxy | Grants authority to update user proxies. | Yes | No |
| Stop | Grants authority to stop the execution of Connect:Direct. | Yes | No |
| Refresh | Grants authority to refresh the Connect:Direct server initialization parameters. | Yes | No |
| Trace | Grants authority to access the Trace utility. | Yes | No |

10. Click the **Override** tab to define override authority.

11. Change the value of one or more of the override privileges to **Yes** to grant access to the override function. Refer to the following table for the name, definition, and valid values for each of the override privileges:

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Execution Priority | Grants authority to override the default execution priority in a Process. | Yes \| No \| All |
| Remote Node ID | Grants authority to use the remote node ID parameter on the Process or when submitting the Process. | Yes \| No |
| File Attributes | Grants authority to override the system's default file attributes when creating files using a **copy** Process. | Yes \| No |
| ACL Update | Grants authority to define access–allowed and access–denied entries in the Access Control List (ACL) for a file created using a **copy** Process. | Yes \| No |
| CRC | Grants authority to override the CRC enabled state in node and Process statements. | ON \| OFF \| Blank |

12. Click **OK** to save the new user authority.

## Modifying a User Authority

To modify user authority information for Process statement and control functionality:

1.  From the **Admin** menu of Connect:Direct Requester, select **User Authorities**. The **User Authorities** dialog box is displayed.

2.  Double-click the user authority to edit.

3.  Change the user authorities as necessary:

    a.  On the **Main** tab change the values used to determine what Process statements the user can authorize and identify any Process statement restrictions for a user.

    b.  To restrict functions to certain directories, click the **Directory** tab. If you do not specify a directory, the user can perform the task from any directory, regardless of whether the request is submitted from the PNODE or SNODE. The remote user proxy can override the directory specification.

    c.  To change permissions for a user to perform Connect:Direct administrative and control functions, click the **Admin** tab.

    d.  To change permissions for a user to override settings for Connect:Direct nodes, click the **Overrides** tab.

4.  Click **OK** to save the changes.

## Deleting a User Authority

To delete a user authority:

1. From the **Admin** menu of Connect:Direct Requester, select **User Authorities**. The **User Authorities** dialog box is displayed.

2. Select the user you want to delete and click **Delete**.

# Defining Remote User Proxies

The secure point-of-entry proxies definition contains remote user information for operations initiated from remote Connect:Direct nodes. It defines a proxy relationship between a user at a remote Connect:Direct node and a local user ID. This proxy relationship enables users at remote Connect:Direct nodes to submit work to their system that interacts with the Windows node without explicitly defining user IDs and passwords for the specified Windows system in the Processes. Several proxies can use the same local user account with overrides specific to each proxy.

Connect:Direct names each definition of this type according to the remote node and submitter ID. Each definition contains the User ID and password for a local account to use when the specified remote user submits Processes that interact with this node.

**Note:** When you update the password for the specified local user account on your Windows system, you must update the corresponding Connect:Direct secure point-of-entry proxy as well. This process is not automated.

The user proxy definition specifies the following Connect:Direct remote user information:

✦ Remote Connect:Direct user ID and remote Connect:Direct node name. You can also set a generic User ID called <ANY USER> and node name <ANY NODE> to allow all of your remote users to connect through one proxy.

✦ Local user ID and password to use with submitted operations from the remote Connect:Direct node.

✦ Any directory restrictions for Connect:Direct operations.

When a remote user submits a Connect:Direct Process that does not have a remote node security ID specified, Connect:Direct for Windows for Windows accesses the proxy definition that corresponds to the specified user and logs on to the Windows server using the account information specified in the proxy definition. If a proxy definition is not available for a given remote node/submitter ID combination, the Process must explicitly specify the user ID and password to use on the local node. Use the remote node ID security parameter in the **Process** statement to provide the necessary security information to execute Processes with this node.

## Creating or Modifying a Remote User Proxy

To create a proxy for a remote user:

1. From the **Admin** menu of Connect:Direct Requester, select **User Proxies**. The **User Proxies** dialog box is displayed.

2. Add or modify a remote user proxy:

   a. To modify an existing proxy, double-click the Remote User name to modify.

   b. To add a new remote user proxy, click **Insert**.

   The **New User Proxy** dialog box is displayed.

3. Type or modify the following remote user proxy information:

   a. For a new remote user proxy, type the remote node user ID or select <ANY USER> to define a generic user ID.

   > **Note:**   You can enter a user ID in UPN format such as user@csg.stercomm.com.

   b. Type the submitter ID, a specific node name, or select <ANY NODE> to define a generic node.

   c. Type the local user ID to map to the remote node or a specific user ID in the **Userid** field.

   d. Type the password associated with the local user ID to which the proxy is mapped.

   e. Retype the password to verify it.

4. If necessary, click the **Directories** tab to define the directory restrictions for the remote user.

5. Select whether the proxy user has permission to perform **Copy Send** or **Copy Receive** operations (Yes | No), or if the permissions for the proxy user are the same as for the real user (User).

6. Modify one or more of the following functions:

   a. Specify the directory the user can use to copy files from and use as a source directory in the **Upload** field. If no directory is specified, the user can copy files from any directory.

   b. Specify the directory the user can use to copy files to and use a destination directory in the **Download** field. If no directory is specified, the user can copy files to any directory.

   c. Specify the directory containing the Processes the user has authority to submit in the **Process** field. If no directory is specified, the user can submit Processes from any directory.

   d. Specify the directory containing the programs the user has authority to run in the **Program** field. If no directory is specified, the user can run programs from any directory.

7. Click **OK** to save the remote user proxy information.

## Deleting a Remote User Proxy

To delete user proxy information for an existing remote Connect:Direct user:

1. From the **Admin** menu of Connect:Direct Requester, click **User Proxies**. The **User Proxies** dialog box is displayed.

2. Select the remote user you want to delete from the list.

3.  Click **Delete** to delete the remote user.

4.  Click **Close** to complete the operation and close the **User Proxies** dialog box.

# Chapter 3

# Defining and Managing the Connect:Direct Network

Connect:Direct uses a network map to identify the remote nodes that can communicate with the local node. Two methods are available to update the network map.

✦ Connect:Direct Requester provides a graphical interface to create network map entries. This chapter provides instructions on using Connect:Direct Requester to create and update network map entries.

✦ The Configuration Tool enables you to extract a Connect:Direct for Windows network map as a text file from the Windows Registry. You can then update the network map and insert it into the Registry to apply it. This chapter provides instructions on using the Configuration Tool to extract configuration information.

## About Defining the Network Map

You create a remote node entry in the network map for each remote node with which the local node communicates. Each network map entry contains information about the remote node, such as the remote node name, the operating system type, the communications mode, and the communications path to use when connecting to the remote node. The communications mode and communications path are separately defined information that can be associated with one or more remote nodes.

The remote node definition, the communications mode, and communications path definition are the three components of each network map entry. This chapter defines these three components and provides information for defining, modifying, or deleting them.

---

**Note:** The network map is not mandatory if the initialization parameter **netmap.check** is set to **n** and all remote nodes have TCP/IP paths that you identify by their IP address or host name.

---

# Defining Remote Node Information

You define a network map entry for each remote node that communicates with the local node. The network map defines the following components:

✦ **Remote node definition**—Information about remote Connect:Direct nodes that the local node communicates with, including retry parameters, maximum session connections, and network information. To create a remote node definition, complete the procedure *Creating or Modifying a Remote Node Definition* on page 32.

✦ **Communications mode**—Information relating to session characteristics for a protocol, such as TCP/IP or APPC mode information. A communications mode can be associated with one or more communications paths or remote node definitions. To define a communications mode, complete the procedure *Adding or Modifying Communications Mode Settings* on page 37.

✦ **Communications path**—Transport and protocol information about the available communications paths and their attributes. The protocol information includes protocol type, such as TCP/IP, and specific protocol type information, such as the name of the TCP mode for TCP/IP protocols. The protocol information also includes the LU 6.2. protocol type and specific protocol information, such as Net ID, PU Name, and LU name for LU 6.2 protocols. Each communications path definition specifies a default communications mode. To define a communications path, complete the procedure *Creating or Modifying a Communications Path Definition* on page 40.

Each remote node definition can be related optionally to one communications mode and can be related to multiple communications paths, allowing multiple communications paths and modes to be available for use when communicating with the named remote node. These communications paths and modes can be shared with other remote nodes and are not specific to one remote node definition. Communications paths are used when establishing sessions with a remote Connect:Direct node. Connect:Direct tries each communications path named, in the order it is listed, until either a session is established or all paths have been tried. You must designate at least one path.

When defining, modifying, or deleting network map entries, be aware of the dependencies among the three components and the manner in which they refer to each other. For example, before you name a communications mode in your communications path definition, the communications mode must exist.

## Creating or Modifying a Remote Node Definition

The remote node definitions contain information for remote Connect:Direct nodes that the local Connect:Direct node communicates with.

To create or modify a remote node definition in the network map:

1. From the **Admin** menu of Connect:Direct Requester, select **Netmap** to open the network map.

2. Add or modify a remote node definition:

   a. To add a node, from the **Netmap** menu, select **Insert** and type a node name in the **Name** field.

   b. To modify an existing node, double-click the node name in the **Netmap** window.

   The **Netmap Node Properties** dialog box is displayed.

3. Type or select the information for the fields on the **Main** tab. Refer to the following table for the name, definition, and valid values for each field.

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Name | Specifies the name of the remote Connect:Direct node. If you are modifying a node in the network map, this field cannot be edited. | A 1–16 alphanumeric character string |
| Operating System | Specifies the operating system for the remote node. | MVS \|Netware \| OpenVMS \| OS/2 Warp \| OS/390 \| OS/400 \| Tandem \| UNIX \| VM \| VSE \| Windows |
| Max Pnode Sess | Specifies the maximum concurrent connections for all remote nodes where the local Connect:Direct for Windows node is the originator of the Connect:Direct Process. The maximum number of concurrent sessions is limited to the lesser of the values defined in the initialization parameters file and the network map definition for a given node. | A numeric value from 0–255. The default is 1. For a workstation version of Connect:Direct for Windows, maximum PNODE sessions are limited to 1. |
| Max Snode Sess | Specifies the maximum concurrent connections, where the local Connect:Direct for Windows node is the partner, or secondary, node cooperating with a remote Connect:Direct node to execute a Process. The maximum number of concurrent sessions is limited to the lesser of the values defined in the initialization parameters file and the network map definition for a given node. | A numeric value from 0–255. The default is 1. For a workstation version of Connect:Direct for Windows, maximum SNODE sessions are limited to 2. |
| Default Class | Specifies the default session class used to start Session Managers. A Process executes on the specified class or any higher session class. This value overrides the equivalent value for this node in the initialization parameters. | A numeric value from 1 to the value of maximum concurrent local node connections (sess.pnode.max). The default value is 1. The value cannot be greater than the maximum number of local sessions with primary control. |
| Short Term Retry Attempts | Specifies the number of retry attempts in the event of a short-term connection failure. Long-term retry parameters are used after the number of short-term attempts you specify has been reached. | A numeric value from 1–255. The default is 10. |
| Short Term Retry Interval | Specifies the amount of time to wait between each short-term retry attempt. | A 24-hour time value formatted as *hh:mm:ss*. The maximum value is 23:59:59. The default is 00:00:10 or 10 seconds. |

| Field Name | Field Definition | Valid Values |
| --- | --- | --- |
| Long Term Retry Attempts | Specifies the number of retry attempts after all of the short-term retry attempts are used. | A numeric value from 0–255.<br><br>The default is 0. |
| Long Term Retry Interval | Specifies the amount of time to wait between each long-term retry attempt. | A 24-hour time value formatted as *hh:mm:ss*. The maximum value is 23:59:59.<br><br>The default is 00:03:00, or 3 minutes. |

4.  To configure TCP/IP settings, click the **TCP/IP** tab.

5.  Set the TCP/IP attributes. Refer to the following table for the name, definition, and valid values for each field on the **TCP/IP** tab.

| Field Name | Field Definition | Valid Values |
| --- | --- | --- |
| Host/IP Address | Specifies the host name or IP address of the remote node. Alias names are not supported. | A numeric value in the format *nnn.nnn.nnn.nnn (IPv4)* or *nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn: nnnn (IPv6)* or the *host name*.<br><br>For more information on specifying IP addresses and host names, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |
| Port/Service | Specifies the communications port number for Connect:Direct if it differs from the default value specified in the initialization parameters. | A numeric value in the format *nnnnn*, where *nnnnn* is a positive integer from 0–65535. |
| Mode Override | Select the name of the network map TCP/IP communications mode definition record used when communicating with this remote node. If this parameter is not specified, its value defaults to the last TCP/IP mode in the list. | Name of a defined TCP/IP communications mode. |

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Alt Comm Outbound | Alternate communication address (communication path) used for outbound Processes. This parameter provides the alternate addresses for a remote node that has multiple NIC cards. When the local node is the PNODE, the alternate addresses are tried (starting with the first IP address listed) if an initial attempt to the primary address fails. After a connection has been established, if the connection is subsequently lost, attempts to reestablish the connection through the retry mechanism use the same address as the initial connection.<br><br>When the local node is the SNODE, the alternate addresses are used in the Netmap check.<br><br>**Note:** This parameter should not be used in a Connect:Direct for Windows outbound Process if the SNODE is Connect:Direct/Plex. | Fully-qualified host name or IP address and port number.<br><br>A comma separates the list of alternate communication paths as shown in the following example:<br><br>salmon;9400, 10.20.40.65;9500<br><br>The list is processed from the top down.<br><br>For more information on specifying IP addresses and host names, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |
| Alternate Comminfo | Provides support for establishing netmap-checked sessions with high-availability (especially load balancing) systems with multiple IP addresses, such as Connect:Direct/Plex z/OS. Use this parameter to list all IP addresses or host names that are part of the multiple IP address environment.<br><br>For Connect:Direct/Plex, this list should include the address of each Connect:Direct/Server with a different IP address from the Connect:Direct/Plex Manager.<br><br>If a remote node has more than one outgoing IP address (as in a load balancing environment), specify all of the remote node's possible outgoing addresses in the **Alternate Comminfo** field so that those outgoing IP addresses are contained in the local node's netmap entry for that remote node. This allows netmap checking to succeed when the remote node connects to the local node using any of the possible outgoing IP addresses specified. | *hostname1/IP address, hostname2/IP address, hostname3/IP address*<br><br>host name—Host name associated with the IP address. For example:<br><br>hops (where hops is a machine on the local domain)<br><br>hops.csg.stercomm.com (fully-qualified host name)<br><br>*nnn.nnn.nnn.nnn* or *nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn*—IP address of a machine running Connect:Direct<br><br>For more information on specifying IP addresses and host names, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |

6. To configure the APPC settings, click the **APPC** tab.

7. Set the APPC attributes. Refer to the table for the name, definition, and valid values for each of the fields on the **APPC** tab:

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Network Name | Specifies the name of the SNA network in which the remote Connect:Direct node resides. | SNA network name |
| Partner LU | Specifies the name of the partner node to be used as the remote connection end point for the LU-to-LU session. This is the VTAM APPLID for host connections or partner LU name for LU 6.2 only. | A 1–8 character alphanumeric string |
| Mode Override | Specifies the name of the network map APPC communications mode definition record that is used when communicating with this remote node. If this parameter is not specified, its value defaults to the last APPC mode in the list. | Name of a defined APPC communications mode |

8. To identify the communications path, click the **Communication Paths** tab.

   The **Path Properties** dialog box is displayed.

9. Perform the following actions as required:

   a. To add a path defined in the network map, select a path name from the **Available Paths** box and click the right arrow button.

   b. To view the properties of a path, select the path from the **Available Paths** box and click **Properties**.

   c. To add a new path to the network map, click **New**.

   d. To delete a path, select the path in the **Available Paths** box and click **Delete**.

   e. To select all available paths, click **Add All**.

   f. To remove a selected path, select the path in the **Selected Paths** box and click the left arrow button.

   g. To remove all selected paths, click **Remove All**.

10. To add a description of the node, click the **Description** tab. Connect:Direct does not use this information. Refer to the following table for the name, definition, and valid values for each field.

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Name | Specifies the name of the Connect:Direct administrator or operator for the remote node. | A 1–49 alphanumeric character string |
| Phone Number | Specifies the phone number of the administrator or operator for the remote node. Do not use blanks in this string. | A 1–39 alphanumeric character string |
| Node Description | Specifies any additional information you want to include specific to the remote node. | A 1–127 alphanumeric character string |

11. Click **OK** to save the changes.

# Adding or Modifying Communications Mode Settings

You can add or modify the settings for a communications mode object. These values override the values in the initialization parameters file.

To add or modify TCP/IP protocol settings:

1. From the **Admin** menu of Connect:Direct Requester, select **Netmap** to open the network map.

2. Add or modify a mode definition:

   a. To modify an existing mode definition, double-click the mode.

   b. To add a new mode definition, right-click in the mode box and click **Insert**.

3. Modify the mode settings as required. If you are defining a new object, select either TCP/IP or APPC as the protocol to associate with the communications path. Refer to the following table for the definitions, descriptions, and valid values for each field.

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Name | Specifies the name of the remote node and communications path object types. If you are modifying settings for a remote node, the **Name** field is not blank. You cannot change the name. | A 1–48 alphanumeric character string |
| Protocol | Specifies the type of communications protocol. | TCP/IP | APPC |

4.  To add or update TCP/IP settings, click the **TCP/IP** tab. Refer to the following table for the name, definition, and valid values for each field.

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Buffersize | Specifies the buffer size for transmitting data to and from the remote Connect:Direct node. This value overrides the value in the initialization parameters file. | A numeric value from 256–65536. The default is 65535. |
| Pacing Send Count | Specifies the number of send operations to perform before automatically waiting for a pacing response from the remote Connect:Direct node. A value of 0 indicates that there is no pacing. This value overrides the value in the initialization parameters file. | A numeric value from 0–63. The default is 0. |
| Pacing Send Delay | Specifies the amount of time Connect:Direct waits before sending each outbound data buffer to the remote node. This can prevent flooding the network. A value of 0 indicates that Connect:Direct sends each data buffer as soon as possible. This value overrides the value in the initialization parameters file. | An integer from 0–86400. The default is 0. |
| CRC | Specifies whether CRC checking is turned on. The default value for the local node is OFF. The default value for the remote node is blank. The remote node defaults to blank to simplify the use of the crc.override parameter. When crc.override is enabled in the initialization parameter, only the nodes that require a different configuration need to be changed. | OFF | ON | blank |

5.  To add or update APPC settings, click the **APPC** tab and type the following values:

    a.  Type or select any valid RU size between 256 and 4096 in the **Max RU Size** field. The default is 4096.

    b.  Specify the largest permissible pacing window size for sessions in this mode grouping in the **Max Pacing** field. Connect:Direct sends this number of data buffers before waiting for an acknowledgment from the remote Connect:Direct node. A value of 0 indicates no pacing. The default is 7 and the maximum is 63.

    c.  Specify the maximum sessions to be allowed in this mode group. The default is 8. Specify if you are using dependent LUs as the communications path.

6.  Click **OK** to save the mode properties.

## Assigning a Mode to a Communications Path

You can update the communications mode information for a remote node definition at any time. When you configure a communications path, you can either associate an existing mode with it, or you can define a new mode.

## Defining a New Mode for a Communications Path

To define a new mode for a communications path:

1. From the **Admin** menu of Connect:Direct Requester, select **Netmap** to open the network map.

2. Right-click the **CommPath** box and click **Insert**.

3. Type the name of the communications path in the **Name** field.

4. Click **New**.

5. Type the name of the new mode in the **Name** field.

6. Select TCP/IP or APPC to identify the protocol to associate with the mode.

7. To add TCP/IP settings, click the **TCP/IP** tab. Refer to the following table for the name, definition, and valid values for each field.

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Buffersize | Specifies the buffer size for transmitting data to and from the remote Connect:Direct node. This value overrides the value in the initialization parameters file. | A numeric value from 256–65536. The default is 65535. |
| Pacing Send Count | Specifies the number of send operations to perform before automatically waiting for a pacing response from the remote Connect:Direct node. A value of 0 indicates that there is no pacing. This value overrides the value in the initialization parameters file. | A numeric value from 0–63. The default is 0. |
| Pacing Send Delay | Specifies the amount of time Connect:Direct waits before sending each outbound data buffer to the remote node. This can prevent flooding the network. A value of 0 indicates that Connect:Direct sends each data buffer as soon as possible. This value overrides the value in the initialization parameters file. | An integer from 0–86400. The default is 0. |
| CRC | Specifies whether CRC checking is turned on. The default value for the local node is OFF. The default value for the remote node is blank. The remote node defaults to blank to simplify the use of the crc.override parameter. When crc.override is enabled in the initialization parameter, only the nodes that require a different configuration need to be changed. | OFF \| ON \| blank |

8. Click **OK** to close the **Mode Properties** dialog box.

9. Click **OK** to close the **Path Properties** dialog box.

## Creating or Modifying a Communications Path Definition

The communications path defines the physical communications path between the local Connect:Direct node and one or more remote Connect:Direct nodes.

If you install SNA software and select APPC support during installation, two communications path objects are created in the Windows Registry: one for TCP/IP and one for APPC. If you do not have APPC support, only a TCP/IP default communications path object is created.

1. To create or modify a communications path definition:

   a. To modify a communications path definition, double-click the definition in the **CommPath** window

   b. To add a new communications path, right-click the **CommPath** box and click **Insert**.

2. The **Path Properties** dialog box is displayed.

3. If you are defining a new communications path, type the name of the communications path in the **Name** field, and select the protocol to associate with it: **TCP/IP** or **APPC**.

4. Define a new mode or assign an existing mode to the communications path. Refer to the table in step 3 for the name, definition, and valid values for each field on the **Main** tab:

5. To associate an existing mode with the communications path:

   a. Select the mode from the **Selected Mode** drop-down list.

6. To define a new mode to associate with the communications path:

   a. Click **New**.

   b. Type the name of the new mode in the **Name** field.

   c. Select the protocol to associate with the mode: **TCP/IP** or **APPC**, and click **OK**.

7. To add or update TCP/IP settings, click the **TCP/IP** tab. Refer to the following table for the name, definition, and valid values for each field.

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Buffersize | Specifies the buffer size for transmitting data to and from the remote Connect:Direct node. This value overrides the value in the initialization parameters file. | A numeric value from 256–65536. The default is 65535. |
| Pacing Send Count | Specifies the number of send operations to perform before automatically waiting for a pacing response from the remote Connect:Direct node. A value of 0 indicates that there is no pacing. This value overrides the value in the initialization parameters file. | A numeric value from 0–63. The default is 0. |

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Pacing Send Delay | Specifies the amount of time Connect:Direct waits before sending each outbound data buffer to the remote node. This can prevent flooding the network. A value of 0 indicates that Connect:Direct sends each data buffer as soon as possible. This value overrides the value in the initialization parameters file. | An integer from 0–86400. The default is 0. |
| CRC | Specifies whether CRC checking is turned on. The default value for the local node is OFF. The default value for the remote node is blank. The remote node defaults to blank to simplify the use of the crc.override parameter. When crc.override is enabled in the initialization parameter, only the nodes that require a different configuration need to be changed. | OFF | ON | blank |

8. To add or update APPC settings, click the **APPC** tab.

   The following table shows the name, definition, and valid values for each field on the **APPC** tab:

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Adapter Number | Indicates whether the Token Ring adapter number to be used for this connection is Primary or Alternate. | <u>Primary</u> | Alternate |
| Adapter Address | Specifies the token ring address of the remote node. This address consists of hexadecimal digits. Enter all zeros if the local node is the same as the remote node. | Valid remote node token ring address |
| Local SNA Network | Specifies the SNA network name of the local Connect:Direct node. This name must match the name of the SNA network in which the local Connect:Direct node resides. | Valid SNA network name |
| LU Name | Specifies the name of a Local APPC LU configured within Microsoft SNA Server. This is a required parameter. | Valid LU alias name |

9. Click **OK** to save the changes.

# Deleting a Network Map Entry

To delete a network map entry:

1. From the **Admin** menu of Connect:Direct Requester, select **Netmap**. The **Netmap for Nodename** dialog box is displayed.

2.  Right-click the network map entry you want to delete and click **Delete**. You are prompted with a dialog box asking if you are sure you want to delete the entry.

3.  Click **Yes** to delete the entry. You are returned to the **Netmap for *Nodename*** dialog box, and the entry is listed with a red bar to the left of it.

4.  Right-click to open the shortcut menu and click **Apply**. The selected network map entry is deleted from the network map list.

# Validating a Network Map Entry

After you create a network map entry, you can validate the entry to ensure it was created correctly. For example, if you typed an invalid entry in a field, or accidentally deleted a communications path, the validation function generates an error and explains why you received it.

To validate a network map entry:

1.  From the **Admin** menu of Connect:Direct Requester, click **Netmap**. The **Netmap for *Nodename*** dialog box is displayed.

2.  Select the network map entry you want to validate.

3.  From the **Netmap for *Nodename*** window, right-click to open the shortcut menu and click **Validate**. The **Output Window** opens as Connect:Direct validates the network map entry.

4.  After you validate the network map entry, close the **Netmap for *Nodename*** window.

# Viewing a Network Map Entry as Text

To view a network map entry as text:

1.  From the **Admin** menu of Connect:Direct Requester, click **Netmap**. The **Netmap for *Nodename*** dialog box is displayed.

2.  Select the network map entry you want to view as text.

3.  From the **Netmap for *Nodename*** window, right-click to open the shortcut menu and click **Text View**. A view-only copy of the network map entry is displayed.

4.  Right-click again to open the shortcut menu and select the **List View** option to view the network map entry in its original presentation.

# Applying a Network Map

To apply a network map file to the node:

1.  From the **Admin** menu, select **Netmap** to open the network map file.

2.  From the **Netmap** menu, select **Validate**. If no errors are found, the output window contains the message *Netmap validation Successful*.

3.  From the **Netmap** menu, select **Apply**.

4.  Select the node name and click **OK** to apply the network map.

# Printing and Viewing Node and Network Map Definitions

After you set up a node and define a network map, you can use the Configuration Tool to extract the network map and user authorization information from the Registry as plain text files. You can then update the parameters in the extracted files for distribution to the nodes in an environment. After you customize the files with site-specific parameters, you can insert them into the Windows Registry where they are applied while the server is running, or you can apply them during a new Connect:Direct installation.

To extract the configuration files from the Windows Registry on the target node:

1.  Click **Start** and point to **Programs > Sterling Commerce Connect:Direct v4.4.00** > **CD Configuration Tool**.

2.  From the **File** menu, select **Extract**. The **Local Node Properties** dialog box is displayed:

3.  Click **OK**. The network map and User Authorization information for the node are extracted as as cascading windows named Map and User.

4.  To save the extracted user authorization information:

    a.  Click the **User** dialog box to select it.

    b.  From the **File** menu, click **Save As**.

    c.  Select the directory where you want to save the file.

    d.  If necessary, type a different file name in the **File name** field and click **Save**.

5.  To save the extracted network map information:

    a.  Click the **Map** dialog box.

    b.  From the **File** menu, click **Save As**.

    c.  Select the directory where you want to save the file.

    d.  If necessary, type a different file name in the **File name** field and click **Save**.

# Viewing the Sample Configuration Files

When you want to update configuration information, you can use the sample configuration files as a template. These text files contain the valid keywords for a network map and user authorizations. You can customize these files to configure nodes in an enterprise.

---

**Note:**   To automate the installation of Connect:Direct for Windows, you can perform a silent installation that requires no user interaction and that applies the User.cfg and Map.cfg files during the installation. See the *Connect:Direct for Windows Getting Started Guide* for more information on silent installations.

---

Two sample configuration files are included with the Configuration Tool:

✦ **Map.cfg file**—Network map objects are created in the Registry during installation. These objects contain the remote node definitions, communications path definitions, and communications mode definitions.You can update the network map on your nodes by customizing the sample Map.cfg file and inserting it into the Windows Registry.

✦ **User.cfg file**—Use the parameters in User.cfg to build user functional authorities and user proxies. Connect:Direct applies the information in this file to authorize local and remote users to issue commands and Process statements and to perform tasks. Use the Configuration Tool to create authorizations for each user, including encrypted passwords for user proxies. After you have created the user authorizations, you can insert the User.cfg file into the Windows Registry.

# Customizing Configuration Files

The Configuration Utility is used to customize configuration files for your environment and prepare them for rollout. Not only can you edit configuration information using this tool, but you can also add encrypted passwords to user proxies and validate the configuration files for use with Connect:Direct.

## Editing Configuration Files

To prepare configuration files for rollout, edit them to meet your configuration needs.

To edit the configuration files:

1. From the **File** menu, select **Open**.

2. Select the drive and directory where the configuration file is located.

3. Select the file and click **Open**.

> **Note:** If you want to save the original file as a template for building future configuration files, save the file with a different name before you alter any of the configuration information.

4. Edit the parameters as necessary.

5. From the **File** menu, select **Save**.

## Entering Encrypted Passwords for User Proxies

You can set up and enable user proxies on the local Connect:Direct server that allow remote users to log on to the local server without revealing their password. User proxies improve security on the remote node by protecting remote users' passwords.

The Configuration Utility can be used to insert encrypted passwords into the proxy section of the USER.CFG file. When you enter a password, the Configuration Utility encrypts it for you.

To add a local user ID and encrypted password:

1. Open the User.cfg file.

2. Place the cursor on the line following the Proxy section header.

3. From the **Tools** menu, select **Password**. The **Logon Information** dialog box is displayed.

4. Type the User ID and password in the **Userid** and **Password** fields.

5. Re-type the password in the **Verify Password** field.

6. Click **OK**. The encrypted password is inserted as the **LocalPassword** parameter value and the User ID is inserted as the **LocalUserid** parameter.

## Validating Configuration Files for Use with Connect:Direct

After you have customized a file, validate the configuration to ensure that the file can be used with Connect:Direct. When you validate a file, error messages describe any errors and each error is highlighted so you can easily determine which information must be revised.

To validate a configuration file:

1. Open the configuration file to validate.

2. From the **Tools** menu, select **Validate**.

   a. If no errors are detected, a message indicating no errors is displayed. Click **OK** to return to the file.

   b. If errors are detected, an error message is displayed with the error identifier and an explanation of the problem. The error is highlighted in the configuration file. Click **OK** to close the error message.

3. Edit the configuration information to clear each error.

4. Repeat this procedure until no errors are returned.

---

**Note:** When you save a configuration file, the Configuration Utility automatically validates it. You cannot save a file if it contains invalid information.

---

# Applying Updated Configuration Information

You can update the network map and user authorizations by inserting the updated configuration files in the Windows Registry. After the configuration files are inserted, the settings are applied when the server is running.

---

**Note:** You cannot use cdconfig to change existing objects in the network map and user authorizations. It can only be used to create new objects (netmap entry or user auth name).

---

To insert the updated network map file into the Windows Registry, type the following command at the command prompt, in the directory where Connect:Direct for Windows is installed:

```
cdconfig /q /nnodename /i /fmap.cfg
```

When inserting a file, the Configuration Utility uses the following switches:

✦ /q—To silently run the Configuration Utility while the file is extracted

✦ /n—To identify the node from which to extract the network map or user authorizations (/n*nodename*)

✦ /i—To insert the configuration file into the Registry

✦ /f—To identify the file to insert into the Registry (Map.cfg or User.cfg)

✦ /u—To extract a user configuration file (user.cfg) file from a specified node as illustrated in the following sample:

```
cdconfig /nLinji3300 /u /fTestUser.cfg
```

✦ /m—To extract the network map configuration file (map.cfg) from a specified node as illustrated in the following sample:

```
cdconfig /nLinji3300 /m /fTestMap.cfg
```

# Stopping Connect:Direct for Windows

This chapter describes how to stop Connect:Direct for Windows.You can stop Connect:Direct for Windows from Connect:Direct Requester or the CLI, and by using the Windows Services facility or the Admin Tool Utility.

## Stopping Connect:Direct for Windows from Connect:Direct Browser

You can stop the Connect:Direct for Windows server in the following ways:

✦ Issue the **Stop** command from Connect:Direct Requester or the CLI.

✦ Use the Services facility on the Windows Control Panel.

✦ Use the Admin Tool Utility.

To stop Connect:Direct for Windows from Connect:Direct Requester:

1. From the **Admin** menu of Connect:Direct Requester, select **Netmap** to open node that you want to stop.

   a. From the **Admin** menu, select **Stop Node**. The **Stop Server** dialog box is displayed.

   b. Select one of the stop server options:

   • **Quiesce after active processes complete**—Stops the server when all executing Processes are complete. No new Processes are started.

   • **Terminate processes at end of step**—Stops Connect:Direct when all executing Process steps are complete. Writes statistics records, closes files, and shuts down Connect:Direct.

   • **Immediate orderly shutdown**—Stops all activity, terminates Processes and connections, writes statistic records, closes files, and shuts down Connect:Direct.

   • **Force termination now**—Forcibly terminates Connect:Direct. No attempt is made to terminate executing Processes or to write statistic records. All server resources are released. This is the least desirable way to stop the server.

   c. Click **OK** to stop the node. If the server is stopped, the icon in the Control Pad nodes view is unavailable.

# Stopping Connect:Direct Using the CLI

To stop Connect:Direct for Windows from the Connect:Direct for Windows CLI, issue the **stop** command. Refer to *Stopping Connect:Direct* on page 179 for more information about the **stop** command options.

# Stopping Connect:Direct from the Services Facility

To stop Connect:Direct for Windows from the Windows Services facility:

1.  Click **Start > Settings** > **Control Panel >Administrative Tools** > **Services**.

    a.   Select the Connect:Direct node that you want to stop.

    b.   Click **Stop**. The display changes to indicate that Connect:Direct has stopped.

    **Note:**   Connect:Direct for Windows does not implement the Pause and Continue functions on the Services applet.

# Stopping Connect:Direct from the Admin Tool Utility

To stop Connect:Direct for Windows from the Admin Tool Utility:

1.  Select the Connect:Direct for Windows server you want to stop.

2.  Click the red traffic light icon on the toolbar.

# Creating Processes

A Process is a set of statements grouped together to perform a series of Connect:Direct tasks. You define the tasks to perform and save the Process to a file. Finally, you use the Submit Process command to execute all the tasks defined in the Process.

Processes allow you to automate routine tasks, execute multiple tasks with one command, save the Process to a file for future use, and build blocks of work based on conditions. Before you build a Process, decide what tasks you want to perform and what nodes you want to use.

Building a Process requires the following tasks:

✦ Creating a Process statement

✦ Defining or modifying Process options

✦ Building the group of tasks that perform the work

✦ Validating Process content

✦ Saving the Process file

## Command Types

When you build a Process, you add command types to define the tasks you want to perform. The following table describes the command types:

| Command Type | Description |
| --- | --- |
| Copy | Transfers data between two specified nodes. The Copy statement identifies the source files, applies any pre-transfer attributes such as compression or checkpointing, transfers the file to the destination, and saves the file to the new file name. |
| | Use the Copy statement to create a new file or append data to an existing file. To use the Copy statement in a Process, identify the PNODE and the SNODE. Identify the source file and, if symbolic variables are used, assign values to the variables before the Process is submitted or specify built-in variables. For more information on built-in variables, see |
| Run Task | Runs programs or operating system commands on the specified node. The Process will suspend execution until the Run Task completes. Use the Run Task statement to process data before or after it is transferred or to launch programs when a file is created. |
| | Perform the following types of tasks using the Run Task statement: |
| | z/OS—Attach user subtasks or programs. |
| | OS/400—Execute OS/400 CL commands to call programs, submit jobs, send notification to a user, create save files, or restore save files. |
| | Tandem—Execute user-written or system programs during Process execution. |
| | UNIX—Invoke a UNIX command shell to execute one or more UNIX commands. |
| | VSE—Attach user programs or subtasks during Process execution. |
| | OpenVMS—Create a detached OpenVMS Process that executes one or more DCL commands or command procedures. |
| Run Job | Executes batch jobs or commands on a specified node. These jobs or commands run asynchronously, running in the background while the Process continues execution. The Process does not wait for the job to complete. |
| | The return code associated with a Run Job statement indicates the success or failure of the statement to launch the job or command and not the success or failure of the batch job or command. |
| | Use the Run Job command to perform the following types of tasks. |
| | z/OS—Submit jobs to the internal reader, a facility that transfers jobs to the job entry subsystem (JES). |
| | NetWare—Start NetWare Loadable Modules (NLMs) and send jobs to the print queue of the NetWare server. |
| | OS/400—Submit an OS/400 CL command to run as a separate job through the SBMJOB command. |
| | Tandem—You cannot execute Tandem commands using Run Job. |
| | UNIX—Invoke a UNIX command shell and execute UNIX commands. |
| | VSE—Submit jobs through the VSE virtual reader, a facility that transfers jobs to VSE/POWER. |
| | Windows—Start programs that will run in a Windows-on-Windows VDM, or MS-DOS applications. |
| Submit | Submits a Process from within an executing Process on the PNODE or the SNODE. To use the Submit statement, the Process must reside on the node you are submitting the Process to. Use the Submit statement to execute a Process on the SNODE which would, in turn, submit a Process to a third node. |

| Command Type | Description |
|---|---|
| Conditional | Several conditional statements are available to allow you to perform tasks based on conditions that you establish. Use the following commands to create the conditions necessary to meet your requirements: |
| | If—Executes a block of statements based on the results of a previous statement. |
| | Else—Defines a block of statements that execute when the If statement is not true. |
| | Eif—Marks the end of the If statement and any statements based on the If condition. |
| | Goto—Executes a jump to a specific statement that occurs later in the Process. |
| | Exit—Bypasses all remaining steps in a Process and ends the Process. |

# Establishing Preferences

Throughout your session with Connect:Direct Requester, certain predefined preferences are in effect. Connect:Direct establishes default user preferences during installation. You can modify preferences at any time to more accurately reflect how you work. Preferences save you time, since preferences are used for all instances, except when you override the values.

To establish preferences for a Connect:Direct session:

1. From the Connect:Direct Requester Main Window, select **Options** from the **Tools** menu. The **Options** dialog box is displayed.

2. To set general preferences, change values in the following fields:

| Field | Description |
|---|---|
| Reload last saved workspace at startup | When selected, Connect:Direct Requester displays the last saved workspace when you log in. |
| Track processes in the execution status window | To obtain summary information about a Process and display the information in the Execution Status Window. This option establishes the default for the Submit Process page. |
| Enable in-place editing | To change the label field on the Program Definition Editor (PDE) directly and eliminate the need to edit the labels through the Process Properties page. |

| Field | Description |
|---|---|
| Activity log | Use the options in this window to display the activity log in the Output window, save the activity log to a file, append data to the log file, or create a new log file every time you start Connect:Direct. |
| | Check **Enable the activity log in the output window** to display the activity log in the output window. |
| | Check **Write the activity output to a file** and type the name of the file in the box provided to write the activity log to a file. Select the appropriate button to create a new file every time you start Connect:Direct Requester or to append activity to the existing log file at start up. |

3.  To set Process preferences, click the **Process Defaults** tab and change the values in the following fields:

| Field | Description | Valid Values |
|---|---|---|
| Hold | To hold Processes in the Hold queue in HI (Held Initially) status until explicitly released. A Process submitted with Hold set to **Yes** is placed in the Hold queue even if you specify a Start Time. | No—Execute a Process as soon as resources are available.<br>Yes—Hold a Process.<br>Call—Hold a Process until the remote node connects to the local node. At that time, the Process is released for execution. |
| Retain | Processes are retained in the Hold queue in HR (Held for Retain) status after execution. You can release the Process for execution later through the Process Monitor function or explicitly delete it. | No—Do not retain Processes after execution.<br>Yes—Retain Processes after execution.<br>Initial—Retain Processes in the Hold queue in HR status for automatic execution every time the Process Manager initializes. Do not provide a start time parameter when you choose this option. |
| Plexclass | Specifies the class that directs the Process to only certain servers in a Connect:Direct/Plex. This parameter is only used in a Connect:Direct/Plex. | Name of the class. |

| Field | Description | Valid Values |
|---|---|---|
| CRC | ◆ Specifies whether CRC checking is turned on.<br><br>◆ The default value for the local node is OFF. The default value for the remote node is blank.<br><br>◆ The remote node defaults to blank to simplify the use of the crc.override parameter. When crc.override is enabled in the initialization parameter, only the nodes that require a different configuration need to be changed. | ◆ OFF \| ON \| blank<br><br>◆ OFF—Do not perform CRC checking.<br><br>◆ ON—Perform CRC checking.<br><br>◆ blank—Use the value defined in crc.override. |
| Priority | The preferred priority of a Process in the TCQ. Connect:Direct uses the priority parameter for Process selection. A Process with a higher priority is selected for execution before a Process with a lower priority. | The valid range is 1–15, where 15 is the highest priority. |
| Class | The preferred session class on which a Process can execute. A Process can execute in the class specified or any higher session class. | Values range from 1 to the maximum number of local node sessions in the network map definition. |
| Notify Userid | Specify the computer name of the user to receive Process completion messages. | A valid computer name. |
| Accounting Data | Type an arbitrary string to be used as the preferred accounting information for the PNODE or the SNODE or both nodes. | The maximum length of the string is 256 characters. |

4. To set Work List preferences, click the **Work List** tab and change the values in the following fields:

| Field | Description | Valid Values |
|---|---|---|
| Track worklist is the worklist status window | Determine if Work List statuses are reported in the status window. | On or Off |

---

| Field | Description | Valid Values |
|-------|-------------|--------------|
| Max Delay for Serial Execution | Identifies the maximum amount of time to wait. This parameter is useful when the Process is submitted by a batch file and you want to suspend further execution of the batch file until the submitted Process completes. | Unlimited—The command processor waits until the Process completes execution.<br>Time and hh:mm:ss—Select this option and type the time to wait for a Process to complete. |

5.  To set directory preferences, click the **Directories** tab and change the values in the following fields:

| Field | Description |
|-------|-------------|
| Processes | To define the default directory for Process files |
| Work Lists | To define the default directory for Work Lists. |
| Process Monitors | To define the default directory for Process Monitor files. |
| Statistics Monitors | To define the default directory for Statistics Monitors. |
| Start "Save as" in these directories | Select this check box to make these directories the default when saving. |

6.  To set file type preferences, click the **File Types** tab and select the file extensions from the list to associate with Connect:Direct:

- ◆ .cdp (Process Definitions)
- ◆ .cdw (Work Lists)
- ◆ .cds (Statistics Monitors)
- ◆ .cdm (Process Monitors)
- ◆ .cdn (Network Map)

7. To set statistics preferences, click **Select Stat Defaults** and change values in the following fields:

| Field | Description | Valid Values |
|---|---|---|
| ◆ Monitor | ◆ Determine if all statistics are monitored or only selected statistics, based on criteria or for Step and Process completion only. If you choose **All**, indicate the time range for the selection. | ◆ All Statistics For Last _ hrs:min<br>◆ Filter Using Selection Criteria Pages<br>◆ Step and Process Completion Statistics Only |
| Refresh Display | Identify whether you want the Statistics Monitor display to be refreshed. Specify the interval in minutes between refreshes if you choose the Refresh every _ minutes. | Refresh every _ minutes<br>Refresh on open<br>Autoscroll—Display the latest statistics |

8. To set Process Monitor preferences, click **Select Proc Defaults** and change values in the following fields:

| Field | Description | Valid Values |
|---|---|---|
| Monitor | Determine if all Processes will be monitored or only selected Processes, based on selection criteria. | All Processes<br>Filter Using Selection Criteria Page |
| Refresh Display | Identify whether you want the Process Monitor display to be refreshed. Specify the interval in seconds between refreshes if you choose the Refresh every _ seconds. | Refresh every _ seconds<br>Refresh on open |

9. Click **OK**.

# Creating a Process Statement

Every Process begins with a Process statement that defines general information about the Process.

To create a new Process statement from Connect:Direct Requester:

1. From the **File** menu, select **New > Process**.

2. Type a Process name, from 1 to 8 alphanumeric characters, in the **Process Name** field.

3. If all work will be performed on the PNODE, type or select the name of the PNODE in the **PNODE Name** field.

4.  To issue a warning message if an attempt is made to submit the Process on a different PNODE, click **Warn if submitted to a different node**.

5.  To issue a warning message if an attempt is made to submit the Process on a PNODE with a different operating system, click **Warn if submitted to a different operating system**.

6.  Specify the SNODE in one of the following ways:

    a.  Select the node from the drop-down menu.

    b.  Type the name of a Connect:Direct node.

    c.  Specify an IP address and port, using the following format:

    ```
    hostname|IPaddress;portnumber|servicename
    ```

    **Note:**   For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*.

7.  Click **OK**.

8.  If desired, change the options for the Process. Refer to *Defining Command Options* on page 62

9.  Add commands as necessary to the Process. Refer to *Creating Commands* on page 56.

**Note:**   When you create a Process, the PEND statement is added to the end of the Process and is a required statement that marks the end of a Process. Do not edit or delete the PEND statement.

# Creating Commands

You can add commands to a Process statement to run a Process, run a task, or run a job. When you add a command to a Process, you use the preferences you defined or you set unique values for each command, including when a command runs, the priority and class for the command, the TCQ status in which to place a command, the values for CRC checking or CDPlexclass status, who is notified when a command is run, security options, and accounting information. Use the procedures in this section to add a command to a Process and change options as desired.

## Adding a Copy Command

The Copy statement transfers data between two nodes. The Copy statement identifies the source files, applies any pre-transfer attributes such as compression or checkpointing, transfers the file to the destination, and saves the file to the new file name.

Use the Copy statement to create a new file or append data to an existing file. To use the Copy statement in a Process, identify the PNODE and the SNODE. Identify the source file and, if symbolic variables are used, assign values to the variables or specify built-in variables before the Process is submitted. Connect:Direct for Windows supports the string (*) and character (?)

wildcards, allowing you to copy multiple files from a source directory to a target directory with a single copy statement.

---

Note: The list of files to be copied is generated at the start of a wildcard copy. When a Process restarts, in a wildcard copy statement, the step restarts with the first file that is not completely copied. If you are using checkpointing, the statement restarts at the past checkpoint of the file that is not completely copied.

---

To add a Copy command to a Process:

1. Open a Process file.

2. From the **Process** menu, select **Insert > Copy**.

3. To identify the step within the Process, type a label of up to 8 alphanumeric characters in the **Copy Statement Label** box.

4. To copy the file to the SNODE, select **Send**.

5. To copy a file from the SNODE, select **Receive**.

6. Type the name of the source file in the **Source Filename** field.

7. Enter the name of the file in the **Destination Filename** field.

8. Select one of the following destination disposition options:

   ◆ **NEW**—To create a new file at the destination

   ◆ **RPL**—To append the information to an existing file if it exists or to create a new file, if the file does not already exist.

   ◆ **MOD**—To append the transferred information to an existing file.

9. To enter compression and checkpoint restart options, click the **Transfer** tab and select from the following options:

   a. In the Compression window, select one of the following compression types:

| Compression Option | Description | Valid Values |
|---|---|---|
| None | Turn on this option if you do not want compression. | None |
| Primary Char | Turn on this option to compress repetitive characters. Specify the primary character to compress. If the character is repeated 2–63 times, the characters are compressed to one byte. If other characters are repeated 3–63 times in succession, the characters are compressed to two bytes. | The hex or character to compress. |

| Compression Option | Description | Valid Values |
|---|---|---|
| Extended | Turn on this option to compress repetitive strings. Extended compression produces the best compression results. Specify this type of compression when lines are limited and data is repetitive. <br><br> The Window value identifies the compression for windows. The greater the window size, the greater the degree of compression. A window size of 8 uses 1K of memory while a window size of 15 uses 128K of memory. <br><br> Memory identifies how much virtual memory is allocated to maintain the internal compression state. | **Comp Level**—Select the level of compression from 1–9. <br><br> **Window**—Select the window size level from 9–15. <br> The Default is 13. <br><br> **Memory**—Select a memory compression from 1–9. <br> The Default is 4. |

b.  To use checkpoint restart, select one of the following options:

- **Default**—To use checkpoint restart options defined in the default Process.

- **None**—To turn off checkpoint restart.

- **Check**—To eliminate the need to retransmit an entire file in the event of a transmission failure. If a copy procedure is interrupted, Connect:Direct restarts that copy at the last checkpoint. If you turn on this option, type the interval at which to mark a checkpoint restart and turn on either Kilobytes or Megabytes to indicate the measurement to use.

10. To override the preferences or provide additional parameters to describe the sending and receiving files:

---

**Note:**   These tabs reflect the operating system of the sending and receiving file location.

---

a.  To define options for the sending file, click the **From <*Operating system*>** tab, and type the sysopts parameters. Refer to the online help for more information.

b.  To define options for the receiving file, click the **To <*Operating system*>** tab, and type the sysopts parameters. Refer to the Help for more information.

11. Click the **Comment** tab and add a description of the statement. This information is optional.

12. Click **OK**.

## Creating a Run Task Command

You can run programs and commands by adding the Run Task command to a Process. The Run Task command executes programs on the PNODE or the SNODE.

To add a Run Task command to a Process:

1. Open a Process file.

2. From the **Node** menu, select **Run Task**.

3. Select the node where the program or command will execute in the **Run On** field.

4. Do one of the following to select the program used by the node in the **Program** field:

    ◆ Type **Windows** for a Windows node.

    ◆ Type **UNIX** for a UNIX node.

    ◆ For OS/400, type **cmd(CL command) [parameter for OS/400 SBMJOB command]**

    ◆ For z/OS, type the name of the program to be attached as a subtask in upper case letters.

    ◆ For Tandem, type the name of the object file.

    ◆ For VSE, type the name of the program to be attached as a subtask in upper case letters.

    ◆ Type **NetWare** for a NetWare node.

5. Use the optional parameters and commands as necessary, for the operating system you selected in Step 4. Refer to the Help for syntax.

6. Select the **Track execution in output window** check box to view activity during execution.

7. Continue defining command options. Refer to *Defining Command Options* on page 62.

## Creating a Run Job Command

The Run Job command executes batch jobs or commands on the specified node. Jobs submitted using the Run Job command run in the background. The return code associated with the run job statement indicates the success of the Run Job command and not the success of the batch job or command.

Use the Run Job command to perform the following types of tasks.

✦ Submit jobs in an z/OS environment to the internal reader, a facility that transfers jobs to the job entry subsystem (JES).

✦ Submit a job at the OpenVMS node in an OpenVMS environment.

✦ Submit an OS/400 CL command to run as a separate job through the SBMJOB command.

✦ Invoke a UNIX command shell and execute UNIX commands.

✦ Submit jobs through the VSE virtual reader, a facility that transfers jobs to VSE/POWER.

✦ Start programs in the Windows environment.

> **Note:** You cannot execute Tandem commands using Run Job.

Run batch jobs on the PNODE or the SNODE using the Run Job command. To add a Run Job command to a Process:

1. Open a Process statement.

2. To add a new Run Job statement to a Process:

   a. From the **Process** menu, select **Insert > Run Job**.

   b. Type the statement label in the **Run Job Statement Label** field.

3. Type the file name. This field is only valid for specific operating system types.

4. Select the executing node in the **Run On** field.

5. Select **Track execution in output window** check box if you want to monitor execution of the command.

6. Select the node where the job will execute.

7. Type the file name based on the operating system used by the node, in the **Filename** field. This field is only valid for specific operating system types:

   ◆ For Windows nodes, type **Windows**.

   ◆ For z/OS nodes, type the data set that contains the job in the format: DATASETNAME | DATASETNAME(MEMBER).

      The data set and member must be in uppercase. If the data set is a PDS, specify the member. The data set containing the job must exit on the z/OS node where the job will execute. A data set containing JCL is limited to a record length of 54 bytes.

   ◆ For OS/400 nodes, type **OS/400**.

   ◆ For VSE nodes, type the name of the member in the LIBDEF source chain that contains the job to be submitted in the format: | membertype(member). If the membertype is not specified, this job must be cataloged with a membertype of J.

   ◆ For UNIX, type **dummy**.

   ◆ For Open VMS, type **PGM=VMS**.

8. Select the **Track execution in output window** check box to view activity during execution.

9. Continue defining the command. Refer to *Defining Command Options* on page 62.

## Adding a Submit Process Command

To add a Submit Process command to a Process:

1. Open a Process file.

2. From the **Node** menu, select **Submit Process**.

3. Select the name of the node where the Process file is located in the **Location** field. You can submit from the default node or select another node.

4. In the **Filename** field, type the full path and Process file name. If you are submitting a Process on a Windows node, you can click the browse button to locate the Process file.

5. To override the Process name, type a 1–16 alphanumeric string in the **New Name** field.

6. If you want the Process to execute with a different SNODE, enter or select the SNODE in the **SNODE** field. The SNODE you define here overrides the SNODE defined in the Process file.

7. Select the **Track execution in output window** check box to view activity during execution.

8. Continue defining the command. Refer to *Defining Command Options* on page 62.

## Adding an If Statement

The If statement executes a block of statements based on the results of a previous statement. The results are categorized by a return code. The If statement checks the value of the return code and executes the block if the statement is true.

You must use the Eif statement (endif) with the If statement. If the conditions of the If statement are not satisfied, use an Else statement to designate the execution of alternate instructions.

To add an If statement to a Process:

1. From the **Process** menu, select **Insert > If**.

2. Enter the statement label in the **If Statement Label** field.

3. Select a step label to base the operator and value.

4. Select the type of comparison statement in the **Operator** field.

5. Select one of the following return codes in the **Value** field.

| Return Code | Description |
| --- | --- |
| 0 | Indicates successful completion of the stop |
| 4 | Indicates a warning |
| 8 | Indicates an error |
| 16 | Indicates a catastrophic error |

6. To add comments, click the **Comments** control tab. This information is optional.

7. Click **OK** to save the statement. The If statement is displayed in the Process window.

8. Create the block of statements that executes based on the If statement. You can add an Else statement to execute a block of statements if the condition in the If statement is not satisfied.

9. Add an Eif statement to the end of the block to designate the end of the block of statements.

## Adding an Else Statement

The Else statement defines a block of statements that execute when the If statement is not true. The Else statement is only valid when used in combination with the If statement.

To add an Else statement to a Process:

1. From the **Process** menu, select **Insert > Else**.

2. If desired, add an optional comment to the Process. This information is optional.

3. Click **OK** to save and exit the **Else** dialog box. The Else statement is displayed in the Process window.

### Adding an Eif Statement

The Endif statement (Eif) marks the end of the If statement and any statements based on the If condition. The Endif statement is only valid when used in combination with the If statement.

To add the Endif statement to a Process:

1. From the **Process** menu, select **Insert > End If**.

2. If desired, type a comment for the Process. This information is optional.

3. Click **OK** to save and exit the **Endif** dialog box. The Endif statement is displayed in the Process window.

### Adding or Modifying a Goto Statement

The Goto statement executes a jump to a specific statement that occurs later in a Process. This statement cannot be used to loop to a statement earlier in the Process. Use the Goto statement with the step label to define the location of the statement in the Process.

 To add or modify a Goto statement in a Process:

1. Take one of the following actions:

    ◆ To create a new Goto statement, from the **Process** menu, select **Insert** > **Goto** and enter the statement label in the **Target Label** field.

    ◆ To modify an existing statement, double-click the Goto statement in the Process window.

2. To add or modify comments, click the **Comments** tab and type the comment. This information is optional.

3. Click **OK** to save the Process. The Goto statement is displayed in the Process window.

### Adding or Modifying an Exit Statement

The Exit statement bypasses all remaining steps in a Process and ends the Process.

To add an Exit statement to a Process:

1. From the **Process** menu, select **Insert** then **Exit**.

2. If desired, click the **Comments** tab and add an optional comment to the Process. This information is optional.

3. Click **OK** to save the Process and exit the **Exit Statement** dialog box. The Exit statement is displayed in the Process window.

# Defining Command Options

After you create a statement, you then set the Process options including when the Process is submitted, how a Process is handled in the TCQ, if a user is notified when a task is complete, who has access to the Process, and any defining comments associated with the Process.

To define statement options, perform one or more of the following procedures:

✦ Identify how tasks are managed and how resources are allocated. See *Setting the Control Functions for a Command or Process* on page 63 for instructions.

✦ Identify user IDs and passwords needed to access the SNODE and the PNODE. See *Setting Security Options* on page 64.

✦ Define variable values. See *Assigning Values to Symbolic Variables* on page 65.

✦ Specify accounting data. See *Specifying Accounting Data* on page 67 for instructions.

✦ Add a comment about the statement. See *Adding or Modifying Comments* on page 67. This information is optional.

## Setting the Control Functions for a Command or Process

When you run a command or submit a Process, you have the option to set many control functions. If you plan to use a similar control settings for most tasks, you can define default options once. Then these values are used as the default values for each new command or Process you define. You can also set the following control options when you define a Process, a Send/Receive file command, a Run Task command, or a Run Job command.

✦ The date and time for running a Process or command.

✦ Placing a Process or command in the TCQ Hold queue for future execution.Specify whether the Process is held until you release it or until a connection is established with the SNODE.

✦ Activate Plexclass or CRC checking.

✦ Priority options to define how tasks are managed and how resources are allocated.

✦ A user ID to receive messages when a Process finishes. The method of notification depends upon the capabilities of the operating system.

To add control functions for a Process or command:

1. Open the Process or command and click the **Control** tab.

2. To specify a run date, select one of the following start dates:

   ◆ **Today**—If you want the program to run today.

   ◆ **Date**—To specify a date to run the Process. Click the selection arrow and click a date on the calendar to specify the date.

   ◆ **Day**—If want to run the Process on a certain day; then select a day of the week from the drop-down box.

3. To specify the time to run the task, select one of the following:

   ◆ **Immediate**—To run the Process immediately. This option is only available if you selected **Today** or **Date** in the **Start Date** field.

   ◆ **Time**—To run the Process at a specific time on the start date you selected. Type a time to start the Process in the format hh:mm:ss.

4.  To place the Process in the Hold queue, select one of the following options in the **Hold** field:

    ◆   **Yes**—To hold the Process in the queue in Held Initially status (HI) until explicitly released.

    ◆   **No**—If you do not want to place the Process in the Hold queue. Process executes as resources are available.

    ◆   **Call**—To hold the Process until a connection is established between the PNODE and the SNODE. The Process executes if another Process establishes connection between the nodes.

5.  To place the Process in the Retain queue, select one of the following options in the **Retain** field:

    ◆   **No**—If you do not want to retain the Process after execution.

    ◆   **Yes**—To retain the Process in the Hold queue in Hold Retain status (HR) after execution. You can release the Process later from the Process Monitor.

    ◆   **Initial**—To retain the Process in the Hold queue in HR status for automatic execution every time the Process Manager initializes. Do not specify a start time with this option.

6.  If you are sending Processes to a location that supports CD Plexclass, type the class value of the remote node, from 1 to 8 characters, in the **Plexclass** field.

7.  In the **CRC** field, select one of the following options:

    ◆   **Blank**—To use the default value for the Process that was configured in the Initparms and the Netmap entry for the remote node.

    ◆   **OFF**—To turn off CRC checking.

    ◆   **ON**—To turn on CRC checking.

8.  To change the TCQ priority, type a value in the **Priority** field from 1 to 15, where 15 is the highest priority.

9.  To change the preferred session class, type the preferred session class in the **Class** field, from 1 to the maximum local sessions defined in the network map.

10. When you finish, modify other options or click **OK** to save your changes and close the dialog box.

---

**Note:**   If you select **Yes** for RETAIN and you specify a start time, HOLD status takes precedence. If you set HOLD to **No** or **Call**, and set RETAIN to **Yes**, HOLD is ignored.

---

11. Type the user ID in the **Notify Userid** field.

12. Click **OK** to close the dialog box or click one of the other tabs to continue modifying Process options.

## Setting Security Options

Use the Security options to specify the user IDs and passwords needed to access the PNODE and the SNODE.

To identify the user ID and passwords for the PNODE and SNODE:

1. Open the Process or command and click the **Security** tab.

2. To set security for the PNODE:

   a. Type the PNODE user ID in the **PNODE Userid** fields.

   b. Type the PNODE password in the **Password** field.

3. To set security for the SNODE:

   a. Type the SNODE user ID in the **Snode Userid** field.

   b. Type the SNODE password in the **Snode Password** fields.

   c. To change the password for the user ID on the SNODE, type the new password in the **New Password** field.

   d. Type the new password a second time in the **Verify New Password** field to validate the change.

4. Click **OK** to close the dialog box or click one of the other tabs to continue modifying Process options.

## Assigning Values to Symbolic Variables

Assign values to all symbolic variables before execution. The values are then substituted during execution whenever the symbolic variable is encountered.

To assign values to symbolic variables:

1. Open the Process or command and click the **Variables** tab.

2. To create a new symbolic variable, type the symbolic variable name in the **Variable Name** field. Symbolic variable names are case-sensitive. Enter the symbolic variable exactly as used in the Process.

3. To change a symbolic variable, select the symbolic variable statement in the **Variable** list. If necessary, edit the variable name.

4. Type the symbolic variable value in the **Variable Value** field or edit the existing value.

   You can also specify the following built-in variables for the variable value; these values must be enclosed in quotation marks:

| Built-in Variable | Value |
| --- | --- |
| %JDATE | Specifies the date the Process was submitted in Julian format. The variable is resolved as the submission date of the Process in the format yyyyddd. Among other uses, the value returned is suitable for constructing a file name on the node receiving the file.<br><br>**Note:** The value of the variable is resolved at Process submit time. The value will correspond to the date on which the Process was submitted, regardless of when or how many times the Process is actually executed. |
| %NUM1 | Specifies the submission time of the Process in a 6-digit numeric-value format of minutes, seconds, and fraction of seconds. |
| %NUM2 | Specifies the submitted time of a Process as 1 hex digit |
| %PNODE | PNODE name where the submit occurs |
| %PRAND | Pseudo-random number (6 hex digits) |
| %SUBDATE | Specifies the date the Process was submitted in Gregorian format. The variable is resolved as the submission date of the Process in the format cyymmdd where c is the century indicator and is set to 0 for year 19yy or 1 for year 20yy.<br><br>The value returned can be used to create a file name on the node receiving the file. |
| %SUBDATE1 | Use this parameter to substitute the submitted date in the yyyymmdd date format. |
| %SUBDATE2 | Use this parameter to substitute the submitted date in the yyyyddmm date format. |
| %SUBDATE3 | Use this parameter to substitute the submitted date in the mmddyyyy date format. |
| %SUBDATE4 | Use this parameter to substitute the submitted date in the ddmmyyyy date format. |
| %SUBTIME | Specifies the time the Process was submitted. The variable is resolved as the submission time of the Process in the format hhmmss. The return value can be used to create a file name on the node receiving the file.<br><br>**Note:** The value of the variable is resolved at Process submit time. The value will correspond to the time at which the Process was submitted, regardless of when or how many times the Process is actually executed. |
| %USER | Specifies a variable that resolves to the user submitting the Process |

5.  To remove a symbolic variable, select the variable and click **Delete**.

6.  Click **Add** to save and add the variable to the **Variable** list.

7.  Click **OK** to close the dialog box or click one of the other tabs to continue modifying Process options.

## Specifying Accounting Data

Accounting Data is a free-form, user-defined field that sets up accounting and tracking information about Process execution and data transfers. You can track data transfers by cost centers, department numbers, satellite locations, or any other type of code or identification that would benefit the management of data tracking.

To specify accounting data:

1.  Open the Process or command and click the **Accounting** tab.

2.  Type the information in the **PNODE** field to specify accounting data for the PNODE. The maximum length of the string is 256 characters.

3.  Type the information in the **Snode** field to specify accounting data for the SNODE. The maximum length of the string is 256 characters.

4.  Click **Reset to Defaults** to reset to values specified in Process Defaults page of the Options dialog box.

5.  Click **OK** to close the dialog box or click one of the other tabs to continue modifying Process options.

## Adding or Modifying Comments

Use comments to explain the context of a statement. Adding comments is helpful to explain what the command performs; however, this information is optional. When you view a Process in text format, the comments appear before the associated statement.

To add comments to a Process statement:

1.  Select the statement you want to add comments about in the Process window. To add comments to the Process statement, from the Process window, select **Properties**.

2.  From the **Process** menu, select **Statement Properties**.

3.  Click the **Comment** tab.

4.  Type the text in the **Comment** field.

5.  Click **OK** to save the changes.

# Validating Process Content

When you finish creating or modifying a Process, validate the content of the Process. Validating Process content checks the syntax for errors or missing information. Validation does not check the content of the statements, only that they are formatted correctly. The Process validation sends messages to the Output window. A Validation Successful message means that the syntax is formatted correctly.

To validate Process content:

1.  Open the Process file.

2.  From the **Process** menu, select **Validate**.

3.  View the messages displayed in the Output window. If messages indicate invalid statements, edit the statements and validate the content of the Process again.

# Managing Processes Using a Work List

The Work List is a document containing a sequential list of Process submit requests. It may contain any of the commands that you can submit to the TCQ: send and receive file, submit local or remote Process, submit local or remote program, and submit remote batch job. When you create these Processes, you can submit them directly to the TCQ or place them in a Work List for later submission.

Using Work Lists saves you time and effort by automating routine or repetitive submission tasks. You can build a Work List to periodically submit related work as a single work flow. Your Work List can serve as a library of related or unrelated Processes.

Use Work Lists in the following way:

✦   Submit the entire Work List to the TCQ.

✦   Submit selected items from the Work List.

✦   Create, save, open, close, update, or print a Work List.

✦   Embed a Work List into another application and submit it using object linking and embedding (OLE).

✦   Specify substitution variables at the time you submit the Work List.

## Creating a Work List

To create a new Work List:

1.  From the File menu on the Connect:Direct Requester Main Window, select **New > Work List**.

2.  Add Work List items as desired.

3.  If you want an automatic status monitor established for the Work List as it executes:

    a.  From the **WorkList** menu, select **Work List Properties**.

    b.  Activate **Auto Monitor**.

    c.  If you want the selected units of work to execute serially, turn on **Serial Execution**. This feature causes the Submit action to wait until each unit of work is completed before submitting the next unit of work in the Work List. Processes are considered successfully completed if they do not have a status of HI, PE, or EX.

4.  To define substitution variables for the Work List:

    a.  Click the **Variables** tab.

    b.  To add a new variable, type a name and value in the appropriate boxes and click **Add**.

    c.  To modify a variable, double click the variable. Type a new name or value and click **Add**.

    d.  To delete a variable, highlight the variable and click **Delete**.

5. To add a comment describing the Work List, click the **Comment** tab and type the information in the dialog box provided. This information is optional.

6. Click **OK**.

## Updating a Work List

When you create a Process, a Send/Receive File Command, a Run Task Command, or a Run Job Command, you can add the command to a Work List. Use the Work List to define substitution variables, which you can set at submit time. All the variables must have assigned or default values before the Work List is submitted.

To update a Work List:

1. Create a command. Refer to *Creating Commands* on page 56.

2. Select **Add to Worklist** from the command you are creating.

## Submitting a Work List

Once you create a Work List, perform one of the following actions.

To submit the Work List:

✦ To submit all items in the Work List, select **WorkList > Submit**.

✦ Highlight the items to submit and select **WorkList > Submit Selected**.

## Monitoring a Work List Submission

When you submit a Work List or a task from the Work List, the Work List or task is submitted to the TCQ. If you activate Auto Monitor, a **Work List** status window displays the work items as they are submitted.

## Canceling Work List Operations

Although you cannot close a Work List until all of its tasks are executed, you can cancel the execution of the tasks in a Work List.

To cancel execution of Work List operations:

1. On the **Work List** menu, click **Cancel Execution**.

2. The system stops waiting on processes or programs and will not submit any more tasks. Any tasks that have already started executing will finish, but no other tasks will execute.

## Editing a Work List In Text Format

You can edit the text of a Work List to change command statements.

**Note:** Editing the text of a Work List is a task for experienced users.

To edit a Work List:

1.  Open the Work List that you want to edit.

2.  On the **Work List** menu, click **Edit/View Text**.

3.  Edit the text as necessary.

4.  Close the window. The program displays a message asking if you want to save your work.

5.  Click **Yes**. Your Work List is validated and saved. Any errors found during validation are displayed at the bottom of your screen.

## Inserting a Task into a Work List

To insert a task into a Work List operation:

1.  On the **Work List** menu, click **Insert**.

2.  Select one of the following options to identify the task type to add to the Work List:

    ◆   Submit from File

    ◆   Send/Receive

    ◆   Run Task

    ◆   Run Job

    ◆   Local Program

    ◆   Comment

3.  Type any arguments in the box.

4.  Enter the information for the Process in its dialog box.

5.  Click **OK**. The **Add to Work List** window is displayed.

6.  Click **OK**.

## Validating a Work List

To validate a Work List operation:

1.  Open the Work List that you want to validate.

2.   Select the tasks you want to validate.

3.  On the Work List menu, click **Validate**. The validation information is displayed at the bottom of your screen.

## About the Work List Edit/View Text Window

The **Work List Edit/View Text** window enables you to see the full text of all of the tasks in a Work List. You can edit the text directly if necessary. A keyword must be placed on a line by itself.

If you choose to edit the text, use the following keyword and syntax rules:

| Keyword | Description |
|---------|-------------|
| Set | Work Task List |
| Submit | Ad Hoc Process Task |
| Submit File= | Submit Process Task |
| Run | Local Program Task |

The following table lists the syntax types and restrictions:

| Syntax Type | Restrictions |
|-------------|--------------|
| Work List Task Syntax | Must have the SET keyword<br>May have AutoMonitor and Serial Execution<br>May have variables |
| Local Program Task Syntax | Must have the RUN keyword<br>Must have the FILE keyword<br>Must have arguments or a working directory |
| Submit File Task Syntax | A Submit File task must have SUBMIT FILE= followed by the file name |
| Submit Run Task Syntax | Must have PEND at the end of the Run Task command |

# Saving a Process

When you have finished creating or editing a Process, save the Process for future use.

To save a Process:

1. From the **File** menu, select **Save**.

2. Type a name for the Process including the.CDP extension.

Processes are stored in the Process directory.

# Copying a Process

You can use a Process as a template by copying the Process, making changes to the copy, and saving the copy to a new Process file.

To copy a Process:

1. Open the Process file.

2. From the **File** menu, select **Save As**.

3.  Save the Process with a new file name.

4.  Change the Process statements. To change the Process statement, press Enter to access the **Process Properties** dialog box.

5.  To save the Process file with the changes, from the **File** menu, select **Save**.

# Changing a Submitted Process

You can change the following Process options once a Process is submitted:

✦  SNODE

✦  Hold

✦  Execution

✦  Class

✦  Priority

To change a Process:

1.  Right-click the open **Process Monitor** window and select Change Process. The **Change Process** dialog box is displayed.

2.  Make the necessary changes.

3.  Click **OK**.

# Managing Files with Connect:Direct File Agent

Connect:Direct File Agent is a component of Connect:Direct that provides unattended file management. Before using Connect:Direct File Agent, you must plan how to configure it to automate file management for your site. After planning what you need to accomplish, configure File Agent to connect to a Connect:Direct server, watch the directories that files of interest will be added to, and submit a specified Connect:Direct Process to the server when a file is detected.

## Connect:Direct File Agent

Connect:Direct File Agent provides monitoring and detection capabilities that enhance the automation you accomplish with Connect:Direct Processes. You cannot create Processes with File Agent; however, File Agent variables can pass arguments to a Process. File Agent does not delete, copy, or move files directly, but it helps you accomplish such tasks by submitting the Process you specify in the configuration to the Connect:Direct server. Before you configure File Agent, you must create and test the Connect:Direct Process that you intend to specify as the default Process in the File Agent configuration.

Using the File Agent configuration interface and Help system, you define the *default configuration file* (Default_Config.ser). The default configuration file defines the Connect:Direct server that Connect:Direct for Windows communicates with; the directory, or directories, that File Agent monitors; and how a file added to a watched directory or a detected system event is processed.

You can configure Connect:Direct File Agent to operate in either of the following ways:

✦ Watch for any file to appear in one or more *watched* directories and submit the default Process after detecting the newly added file.

✦ Override the default Process specified and apply either *watched file event* rules (Submit Process rule) or *system event* rules that are enabled for the configuration. File Agent applies a *watched file event* rule to a detected file by checking file properties to determine whether criteria specified by the rule are met. A *system event* rule checks whether a system event meets criteria specified by the rule. If all criteria for a rule are met, File Agent submits the Connect:Direct Process associated with that rule.

You can create File Agent rules based on the following properties:

✦ Full or partial name of the file detected in a watched directory

✦ Size of the file detected in a watched directory

✦ System event title

✦ System event contents (as included in a stack trace)

You can specify more than one rule in a File Agent configuration; each rule can have File Agent submit a different Process.

---

**Note:**   Although you can create multiple rules as part of a File Agent configuration, File Agent processing ends when all criteria for a rule are met. Therefore, you should specify rules so that those with more specific criteria (properties) are listed first in the configuration.

---

For optimum performance, you should configure Connect:Direct File Agent to communicate with the Connect:Direct node where it is installed. You can configure File Agent to use continuous signon and remain connected to the API port for the Connect:Direct server at all times, or configure it to connect to the port only when it needs to. File Agent is available on UNIX, Windows, and OS/390 (and z/OS) operating systems. When you use Connect:Direct for UNIX or Windows, the watched directory is a UNIX path name or a Windows path to the directory. When you use Connect:Direct with OS/390 and z/OS, the watched directory can be a fully specified HFS path name for a file or a directory, a fully specified MVS data set name, a partial MVS data set name, or the name of a partitioned data set (PDS) or partitioned data set extended (PDSE).

File Agent can monitor multiple directories, including local and network directories. File Agent scans the watched directories you specify in the configuration for newly added files (unless you specify a rule to force other operation). By default, Connect:Direct File Agent scans a watched directory once each minute. For example, if you start File Agent at 1:00 p.m., a file added to that watched directory at 12:55 a.m. is not detected. If you start File Agent at 1:00 p.m., and a file is placed in the watched directory at 1:01 p.m., then File Agent detects this newly added file. File Agent detects a file only one time, unless the file is accessed and saved with a later timestamp.

Using Connect:Direct File Agent requires an understanding of Connect:Direct Processes, operating systems, and scripting (for regular expression operator use with File Agent rules).

## Running File Agent

You can run File Agent from a UNIX or MS-DOS command line, configure it to start automatically as a Windows Service at system startup, or configure it to run from a Windows shortcut. Use the command line to verify that File Agent is working correctly or to specify an alternate configuration file. After you run File Agent from the command line to verify that File Agent is operating correctly, run it using the method that requires the least user intervention.

When File Agent runs as a Windows service, it is fully automated, requiring little user intervention. On UNIX, you can modify the initialization sequence of the computer to call the cdfa.sh script and run Connect:Direct File Agent whenever you restart the computer. On OS/390 or z/OS, you must run the appropriate job to start the File Agent configuration interface, or start or shutdown the File Agent.

Depending on your network and how you use Connect:Direct, there might be more than one File Agent running (on different hosts). The first File Agent that connects to a Connect:Direct server

becomes the File Agent *gate keeper*. The gate keeper port is used to keep track of locations monitored in case multiple File Agents are configured to watch a single directory. The gate keeper ensures that only one File Agent detects a file that appears in a watched directory.

---

*Caution:* When running multiple File Agents (on multiple hosts), each File Agent must use a unique gate keeper port number. If multiple agents use the same gate keeper port number, the first File Agent connects, becomes the gate keeper, and operates as configured. However, if another File Agent with the same gate keeper port number connects subsequently, monitoring and rule processing for that File Agent may not occur as configured.

---

## File Agent Logging Capabilities

File Agent logging capabilities vary by platform. Running File Agent from a command line using the verbose argument turns on File Agent logging when it is available. When you run File Agent as Windows service, no information is logged unless an error occurs.

Connect:Direct File Agent provides the following levels of status information when logging is available on a platform:

✦ System log—Shows all system activity. This log is created only when you specify verbose mode or if an error occurs. If you are running verbose mode from the command line, this log information is shown in the command line window. If you are not running in verbose mode, the system log appears in the snaps directory (located in the File Agent installation directory), which is created when the first event occurs.

✦ First Failing Status (FFS) log—One or more logs created when an error occurs. The snaps directory is created as needed and contains the FFS logs for any errors encountered by File Agent.

Using trace commands provided for your platform also captures details related to File Agent operation.

## File Agent Configuration Interface and Help

Instructions for configuring File Agent are available in the online Help system that you access from the configuration interface. Field-level Help is displayed in the bottom pane of the configuration interface. Clicking **Help** displays the online configuration procedures.

---

# Planning the File Agent Configuration

Before you begin configuring File Agent, you must use Connect:Direct to choose or create the Processes that perform the actions you want to automate. You will need to carefully configure File Agent to connect to the Connect:Direct server and to monitor and detect conditions (such as a file

addition to a directory). At detection, Connect:Direct File Agent submits the Process for executing actions that need to be performed in response to those conditions.

Refer to *Connect:Direct File Agent Configuration Examples* to review some configuration scenarios that can help you plan your File Agent configuration. When you configure File Agent, it is best to take an incremental approach; that is, first specify the server connection, a default Process, and the watched directories. Run a test from the command line to ensure that the default File Agent configuration is working correctly. After a successful test of the default configuration, you can run the File Agent Configuration Interface again to start building and testing any File Agent rules that you want to apply, one by one. After you successfully create a default configuration, you can use the file as the basis for other configuration files.

Use the *File Agent Worksheet* to gather the information you need to configure File Agent. Contact your system administrator for the site-specific information you will need to establish a connection to the Connect:Direct server. As you complete the worksheet, run the File Agent Configuration Interface and use the File Agent Help system to learn about entering parameters. The Help system and field-level help provide complete descriptions of parameters and arguments you can specify in the configuration file. Make copies of this worksheet if you have to configure File Agent on multiple Connect:Direct servers.

The following diagram illustrates the flow of steps for setting up File Agent for use in a production environment.
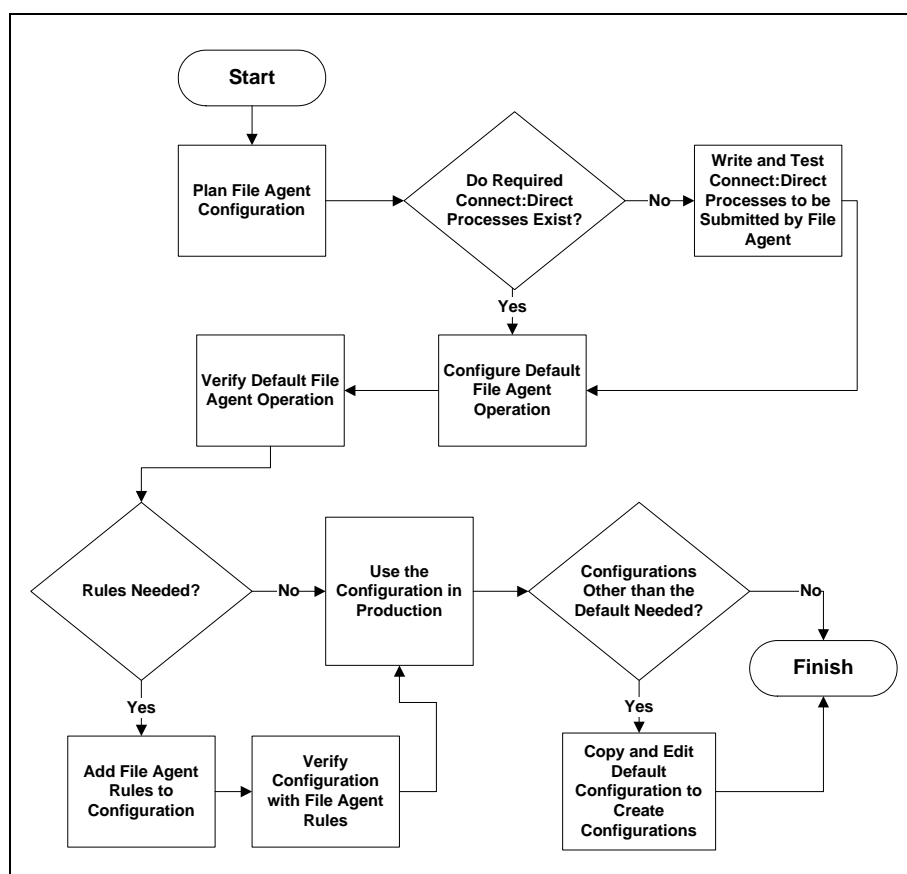
# File Agent Worksheet

| **Connect:Direct Server Connection Information** | |
| --- | --- |
| Userid for API (for connecting to the Connect:Direct server) **Required**<br>Must match the user ID used to submit the default Process. | _____ |
| Password for API (for connecting to the Connect:Direct server) **Required**<br>Must match the password used to submit the default Process. | _____ |
| API host DSN name (name of the host on which the Connect:Direct server is located) **Required** | _____ |
| API port (default =1363)<br>1–5 digit port number that Connect:Direct File Agent uses to connect to the Connect:Direct server API. **Required** | _____ |
| Gatekeeper port (default=65530)<br>Port used to track directory monitoring and ensure that multiple File Agents do not monitor a single directory. **Required** | _____ |
| Gate keeper DNS name (optional)<br>(default=127.0.0.1) | _____ |
| **Default Process and Watched Directory Information** | |
| Watched directories:<br>For Windows and UNIX, one or more valid specifications of paths (Windows) or pathnames (UNIX). For OS/390 and z/OS, one or more fully specified HFS pathnames of a file or directory, or a full or partial MVS data set name. **Required**<br>List one valid entry per line. | |
| | |
| | |
| | |
| | |
| | |

| Default Process and Watched Directory Information | |
|---|---|
| Default Process:<br><br>Windows and UNIX: Valid path and name of the file that contains the default Process on the Connect:Direct server.<br><br>OS/390 and z/OS: Member Name in DMPUBLIB<br><br>**Note:** If you do not specify a default Process or create a rule, no processing is performed when a file or event is detected. | _____<br><br>_____ |
| Default arguments (See Help for complete list.)<br>Argument string to pass to the default Process in the following format:<br>&FA_*XXXX_XXX.*<br>**Note:** The percent sign (&) and period (.) are required. | _____ |
| Error Process: | _____ |
| Error arguments | _____ |
| Process class (default=1) **Required** | _____ |
| Process priority (default=1) | _____ |
| Watched file interval (default=1 minute) | _____ |
| File completion delay (default=1 minute) | _____ |

**Note:** If you are using X Windows, the X11 display variable is used to connect to the GUI server for terminal emulation. The File Agent Configuration Interface will display on the monitor specified for the X11 display variable. If you want to display the File Agent Configuration Interface on a Windows computer, you must specify the network ID of the terminal you want to use for displaying the File Agent Configuration Interface.

# Connect:Direct File Agent Configuration Examples

The following examples illustrate three typical scenarios for using Connect:Direct File Agent. Fields that are not required to be set for the operation demonstrated in the example are not included in the tables of configuration parameters. Required fields are indicated by an asterisk(*) in the File Agent Configuration Interface, and all fields are described in field-level Help. You can use the sample scenarios to become familiar with settings for parameters you must set using the configuration interface in order to accomplish watched directory monitoring and file detection needs.

See the *File Agent Worksheet* on page 77 for a description of the parameters required to establish the connection. The tables of sample data for these scenarios assume that you have already configured the site-specific parameters required to establish a connection to the Connect:Direct server where File Agent is installed. The sample scenarios also assume that the Connect:Direct Processes that will perform tasks associated with Connect:Direct File Agent file detection activities have been created.

## Detecting a File Added to a Watched Directory on an OS/390 or z/OS System

Some users need to access a report file that is expected to be transferred to a location that only administrators can access. File Agent can be configured to perform the processing on an OS/390 or z/OS system:

✦ Monitor the watched data set called EASTERN.Q1.REPTS.

✦ Submit a default Process called DEFPROC. The default Process has been created to copy a file detected in the watched data set to a specified location for access by users.

| Tab | Field | Sample or Description |
|---|---|---|
| **File agent** | Watched directories | Type EASTERN.Q1.REPTS to specify the fully qualified MVS data set name to watch. |
| | Default Process | Type DEFPROC, the member name for the Process in DMPUBLIB. |
| | | **Note:** If no default Process is specified and the file does not match a rule, then no processing occurs. |

## Detecting a File by File Size on a Windows System

Customer transaction files are regularly transferred into the Windows directory c:\monthend\datafile. Files larger than 1 MB require special processing that will automatically be performed on files in a certain directory.

The sample values in the following table accomplish the following processing:

✦ Monitor the watched directory c:\monthend\datafile

✦ Apply the rule titled Find big file to detect files larger than 1 MB.

✦ Override the default Process and submit a Process that is associated with the Check file size rule. This fixbigfile.cdp Process will copy a file larger than 1 MB from the c:\monthend\datafile directory to the c:\reprocess directory.

✦ Pass the path and file name of the file that meets the criteria for the find big file rule to the Process fixbigfile.cdp.

| Tab | Field or Dialog Box | Actions and Sample Entry |
|---|---|---|
| **File agent** | **Watched directories** field | Specify the directory to watch. Type the directory path for the directory that File Agent will watch:<br>**c:\monthend\datafiles** |
| **Rules** | **Create rule** dialog box | Click **New** and type the name you want to give the rule in the field:<br>**find big file**<br>Click **find big file** in the list of rules and click **Edit**. |
| | **Match criteria list for rule "find big file"** | Specify the criterion to check for a detected file. Select the default criterion name, **Not enabled: system event title matches " "** and click **Edit match**. |
| | **Edit match criterion for rule "find big file"** dialog box | ◆ Click **Enabled** to enable the criteria you are about to specify.<br>◆ Click **Size of the newly arrived file**<br>◆ Click **Matches** to display the options for the comparison. Click **Greater than** to define the how the file size should compare.<br>◆ Type **1024** in the **Compare size** field and click **OK**. |
| | **Process name** field | Scroll down to view the **Submit Process information for watched file event rule "find big file"**. Type the directory path and filename for the Process that File Agent is to submit when the rule Find big file results in a match:<br>**c:\processes\fixbigfile.cdp**<br>Click **Done** and then click **Save**. |

## Detecting a System Event by Title on a Windows System

IndexOutOfBoundsException is the title of an event that indicates a number is outside of an expected range. In the following example, File Agent is used to detect an event with IndexOutOfBoundsException in the title, pass a string (the event title) to a Connect:Direct Process, and then submit a Process to the Connect:Direct server that will perform actions the environment requires for this type of event. In this scenario, the event IndexOutOfBoundsException could indicate activity that a network administrator should investigate. Because the site uses a Connect:Direct mailbox system, the configuration will include the administrator's account to be notified when Connect:Direct File Agent submits a Process for the IndexOutOfBounds rule.

The sample values in the following table accomplish the following processing:

✦ Override the default Process and submit \processfolder\oo_boundserrproc.cdp

✦ Send a message to the administrator's mailbox system account after submitting the oo_boundserrproc.cdp Process for the rule.

| Tab | Dialog Box, Window, or Field | Description/Example |
|---|---|---|
| **Rules** | **Create rule** dialog box | Type **index out of bounds** as the name of the rule you are creating. |
| | **Match criteria list for rule "index out of bounds"** window | Select the default criteria **Not enabled: System event title matches " "** and click **Edit match**. |
| | **Edit match criterion for rule "index out of bounds"** dialog box | ◆ Click **Enabled** to enable the criteria you are about to specify.<br><br>◆ Click **System event title** as the criterion to match for the rule.<br><br>◆ Click **Matches** on the drop-down field to see the options for comparison to a string.<br><br>◆ Click **Contains** to specify how the compare string should relate to a system event title that File Agent detects.<br><br>◆ Type **IndexOutofBounds** as the **Compare String** to indicate that the system event title should include this string.<br><br>◆ Click **OK**. |
| | **Submit Process information for system event rule "index out of bounds"** window | Type information into the fields that will define the Process to submit and the mailbox user to notify after the Process is submitted. |
| | **Process name** field | Type **c:\processfolder\errproc.cdp** to specify the path and file name for the Process File Agent submits when a file meets the rule criteria. |
| | **Notification userid** field | Type **adminjim@company.com** to specify the user to notify when File Agent submits the Process. |

## Passing the UNIX Pathname for a Detected File to a Process

Because Connect:Direct File Agent can watch multiple directories for the appearance of a new file, the Connect:Direct Process that File Agent is to submit to the server at the appearance of a new file might need to reference the Windows Path or UNIX pathname for the detected file as part of commands and statements in the Process.

In the following example, a UNIX pathname is passed to the default Process, copynewfile.cdp.

| Tab | Dialog box, Window, or Field | Sample Entry |
|-----|------------------------------|--------------|
| **File agent** | **Watched directories** | Type one UNIX pathname per line for each location File Agent is to watch for the appearance of files:<br>**user/bin/monthend/**<br>**quartend/easterndiv/errorfiles**<br>**managers/special/reports**<br>File Agent checks these pathnames for new files. |
| | **Default process** | Type the UNIX pathname and filename for the Connect:Direct Process that File Agent is to run when a file appears in any watched directory specified:<br>**user/bin/admin/copynewfile.cdp**<br>The pathname where File Agent detected a new file is passed to this Process. |
| | **Default arguments** | Type the File Agent variable for passing a UNIX pathname or Windows path, including the leading percent sign (%) and the ending period (.):<br>**&FAP=%FA_PATH_FOUND.**<br>In this example, **&FAP** is the variable to which File Agent will pass the UNIX pathname where the file was detected. **%FA_PATH_FOUND.** is the File Agent variable used to indicate the information to pass to the Connect:Direct Process. |

# Chapter 7

# Managing Processes

Connect:Direct for Windows provides the following tools to manage Processes:

✦ Process Monitor—Enables you to view Processes in the TCQ, release held Processes, change the status of a Process, and delete a Process. After you submit a Process, it is placed in the Transmission Control Queue (TCQ). The Process Monitor

✦ Process Notification Utility—Enables you to change the notification method. You define the method of notifying users of Process execution when you install Connect:Direct for Windows.

✦ Windows Event Logging—Logs informational, warning, and error messages.

✦ Messages—If you need to troubleshoot the meaning of an error message, you can view more information about an error message with the Message Lookup.

✦ SNMP—If you use Microsoft Windows SNMP to capture messages, you must identify the messages you want to include and determine if messages are trapped or logged to the event log.

✦ Activity Log—This document contains a list of Connect:Direct activities, including every significant activity that you have requested from the time the activity log is opened until it is closed. Each activity record logged is maintained with the significant data associated with it. For example, when you save a file, the Save activity record is stored, with the file name of the saved document.

The Activity Log is created at startup and is enabled by default. It can be viewed in the **Output** window by clicking the Activity Log tab. You may create, save, open, close, and print Activity logs. To save an activity log, you must specify that you want the activity log written to a file. When you open a new activity log, the currently active Log is inactivated. You must close Connect:Direct Requester in order to open an activity log.

## Understanding the TCQ

After you submit Connect:Direct Processes, they are stored in the TCQ. The TCQ controls Connect:Direct Process execution. As sessions are available, the TCQ releases Processes to begin execution according to the scheduling parameter values and the class and priority of each Process. Use the Process Monitor to manage and view the status of submitted Processes.

## TCQ at Server Startup

The initialization parameter **tcq.start** determines what the TCQ does with existing Processes. The default value is **tcq.start=<u>w</u>** (warm start), which specifies that all existing Processes in the TCQ are retained. A warm start restarts any Processes submitted with the **Process** statement parameter **retain=initial**. You can change the parameter to **tcq.start=c** (cold start) to delete all existing Processes in the TCQ when the server restarts.

# TCQ Logical Queues

As Connect:Direct Processes are submitted, they are placed in one of the four TCQ logical queues: Execution, Wait, Timer, and Hold. As sessions are available, the TCQ releases Processes to begin execution according to the Process class and priority, as shown in the following figure:
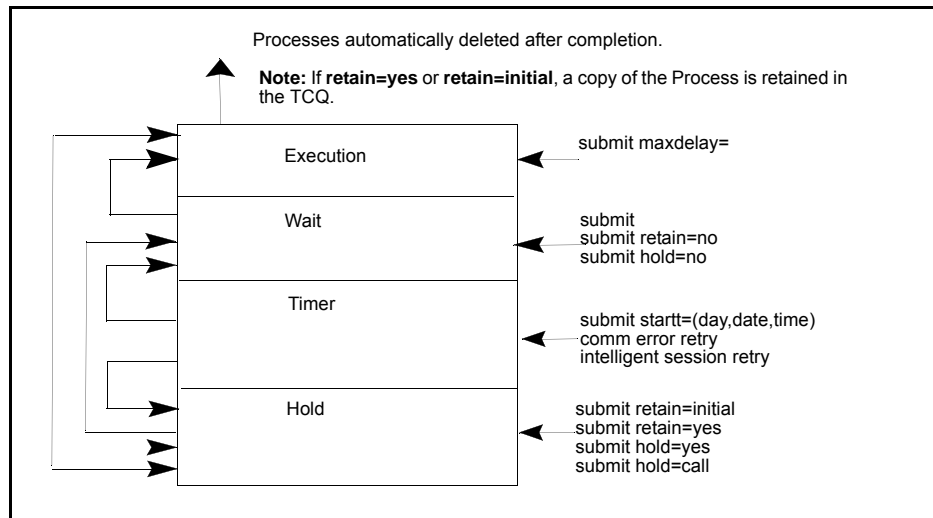


Each Process in the queue is assigned a status. The queues and status values are identified in the following sections.

## Execution Queue

Processes are placed in this queue after the connection to the SNODE occurs. Processes typically come from the Wait queue, but also can be placed in the Execution queue by a **submit** command with **maxdelay=** specified. After a Process successfully finishes, it is automatically deleted from the Execution queue. When a session is interrupted, the Process moves from the Execution queue to the Timer queue, if retry values are specified. If connection is not made before the retry values are exhausted or if retry values are not specified, the Process moves to the Hold queue with a status of HE. The following table displays the status values assigned in the Execution queue:

*Connect:Direct for Windows System Guide*

| Status | Comment |
| --- | --- |
| EX | Process is executing between two Connect:Direct nodes. |
| PE | Processes waiting for Process start messages to be exchanged between the PNODE and the SNODE. This is the initial queue status when a Process is submitted with **maxdelay=** specified. |

## Wait Queue

Processes are placed in the Wait queue while waiting for an available connection between the PNODE and the SNODE. Processes can come to the Wait queue from the Hold queue or the Timer queue. Processes also can be placed in the Wait queue by a **submit** command with no parameters specified, **submit** with **retain=no**, or **submit** with **hold=no**. After the connection is made, Processes automatically move to the Execution queue. The following table displays the status values assigned in the Wait queue:

| Status | Comment |
| --- | --- |
| WC | The Process is ready to execute, but no session is available. This Process runs as soon as a new session is created or an existing session becomes available. |
| WR | The Process is in retry status. The number of retries and intervals between retries is specified in the network map. |
| WA | The initial queue status when a Process is submitted without HOLD or RETAIN specified. This Process is ready to execute as soon as possible. |

## Hold Queue

Processes are placed in this queue while waiting for operator intervention before progressing to the Wait queue. This queue enables operators of the PNODE and SNODE to coordinate and control Process execution.

Processes are placed in the Hold queue by a **submit** command with **retain=initial**, **retain=yes**, or **hold=yes** parameters specified. Processes submitted with **hold=call** also are placed in the Hold queue. Processes are moved from the Timer queue to the Hold queue by a **change process** command with **hold=yes** specified. Processes are moved from the Hold queue to the Execution queue by a **change process** command with the **release** parameter specified.

The following table displays the status values assigned in the Hold queue:

| Status | Comment |
|--------|---------|
| HC | The Process was submitted with **hold=call** specified. A session started from the remote node causes the Process to be moved to the Wait queue in WC status. The Process is placed in the Execution queue when the Process is selected for execution. |
| HI | The Process was submitted with **hold=yes**. The Process can be released later by a **change process** command with **release** or **hold=no** specified. |
| HE | A session error or other abnormal condition occurred. |
| HO | A **change process** command with **hold=yes** was specified. |
| HR | The Process was submitted with **retain=yes** or **retain=initial** specified and has already executed. The Process can be released later by a **change process** command with **release** specified. |
| HS | The Process was suspended due to a server shutdown. |

## Timer Queue

Processes are placed in this queue by a **submit** command with the **startt** parameter specified. Processes in the Wait for Start Time (WS) status are waiting for the start time to arrive before moving to the Wait queue. Processes also are placed in the Timer queue in Retry (WR) status after an error, such as a line break or a lost connection. Connect:Direct automatically tries to execute the Process again based on the number of times to retry and the delay between retries as specified in the **submit** command, **Process** statement, network map parameters, or initialization parameters. Processes move from the Timer queue to the Wait queue. A **change process** command with **hold=yes** specified moves the specified Process from the Timer queue to the Hold queue.

The following table displays the status values assigned in the Timer queue:

| Status | Comment |
|--------|---------|
| RE | The Process is in retry status. The number of retries and intervals between retries is specified in the network map. |
| WS | The Process was submitted with a start time (**startt**) or date that has not expired. When **startt** is reached, the Process is placed in the Wait queue for scheduling for execution. |

# Viewing Processes in the TCQ

To view the Processes in the TCQ, use a Process Monitor. You can use more than one monitor to view different queues or to look at Processes based on different criteria. You can arrange Process Monitors and save the Workspace view so that when you enable that view, the Monitors are

automatically displayed. You can define how you want to display the Process Monitor, what types of Processes you want to view, and what queues you want to view.

## Creating a Process Monitor

To create a Process Monitor:

1. From the **File** menu, select **New** > **Process Monitor**.

2. Do one of the following:

   ◆ To monitor all Processes, select **All Processes**.

   ◆ To monitor only those Processes that meet certain criteria, select **Filter Using Selection Criteria Pages**. Define the criteria based on the options on the remaining property pages. Refer to *Monitoring Processes Based on Selection Criteria* on page 87.

   ◆ To refresh the monitor at specified intervals, select the **Refresh every _ seconds** option and enter the interval from 1-999 seconds.

   ◆ To monitor new data that was created since the Process Monitor was closed, select **Refresh on open**.

3. Select the node on which to monitor Processes in the **Node** field.

4. Click **OK**.

## Monitoring Processes Based on Selection Criteria

You can monitor Processes based on selection criteria. Select the criteria to use to include in a Process Monitor. Select one or more of the following filters: the status of a Process in the queue, the remote nodes included in a Process, the user who submitted a Process, or a Process name or number.

To identify what Processes to include in a Process Monitor:

1. Open a Process Monitor window.

2. Select **Filter Using Selection Criteria Pages**.

3. To include Processes in a Process Monitor based on TCQ queue:

   a. Click **Status**.

   b. Select the status types you want to monitor. You are not limited to the number of status types you can request. The status types are defined in the following table:

| Status | Description |
| --- | --- |
| Execution | Processes that are being serviced by the session manager. |
| Pending Execution | The Process was submitted with the maximum delay option set to zero. |

| Status | Description |
|---|---|
| Waiting Connection | The Process is ready to execute as soon as a session is available. |
| Waiting Start Time | The Process is waiting in the Timer Queue because it was submitted with a start time or date that has not expired. When the start time is reached, the Process is placed in the Wait queue to schedule for execution. |
| Held Suspension | The operator issued a delete process request with hold set to Yes. |
| Timer Retry | The Process is waiting for a retry attempt. |
| Held for Call | The Process was submitted with the Hold option set to Call. A session started from either node moves the Process to the Wait queue in WC status. The Process is placed in the Execution queue when the Process is selected for execution. |
| Held due to Error | A session error or other abnormal condition occurred. |
| Held Initially | The Process was submitted with the Hold option set to Yes. |
| Held by Operator | A change process request with hold set to Yes has been issued. |
| Held by Retain | The Process was submitted with retention set to Yes or Initial. |
| Select All | To monitor all status types. |
| Unselect All | To deselect all status types. |

4. To view Processes based on Process name:

   a. Click the **Process** tab.

   b. Select the Queue to search in the **Queue(s)** field.

   c. To monitor a Process by name, click the **New** icon, type the name of the Process, and press **Enter**. Repeat this step to add additional Process names.

   d. To monitor a Process by number, click the **New** icon, type the Process number, and press **Enter**. Repeat this step to add additional Process numbers.

5. To view Processes based on a remote node:

   a. Click the **Nodes** tab.

   b. To select all remote nodes defined in a network map:

   • Select the network map node in the **Netmap Nodes** field.

   • Click the right arrow or select **All** to select all network map nodes.

   • To type a remote node name, click the **New** icon, type the remote node name, and press **Enter**. Repeat this step to add remote node names.

6. To view Processes based on a user ID:

   a. Click the **Submitter** tab.

   b. Type the user ID or user proxy in the **User** field.

   c. Type the node name in the **Node** field and click **Add**. Continue entering node names and clicking **Add** until you have added all user IDs you want to monitor.

7. Click **OK**.

## Opening a Process Monitor File

To open a Process Monitor file:

1. Select **Files > Open**.

2. From the Files of type drop down box, select Process Monitors (*.cdm) to display only Process Monitor files.

3. Locate and select the Process Monitor file to open.

## Saving a Process Monitor

Saving a Process Monitor to a file lets you use the same format and monitor criteria again. When you save a Process Monitor, you are saving the criteria and the format of the Process Window; not the Process information displayed at the time you save the monitor.

To save the Process Monitor:

1. From the **File** menu, select **Save**.

2. Type the name of the Process monitor with the extension .CDM.

3. Click **OK**.

## The Process Monitor Output

Each line in a Process Monitor is a summary record of the current status of the Process in the TCQ. Depending on the status of the Process or the step being executed, some fields may be blank. The lines are numbered sequentially. A "W" to the left of the sequence number indicates the Process has met the warning conditions you established.

The following information is displayed:

| Field | Content |
| --- | --- |
| Name | Process name. |
| Number | Process number assigned when the Process is placed in the TCQ. |
| Status | Status of the Process in the queue. |
| Step Name | Currently executing step of the Process. |

| Field | Content |
| --- | --- |
| Queue | Logical queue where the Process is currently located (Execution, Hold, Wait, or Timer). |
| Byte Count | Number of data bytes read or written. |
| Submit Node | Node name from which the Process was submitted. |
| Submitter | User ID of the person who submitted the Process. |
| Pnode | Primary node in the Process. |
| Snode | Remote or partner node in the Process. |
| Message | Message associated with the current Process step. |
| Source File | Name of the source file. |
| Submit Date/Time | Date and time the Process was submitted. |
| Schedule Date/Time | Date and time the Process was scheduled to be submitted. |
| Retain | Identifies if the Process is to be retained in the TCQ for future submission. |
| Function | Type of Process statement currently executing (COPY, RUNJOB, RUNTASK, SUBMIT). |
| CC | Completion Code.<br>0—Success<br>16—Severe error |
| FDBK | Feedback code. |
| Message Text | Message short text. |
| Message Data | The substitution variables with their values for the short text. |
| Log Date/Time | The date and time the Process record was created. |
| Hold | The hold status of the Process. |
| Class | Session class on which the Process is executing. |
| Prty | TCQ priority of the Process. |
| Local node | Indicates whether the local node is the SNODE or the PNODE in the Process. |
| From Node | Indicates whether the local or remote node is the FROM node in a COPY. |
| Compression | Compression factor used in a copy step. |
| Checkpoint Restart | Indicates use of checkpointing in a copy step. |
| Restart | Indicates whether the Process was restarted. |
| Source Disp 1 | Source file disposition parameters. |
| Source Disp 2 | Source file disposition parameters. |

| Field | Content |
| --- | --- |
| Source Disp 3 | Source file disposition parameters. |
| Record Count | Number of data records read or written. |
| Exec | The operating system execution priority for the Process. |
| Total Bytes | Number of data bytes read or written. |
| Bytes Sent/Received | Number of data bytes plus RU bytes sent. |
| RUs Sent/Received | Number of request/response units sent. |
| Dest File | Name of the destination file. |
| Dest Disp 1 | Destination file disposition parameters. |
| Dest Disp 2 | Destination file disposition parameters. |
| Dest Disp 3 | Destination file disposition parameters. |

### Using the Output Display

Through the output display, you can accomplish the following tasks:

✦ To view the detail fields associated with a Process, double click on the Process summary line.

✦ To view the condition which caused the warning flag on a Process, select the flagged Process. The condition is posted at the bottom of the display.

> **Note:** You cannot change the size of the rows.

✦ To change the order of the fields in your display, select the heading of the field you want to move, press the Shift key, point the mouse at the heading, and drag and drop the column to its new location. You may move any column. After you close the reformatted display, that format becomes your default format.

# Notifying

When you install Connect:Direct for Windows, you identify the method used to notify a user of Process execution. If you want to change the method used to notify selected users when a Process executes, use the Change Notification utility. This application updates the Registry entries used by Connect:Direct to perform the specified notification.

## Notification Methods

Connect:Direct for Windows provides two notification methods:

✦ NT Broadcast—NT Broadcast notification is performed using the Windows *net send* command.

✦ SMTP—E-Mail notification is performed using Simple Mail Transfer Protocol (SMTP) notification, a simple ASCII protocol.

## NT Broadcast

Connect:Direct for Windows uses the Windows *net send* command to accomplish notification through NT Broadcast. The notification is sent to the specified user or users attached to a particular computer or domain on behalf of the user context that Connect:Direct is running in.

After the notification ID is specified from within the Process Control Options category of the **Process Definition** dialog box, Connect:Direct supplies the user as an argument in the *net send* command. If the specified recipient cannot be located, or is not logged on, the notification fails and is not attempted again.

## SMTP Notification

Connect:Direct uses the SMTP notification method and exchanges E-Mail using TCP/IP and a message transfer agent (MTA). The SMTP standard is one of the most widely used upper layer protocols in the Internet Protocol stack. This protocol defines how to transmit messages (mail) between two users. SMTP uses spooling to allow mail to be sent from a local application to the SMTP application, which stores the mail in some device or memory. Once the mail has arrived at the spool, it is queued. A server checks to see if any messages are available and then attempts to deliver them. If the user is not available for delivery, the server tries later. Eventually, if the mail cannot be delivered, it will be discarded or returned to the sender.

## Changing Process Notification

To change the Process notification setup, perform the following steps:

1. Click **Start** and point to **Programs**.

2. Select **Sterling Commerce Connect:Direct v4.4.00 > CD Change Notification Utility**.

3. The **Connect:Direct Notification** dialog box is displayed.

4. In the **Node Name** field, type or select the local node you are configuring. When you select a node, the current method of notification is displayed in the **Transport** field.

5. NT Broadcast is the default setting for the notification methods. If want to use SMTP as the notification method, select SMTP in the **Transport** field and type the following information in the fields provided:

◆ **Host Address**—The address of the SMTP server as either the DNS name or the IP address. For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*.

◆ **Host Port**—The port to use to connect to the SMTP server. The default is 25.

- ◆ **Sender**—The E-Mail address you wish to use as the sender.

- ◆ If you want to activate authentication, turn on **Authentication** and provide a User ID and password to use to sign on to the SMTP server.

6. Click **Save**.

7. Click **OK**.

# Viewing Process Statistics

Connect:Direct records the history of a Process in a single relational database. You can review this information to examine details about server execution. Connect:Direct for Windows also uses the Windows Event Logging facility to log certain messages that the server generates during execution. Connect:Direct selects specific record IDs or event types when logging statistics. Connect:Direct assigns these record IDs severities and passes them on to the Windows Event Logging facility. The **event.log** initialization parameter controls the statistics IDs that Connect:Direct sends to the Windows event log.

The statistics file stores information about all events that take place within the Connect:Direct server for a specific period of time. The amount of time is determined by the value specified for the **stat.max.age** initialization parameter. Each record within the statistics file consists of fields that contain general information about the record and a field that contains the statistics or audit information to log.

In the fixed portion, the following fields are defined for the statistics files:

| Field | Description |
| --- | --- |
| EntryDateTime | Date and time that the record was inserted into the statistics file. The format is YYYYMMDD HHMMSSTTT. |
| RecID | A 4-character ID that describes the type of record. |
| RecCat | A 4-character ID that describes the category of record. |
| PrcName | Name given to the Process when it is submitted. |
| PrcNumber | Number assigned to the Process when it is submitted. |
| StartTime | Start time of a Process step, if this is a Process step statistics record. |
| StopTime | Stop time of a Process step, if this is a Process step statistics record. |
| SubmiterNode | Node that the Process was submitted on. |
| SubmitterUserID | User that submitted the Process. |
| RemoteNode | Remote node for the Process. |
| ConditionCode | Condition code for the statistics record. |
| AuditInfo | Variable portion of the statistics record. |

If the existing statistics file cannot be extended, the server issues a message to the Windows event log and terminates immediately. The server terminates all activity as if issuing a **stop** command with the **immediate** parameter. Any statistics records that are usually written during **stop immediate** processing are not created in this situation.

## The Statistics Monitor Window

The **Statistics Monitor** window displays the following information:

| Field | Definition |
| --- | --- |
| Log Date/Time | Specifies the date and time the statistics record was created. |
| Type | Specifies whether the record is event or Process related. CAEV indicates that the record is related to an event. CAPR indicates that the record was related to a Process. |
| RecID | Specifies the type of statistics record generated. |
| CC | Specifies the completion code. |
| FDBK | Specifies the feedback code for the module. |
| MSGID | Specifies the Message ID. |
| PName | Specifies the Process name. |
| PNum | Specifies the Process number. |

Each row is a statistics record. Select the row to view statistics record details. You can also modify the appearance of the Statistic Monitor window.

## Deleting Statistics Records

The server deletes statistics records based on the value specified for the statistics initialization parameter **stat.max.age**. The **stat.max.age** parameter controls the amount of time that the server retains the statistics record in the database. When statistics records reach the value specified by the **stat.max.age** parameter, the server automatically deletes them. To retain all of your records, back up your database regularly or set **stat.max.age=** to disable automatic deletion.

## Controlling Statistics File Content

Use the **log.commands** and **log.select** initialization parameters to control whether the statistics file logs output from all commands and whether the file logs commands that select Processes and select statistics. See *About Statistics Parameters* on page 147 for a description of the statistics parameters.

## Creating a Statistics Report

To create a statistics report:

1. To open a new statistics monitor, from the **File** menu of Connect:Direct Requester, select **New > Statistics Monitor**.

2. To modify an existing statistics monitor, select **File > Open** from Connect:Direct Requester and select from the Statistic Monitor directory.

3. To view statistics for all Processes executed over a specific period of time:

    a. Select **All Statistics for Last _ hrs.min**.

    b. Enter the time period in hours and minutes.

    c. Click **OK**.

4. To view only those Processes that meet certain criteria:

    a. Select **Filter Using Selection Criteria Pages**.

    b. Define the criteria on the remaining property pages. Refer to *Selecting Statistics Based on Process Names or Numbers* on page 96.

5. To view statistics based on step and Process completion only select **Step and Process Completion Statistics Only**.

6. To refresh the monitor at specified intervals:

    a. Select **Refresh every _ minutes**.

    b. Type the interval in minutes (1–60).

7. To view new statistics that were created since the Statistic Monitor was closed, select **Refresh on open**.

8. To view the last statistic as it occurs, select **Autoscroll**.

9. If necessary, select the node to monitor in the **Node** field.

10. Click **OK**.

## Selecting Statistics Based on Process Names or Numbers

You can select Statistics based on Process names or Process number. The Process number is the number assigned by Connect:Direct when the Process is submitted.

To view Statistics based on Process name:

1. Open or create a Statistics Monitor.

2. Select **Filter Using Selection Criteria Pages**.

3. Click **Add**.

4. To view statistics based on Process name:

   a. Click the **Process** tab.

   b. Click the new icon.

   c. Type the Process name and press **Enter**.

   d. Continue clicking the new icon, typing Process names, and pressing **Enter** until you have added all Process names you want to monitor.

5. To view Statistics based on Process number:

   a. Click the new icon.

   b. Type the Process number and press **Enter**.

   c. Continue clicking the new icon, typing Process numbers, and pressing **Enter** until you have added all Process numbers you want to monitor.

6. To view statistics based on a user ID:

   a. Click the **Nodes** tab.

   b. Click the new icon.

   c. Type the user ID or user proxy and press **Enter**.

   d. Repeat these steps until you have added all user IDs you want to monitor.

7. To view statistics based on who submitted a Process:

   a. Click the **Submitter** tab.

   b. Type the user ID or user proxy.

   c. Type the node on which the user is located.

   d. Click **Add**.

   e. Repeat these steps until you have added all submitters you want to monitor.

8. To view statistics based on a range of date or time:

   a. Click the **Ranges** tab.

   b. Activate the **Date/Time Range** option.

   c. Select one of the following options:

| Option | Description |
|---|---|
| Range | Type the time range to monitor in hours and minutes. |
| Start and Stop | Type the beginning and ending date and time range or select the date from the calendar in the **Start** and the **Stop** fields. Type the date in the format mmm/dd/yyyy and the time in the format hh:mm:ss Xm. |
| Start Only | Type the beginning date and time range or select the date from the calendar in the **Start** field. Type the date in the format mmm/dd/yyyy and the time in the format hh:mm:ss Xm. |
| Stop Only | Type the ending date and time range or select the date from the calendar in the **Stop** field. Type the date in the format mmm/dd/yyyy and the time in the format hh:mm:ss Xm. |

d. To select statistics based on a condition code, select the condition code on which the condition is based and select a conditional operator as detailed in the following table:

| Option | Description |
|---|---|
| Conditional Code Range Delimiters | If you want to limit the statistics based on error code values, select the condition code delimiters in the first drop-down box including:<br><br>◆ Equal to<br><br>◆ Greater or Equal to<br><br>◆ Greater than<br><br>◆ Less or Equal to<br><br>◆ Less than<br><br>◆ Not Equal to |
| Conditional Code Range Error Codes | If you want to limit the statistics to selected error codes, select the error code value from this drop-down box including:<br><br>◆ 0<br><br>◆ 4<br><br>◆ 8<br><br>◆ 16<br><br>◆ Any |

9. To generate statistics for specific source or destination files:

   a. Click the **Files** tab.

   b. Click the **Add** icon for the file type: either source or destination files.

   c. Type the name of the file including the full path name.

      d.   Press **Enter**.

      e.   Repeat this procedure until all files that you want to monitor are identified.

10. To generate statistics based on record types:

      a.   Click the **Records** tab.

      b.   Select a record category from the drop down menu for the **Record Category** field.

      c.   Select a record from list. To remove a selection, click the item again.

      d.   To select all the records, click **Select All**.

      e.   To clear all of selections, click **Unselect All**.

11. To generate statistics based on user-defined records:

      a.   From the **Records** tab, click **Add** in the **User Defined Records** dialog box.

      b.   Type the first four characters of the message ID.

      c.   Press **Enter**.

      d.   Repeat these steps until you have added all records you want to monitor.

12. Click **OK**.

## Statistics Report Output

Each line in the Statistics report represents a statistics record. The following information is displayed for each record type:

| Field | Description |
| --- | --- |
| Log Date/Time | The date and time the statistics record was created. |
| Type | The record category. |
| | CAEV—Specifies that the record is related to an event, such as a Connect:Direct shutdown. |
| | CAPR—Specifies that the record is related to a Process. |
| Rec ID | The type of statistics records, such as Copy Termination records or Connect:Direct initialization event records. Identifiers are as follows: |

| Record ID | Category |
| --- | --- |
| CHGP | Event |
| COAC | Event |
| CRHT | Event |
| CSTP | Event |
| CTRC | Process |
| CTRM | Event |

| Field | Description | |
|---|---|---|
| Rec ID (cont'd) | **Record ID** | **Category** |
| | CUKN | Event |
| | CXIT | Event |
| | DELP | Event |
| | FLSP | Process |
| | FMRV | Process |
| | FMSD | Process |
| | GPRC | Process |
| | IFED | Process |
| | LIEX | Event |
| | LIOK | Event |
| | LWEX | Event |
| | NAUH | Event |
| | NMOP | Event |
| | NUIC | Event |
| | NUTC | Event |
| | NUIS | Event |
| | NUTS | Event |
| | PERR | Process |
| | PFLS | Process |
| | PRED | Process |
| | PSAV | Event |
| | PSED | Process |
| | PSTR | Process |
| | RJED | Process |
| | RNCF | Process |
| | RTED | Process |
| | SBED | Process |
| | SELP | Event |
| | SELS | Event |

| Field | Description | |
|---|---|---|
| Rec ID (cont'd) | **Record ID** | **Category** |
| | SEND | Event |
| | SERR | Event |
| | SHUD | Event |
| | SIGC | Event |
| | SSTR | Event |
| | STOP | Event |
| | SUBP | Event |
| | TRAC | Event |
| | UNKN | Event |
| | USEC | Process |
| | xxxx | Event |

Other fields displayed depend on the record type.

You can accomplish the following tasks through this display:

✦   If you want to redefine the selection criteria, click the **Criteria** button.

✦   To see the detail fields associated with the Process, double-click on the Process summary line.

✦   To change the size of a field in your display, point the mouse at the boundary edge of the heading of the field you want to change. While holding down the mouse button, move the mouse horizontally until the width of the field is as desired.

✦   To change the width of a row, move the cursor to the line between any of the line numbers. While holding down the mouse button, move the mouse vertically until the width of the row is as desired.

✦   To change the order of the fields in the display, select the heading of the field you want to move, press **SHIFT**, point the mouse at the heading, and drag and drop the column to its new location. You can move any column.

After you close the reformatted display, that format becomes the default format.

# Understanding the Windows Event Logging Facility

Connect:Direct for Windows uses the Windows Event Logging facility to log informational, warning, and error messages that the server generates during execution. A subset of the Connect:Direct statistics records are also sent to the event log. The **event.log** initialization parameter determines which of these records to log.

Merging this critical error information with the event log enables the system administrator to have a single point of focus for error information from all Windows subsystems. The Event Logging

facility also allows for easy correlation of the various error messages that both Connect:Direct and Windows generate.

---

*Caution:*   Connect:Direct can generate numerous event records. For control purposes, define a large event log, use the **event.log** initialization parameter to reduce the number of events Connect:Direct generates, or define the event log to be wraparound.

---

The event log receives the following information from Connect:Direct for Windows:

| Information | Connect:Direct Record IDs | Windows Event Type |
|---|---|---|
| Server initialization/termination | NUIC and NUTC | Informational |
| Communications session start/end | COAC and SEND | Informational |
| Communications error | SERR | Error |
| Connect:Direct Process start/end | PSTR and PRED | Informational |
| Connect:Direct Process step information | CTRC, PSED, RJED, RTED, and SBED | Informational |
| Connect:Direct Process flush/error | PERR and PFLS | Warning |

## Controlling Event Log Content

Use the **event.log** initialization parameter to control the Connect:Direct statistics IDs that Connect:Direct sends to the event log. It specifies the statistics IDs the system sends to the Windows event log. You can specify a list of Connect:Direct Statistics IDs or the keyword All. If you specify more than one statistics ID, use a comma to separate IDs in a list. The default is All.

The following sample uses the **event.log** initialization parameter to log Process start and stop, Process flush, and Process errors to the event log.

```
event.log=PERR,PFLS,PRED,PSTR
```

## Viewing the Event Log

To view the Windows Event Log:

1.  Click **Start** and point to **Settings > Control Panel** > **Administrative Tools** > **Event Viewer**. The **Event Viewer** dialog box is displayed.

2.  Double-click the event in the list you want to view. The **Events Properties** dialog box is displayed.

    The Event Detail fields are display only. They include the date and time, the application that is the source of the event, the type of event, and the description for the event ID generated.

3.  Click the up or down arrow to continue viewing details of the list of events.

4.  Click **OK** to exit the **Event Properties** dialog box.

---

Filtering the Event Log

The Windows Event Viewer utility also enables you to filter the event log. If you want to view a category of events in the log, you can sort the event log by using the Filter Events option. You can filter the event log by specifying settings that control a range of events by date and time. You can also filter the event log by particular event types, application source and category, particular user, computer, and Event IDs. See the Windows documentation for more information on the Event Viewer and the filter functionality.

Use the Connect:Direct Message Lookup Utility to assist you in troubleshooting. Enter a message ID to access the short and long text explanations for Connect:Direct messages.

> **Note:**   You can also view messages with the **select message** command from the CLI. See *Submitting Processes Using the CLI* on page 153 for more information about the CLI commands.

# Viewing Messages

Use the Connect:Direct Message Lookup Utility to view the short and long text explanations for error messages.

To view messages with the **Message Lookup Utility**:

1.  Select **Start > Programs >Sterling Commerce Connect:Direct Windows v4.4.00> CD Message Lookup**. The **Connect:Direct Message Lookup** dialog box is displayed.

2.  Enter the message ID in the **Message ID** field.

3.  Click **Lookup** to display the short message text and long message text.

# Sending Messages to a SNMP Management System

The Simple Network Management Protocol (SNMP) defines a set of protocols that describe management data and the protocols for exchanging that data between systems. Connect:Direct for Windows provides an extension agent that works with Microsoft Windows SNMP Extensible Agent to send traps to a configured Network Manager. See Microsoft documentation for details on configuring the SNMP service to send traps.

Connect:Direct for Windows enables messages to be written to the event log. Traps are triggered by Connect:Direct for Windows messages. Certain messages are associated with trap events. However, you can also define any message to be a trap event. When Connect:Direct for Windows starts, it processes the configuration file to create a table of trap-event messages. When a trap-event message occurs, it is forwarded to the extension agent. The extension agent works with the Microsoft Windows SNMP Extensible Agent to forward the trap to all configured network managers. The network managers who receive the messages are defined when you install Microsoft SNMP.

To activate SNMP trapping, set the SNMP initialization parameter to **Y**. The default is **N**.

When a trap-event message occurs, Connect:Direct for Windows writes the message to the application event log.

To activate event logging, set the initialization parameter called EVENTLOG to **Y**. The default is **N**.

The configuration file is used by both SNMP and event logging. It is specified by the SNMP.DSN initialization parameter and the default file name is:

*installation directory*/snmp/snmp.ini

where *installation directory* is the location where you installed Connect:Direct for Windows.

The configuration file enables you to customize the events that can be trapped. By default, all events are enabled. If you want to disable some events or define other messages to be trap events, edit this file with any standard editing tool.

Do one or more of the following to customize the configuration file:

✦ To turn off an event, set its value to N.

✦ To turn off all status traps, set sendStatusTraps=No.

✦ To turn off all alarm traps, set sendAlarmTraps=No.

✦ To define a list of Connect:Direct messages that will produce an alarm trap, use the following: USERMESSAGEALARM = (message ID, message-list)

✦ To define a list of Connect:Direct messages that will produce a status trap, use the following: USERMESSAGESTATUS = (message ID, message-list)

When the server is started, the SNMP and EVENTLOG initialization parameters are checked. If the SNMP and EVENTLOG parameters are not specified, SNMP and event logging are not activated.

## Identifying the Trap Variables

Traps are defined as alarm or status alerts that enable the network manager to display the trap in the appropriate color on the network manager console. Alarm trap variables signal events that are critical to the operation of Connect:Direct for Windows. Status trap variables signal events that are not critical to the operation of Connect:Direct for Windows but show valuable information. The tables in the following sections describe the predefined traps, the message that triggers the trap, and a description of the trap and associated text.

The categories for trap variables are as follows:

✦ Type events

✦ Initialization events

✦ Shutdown events

✦ API events

✦ Execution events

✦ License Key events

✦ Miscellaneous events

Valid values for all events are **YES** (or **Y**) to enable and **NO** (or **N**) to disable.

This section identifies the Connect:Direct traps supported and identifies the messages associated with the trap. Connect:Direct traps are defined in two groups: status traps and alarm traps.

✦ Status traps are not critical to the operations of Connect:Direct for Windows but display important information.

✦ Alarm traps are more serious, such as license expiration.

Following are lists of Connect:Direct for Windows specific or predefined traps and the message that triggers the event. All traps are enabled by default. If any of the traps need to be disabled, specify them in the SNMP.DSN parameter with the following syntax:

```
Trap-name= Yes / No (YES=Enable and NO=Disable)
```

## Type Events

Use the events in the following table to enable or disable all alarm or status events.

| Trap Event | Description | Event |
|---|---|---|
| sendAlarmTraps | NO—Disables all alarm trap variables regardless of individual settings. YES—Enables all alarm trap variables, allowing you to disable individual alarm trap variables. | Alarm |
| sendStatusTraps | NO—Disables all status trap variables regardless of individual settings. YES—Enables all status trap variables, allowing you to disable individual status trap variables. | Status |

## Initialization Events

The following table details status events that occur at initialization:

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| initializationComplete | This node was initialized successfully. | NUIC00I |
| snaNotAvailable | The SNA support is temporarily unavailable because the APPC/SNA feature could not be initialized. | LSNA003I |
| snaNowAvailable | SNA/APPC has been successfully initialized. | LSNA004I |
| tcpNotAvailable | The TCP support is temporarily unavailable because the TCP environment could not be initiated. | LTCP003I |
| tcpNowAvailable | The connection to TCP/IP has been successful and all TCP functions are supported. | LTCP004I |

## Shutdown Events

The following table details status events that occur at shutdown:

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| shutdownRequested | A Connect:Direct for Windows Stop command has been issued. Type can be QUIESCE, STEP, IMMEDIATE, or FORCE. | LSMG421I |
| normalShutdownComplete | Connect:Direct for Windows termination has been completed successfully. | NUTC001I |

The following table details the alarm event that occurs at shutdown:

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| abnormalShutdown | An abnormal termination of Connect:Direct for Windows has occurred. | LSMG419I |

## API Events

The following table details the status event that occurs from the API:

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| maxUserReached | MAX.API.CONNECTS has been reached. | LCOA009I |

## Execution Events

The following table details status events that occur when a Process executes:

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| processRetryExceeded | The Process has exceeded the retry threshold and has been placed in the Hold queue. | SVTM506I |
| tcqMovement | A Connect:Direct for Windows Process has been moved to the Hold queue due to errors during Process execution. | LPMM001E LSMG007E SVTM105I |
| userMessageStatus | A user defined Connect:Direct message has been issued. | Any Connect:Direct message |

The following table details alarm events that occur when a Process executes:

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| processFailure | A Process has failed with a return code greater than 4 due to abnormal session termination, network map check failure, or FM security failure. | LSMG420I |
| maxPnode | The maximum number of PNODE Processes has been reached. | LPMM055E |
| processNot Started | Process was not started because session manager could not be initialized. | LSMG418I |
| maxSnode | The maximum number of SNODE Processes has been reached. | LSMI013E |
| userMessageAlarm | A user-defined Connect:Direct message has been issued. | Any message ID |

## Miscellaneous Events

The following table details other status events:

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| tracesEnabled | Connect:Direct traces enabled. | STRA029I |
| netmapUpdate | The Connect:Direct network map has been updated. | SMUP192I |
| authFileUpdate | Dynamic update of the Connect:Direct AUTH has occurred. | SAFC006I |
| initparmRefresh | Dynamic update of the Connect:Direct INITPARM has occurred. | SITA993I |
| changeProcess | A CHANGE PROCESS command has occurred. | SOPB017I |
| deleteProcess | A DELETE PROCESS command has occurred. | SOPC011I |

## License Key Events

The following table details license key alarms events:

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| apInitFailure | An initialization error occurred when processing the license file. | APSM012E |
| apKeyValidFailure | The license file is invalid. | APSM008E |
| emergencyKey | The EMERGENCY-KEY is being used. | APSM998I |
| productHasExpired | The Connect:Direct product license has expired. | APSM009E APSM010E |

*Connect:Direct for Windows System Guide*

| Trap Variable | Description | Trap Trigger |
|---|---|---|
| productWillExpire | The Connect:Direct product will expire in x number of days. | APSM011E |
| generalAPFailure | An unexpected failure has occurred while processing the license file. | APSME |

# Using the Activity Log

This Activity Log contains a list of Connect:Direct activities, including every significant activity you have requested from the time the activity log is opened until it is closed. Each activity record logged is maintained with the significant data associated with it.

For example, when you save a file, the Save activity record is stored, with the file name of the saved document. The Activity Log is created at startup. It is minimized and located in the lower left-hand corner. You may create, save, open, close, and print Activity logs. When you open a new activity log, the active Log that is already open is deactivated.

## Opening an Activity Log

To open an activity log, click the **Activity Log** in the bottom pane of the Connect:Direct Requester window. The log is displayed in a pane at the bottom of the **Connect:Direct Requester** dialog box.

## Saving an Activity Log

The activity log information is lost when you close Connect:Direct Requester unless you set the activity log information to write to a file.

To write the activity log to a file automatically:

1. From the Connect:Direct Requester Main Window, select **Tools > Options**. The **Options** dialog box is displayed.

2. To display the activity log in the output window, check **Enable the activity log in the output window.**

3. To write the activity log to a file:

   a. From the Options dialog box, check **Write the activity output to a file**.

   b. Type the name of the file in the box provided.

   c. If you want to create a new activity log every time you start Connect:Direct Requester, select **Create new file at startup**.

   d. If you want to append activity to the existing log file at start up, select **Append to existing activity file**.

4. Click **OK**.

# Managing a Connect:Direct Server

The Connect:Direct Admin Tool enables you modify a Connect:Direct server and databases. You configured the server when you installed Connect:Direct for Windows. If you want to change the server configuration, use the Admin Tool utility. You can also start and stop the server using this utility.

## Starting the Admin Tool Utility

To start the Admin Tool, perform the following steps:

1. Click **Start** and point to **Programs**.

2. Click **Sterling Commerce Connect:Direct v4.4.00 > CD Admin Tool**. The **Admin Tool** main window is displayed.

The main window contains an icon representing the local computer and a traffic light icon representing each Connect:Direct node installed on the computer. A running server is represented by a green light, a stopped server is represented by a red light, and a server in an intermittent state (starting or stopping) is represented by a yellow light.

## Understanding the Toolbar

The Admin Tool toolbar provides icons to perform frequently performed actions. The following table describes the Admin Tool utility icons:

| Select | To |
|--------|-----|
|  | Start a server. The traffic light icon displays a green light when the server is starting or running. |
|  | Stop a server. The traffic light icon displays a red light indicating the server is not running. |

| Select | To |
|---|---|
| ⬜ | Set general properties for a Connect:Direct for Windows server. Stop the server before setting general properties. |
| ❓ | Display the **About Admin Tool** dialog box. |
| ▶❓ | Click on a menu bar item, or toolbar item, to access a description of its function. This function does not work when a dialog box is open. |

# Starting and Stopping Connect:Direct Servers

Use the Admin Tool utility to start and stop Connect:Direct servers.

## Starting a Connect:Direct Server

To start a Connect:Direct server, perform the following steps:

1. Select the server you want to start.

2. Select **Start** from the **Server** menu. The server icon changes to yellow and then changes to green if the server was started. If the server is unable to start, the light changes back to red.

## Stopping a Connect:Direct Server

To stop a Connect:Direct server, perform the following steps:

1. Click the running server you want to stop.

2. Select **Stop** from the **Server** menu or click the red light icon.

   The server icon changes to yellow and then changes to red when the server is stopped.

---

*Caution:*   Stopping the Connect:Direct service is similar to issuing a **stop force** command. Any currently executing Processes are interrupted.

---

# Configuring a Connect:Direct Server

To configure general, TCP/IP, SNA, and database properties for the server you selected, use the **Properties** dialog box. Perform the following steps to access the **Properties** dialog box.

---

*Caution:*   You must stop the Connect:Direct service prior to configuring any properties. Stopping the Connect:Direct service is similar to issuing a **stop force** command. Any currently executing Processes will be interrupted.

---

1.  Click the server you want to configure.

2.  If the server is running, click the stop icon to stop the server.

3.  From the **Server** menu, select **Initialization Properties**. The **Properties** dialog box is displayed.

4.  Select one of the following **Service Startup** methods from the pull-down menu:

    ◆   Automatic to start Connect:Direct every time the system starts.

    ◆   Manual to start Connect:Direct manually.

    ◆   Disabled to prevent Connect:Direct from being started.

5.  If necessary, change the default settings on the General properties page. Refer to the following table for a description of the General fields:

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Max API Connections | Specifies the maximum number of concurrent client API connections permitted for the local node. The value for this parameter is for the maximum number of connections by *different* userids. There is no limit to the number of API connections by a single userid. | A numeric value from 1–255. The default is 10. |
| Max Pnode Sessions | Specifies the maximum concurrent connections for all remote nodes where the local Connect:Direct for Windows server node is the originator of the Connect:Direct for Windows Process. The value entered in the **Max Pnode Sess** field of the Network map remote node object overrides this value. | A numeric value from 1–255 for the server version of Connect:Direct for Windows. The default is 1 for the server version. The workstation version of Connect:Direct for Windows is limited to 1 PNODE session. |
| Max Snode Sessions | Specifies the maximum concurrent connections, where the local Connect:Direct for Windows server node is the partner node cooperating with a remote Connect:Direct node to execute a Process. The value entered in the **Max Snode Sess** field of the network map remote node object overrides this value. | A numeric value from 1–255 for the server version of Connect:Direct for Windows. The default is 1 for the server version. The workstation version of Connect:Direct for Windows is limited to 2 SNODE sessions. |
| TCQ Startup | Specifies how the Connect:Direct for Windows server executable starts with respect to the TCQ. | **Warm** to retain all existing Processes in the TCQ at server startup. If you select a warm start, all Processes in executing state when the server was stopped will restart when the server is restarted. **Cold** to delete all existing Processes in the TCQ at startup. |

| Field Name | Field Definition | Valid Values |
|---|---|---|
| TCQ Max Age | Specifies the maximum number of days a Process with Held due to Error (HE) status remains in the TCQ before it is automatically deleted. | A numeric value from 0–30. The default is 30 days. Connect:Direct does not automatically delete Processes when you specify 0. |
| Stat Max Age | Specifies the maximum age (in days) that statistics records are allowed to reach before the system automatically deletes them. | A numeric value from 0–30. The default is 7 days. Connect:Direct does not automatically delete statistics records when you specify 0. |

6.  Click the **TCP/IP** tab to access the TCP/IP properties page. The TCP/IP properties page is displayed:

7.  If necessary change the default values of TCP/IP fields. Refer to the following table for a description of the TCP/IP fields:

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Enable TCP/IP Support | Enables TCP/IP support. | enabled | disabled |
| API IP Address | Specifies the IP address that the Connect:Direct Requester or user-written API programs use to establish client sessions with the Connect:Direct server. | A numeric value in the format *nnn.nnn.nnn.nnn (IPv4)* or *nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn (IPv6)*, or the host name. For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |
| API Port | Specifies the port number that the Connect:Direct Requester or user-written API programs use to establish client sessions with this Connect:Direct server. | A numeric value in the format *nnnn*, where *nnnn* is a positive integer from 0 to 65535. The default is 1363. |

| Field Name | Field Definition | Valid Values |
|---|---|---|
| API Additional Addresses and Ports | Specifies additional addresses and ports that the Connect:Direct Requester or user-written API programs use to establish client sessions with this Connect:Direct server. Multiple address/host names (and combinations with port numbers) can be specified in this field. The port is separated from its corresponding address/host name with a semi-colon (;), and each address/host name and port combination is separated by a comma (,). A space may be added after the comma for readability. | A numeric value in the format *nnn.nnn.nnn.nnn;nnnn (IPv4)* or *nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn;nnnn (IPv6)*, or the *host name;nnnn*. For example, 10.20.9.175;2363, fd00:0:0:2014::7; 2364 This is an optional field. For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |
| Host IP Address | Specifies the IP address for incoming communications from remote Connect:Direct nodes to this Connect:Direct server node. | A numeric value in the format *nnn.nnn.nnn.nnn (IPv4)* or *nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn (IPv6)*, or the *host name*. For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |
| Host Port | Specifies the port number for incoming communications from remote Connect:Direct nodes to this Connect:Direct server node. | A numeric value in the format *nnnn*, where *nnnn* is a positive integer from 0 to 65535. The default is 1364. |
| Host Additional Addresses and Ports | Specifies additional IP addresses and ports for incoming communications from remote Connect:Direct nodes to this Connect:Direct server node. Multiple address/host names (and combinations with port numbers) can be specified in this field. The port is separated from its corresponding address/host name with a semi-colon (;), and each address/host name and port combination is separated by a comma (,). A space may be added after the comma for readability. | A numeric value in the format *nnn.nnn.nnn.nnn;nnnn (IPv4)* or *nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn;nnnn (IPv6)*, or the *host name;nnnn*. For example, 10.20.9.175:2364, mdallas;1364 This is an optional field. For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Buffer Size | Specifies the data buffer size for transmitting data to and from a remote Connect:Direct node if the value is not in the network map entry. The value entered in the Buffer Size field of the network map TCP/IP Communication Mode object overrides this value. | A numeric value from 256–65536.<br>The default is 65535. |
| Enable Dynamic Addressing (DHCP) | Enables dynamic addressing, so that the DHCP server will assign an IP address when it is requested for the Connect:Direct server node. | Disabled | Enabled |

8. Click the **SNA** tab to open the SNA properties page.

9. If necessary, change SNA field values. Refer to the following table for a description of the SNA fields:

| Field Name | Field Definition | Valid Values |
|---|---|---|
| Enable SNA Support | Enables SNA support. | Enabled or Disabled<br>If SNA is activated at installation, the default value is enabled. |
| Local LU Name | Specifies the Local LU name for the Connect:Direct server node. | Local LU Name<br>If you want to monitor multiple LU's, leave this field blank. The default value is the local LU name. |

10. Click the **Database** tab to open the Database properties page.

To change the current database configuration:

11. Click **Modify Settings** to invoke the Connect:Direct Database Wizard utility. Use the Database Wizard utility to configure the ODBC and database environments used by Connect:Direct for Windows.

12. Select one of the following database types:

   ◆ MySQL

   ◆ Microsoft SQL Server

| If You Selected | | Perform the following steps |
|---|---|---|
| MySQL | 1 | Identify the TCQ/Stats data source in the **TCQ/Stats Datasource** field. |
| | 2 | Identify the data source for message in the **Message Datasource** field. |
| | 3 | Identify the name of the database in the **Database Name** field. |
| | 4 | Identify the name of the host in the **Host Name** field. |
| | 5 | Identify the port number in the **Port** field. |
| | 6 | Type the system administrator login information including user ID and password in the fields provided. |
| | 7 | Click **Next**. |
| Microsoft SQL Server | 1 | Identify the TCQ/Stats data source in the **TCQ/Stats Datasource** field. |
| | 2 | Identify the data source for message in the **Message Datasource** field. |
| | 3 | Identify the database name in the **Database Name** field. |
| | 4 | Specify the SQL server by either: |
| | | ◆ Typing the name of the SQL server |
| | | ◆ Selecting the name of the SQL server from the pull-down list. Any local instance names found on the system are listed. |
| | | ◆ Clicking (...) to select a SQL Server from the network. Any SQL Servers running in the current in the Windows domain are listed, which may include local instance names. |
| | 5 | The **ODBC Driver** field defaults to **SQL Native Client** if it is installed. If this driver does not exist, the only option is the **SQL Server** ODBC driver. |
| | 6 | To enable **SQL Server Authentication**, select it and type the system administrator login information including user ID and password in the fields provided. |
| | 7 | Click **Next**. |

8. When you validate the information entered, click **Finish**.

9. Click **Yes** to build the database.

10. Click **OK**.

11. If you want to update the license management key:

   a. Click the **License Management** tab

   b. Select the new license management key.

12. Click **OK**.

# Working with Active Directory

When you install Connect:Direct on a computer, you can add the node to Active Directory during the installation. However, you can use the Admin Tool to add or delete Active Directory entries

after the initial installation. You can also view or print a report that lists all nodes that are registered in Active Directory. In order to add or delete a node from Active Directory, you must be a member of the Enterprise Admins group.

> **Note:**   You can only add or delete Active Directory entries if Active Directory services have been implemented in your environment.

## Adding an Active Directory Entry

To add a node entry to Active Directory, perform the following procedure:

1.   Start the Admin Tool utility.

2.   Select **Create** from the **Active Directory** menu. The **Create Active Directory Entry** dialog box displays a list of all Connect:Direct nodes that are in Active Directory.

3.   Select the node to add and click **Add**.

## Deleting an Active Directory Entry

To delete a node entry from Active Directory, perform the following procedure:

1.   Start the Admin Tool utility.

2.   Select **Delete** from the **Active Directory** menu. The **Delete Active Directory Entry** dialog box displays a list of all Connect:Direct nodes that are in Active Directory.

3.   Select the node to delete and click **Remove**.

## Creating an Active Directory Report

To create a report that identifies all nodes registered in Active Directory, perform the following procedure:

1.   Start the Admin Tool utility.

2.   Select **Report** from the **Active Directory** menu. The Active Directory Report is displayed.

# Troubleshooting

Use the following information when troubleshooting problems with Connect:Direct for Windows and its components.

## Diagnosing Server Problems Using Traces

If you have a problem with a server, you can start a trace. A trace provides detailed information about Connect:Direct activity and assists in diagnosing problems related to Connect:Direct operations. The trace facility writes this information to a file. You can set the size of the output file and specify a wrap if the file reaches its maximum size.

You can set a trace to one of three levels: basic, intermediate, and full. A basic trace captures the function entry and exit only. An intermediate trace captures the function argument and its start and stop. A full trace captures internal control blocks and the function argument, start, and stop.

You can create a trace using Connect:Direct Requester or the CLI. The procedures in this chapter describe how to define a trace using Connect:Direct Requester. Refer to *Using Traceoff* on page 176 and *Using Traceon* on page 177 for instructions on using the CLI to define or turn off traces.

### Defining a Trace

To define a Connect:Direct trace:

1. Start Connect:Direct Requester.
2. From the **Admin** menu, select **Tracing**. The **Trace Properties** dialog box is displayed.
3. To set trace information for selected functions, turn on tracing for the specific Connect:Direct events.

    ◆ **CMGR**—Capture the interaction between clients and the server.

    ◆ **PMGR**—Capture server Process changes.

    ◆ **MAIN**—Capture server initialization and termination.

    ◆ **STAT**—Capture statistics activity.

◆ **SMGR**—Capture the execution of Processes and the interaction of the server with other nodes.

◆ **COMM**—Capture interactions with external communications facilities invoked from Session Manager.

4. To set tracing for all available events, click **Full Tracing**.

5. To turn off all tracing, click **All Off**.

6. **Generate a configuration report** is selected by default. If you do not want to generate a configuration report, turn off this option.

7. If you select **Full Tracing** or set **SMGR** or **COMM** category to **Full**, you can define advanced trace settings. To define advanced settings:

   a. Click the **Advanced** tab.

   b. Identify the node to perform an advanced trace on: PNODE, SNODE or both.

   c. If you want to perform a trace based on a Process name, a Process number, or a Process destination, click the **New** icon in the appropriate field and type up to four names, numbers, or destinations.

8. To specify output file information:

   a. Click the **File** tab.

   b. Enter the output file name. Do not select the name of an active trace file. A file can only be used for one trace at a time. The default file name is CDTRACE.CDT.

   c. Enter the maximum file size allowed in the **Max Size** field. Select **Kilobytes** or **Megabytes** to specify the size unit.

   d. **Wrap tracing output when max file size is reached** is selected by default. If you do not want the tracing output to wrap when the maximum file size is reached, turn off this option.

9. Click **OK**.

## Stopping a Trace

To stop a trace:

1. Select **Admin > Tracing**.

2. Set the trace that you want to stop to **OFF**.

3. Click **All Off** to turn off all tracing.

4. Click **OK**.

## Optional Startup Parameters for Traces

The following paragraphs explain the syntax of valid startup parameters for Connect:Direct. All of these parameters are optional.

| Parameter | Description | | |
|---|---|---|---|
| **-ttyp=** | Specifies the type of trace to start during initialization. You must provide this parameter in order to generate any trace output. Any combination of the following options are valid. | | |
| | | **Options** | **Description** |
| | | **c** | Starts a Command Manager trace. This trace affects all Command Managers started on behalf of users logging in to the Connect:Direct for Windows server. It shows both the command processing flow and the interactions between the server executable and the Connect:Direct for Windows. |
| | | **p** | Starts a Process Manager trace. This trace shows all events related to the Process Manager and to Session Manager startup by the main server executable. |
| | | **s** | Starts a Session Manager trace. This trace shows all Session Manager processing, including file and communications API access, but does not display information that is sent across the communications line. If you do not specify any further qualifying parameters, the system traces all Session Managers. |
| | | **o** | Starts a Communications trace. This trace shows all the communications data that flows across the network between the local and remote Connect:Direct Session Managers. |
| | | **m** | Starts a trace of the main-line logic that includes initialization and termination |
| | | **a** | Starts a trace of the Statistics subsystem. This trace shows all records sent to the Statistics thread. |
| | | **x** | Starts all trace types. Specifying a value of x is the same as specifying **cpsoma**. |

| Parameter | Description |
|---|---|
| **-tlvl=** | Specifies the trace level. The values are: |

| | | Options | Description |
|---|---|---|---|
| | | **b** | Starts a basic trace that includes only module entry and exit records. This is the default. |
| | | **i** | Starts an intermediate trace that shows all trace data produced by **b** plus function arguments |
| | | **f** | Starts a full trace that shows all trace data produced by **i** plus certain Connect:Direct control blocks and internal information. |

| Parameter | Description | |
|---|---|---|
| **-twrp** | Specifies that the trace file should wrap once it reaches its maximum size. | |
| **-tfil=*filespec*** | Specifies the fully qualified name of the trace file that receives output. The default is CDTRACE.CDT in the directory where the Connect:Direct for Windows server executable resides. Windows Services treats a backslash (\) as an escape character, so type two backslashes for each backslash in the file path. | **Example**<br>  **-tfil=**c:\\*users*\\*default*\\*trace.it* |
| **-tfsz=** | Specifies the maximum file size the trace file can reach before it wraps if -twrp is also specified. If -twrp is not specified, -tfsz indicates that the trace file will terminate when it reaches this size.<br>The default is -tfsz=100**K**. | |

| | | Options | Description |
|---|---|---|---|
| | | **bytes** | You can specify the file size in bytes. |
| | | *bytes***K** | You can specify the file size by using the **K** (x1000) suffix. |
| | | *bytes***M** | You can specify the file size by using the **M** (x1000000) suffix. |

For Session Manager (**s**) or Communications (**o**) trace types, the following parameters are also available:

| Parameter | Description |
|---|---|
| **-tpnum=** | Requests a trace of up to eight Process numbers upon the initiation of the Process. If you specify -tpnum, Connect:Direct will only trace the specified Processes. |

| Parameter | Description | | |
|---|---|---|---|
| | | **Options** | **Description** |
| | | *pnum* | Process number. |
| | | *(pnum1,pnum 2,...,pnum8)* | Trace of up to eight Process numbers. |
| **-tpnam=** | Requests a trace of up to eight Process names upon the initiation of the Process. If you specify -tpnam, Connect:Direct will only trace the specified Processes. | | |
| | | **Options** | **Description** |
| | | *pnam* | Process name. |
| | | *(pnam1,pnam 2,...,pnam8)* | Trace of up to eight Process names. |
| **-tdest=** | Requests a trace of up to eight Connect:Direct remote node names that have interaction with this local node. This parameter generates trace information when Connect:Direct submits a Process that is destined for the specified remote node or when the specified remote node establishes a session with this local node. If you specify -tdest, Connect:Direct will only trace the specified remote node names. | | |
| | | **Options** | **Description** |
| | | *dest* | Destination node name. |
| | | *(dest1,dest2,. ..,dest8)* | Trace of up to eight destination node names. |
| **-tlnode=**<br>*or*<br>**-tpnode=** | Requests a trace of initiating node sessions only. This parameter modifies the effect of the *-tpnam*, *-tpnum*, and *-tdest* parameters by only allowing traces of Processes submitted on the local node.<br>**Note:** Connect:Direct for Windows also supports **-tpnode** in place of **-tlnode**. | | |
| **-trnode=**<br>*or*<br>**-tsnode=** | Requests a trace of remote node sessions only, that is, a session initiated by a remote node. This parameter modifies the effect of the *-tpnam*, *-tpnum*, and *-tdest* parameters by only allowing traces of Processes submitted on remote Connect:Direct nodes.<br>**Note:** Connect:Direct for Windows also supports **-tsnode** in place of **-trnode**. | | |

# Recovering from Problems

Connect:Direct provides several ways to recover from a system malfunction, such as an abnormal termination of a connection between local and remote Connect:Direct nodes. Process recovery enables you to resume execution as quickly as possible and to minimize redundant data transmission after a system failure.

Connect:Direct uses the following facilities to address errors for Process recovery:

✦  Process step restart

✦  Automatic session retry

✦  Checkpoint/restart

✦  **Run Task** restart

## Process Step Restart

As a Process executes, Connect:Direct records the step that is executing in the TCQ. If Process execution is interrupted, the Process is held in the TCQ in retry (RE) status, unless you explicitly delete the Process with the Hold parameter set to No. After all attempts to restart have failed, the Process goes into Held due to Error (HE) status. When a wildcard copy command restarts, it restarts at the file that was being copied when the Process was interrupted.

When you release the Process for execution, Connect:Direct automatically begins execution at the beginning of that statement. No user specification is required for Process step restart. Connect:Direct always performs Process step restart.

## Automatic Session Retry

The network map remote node object has short-term and long-term connection retry parameters. If no value is specified for these parameters, the default values are taken from the initialization parameters file.

✦  Short-term parameters allow a quick succession of retry attempts in the event of a short-term connection failure.

✦  Long-term parameters are used after the number of short-term attempts you specify has been reached. The assumption is that the connection problem cannot be fixed quickly; therefore, Connect:Direct can retry infrequently to save the overhead of connection retry attempts.

If a session error occurs, the Process moves to the Timer queue in retry (RE) status and short-term and long-term wait times begin. After short-term and long-term wait times expire, the Process is moved to the Hold queue.

The **tcq.max.age** initialization parameter specifies the maximum number of days a Process with a Held due to Error (HE) status remains in the TCQ before it is automatically deleted.

## Checkpoint/Restart

Checkpoint/restart is specific to the Process **copy** statement. Connect:Direct records file positioning checkpoint information at specified intervals during a copy operation. If a copy step is interrupted for any reason, it can be restarted at the last checkpoint position.

> **Note:** Connect:Direct must support checkpoint/restart on both the local and the remote node.

The file sender provides positioning information to the receiver when a checkpoint interval is reached. The receiver stores this information, along with its destination file positioning information, in the checkpoint file. The last four sets of positioning information are retained in the checkpoint file. At restart, each set of information is used until the file is successfully repositioned. If repositioning fails, then the entire file is retransmitted.

## Checkpointing Parameter for the copy Statement

The checkpoint parameter in the Process **copy** statement specifies the byte interval at which checkpoints are taken. See the appropriate **copy** chapter of the *Connect:Direct Process Guide* for use of this parameter.

A checkpoint value of no disables checkpointing. If you do not specify a checkpoint value in the **copy** statement, the default is defined by the checkpoint interval initialization parameter, **ckpt.interval.**

## Restart during run task Operations

Connect:Direct for Windows also offers checkpoint/restart capabilities with the **run task** Process statement.

If a **run task** operation is executing on the SNODE and a session failure occurs, the PNODE recognizes the session failure and puts the Process in the Timer queue for retry. The SNODE, however, is still running the **run task** operation and is not notified of the session failure until the operation finishes. The checkpoint/restart feature for **run task** ensures that when the Process restarts on the PNODE, the **run task** operation does not execute a second time on the SNODE.

## Restarting Process Operations

If a **run task** Process step restarts, the node where the operation executes attempts to find the checkpoint information in the TCQ header. If the **run task** step is still executing, the Process that is running for the restart of the step waits for the **run task** operation to finish the first task and proceed to the next step of the Process, if there is one.

When the first Process finishes, it determines that the session under which it was running has been lost and terminates without logging statistics records that indicate a session failure. The second Process records how the **run task** step that was still executing ended and proceeds to the next step in the Process.

Determining Reexecution of the run task Step

If Connect:Direct determines at restart that the **run task** operation ended because it finished before the PNODE restarted the Process, then the **run task** step does not execute again. However, if the **run task** operation did not finish and is not currently running, then the value of the **runtask.restart** initialization parameter determines whether to restart the Process.

If **runtask.restart=yes**, Connect:Direct executes the program again. If **runtask.restart**=**no**, the Process skips the **run task** step.

For a full description of all **run task** statement parameters, go to the Connect:Direct Processes Web site at http://www.sterlingcommerce.com/documentation/processes/processhome.html.

# Working with Sterling Commerce Support to Troubleshoot Connect:Direct Problems

Before calling Sterling Commerce Support, gather information to help support personnel analyze and troubleshoot your problem. Have the following information available when you call.

✦ Network map parameter definitions

✦ Initialization parameter definitions

✦ Statistics report

✦ For SNA sites, gather the following:

◆ If you are using Microsoft SNA software, type **snacfg/print>snacfg.dat** from a command prompt.

◆ If you are using IBM Communications Server SNA software, open the *filename*.acg, where *filename* is the name specified during Communications Server installation.

If you are troubleshooting a Connect:Direct OS/390 server, gather the following information:

✦ Network map parameter definitions

✦ Local Node source

✦ Remote Node source

✦ For SNA connectivity, provide the following:

◆ z/OS LU definition

◆ z/OS PU definition

◆ z/OS APPLID

◆ z/OS LOGMODE definition entry

# Worksheets

Use the worksheets in this appendix to gather and record information needed for Connect:Direct for Windows configuration.

This appendix contains worksheets used during:

✦ Network Map Configuration

  ◆ Communication mode object

  ◆ Communication path object

  ◆ Remote Node object

✦ User authorizations setup

  ◆ User functional authorization

  ◆ Remote user proxy

# Network Map Communications Mode Object Worksheet

The communications mode object defines the protocol and characteristics of sessions that are established using this mode group. Use the information on this worksheet when you create or modify a Communication Mode in your network map. Refer to Chapter 3, *Defining and Managing the Connect:Direct Network*, for field content.

Make a copy of this worksheet for each communications mode in the network.

**TCP/IP Communications Mode**

Communications Mode Name: _____

Buffer Size: _____

Pacing Send Count: _____

Pacing Send Delay: _____

CRC_____

**APPC Communications Mode**

Communications Mode Name: _____

Max RU Size: _____

Max Pacing Size: _____

Max Sessions: _____

# Network Map Communications Path Object Worksheet

The communication path object defines the communications path between the local node and one or more remote nodes. Use this worksheet when you create a communications path in your network map. Refer to Chapter 3, *Defining and Managing the Connect:Direct Network*, for field content. Make a copy of this worksheet for each communications path in the network.

---

**TCP/IP Communications Path**

---

Communication Path Name: _____

Communication Mode: _____

---

**APPC Communications Path**

---

Communications Path Name: _____

Communications Mode: _____

Transport Type:

___ Token Ring

      Adapter Number:      ___ Primary             ___ Alternate

    Address: _____

___ Ethernet

      Adapter Number:      ___ Primary             ___ Alternate

Address: _____

___ SDLC

Local SNA Network: _____

LU Name: _____

---

# Network Map Remote Node Object Worksheet

The Remote Node object provides information about remote nodes to which the local node establishes sessions or that establish sessions with the local node. Use the information on this worksheet when you modify your network map. Refer to Chapter 3, *Defining and Managing the Connect:Direct Network*, for field content.

Make a copy of this worksheet for each remote node.

---

**Main Options**

---

Remote Node Name:  _____

Operating System:  _____

---

**Maximum Sessions**

---

Maximum Concurrent Local Node Session:  _____

Maximum Concurrent Remote Node Session:  _____

Default Class:  _____

---

**Long/Short Retry**

---

Short Term Number of Retries: _____          Interval (Time Between Retries): _____

Long Term Number of Retries: _____           Interval (Time Between Retries): _____

---

**TCP/IP Information**

---

Host Name/IP Address:  _____

Port Number/Service Name: _____

TCP/IP Communication Mode Name:  _____

---

**APPC Information**

---

Remote Network Name: _____

Partner LU Name: _____

APPC Mode: _____

---

## Communications Paths

You must designate at least one path.

Communications Path Names: _____

_____

_____

_____

## Description Information

Contact Name:_____

Contact Phone Number: _____

Comment: _____

# User Functional Authorization Worksheet

User Functional Authorizations control the user's right to issue various Connect:Direct commands and statements or perform tasks through the Connect:Direct Requester. Use this worksheet when making updates to the functional user authorization object in the Registry. Refer to Chapter 3, *Defining and Managing the Connect:Direct Network* for field content.

Make a copy of this worksheet for each Connect:Direct user.

---

**Main Options**

---

Local User ID: _____

Default User Type (choose one): _____

        ___ General User           ___ Administrator

---

**Administrative Options**

---

Indicate whether the administrator has the authority to update the following:

        ___ User Authorizations        ___ Upload

        ___ Remote User Proxies        ___ Download

---

**Directory Restrictions**

---

Specify the directory where you are allowed to perform the specific tasks. If you do not specify a directory, you can perform the specific task from any directory to which the Windows security enables access. This authority is effective regardless of whether the request is submitted from the local or remote system. However, the remote user proxy can override the directory specification.

Upload Directory: _____

Download Directory: _____

Process Directory: _____

Program Directory: _____

**Server Control**

Grants authority to perform the following tasks in Connect:Direct.

___ Stop (Connect:Direct)   ___ Refresh (initialization parameters)   ___ Trace

___ Update Network Map   ___ Update Translation Table

**Statements Authorization**

Grants authority to use the following statements in Connect:Direct Processes.

___ Trace        ___ Copy        ___ Run Job        ___ Run Task        ___ Submit

**Process Control Submit**

Grants authority to manipulate and observe Processes in the TCQ.

|  | NO | YES | ALL |
|---|---|---|---|
| Monitor | ___ | ___ | ___ |
| Change | ___ | ___ | ___ |
| Delete | ___ | ___ | ___ |
| Statistics | ___ | ___ | ___ |

**Overrides Authorization**

Grants authority to override the following Windows defaults.

___ Execution Priority

___ ACL Updates

___ File Attributes

___ Remote Node ID

# Remote User Proxy Worksheet

The Remote User Proxy object defines a relationship between a remote node and a local user ID. Refer to Chapter 3, *Defining and Managing the Connect:Direct Network*, for field content.

Make a copy of this worksheet for each Connect:Direct Remote User Proxy you want to define.

---

**Main Options**

---

Remote Node Name:  _____

Remote User ID: _____

Local User ID: _____

Local User ID Password: _____

       Allow the remote user to:         ___ Upload         ___ Download

---

**Directory Options**

---

Specify the directory where the remote user can perform tasks. The directory properties allow you to restrict the Process directory and data directories that remote users can access. If you do not specify a directory, the directories specified in the functional authorization for the local user account will apply. If there are no directories specified in the local user functional authorizations, the remote user can perform the specific task from *any* directory that the server security enables access.

Upload Directory:  _____

Download Directory:  _____

Process Directory:  _____

Program Directory:  _____

---

# Changing Connect:Direct for Windows Settings

When you install Connect:Direct for Windows, initialization parameters are created in the Windows registry and are used to determine settings at initialization. Initialization parameters (initparms) set the default values of Connect:Direct functions. Initparms determine how Connect:Direct behaves during operation. You can change the default Connect:Direct for Windows settings by changing the value of these parameters.

Initialization parameters are organized in the following groups:

| Category | Description |
|---|---|
| Miscellaneous | Miscellaneous commands describe server path, download and upload directories, dialup entries, and security exits. |
| Local Node Characteristics | Node settings define the name of the local node, and determine default values for functions such as session class, maximum connections, maximum API connections, buffer sizes, and short- and long-term attempts and retries. |
| TCQ Information | TCQ settings determine default values for the Process file directory, remote node run task operations, the length of time a Process is held in error, and how the TCQ handles Processes during server startup. |
| Global Copy Parameters | Global copy settings determine default checkpoint intervals, translation tables and translation directories. |
| Statistics Information | Statistics settings determine the maximum age that statistics records are kept and what commands are logged in the statistics file. |

# Sample Initialization Parameters Format

The following figure illustrates the format of the initialization parameters. The initialization parameters are listed in groups that are labeled in brackets.

```
[Miscellaneous Commands]
server.path=C:\Program Files\Sterling Commerce\Connect Direct v4.4.00\Server\
proc.prio.default=10
exec.prio.default=7
license.management.key=LicEmerg.txt
download.dir=C:\Program Files\Sterling Commerce\Connect Direct v4.4.00\Server\DOWNLOAD\
upload.dir=C:\Program Files\Sterling Commerce\Connect Direct v4.4.00\Server\UPLOAD\
program.dir=C:\Program Files\Sterling Commerce\Connect Direct v4.4.00\Server\PROGRAM\
restrict.cmd=N
security.exit=<None>
notify.level=A
file.exit=<None>
event.log=All
snmp=N
snmp.eventlog=N
snmp.trapstat=N
snmp.dsn=C:\Program Files\Sterling Commerce\Connect Direct v4.4.00\Server\snmp\snmp.ini

[Local Node Characteristics]
max.api.connects=10
conn.retry.stwait=00:00:10
conn.retry.stattempts=10
conn.retry.ltwait=00:03:00
conn.retry.ltattempts=10
contact.name=not specified
contact.phone=not specified
descrip=no description specified
name=CDSWIFTNET-4400
sess.pnode.max=1
sess.snode.max=1
sess.default=1
netmap.check=Y
node.check=B
proxy.attempt=N
protocol=1
tcp.api.port=cdswiftnet;4463
tcp.host.port=cdswiftnet;4464
outgoing.address=<None>
tcp.src.ports=<None>
tcp.src.ports.list.iterations=1
comm.bufsize=65535
pacing.send.delay=00:00:00
pacing.send.count=0
tcp.crc=OFF
tcp.crc.override=N
tcp.max.time.to.wait=00:03:00
tcp.window.size=0
runstep.max.time.to.wait=00:00:00
active.directory.enabled=N
quiesce.resume=N

[TCQ Information]
tcq.max.age=30
tcq.start=W
process.dir=C:\PROGRAM FILES\STERLING COMMERCE\CONNECT DIRECT V4.4.00\SERVER\PROCESS\
runtask.restart=N

[Global Copy Parameters]
ckpt.interval=10240K
xlate.dir=C:\PROGRAM FILES\STERLING COMMERCE\CONNECT DIRECT V4.4.00\SERVER\XLATE\
xlate.send=XLATESND.CDX
xlate.recv=XLATERCV.CDX
disable.cache=N
continue.on.exception=N
ecz.cmprlevel=1
ecz.windowsize=13
ecz.memlevel=4
strip.blanks=I
record.wrap=N
retry.msgids=<None>
retry.codes=<None>

[Statistics Information]
stat.max.age=7
log.select=N
log.commands=Y
stat.sort=Y
```

# Changing Initialization Parameters

You can change the Connect:Direct for Windows initialization settings by editing the initialization parameters.

To edit initialization parameters from Connect:Direct Requester.

1. Click **Start**, then point to **Programs** > **Sterling Commerce Connect:Direct v4.4.00 > CD Requester**.

2. From the **Admin** menu, select **Initialization Parameters**. If you have not attached to Connect:Direct, the **Connect:Direct Attach** dialog box is displayed.

3. Attach to the server.

4. In the initialization parameters file, place the cursor after the = sign following the parameter you want to change.

5. Type the new value. Refer to the parameters tables for the name, definition, and valid values for each command parameter.

6. Select **Apply** from the **Initparms** menu to update and save changes.

7. Close the **Initparms** dialog box by clicking the **X** in the right-hand corner of the dialog box.

8. Click **Yes** to apply the changes and return to the main window.

## About Miscellaneous Parameters

The miscellaneous parameters determine the server path, default Process priority, event log values, and various restricted directories. The following table lists the miscellaneous commands parameters:

| Parameter Name | Parameter Definition | Valid Values |
| --- | --- | --- |
| server.path | Specifies the fully-qualified path name to all Connect:Direct files. Terminate the path name with a trailing backslash (\).<br><br>If you do not provide this parameter, all subdirectories necessary for Connect:Direct operation must be located in the directory where Connect:Direct for Windows was installed. | Valid, fully-qualified path name. |
| proc.prio.default | Specifies the default Connect:Direct Process priority information to assign any time a Process is submitted without the selection priority parameter (selprty) on the **Process** statement. | A numeric value from 1–15, where 15 is the highest priority.<br><br>The default is 10. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| exec.prio.default | Specifies the priority of the Process. The execution priority parameter is used to influence the Windows operating system priority given to the Session Manager when it begins execution of this Process. A Process with a higher priority is assigned a higher operating system priority and receives more system resources.<br><br>Note:    Scheduling Processes with a high execution priority value can affect the execution of other applications on the system. | A numeric value from 1–15, where 15 is the highest priority.<br>The default is 7.<br>These values are mapped to Windows Process priority classes and values. |
| license.management.key | Identifies the name of the license file. If the value is set to NA or is empty, the default of license.key is used. | Name of key file \| NA \| empty field |
| download.dir | Specifies the default directory to copy the destination file to if a **copy** statement does not specify a fully-qualified path. | Valid, fully-qualified path name. The default is *X:\installation directory*\DOWNLOAD |
| upload.dir | Specifies the default directory that source files are copied from if a **copy** statement does not specify a fully-qualified path. | Valid, fully-qualified path name. The default is *X:\installation directory*\UPLOAD |
| program.dir | Specifies the default working directory for a program that is started using a **run task** or **run job** statement when a fully-qualified path is not specified. | Valid, fully-qualified path name. The default is *X:\installation directory*\PROGRAM |
| restrict.cmd | Restricts the use of operating system commands in run task or run job statements by preventing the use of the "CMD" syntax in those statements. In addition, prevents the use of the special characters: "&", "\|", and ">".<br><br>To enable running of **cmd** tasks and the use of special characters, specify **N**. | Y \| <u>N</u> |
| security.exit | Specifies that a security exit is to be used. The security exit parameter specifies whether passticket support is implemented as a user exit during Process execution. | Valid, fully-qualified path name to a user-defined DLL. The default is <NONE>. |
| notify.level | Specifies the level of Process notification based on the Process step return code. If you want notification to occur regardless of the return code, specify a value of A. For notification for a warning level return code of 4 or higher, specify a value of W. For notification for an error level return code of 8 or higher, specify a value of E. | <u>A</u> \| W \| E |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| file.exit | Specifies the name of a user-written Dynamic Link Library (DLL) file. The DLL file opens a source or destination file during processing of the COPY statement and overrides the values specified in the COPY statement. If the DLL file is not in the search path, a fully-qualified path name must be specified. | Valid, fully-qualified path name to a user-defined DLL. The default is <NONE>. |
| event.log | Specifies the statistics IDs the system sends to the Windows event log. Either specify a list of statistics IDs or select **All** to log all IDs to the event log. Use a comma to separate IDs in a list. Refer to *Viewing Process Statistics* on page 93 for a list of statistics IDs. | *statistics ID* | <u>All</u> | (list) |
| snmp | Causes the SNMP trap agent environment to be initialized. | <u>N</u> | Y |
| snmp.eventlog | Initializes the SNMP event logging environment. | <u>N</u> |Y |
| snmp.dsn | Defines the configuration file used to initialize the default traps variables. | configuration file<br><br>The default is installation directory\snmp\snmp.ini. |
| snmp.trapstat | Defines the configuration file used to determine if the SNMP traps are written to the statistics file. | <u>N</u> | Y |

## About the Local Node Parameters

The local node characteristics parameters define the name of the local node and default information used to communicate with a remote node. The following table identifies the local node characteristics parameters:

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| max.api.connects | Specifies the maximum number of concurrent API client connections permitted for the local node by *different* user IDS. There is no limit to the number of API connections by a single user ID. | A numeric value from 1–255. The default is 10. |
| conn.retry.stwait | Specifies the time to wait between retries immediately after a connection failure occurs. The value entered in the Short Term Retry Interval field of the network map remote node object overrides this value. | A 24-hour time value formatted as *hh:mm:ss*. The maximum value is 23:59:59. The default is **00:00:10**, or 10 seconds. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| conn.retry.stattempts | Specifies the number of times to attempt connection after a connection failure occurs. The value entered in the Short Term Retry Count field of the network map remote node object overrides this value. | A numeric value from 0–9999. The default is **10**. |
| conn.retry.ltwait | Specifies the time to wait between long-term retry cycles. This parameter is a long-term connection retry parameter. The value entered in the Long Term Retry Interval field of the network map remote node object overrides this value. | A 24-hour time value formatted as *hh:mm:ss*. The maximum value is 23:59:59. The default is **00:03:00**, or 3 minutes. |
| conn.retry.ltattempts | Specifies the number of times to attempt connection after a connection failure occurs. This parameter is a long-term connection retry parameter. The value entered in the **Long Term Retry Count** field of the network map remote node object overrides this value. | A numeric value from 0–9999. The default is **10**. |
| contact.name | Specifies the name of a contact. | Any name description. |
| contact.phone | Specifies a phone number to use to contact the contact name | Any valid phone number. |
| descrip | Specifies the description of the local node. | Any valid text string. |
| name | Specifies the name of the local node used when identifying the local server to remote nodes and the server object name for which API programs search when locating active Connect:Direct servers. You cannot change the value of this parameter. If you want to change the local node name, you must reinstall Connect:Direct for Windows. | A 1–16 alphanumeric character string specified during installation. |
| sess.pnode.max | Specifies the maximum concurrent connections for all remote nodes to the local node when the local node is the originator of the Process. The maximum number of concurrent sessions is limited to the lesser of the values defined in the Initialization parameter and the Network map definition for a node. | A numeric value from 1–255. The default value is 1 or half of the maximum sessions specified during installation. The workstation version of the product is limited to a maximum of 1 PNODE session. |
| sess.snode.max | Specifies the maximum concurrent connections for all remote nodes where the local node is the partner node of a Process. The maximum number of concurrent sessions is limited to the lesser of the values defined in the Initialization parameter and the Network map definition for a node. | A numeric value from 1–255. The default value is 1 or half of the maximum sessions specified during installation. The workstation version of the product is limited to a maximum of 2 SNODE sessions. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| sess.default | Specifies the default session class for starting session managers. A Process executes on the specified class or any higher session class. If the value specified exceeds **sess.pnode.max**, a warning is issued, and the value is reset to the default value. The value entered in the **Default Class** field of the remote node object overrides this value. | A numeric value from 1 to the value specified for sess.pnode.max.<br><br>The default is 1. |
| netmap.check | Specifies the level of network map checking that occurs for each node that you communicate with.<br>• **Y**—Checks the network map for *all* nodes that Connect:Direct will communicate with to validate the node name and the IP address.<br>• **L**—Checks the network map only for nodes that the local Connect:Direct will initiate sessions with.<br>• **R**—Checks the network map only for remote nodes that will communicate with this node.<br>• **N**—Will not validate any session establishment requests in the network map. | <u>Y</u> \| L \| R \| N |
| node.check | Specifies the level of node checking.<br>• **B**—Checks both the node address and the node name.<br>• **C**—Checks the node name only. | <u>B</u>\| C |
| proxy.attempt | Enables the use of a proxy user ID for a remote node. The use of a proxy user ID offers improved security because neither the local system nor the remote system requires a real user ID from the other side. | Y \| <u>N</u> |
| protocol | Specifies the communications protocol to be used by the local node. | <u>1</u> \| 2 \|3<br><br>**1**—Specifies TCP/IP.<br><br>**2**—Specifies APPC.<br><br>**3**—Specifies either TCP/IP and APPC can be used. |
| tcp.api.port | Specifies one or multiple IP addresses (or host name) and optional port numbers that are used to establish client sessions with this Connect:Direct node. | *IP address* \| *hostname* [*;port number*]<br><br>The default port number is 1363.<br><br>For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| tcp.host.port | Specifies one or multiple IP addresses (or host name) and port numbers that remote Connect:Direct nodes will connect to for Process execution with this local Connect:Direct for Windows node. | *IP address* \| *hostname* [*;port number*]<br><br>The default port number is 1364.<br><br>For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |
| outgoing.address | Specifies a virtual IP address for adjacent nodes in a cluster (in a high-availability environment) to use for netmap checking by the remote node.<br><br>In a Connect:Direct for Windows high-availability cluster, each instance of Connect:Direct for Windows has a separate IP address and a virtual IP address assigned to the cluster. After all of the Connect:Direct for Windows high-availability nodes are configured to bind to the virtual IP address, remote nodes see the single virtual address during a session. If a node in the cluster fails, another high-availability node takes over. Any remote system that is using netmap checking will still allow communications with the high-availability system.<br><br>Be sure remote nodes specify this virtual IP address in their netmap entries for each node in the high-availability cluster. | *IP address* \| *hostname*<br><br>For more information on IP addresses, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| tcp.src.ports | Specifies an IP address or multiple addresses and the ports permitted for the address when going through a packet-filtering firewall. | Valid IP address with an optional mask for the upper boundary of the IP address range and the associated outgoing port number or range of port numbers for the specified IP address, for example:<br><br>(199.2.4.*, 1024), (fd00:0:0:2015:*::*, 2000-3000), (199.2.4.0/255.255.255.0, 4000-5000),(fd00:0:0:2015::0/ 48, 6000, 7000)<br><br>For more information on IP addresses, masks, and ports, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*.<br><br>A wildcard character (*) is supported to define an IP address pattern. If the wildcard character is used, the optional mask is not valid. |
| tcp.src.ports.list.iterations | Specifies the number of times that Connect:Direct scans the list of available ports defined in tcp.src.ports to attempt a connection before going into a retry state.<br><br>This parameter is automatically added to the initialization parameter and is assigned a value of 1. If desired, change this value. | A numeric value from 1–255. The default value is 1. |
| comm.bufsize | Specifies the data buffer size for transmitting data to and from a remote node. For TCP/IP, this value will be overridden by the value in the Buffer Size field of the TCP/IP Communications Mode Object. | A numeric value from 256–65536.<br><br>The default is 65535. |
| pacing.send.delay | Specifies the default time, in milliseconds, to wait between send operations if the value is not in the network map entry. The value entered in the **Pacing Send Delay** field of the TCP/IP Communication Mode object overrides this value. A value of zero indicates that a data buffer should be sent as soon as possible. | A numeric value from 0–86,400,000 (one day in milliseconds).<br><br>The default is 00.00.00. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| pacing.send.count | Specifies the default number of send operations to perform before automatically waiting for a pacing response from the remote node if the value is not in the network map entry. The value entered in the **Pacing Send Count** field of the TCP/IP Communication Mode object overrides this value. Specify zero for no pacing. | A numeric value from 0–32768. <br><br> The default is 0. |
| tcp.crc | Globally turn on or off the CRC function for TCP/IP processes. | ON \| <u>OFF</u> |
| tcp.crc.override | Determines whether node and Process statement overrides for CRC checking are allowed. If this value is set to n, setting overrides for CRC checking will be ignored. | y \| <u>n</u> |
| tcp.max.time.to.wait | Specifies the maximum time to wait for each pending TCP/IP Read on node to node communications. If the value is 0, Reads will not time out. | A 24-hour time value formatted as *hh:mm:ss*. The maximum value is 23:59:59. The default is **00:03:00**. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| tcp.window.size | Specifies the maximum amount of data in bytes that can be sent without receiving an acknowledgement. After a window size of data is sent without acknowledgement, no more data can be sent until an acknowledgement is received. When **tcp.window.size=0** (default), the system's default value is used, which may be too low for a high-latency, high-bandwidth connection, causing slow data transfer rates. To improve performance in this situation, the window size configured for Connect:Direct for Windows must be large enough to allow a packet to be sent and its acknowledgement received without triggering a wait for an acknowledgement. The optimum window size is the smallest quantity of data that does not trigger waits.<br><br>In order for this parameter to take affect, ensure that the following criteria are met:<br><br>◆ RFC1323 support is enabled on the system.<br><br>◆ The system's maximum TCP window size is larger than the configured value.<br><br>The formula for determining the optimum window size is as follows:<br><br>BDP (bytes) = 125 <bandwidth (Mb/s)> <round-trip delay in ms><br><br>**Note:** This system configuration must be done on any Windows system involved in the transfer, regardless of whether it is a PNODE or an SNODE. | A numeric value from 0–2147483646.<br><br>The default is **0**. |
| runstep.max.time.to.wait | Specifies the maximum time to wait for each pending **run task** or **run job** on node to node communications on the remote node only. If the value is 0, the **run task** or **run job** will not time out. This parameter prevents a task from being terminated when the tcp.max.time.to.wait value is reached. When runstep.max.time.to.wait is set to 0, tasks running on the remote node never terminate. When this variable is set to a value, a task is terminated if the remote task has not finished the job at the time interval defined. When a job is terminated, it is placed in the Hold (HE) queue and no retry effort is attempted. The statistics error reported is "FMH71 was not received". | A 24-hour time value formatted as *hh:mm:ss*. The maximum value is 23:59:59. The default is **00:00:00**. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| active.directory.enabled | Specifies whether Connect:Direct for Windows updates its Active Directory entry at startup. This parameter is set automatically at installation. You do not need to update this parameter unless a node is inserted into or removed from the Active Directory after installation. | Y \| <u>N</u> |
| quiesce.resume | Specifies whether testing mode is enabled for Connect:Direct for Windows. To enable the testing mode, you must specify **Y** for this parameter and have a valid NDMPXTBL parameter table in the Server directory where Connect:Direct for Windows is installed. For more information on configuring the parameter table, see *Using Connect:Direct for Windows in Test Mode* on page 191. | Y \| <u>N</u> |

## About Transmission Control Queue Parameters

The Connect:Direct Transmission Control Queue (TCQ) holds submitted Processes. The TCQ information parameters define information about the TCQ, such as the default Process directory. Refer to Chapter 7, *Managing Processes*, for more information about the Connect:Direct TCQ.

The following table identifies the TCQ information parameters:

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| tcq.max.age | Specifies the maximum number of days a Process with Held due to Error (HE) status remains in the TCQ before it is automatically deleted. Processes are not automatically deleted when you specify **0**. | A numeric value from 0–999.<br>The default is 30 days. |
| tcq.start | Specifies how to start the Connect:Direct node with respect to the TCQ. | <u>W</u> \| C<br>**w** (warm)—Specifies to retain all existing Processes in the TCQ at startup.<br>**c** (cold)—Specifies to delete all existing Processes in the TCQ at startup. |
| process.dir | Specifies the default directory that a Process will run from if a **submit** statement does not specify a fully-qualified path. | Valid, fully-qualified path name. The default is *X:\installation directory\*PROCESS. |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| runtask.restart | Specifies whether a run task operation executes on a remote Windows node after a session failure. If a run task operation is executing on the remote node and a session failure occurs, the local node recognizes the session failure and puts the Process in the Timer queue for retry. The remote node is not aware of the session failure until the Process completes. The checkpoint/restart feature for run task ensures that when the Process restarts on the local node, the run task operation does not execute again on the remote node. | Y \| <u>N</u> |

## About Global Copy Parameters

The global copy parameters define default information for the copy operation, such as the number of bytes transmitted in a copy operation before a checkpoint is taken. See the *Connect:Direct Process Guide* for a description of the **copy** Process statement.

The following table identifies the global copy characteristics parameters:

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| ckpt.interval | Specifies the default checkpoint interval used. The interval is the number of bytes transmitted before a checkpoint is taken. The maximum possible value is gigabytes. | no \| *bytes*K \| *bytes*M<br>The default is 10240K bytes. |
| xlate.dir | Specifies the default directory containing the translation table. The default is the XLATE subdirectory where Connect:Direct is installed. | Valid, fully qualified path name. The default is *X:\installation directory*\XLATE. |
| xlate.send | Specifies the name of the default translation table to use when sending data to a remote node. | Valid name for the send translation table. The default is XLATESND.CDX |
| xlate.recv | Specifies the name of the default translation table to use when copying data from a remote node. | Valid name for the receive translation table. The default is XLATERCV.CDX |
| disable.cache | Enables or disables the Windows file cache. | Y \| <u>N</u> |
| continue.on.exception | Specifies whether a Process attempts to continue processing or goes into HOLD status if an abnormal termination occurs during a Connect:Direct session.<br>**y**—Specifies to attempt to continue processing.<br>**n**—Specifies to go into HOLD status. | Y \| <u>N</u> |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| ecz.cmprlevel | Specifies the compression level to use. Level 1 is the fastest method and offers the least degree of compression. Level 9 provides the greatest degree of compression and is the slowest method. | <u>1</u> \| 2 \| 3 \| 4 \| 5 \| 6 \| 7 \| 8 \| 9 |
| ecz.windowsize | Specifies the size of the compression window or history buffer. The greater the window size, the greater the degree of compression, and the greater the amount of virtual memory used. | 9 \| 10 \| 11 \| 12 \| <u>13</u> \| 14 \| 15 |
| ecz.memlevel | Specifies the amount of virtual memory allocated to maintain the internal compression rate. Memory level 1 uses the least amount of memory, but slows processing and reduces the degree of compression. | 1 \| 2 \| 3 \| <u>4</u> \| 5 \| 6 \| 7 \| 8 \| 9 |
| strip.blanks | Determines whether trailing blank characters at the end of each record are removed from a line of text before it is written to the Windows text file or ignored (I). The strip.blanks parameter is ignored when datatype(binary) is specified. | Y \| N \| <u>I</u> |
| record.wrap | **Note:**  This parameter is needed only in certain circumstances because it restructures the data.<br><br>Influences the way that a sending **copy** step works when a logical record size (LRECL) is specified for the remote platform.<br><br>◆ If record.wrap is set to N, and a record length greater than LRECL is encountered in the source file, Connect:Direct for Windows reports an error. This is the desired behavior in most cases.<br><br>◆ If record.wrap is set to Y, and a record length greater than LRECL is encountered in the source file, the record is broken into records of length at most LRECLs before being sent to the remote node. | Y \| <u>N</u> |

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| retry.msgids | Identifies the message IDs to use to support a file allocation retry attempt. | Any of the valid file allocation retry messages. |
| | Since error codes can vary from one operating system to another and the same error code can have different meanings, use message IDs to identify retry conditions when communicating between two different platforms. When a file allocation or open error occurs on either the local or remote node, the PNODE searches for the message ID in the retry.msgids parameters. If the message ID is found, the Process is retried. | |
| | You can perform retry attempts based on codes only, message IDs only, or a combination of the two. | |
| | When a retry condition is detected, the session is terminated cleanly and the Process is placed in the Timer queue. | |
| retry.codes | The codes to recognize as a file allocation retry attempt. File allocation retry enables a Process with a file allocation or open error on either the local or remote node to run the Process again, beginning at the copy step where the error occurred. This feature supports the ability to retry a Process that failed when a file is already in use. | Any valid error code |
| | When a file allocation or open error occurs on either the local or remote node, the PNODE searches for the error or message ID in the retry.codes and retry.msgids parameters. If the error code or message ID is found, the Process is retried. | |
| | Since error codes can vary from one operating system to another and the same error code can have different meanings, use message IDs to identify retry conditions when communicating between two different platforms. | |
| | You can perform retry attempts based on codes only, IDs only, or a combination of the two. | |
| | When a retry condition is detected, the session is terminated cleanly and the Process is placed in the Timer queue. | |

## About Statistics Parameters

The Connect:Direct statistics facility logs information about Connect:Direct operations. The statistics information parameters define the characteristics of the statistics facility, such as the maximum age of a statistics record. The following table identifies the statistics information parameters:

| Parameter Name | Parameter Definition | Valid Values |
|---|---|---|
| stat.max.age | Specifies how many days to store statistics before automatically deleting them. If you set this value to zero, no statistics records are deleted. | A numeric value from 0–365. The default is 7 days. |
| log.select | Specifies whether Connect:Direct logs the **select process** and **select statistics** commands to the statistics file. This specification does not affect the logging of other Connect:Direct commands. | Y | <u>N</u> |
| stat.sort | Determines if statistics are sorted by timestamp date. | <u>Y</u> | N |
| log.commands | Specifies whether Connect:Direct logs all commands issued from Connect:Direct for Windows to the statistics file. This capability enables you to review the statistics file and determine who issued specific commands and what the responses to those commands were. You can override this parameter for the commands that select Processes and statistics with the log.select parameter. | <u>Y</u> | N |

# Specifying IP Addresses, Host Names, and Ports

Connect:Direct for Windows accepts both Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) as well as host names. You can enter IP addresses/host names and ports in several ways, depending on the field you are specifying:

✦ Address or host name only

✦ Port number only

✦ Address/host name with a port number

✦ Multiple address/host name and port combinations

When specifying IP addresses/host names and ports for Connect:Direct for Windows, use the following guidelines.

## IP Addresses

Connect:Direct for Windows accepts both IPv4 and IPv6 addresses. Wherever an IP address is specified in Connect:Direct for Windows, you can use either IPv4 or an IPv6 addresss.

### IPv4 Addresses

IPv4 supports $2^{32}$ addresses written as 4 groups of dot-separated 3 decimal numbers (0 through 9), for example, 10.23.107.5.

### IPv6 Addresses

IPv6 supports $2^{128}$ addresses written as 8 groups of colon-separated 4 hexadecimal digits, for example, 1001:0dc8:0:0:0:ff10:143e:57ab. The following guidelines apply to IPv6 addresses:

✦ If a four-digit group contains zeros (0000), the zeros may be omitted and replaced with two colons (::), for example:

```
2001:0db8:85a3:0000:1319:8a2e:0370:1337
can be shortened as
2001:0db8:85a3::1319:8a2e:0370:1337
```

✦  Any number of successive 0000 groups may be replaced with two colons (::), but only one set
of double colons (::) can be used in an address, for example:

```
001:0db8:0000:0000:0000:0000:1319:58ab
Can be shortened as:
2001:0db8:0000:0000::1319:58ab
```

✦  Leading zeros in a four-zero group can be left out (0000 can be shortened to 0), for example:

```
2001:0db8:0000:0000:0000:0000:1319:58ab
Can be shortened as:
2001:0db8:0:0:0:0:1319:58ab
```

✦  You can write a sequence of 4 bytes that occur at the end of an IPv6 address in decimal format
using dots as separators, for example:

```
::ffff:102:304
Can be written as:
::ffff:1.2.3.4
```

This notation is useful for compatibility addresses.

## Host Names

When you specify a host name, rather than an IP address, Connect:Direct for Windows gets the IP
address from the operating system. The first IP address returned by the operating system is used
regardless of whether it is in IPv4 or IPv6 format.

A host name (net, host, gateway, or domain name) is a text string of up to 24 characters comprised
of the alphabet (A–Z), digits (0–9), minus sign (-), and period (.), for example, msdallas-dt.

The following guidelines also apply:

✦  No blank or space characters are permitted as part of the name.

✦  Periods are allowed only when they are used to delimit components of domain-style names.

✦  Host names are not case sensitive.

✦  The first and last character must be a letter or digit.

✦  Single-character names or nicknames are not allowed.

## Port Numbers

Port numbers can be appended to the end of IP/host addresses when they are preceded by a
semi-colon (;), for example, 10.23.107.5;1364. This convention is specific to Connect:Direct for
Windows and is not an industry standard.

A port number must be in the range of 0 through 65535. Port numbers lower than 1024 are designated as reserved and should not be used. The following examples show port numbers appended to IP/host addresses using these conventions:

```
10.23.107.5;1364
fe00:0:0:2014::7;1364
msdallas-dt;1364
```

# Multiple Addresses, Host Names, and Ports

You can specify multiple IPv4 and IPv6 addresses and host names by separating them with a comma (,). A space can be added after the comma for readability, for example:

```
10.23.107.5, fe00:0:0:2014::7, msdallas-dt
```

You can also specify a port number for each address or host name. The port is separated from its corresponding address/host name with a semi-colon (;), and each address/host name and port combination is separated by a comma (,). A space may be added after the comma for readability. The following example shows multiple address/host name and port combinations:

```
10.23.107.5;1364, fe00:0:0:2014::7;1364, msdallas-dt;1364
```

Multiple address/host names (and combinations with port numbers) are limited to 1024 characters.

# Using Masks for IP Address Ranges

When you specify a value for the **tcp.src.ports** parameter in the initialization parameters file, you can use masks to specify the upper boundary of a range of IP addresses that will use a specific port, multiple ports, or a range of ports. Connect:Direct for Windows supports masks for both IPv4 and IPv6 addresses as shown in the following sample entry from the **initparms.cfg** file:

```
tcp.src.ports=(199.2.4.*, 1024), (fd00:0:0:2015:*::*, 2000-3000), (199.2.4.0/255.255.255.0, 4000-5000),
(fd00:0:0:2015::0/48, 6000, 7000)
```

These sample addresses specify the following information:

(199.2.4.*, 1024)—Any IPv4 address that falls in the range from 199.2.4.0 through 199.2.4.255 will use only port 1024.

(fd00:0:0:2015:*::*, 2000-3000)—Any IPv6 address that falls in the range from fd00:0:0:2015:0:0:0:0 through fd00:0:0:2015:ffff:ffff:ffff:ffff will use a port in the range of 2000 through 3000.

(199.2.4.0/255.255.255.0, 4000-5000)—Any IPv4 address that falls in the range from 199.2.4.0 through 199.2.255.255 will use a port in the range of 4000 through 5000.

(fd00:0:0:2015::0/48, 6000, 7000)—Any IPv6 address that falls in the range from fd00:0:0:2015:0:0:0:0 through fd00:0:0:ffff:ffff:ffff:ffff:ffff  will use port 6000 or port 7000.

As shown in the sample entry above, the wildcard character (*) is supported to define an IP address pattern. You can specify up to 255 unique IP address patterns or up to 1024 characters in length, each with its own list of valid source ports. If the wildcard character is used, the optional mask is not valid.

---

**Note:**   Masks in hexadecimal format are not supported in Connect:Direct for Windows.

---

# Submitting Processes Using the CLI

The Command Line Interface (CLI) provides another tool to submit Process statements and commands to the Connect:Direct server. Refer to Chapter 5, *Creating Processes*, for instructions on using the Connect:Direct Requester to submit Process statements.

If you prefer to use a command line interface, use the CLI to issue requests interactively or you can submit them automatically from batch files or command files. The CLI enables you to perform the following tasks:

✦ Connect to the Connect:Direct server

✦ Issue Connect:Direct commands

✦ Submit a Process

✦ Change Processes

✦ Delete Processes

✦ Retrieve Process status information

✦ Retrieve Process statistics records

## Connecting to a Server

If you want to connect to a Connect:Direct server using the CLI, use the Logon Connection utility. This utility allows you to connect to a node without defining a user ID and user authority in Connect:Direct for Windows. It can be used to run batch oriented jobs.

To use the Logon Connection Utility, first ensure that Java Runtime 1.3 or later is installed. Then, run the LCU.bat file to configure the connection. This batch file enables you to identify the parameters necessary to connect to a server. It then saves this information to a configuration file called cddef.bin or whatever name you specify. If you do not identify a directory to save the file to, it is saved in the root directory. Use the -f parameter to identify the location and file name for the configuration file.

After you create the configuration file, invoke the CLI and include the -f parameter. Before you can use this utility, download Java 1.3 and define a user ID and password in the Windows setup utility.

To create the configuration file:

1. Type the following command to run the LCU batch file.

```
LCU -fx:\directory\filename
```

where x:\directory is the location to save the configuration file and file name is the name of the configuration file.

2. Enter the following information to identify the connection parameters for the server:

   ◆ IP Address of the Connect:Direct server

   ◆ Port number of the Connect:Direct server

   ◆ User ID to use to connect to the server

   ◆ User password of the user ID used to connect to the server.

   The information is automatically saved to the file you identified in step with the x:\directory\filename variable. If you do not enter this variable, the configuration file is saved in the root directory to the file called cddef.bin.

# Invoking the CLI

To invoke the CLI, use the **direct** command, as illustrated in the following example. You must start the **direct** command from the directory where it is installed or it must be defined in the user's path.

```
direct [-nNodeName -uUserid -pPassword -mn -x -zOutputfile.txt]
```

The following table describes the command and its options. If a parameter is not specified, the default value is used:

| Command | Parameters | Description |
| --- | --- | --- |
| direct | | Invokes the Connect:Direct CLI. |
| | -f[pathname and file] | Identifies the file to logon configuration file to use to automatically connect to a Connect:Direct server. |
| | -n[NodeName] | Identifies node name or IP address. |
| | -u[Userid] | Specifies the user ID to use to connect to the node. |
| | -p[Password] | Specifies the password with which to connect to the Connect:Direct node. |

| Command | Parameters | Description |
|---------|-----------|-------------|
| | -m[n] | Specifies the number of lines to display before pausing the statistics and Process information output. This is calculated automatically if a number is not given with the parameter. |
| | | Do not use this option if the output is being piped to a file because the output will be paused with no visible evidence on the screen. |
| | -x | Echoes the command input on the display. The command issued is written immediately before the output. |
| | -z[filename] | Copies command output to the file you specify as well as displaying the output. |
| | -? | Provides help for the command line interface. |

# Terminating the CLI

Terminate the CLI operation by typing **quit**, followed by a semicolon, at the command line as illustrated below:

```
quit;
```

# Command Syntax

The following information describes the general conventions used in the command syntax:

✦ All commands must be followed by a semi-colon.

✦ User names, passwords, and parameters are case sensitive.

✦ Command keywords are not case sensitive.

✦ File names, group names, user IDs, and passwords are defined as variable length strings. Names referring to objects on other nodes can be subject to restrictions imposed by the other node type.

✦ Length restrictions for Connect:Direct node names are specific to a type of field and data. Refer to specific commands in this chapter for restrictions.

✦ If a parameter specifies the word *generic* as a parameter value, you can type a string that includes an asterisk (**\***) or a question mark (**?**) to provide pattern-matching or wildcard matching for parameter values. The asterisk matches zero or more characters, and the question mark matches any single character.

✦ When *list* is a parameter value in the syntax definition, you can specify multiple parameter values by enclosing the group in parentheses and separating values with commas. A list can also include generic values. An example follows:

```
pname=(payproc,arproc,ivproc,a?prod5*)
```

✦ Most CLI commands can be entered using abbreviations. When abbreviations can be used in place of a command, these abbreviations are identified in the command description table.

# Piping Conventions

The Connect:Direct for Windows CLI supports the following piping conventions:

✦ <*filename.ext* uses the file for input

✦ >*filename.ext* uses the file for output

The following example demonstrates the Connect:Direct piping convention being used on a COMMANDS.TXT file containing the Connect:Direct commands you want to issue.

```
submit
test      process snode=dsd.unix
cp        copy from (file=testfile.dat pnode)
               to   (file=test.dat snode)
          pend;
select statistics;
select process;
quit;
```

By typing the following command, you can execute all of the commands in the COMMANDS.TXT file.

```
direct < commands.txt
```

**Note:**   A batch file can also be created containing the *direct < commands.txt* line if you want to execute these commands from a batch file.

# Controlling Process Operation

The following table provides a summary of all available commands that can be used with the CLI:

| Command | Abbreviations | Description |
|---|---|---|
| submit | sub | Submits a Process to the Connect:Direct node. |
| change process | cha, chg, c | Modifies the specified parameters for a nonexecuting Process. |
| delete process | del, d<br>pro, proc, procs, p | Deletes the specified Process from the TCQ. |
| select process | sel<br>pro, proc, procs, p | Retrieves status information about the specified Process. |
| select message | sel<br>mes, msg, m | Retrieves the text explanation of any message Connect:Direct issues. |
| select statistics | sel<br>sta, stat, stats, st | Retrieves statistics records for the specified Process. |
| traceoff | tof, troff | Disables the trace facility. |
| traceon | ton, tron | Enables the trace facility. |
| help | | Lists the supported Connect:Direct commands. |
| quit | qui, q | Stops the Connect:Direct CLI. |
| stop | sto | Stops the Connect:Direct server. |

# Submitting a Process

Use the **submit** command to request that Connect:Direct execute the operations specified in the Process being submitted. Parameters override the same parameters specified in the **Process** statement. The **submit** command also enables you to resolve symbolic parameters found in the Process.

The parameters for the **submit** command are described in the following table:

| Parameter | Description |
|---|---|
| file=*filename* | The name of the Process file, up to 256 characters. If you specify the **file** parameter, you must specify the parameter *before* any other parameter. If you do not specify this parameter, then the text of the Process must follow the **submit** command |
| class=*nn*\| *session number* | The node-to-node session on which a Process can execute. A Process can execute in the class specified or any higher session class. |
| execprty=*nn* | The operating system execution priority, from 1 to 15, of the Process. The execution priority parameter is used to influence the priority given to the Session Manager when it starts this Process. The higher the priority, the higher the Session Manager priority and, therefore, the more system resources. Scheduling Processes to run in the High priority class can have an adverse effect on the execution of other applications in the system. |
| | The values for the execution priority range from 1 to 15 and are mapped to Windows Process priority classes and values. The highest priority is 15. |
| | Only Connect:Direct for Windows supports the execution priority option. |
| hold=yes \| no \| call | The TCQ hold status of the Process. |
| | **yes**—Places the Process in the Hold queue in HI (Held Initial) status until you explicitly release it by a **change process** command. When you specify both **hold=yes** and a startt value, the **hold** specification takes precedence. A Process with **hold=yes** is placed in the Hold queue even if you specify a start time. |
| | **no**—Does not place the Process in the Hold queue. The Process executes as soon as resources are available. |
| | **call**—Holds the Process until the SNODE connects to the PNODE. At that time, the software releases the Process for execution. It also releases the Process when another Process on the PNODE connects to the SNODE. |
| pacct="*pnode accounting data*" | A string, up to 256 characters, to be used as accounting data for the PNODE. Enclose the string in double quotation marks. |
| pnodeid=(*id* , *pswd*) | Security user IDs and passwords at the PNODE. The subparameters can contain 1–48 alphanumeric characters. You must specify both the ID and the password. |

| Parameter | Description |
|-----------|-------------|
| maxdelay=unlimited \| *hh:mm:ss* \| 0 | Causes the command processor to wait until the submitted Process completes execution or the specified time interval expires. |
| | If the time interval expires, the command processor returns a warning status code and message ID. The Process is not affected by the time interval expiration and executes normally. |
| | **unlimited**—Specifies that the **submit** command processor is to wait until the Process completes execution. |
| | *hh:mm:ss*—Specifies that the **submit** command is to wait for an interval no longer than the specified hours (*hh*), minutes (*mm*), and seconds (*ss*). |
| | **0**—Specifies that the submitted Process must begin execution immediately. If the submitted Process cannot begin execution immediately, the **submit** operation fails. |
| newname=*new process name* | Specifies a new Process name, 1–8 alphanumeric characters long, to override the name within the submitted Process. |
| notify=*userid* | The user to receive Process completion messages. |
| sacct="*snode accounting data*" | Accounting data, from 1–256 characters, for the SNODE. Enclose the string in double quotation marks. |
| retain=yes \| no \| initial | Determines whether a copy of the Process is retained in the TCQ for re-execution after the Process executes. |
| | **yes**—Specifies that the software retains the Process in the Hold queue in HR status after execution. Issue a **change process** command to release the Process for execution. |
| | **no**—Specifies that the software is to delete the Process after execution. |
| | **initial**—Specifies that the software is to retain the Process in the Hold queue in HR status for automatic execution every time Connect:Direct initializes. |
| | If **retain=yes**, setting **startt** causes the Process to execute at regular intervals based on the value of **startt**. |
| | If **retain=initial**, do not use the startt parameter. This causes the submit command to fail. |

| Parameter | Description |
|---|---|
| snode=[*nodename*] \| [*hostname* \| *IPaddress* ; *portnumber* \| *servicename*] | Identifies the SNODE. The SNODE name is a 1–16 alphanumeric character string. Specify the node either on the **submit** command or **Process** statement. If you specify the node in this **submit** command, it overrides the node specified in the **Process** statement. |
| | **nodename**—Identifies the remote node object in the Connect:Direct network map. |
| | **hostname \| IPaddress ; portnumber \| servicename**—*S*pecifies an IP address for the SNODE. This is used for TCP/IP connectivity only. |
| | Specify the host name or IP address, a semicolon, and the port number or service name. For more information on specifying valid IPv4 and IPv6 addresses and ports, see Appendix C, *Specifying IP Addresses, Host Names, and Ports*. |
| | If you specify an IP address, you must also specify netmap.check=r or netmap.check=n in your initialization parameters. |
| | **Note:** If you use IPv6 temporary addresses for outbound connections, the connection will fail unless you configure a well-known address for the PNODE server or you disable temporary addresses for the SNODE. |
| snodeid=(*id* [,*pswd* [,*newpswd*]]) | The security user IDs and security passwords on the SNODE. The subparameters can contain one or more alphanumeric characters. |
| | **newpswd**—Specifies a new password value. This subparameter is not supported by all types of Connect:Direct nodes. On z/OS systems only, the user password changes to the new value on the SNODE if the userid and old password are correct. If the SNODE is a UNIX node, the password does not change. |
| | If you specify the password, you must also specify the ID. If you specify a new password, you must also specify the existing password. |
| prty=*nn* | The selection priority of the Process for execution. This priority parameter is used for Process selection. A Process with a higher priority is selected for execution before a Process with a lower priority. The priority value does not affect the priority during transmission. |
| | Values range from 0–15, where 15 is the highest priority. |

| Parameter | Description |
|---|---|
| startt=([date \| day] [,time]) | Identifies the specified date, day, or time to execute the Process. The Process is placed in the Timer queue in WS status. The date, day, and time are positional parameters. If you do not specify date or day, type a comma before the time. |
| | **date**—Specifies the day, month, and year, that you can code as *mm/dd/yyyy* or *mm-dd-yyyy*. You can code month and day as one or two digits and year as two or four digits. If you only specify date, the time defaults to 00:00:00. The current date is the default. |
| | **day**—Specifies the day of the week. Values are today, tomorrow, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. |
| | **time**—Specifies the time of day in *hh:mm*[*:ss*] [*am* \| *pm*] where *hh* is hours, *mm* is minutes, and *ss* is seconds. Seconds are optional. You can specify the hour in either 12- or 24-hour format. If you use the 12-hour format, then you must specify am or pm. The default format is the 24-hour format. |
| | If the time that you specify has already passed, Connect:Direct schedules the Process for the next valid date and time. For example, if you set the Process to start daily at 5:00 PM, or **startt=(, 17:00)**, the Process submits at 5:30 PM and Connect:Direct schedules the Process for the next day. |
| | If you specify only the day value, the time defaults to midnight (00:00:00). This means that if you submit a Process on Monday, with Monday as the only **startt** parameter, the Process does not run until the following Monday at midnight. |
| | Do not set the **startt** parameter if **retain=initial**. This causes the **submit** command to fail. |
| *&symbolic name n="variable string n"* | A symbolic parameter assigned a value. The value is substituted within the Process when the software encounters the symbolic parameter. The symbolic within the Process must be enclosed within quotes. |

## Examples

The following command submits the Process named **payroll.cdp**. Because the command specifies **retain=yes**, the Process is retained in the TCQ after execution. The Process starts the next Monday at 00:00:00. The command specifies Process accounting data for the PNODE.

```
submit file=payroll.cdp retain=yes startt=monday pacct="2003,dept-27";
```

The following command submits the Process named **copyfil.cdp**. Because the command specifies **startt**, the Process executes on the first day of January 2003 at 11:45 a.m.

```
submit file=copyfil.cdp snode=vmcent startt=(01/01/2003, 11:45:00 am);
```

# Changing Process Parameters

Use the **change process** command to modify specified parameters for a *nonexecuting* Process or Processes in the TCQ. Select the Processes to change by Process name, Process number, SNODE name, submitter ID, or a combination of these.

---

Note:    All changes affect the Process in the TCQ, not the original text of the Process as submitted.

---

The **change process** command does the following:

✦ Changes the class, SNODE name, destination node, or priority of a Process

✦ Places a Process in the Hold queue or releases a Process from the Hold queue

The format for the **change process** command follows:

| Command | Parameters |
|---------|------------|
| change process | /* Search Criteria */ |
| | [**pname=***Process name* \| *generic* \| (*list*)] |
| | [**pnumber=***Process number* \| (*list*)] |
| | [**snode=***snode name* \| *generic* \| (*list*)] |
| | [**submitter=**(*node name, userid*) \| *generic* \| (*list*)] |
| | /* The following parameters specify the characteristics of Connect:Direct Processes that you can modify. */ |
| | **class=** \| *session number* |
| | †**execprty=***nn* |
| | **hold=yes** \| **no** \| **call** |
| | **newsnode=***new snode name* |
| | **release** |
| | **prty=***nn* |

## Search Parameters

Specify at least one of the following search criteria parameters:

| Parameter | Description |
|---|---|
| pname=*Process name* \| *generic* \| (*list*) | The name of the Process or Processes. The name can be 1–8 alphanumeric characters long. |
| pnumber=*Process number* \| (*list*) | The Process number of the Process. The Process numbers are assigned when you submit the Process. Valid Process numbers range from 1–999999. |
| snode=s*node name* \| *generic* \| (*list*) | Searches for a Process or Processes by the SNODE (partner) name. The SNODE name can contain 1–16 alphanumeric characters. You can use the IP address of the SNODE as your SNODE name. |
| submitter=(*node name, userid*) \| *generic* \| (*list*) | Searches for a Process or Processes by the node specification and userid of the Process owner. The maximum combined length, including the node name and userid for this parameter, is 66 characters. |

## Change Parameters

You can change one or more of the following characteristics of a Connect:Direct Process:

| Parameter | Description |
|---|---|
| class=*nn* \| *session number* | Changes the node-to-node session on which the Process can execute. A Process can execute on the specified class or any higher session class. |
| execprty=*nn* | The execution priority of the Process. The values for the execution priority range from 1–15 and are mapped to Windows Process priority classes and values. The highest priority is 15.<br>Only Connect:Direct for Windows supports the execution priority option. |
| hold=yes \| no \| call | Moves the Process to the Hold or Wait queue.<br>**yes**—Places the Process in the Hold queue in HO (Held by Operator) status until another **change process** command explicitly releases it.<br>**no**—Places the Process in the Wait queue in WC (Waiting for Connection) status. The Process executes as soon as resources are available.<br>**call**—Places the Process in the Hold queue in HC (Held for Call) status until the SNODE connects to the PNODE or another Process is submitted. At that time, Connect:Direct releases the Process for execution. |
| newsnode=*new snode name* | Specifies a new SNODE name to assign to the Process. |

| Parameter | Description |
|---|---|
| release | Releases the Process from a held state. This parameter is equivalent to **Hold=no**. |
| prty=*nn* | The selection priority in the TCQ. This priority parameter is used for Process selection. A Process with a higher priority is selected for execution before a Process with a lower priority. The priority value does not affect the priority during transmission.<br><br>Values range from 0–15 inclusive, where 15 is the highest priority. |

### Example

The following command changes the SNODE name for any nonexecuting Process named **cdproc** to a new SNODE name, **paris**.

```
change process pname=cdproc newsnode=paris;
```

# Deleting a Process

Use **delete process** to remove a Processes from the TCQ. You can select the Processes to delete by Process name, Process number, SNODE name, submitter, or a combination of these. The format for the **delete process** command follows:

| Command | Parameters |
|---|---|
| delete process | /* Search Criteria */ |
| | [**pname=***Process name* \| *generic* \| (*list*)] |
| | [**pnumber=***Process number* \| (*list*)] |
| | [**snode=***snode name* \| *generic* \| (*list*)] |
| | [**submitter=**(*node name, userid*) \| *generic* \| (*list*)] |
| | /* Processing Parameters */ |
| | [**force=yes** \| **no**] |
| | [**hold=yes** \| **no**] |

## Search Parameters

Use the each parameters to identify the Processes you want to delete. You can delete Processes by name, number, node, or a combination of the criteria.

| Parameter | Description |
|---|---|
| pname=*Process name* \| *generic* \| (*list*) | The name of the Processes, from 1–8 alphanumeric characters, to delete. |
| pnumber=*Process number* \| (*list*) | The number of the Process to delete. The Process number is assigned when the Process is submitted. |
| snode=*snode name* \| *generic* \| (*list*) | The SNODE name of the Processes to delete. The SNODE name can be 1–16 alphanumeric characters long. |
| submitter=(*node name, userid*) \| *generic* \| (*list*) | The submitter node name and user ID of the Processes to delete. The maximum combined length, including the node name and user ID, is 66 characters. |

## Processing Parameters

Define one or more of the following parameters to identify how the delete Processes are managed:

| Parameter | Description |
|---|---|
| force=yes \| <u>no</u> | Forcibly terminate an executing Process. Use this parameter if a Process is in the executing state and is waiting for unavailable resources. |
| | **yes**—Forcibly and immediately terminates the Process or Processes. |
| | **no**—Notifies the partner node of the intent to terminate and terminates the Processes. This is the default. |
| hold=yes \| <u>no</u> | Specifies whether the terminated Process is placed in the Hold queue. |
| | **yes**—Places the Process in the Hold queue in HS (Held Due to Execution Suspension) status after termination. |
| | **no**—Deletes the Process from the TCQ after termination. This is the default. |

## Examples

The following command deletes all Processes submitted by userid **cduser** on node **dallas**. If the Processes are executing, it stops and removes them from the TCQ.

```
delete process submitter=(dallas, cduser);
```

The following command deletes all Processes named **rome** from the TCQ. If the Processes are executing, the command forcibly terminates them.

```
delete process pname=rome force=yes;
```

# Selecting a Process

Use **select process** to display information about Processes in the TCQ. Select Processes by name, number, queue, SNODE, status, submitter, or a combination of these.

## Format

The format for the **select process** command follows:

| Command | Parameters |
|---|---|
| select process | /* Selection Criteria */ |
| | [**pname=**_Process name_ \| _generic_ \| (_list_)] |
| | [**pnumber=**_Process number_ \| (_list_)] |
| | [**queue**=all \| exec \| hold \| wait \| timer] |
| | [**snode=**_snode name_ \| _generic_ \| (_list_)] |
| | [**status**=ex \| hc \| he \| hi \| ho \| hr \| hs \| pe \| re \| wa \| wc \| ws \| (list)] |
| | [**submitter=**(_node name, userid_) \| _generic_ \| (_list_)] |
| | /* Display option */ |
| | [**detail=**yes \| <u>no</u>] |

## Selection Parameters

Specify one or more of the following selection parameters. If you do not specify one of the following selection parameters, all Processes in the TCQ you are authorized to access are selected.

| Parameter | Description |
|---|---|
| pname=_Process name_ \| _generic_ \| (_list_) | Identifies the Process name, up to 8 alphanumeric characters long. |
| pnumber=_Process number_ \| (_list_) | Identifies Processes numbers to select. The Process number is assigned when you submit the Process. Valid Process numbers range from 1–99999. |
| queue=<u>all</u> \| exec \| hold \| wait \| timer | Identifies queue names to select.<br>**all**—Selects all queues. This is the default.<br>**exec**—Selects Processes from the Execution queue.<br>**hold**—Selects Processes from the Hold queue.<br>**timer**—Selects Processes from the Timer queue.<br>**wait**—Selects Processes from the Wait queue. |

| Parameter | Description |
|---|---|
| snode=*snode name* \| *generic* \| *(list)* | Identifies the SNODE name, from 1–16 alphanumeric characters, to select. |
| status=ex \| hc \| he \| hi \| ho \| hr \| hs \| pe \| re \| wa \| wc \| ws \| (*list*) | Selects a Process status to select. If you do not specify a status value, information is generated for all status values.<br><br>**ex**—Selects Processes from the Execution queue.<br><br>**hc**—Selects Processes submitted with **hold=call**.<br><br>**he**—Selects Processes held due to a connection error.<br><br>**hi**—Selects Processes submitted with **hold=yes.**<br><br>**ho**—Selects Processes held by a **change process** command issued with **hold=yes.**<br><br>**hr**—Selects Processes submitted with **retain=yes**.<br><br>**hs**—Selects Processes suspended by a **delete process** command issued with **hold=yes.**<br><br>**pe**—Selects submitted Processes that are awaiting the Session Manager. After the Session Manager initializes, it places the Process in the Execution queue and changes the status to EX.<br><br>**re**—Selects Processes that are waiting for restart after session failure.<br><br>**wa**—Selects Processes that are eligible for execution but not yet selected or running.<br><br>**wc**—Selects Processes that are ready for execution and awaiting an available connection to the SNODE.<br><br>**ws**—Selects Processes waiting in the Timer Queue for a start time. |
| **submitter=(*node name, userid*) \| *generic* \|(*list*)** | Selects Processes by node name and user ID of the submitter of each Process. The maximum combined length, including the node name and userid, is 66 characters. |

## Display Parameter

The following display parameter generates a detailed report of the select process command.

| Parameter | Description |
|---|---|
| detail=*yes* \| <u>*no*</u> | Specifies the type of report generated for the selected Process or Processes. The default is no.<br><br>**yes**—Generates a detailed report<br>**no**—Generates a short report |

## Examples

The following example shows the report information returned when specifying **detail**=yes.

```
================================================================
            SELECT PROCESS
================================================================
Process Name       =>SAMPLE      Class         =>
Process Number     =>38          Priority      =>
Submitter Node     =>CSDPUBS     Pnode         CSGPUBS
Submitter          user1        Snode         CSGPUBS
Retain Process     =>N

Submit Time        =>09:54:33    Schedule Time =>
Submit Date        =>10/30/2002  Schedule Date =>

Queue              =>HOLD
Process Status     =>HI
Message Text       =>
--------------------------------------------------
Process Name       =>SAMPLE      Class         =>
Process Number     =>39          Priority      =>0
Submitter Node     =>CSGPUBS     Pnode         =>CSGPUBS
Submitter          user         Snode         =>CSGPUBS
Retain Process     =>N
Submit Date        =>09:54:35    Schedule Time =>
Submit Date        =>10/30/2002  Schedule Date =>
Queue              =>HOLD
Process Status     =>HI
Message Text       =>
--------------------------------------------------
```

The following example shows the report information returned when specifying **detail**=no or
omitting the **detail** parameter.

```
================================================================
            SELECT PROCESS
================================================================
PROCESS NAME NUMBER USER       SUBMITTER NODE      QUEUE STATUS
----------------------------------------------------------------
SAMPLE        39    user       CSGPUBS             HOLD HI
SAMPLE        39    user       CSGPUBS                 HOLD HI
```

The following command returns status information for Process number 9.

```
select process pnumber=9;
```

# Displaying a Connect:Direct Message

Use **select message** to retrieve the text of any Connect:Direct message.

## Format

The format for the **select message** command follows:

| Command | Parameters |
|---|---|
| select message | Selection Criteria |
| | **msgid=**_message id_ |

The following parameter specifies the message ids to display:

| Parameter | Description |
|---|---|
| **msgid=**_message id_ | The Connect:Direct message ID of the message request. Do not use generic specifications. This parameter is required. |

## Example

Type the following command at the CLI prompt to retrieve the text of the Connect:Direct message LCCC00I:

```
select message msgid=LCCC00I;
```

# Selecting Statistics

Issue **select statistics** to examine records in the Connect:Direct statistics database. The type of information in the output from this command includes such data as copy status and execution events. The search criteria provide flexibility in selecting information you want to retrieve. Additional parameters determine the form in which the information is presented.

When using **select statistics**, specify your selection criteria carefully to avoid displaying excessive volumes of records. If you do not provide selection criteria, all records for the day are retrieved.

The format for the **select statistics** command follows.

| Command | Parameters |
|---|---|
| select statistics | /* Selection Criteria */ |
| | [**ccode=**(*operator*, *code*)] |
| | [**dfile=***destination filename* | (*list*)] |
| | [**pname=***Process name* | *generic* | (*list*)] |
| | [**pnumber=***Process number* | (*list*)] |
| | [**reccat=caev** | **capr** | (**caev,capr**)] |
| | [**recids=***record id* | (*list*)] |
| | [s**node=***snode name* | *generic* | (*list*)] |
| | [**sfile=***source filename* | (*list)]* |
| | [**startt=**([*date* | *day*][, *hh:mm:ss*[*am*|*pm*]])] |
| | [**stopt=**([*date* | *day*][, *hh:mm:ss*[*am*|*pm*]])] |
| | [**submitter=**(*node* name, *userid*) | *generic* | (*list*)] |
| | /* Display option */ |
| | [**detail=**yes | <u>no</u>] |

## Parameters

Provide one or more of the following parameters to determine what statistics are returned:

| Parameter | Description |
|---|---|
| ccode=(*operator*, *code*) | Select statistics records based on the completion code operator and return code values associated with step termination. The condition code operator default is **eq**. You must specify the return code. |
| | Following are the valid completion code *operators*: |
| | eq \| = \| == Equal  (This is the default.) |
| | ge \| >= \| =>Greater than or equal |
| | gt \| >          Greater than |
| | le \| <= \| =< Less than or equal |
| | lt \| <          Less than |
| | ne \| !=        Not equal |
| | Following are the valid completion codes: |
| | 0—Successful execution of the Process. |
| | 4—A warning level error was encountered. The statement probably completed normally but you should verify the execution results. |
| | 8—An error occurred during Process execution. |
| | 16—A Severe error occurred during Process execution. |
| dfile=*destination filename* \| (*list*) | Enables you to search all copy termination records (CAPR category, CTRC record ID) to find those with a destination file name matching the filename or list of filenames specified. |
| | This parameter is not supported by Connect:Direct UNIX. |
| pname=*Process name* \| *generic* \| (*list*) | Selects Process statistics by Process name, a generic name, or a list of names. The name can be 1–8 alphanumeric characters long. |
| pnumber=*Process number* \| (*list*) | Selects statistics by Process number or a list of Process numbers. The Process number is assigned when the Process is submitted. |
| reccat=caev \| capr \| (caev , capr) | Selects statistics based on whether the record category is related to events or to a Process. |
| | The default for this keyword depends on the other search criteria specified. If you specify Process characteristics, such as Process name, Process number, or Submitter, the default is **capr**. If you perform a general search using **startt** or **stopt**, the default is **caev** and **capr**. |
| | **caev**—Specifies that the retrieved statistics file records should include those related to Connect:Direct events, such as a Connect:Direct shutdown. |
| | **capr**—Specifies that the retrieved statistics file records should include those related to one or more Connect:Direct Processes. |

| Parameter | Description |
|---|---|
| recids=*record id* \| (*list*) | Specifies selection by record ID or a list of record IDs. This parameter identifies particular types of statistics records, such as a copy termination records or initialization event records. Following is a list of the record IDs: |
| | APCK—Asset protection check |
| | AUPR—Authorization file processing |
| | CHGP—Change Process command issued |
| | COAC—Communication activated |
| | CMLT—CMGR listen thread terminated |
| | CRHT—Connect:Direct copyright |
| | CSTP—Child Process stopped |
| | CTRC—Copy control record written |
| | CTRM—Child Process terminated |
| | CUKN—Child process unknown status |
| | CXIT—Child process exited |
| | DELP—Delete Process command issued |
| | FLSP—Flush Process command issued |
| | FMRV—Formatted Header (FMH) received |
| | FMSD—Formatted Header (FMH) sent |
| | GPRC—Get Process issued |
| | IFED—If statement ended |
| | IPPR—Initialization parameter processing |
| | LIEX—License has expired |
| | LIOK—Listen okay |
| | LSST—The record ID of a step on the local node |
| | LWEX—License will expire in 14 days |
| | NAUH—Node Authorization check issued |
| | NMOP—Network map file opened |
| | NMPR—The network map is updated through Browser, Control Center, or KQV Interface. |
| | NUIC—Connect:Direct initialization complete |
| | NUIS—Connect:Direct start initialization |
| | NUT1—Connect:Direct phase one termination complete status |
| | NUT1—Connect:Direct phase two termination complete status |
| | NUTC—Connect:Direct termination complete |
| | NUTR—Connect:Direct termination requested |
| | PERR—Process error was detected |
| | PFLS—Process was flushed |
| | PMED—Process Manager ended |
| | PMIP—Process Manager Initprocs thread initialized |

| Parameter | Description |
|---|---|
| recids=*record id* \| (*list*) (continued) | PMMX—Process Manager Max Age thread initialized |
| | PMRC—Process Manager release cell thread initialized |
| | PMST—Process Manager started |
| | PPER—Pipe error |
| | PRED—Process ended |
| | PRIN—Process interrupted |
| | PSAV—Process was saved |
| | PSED—Process step was detected |
| | PSTR—Process has started |
| | QCEX—A Process moved from another queue to the EXEC queue |
| | QCHO—A Process moved from another queue to the HOLD queue |
| | QCWA—A Process moved from another queue to the WAIT queue |
| | QCTI—A Process moved from another queue to the TIMER queue |
| | QCHO—A Process moved from another queue to the HOLD queue |
| | RJED—Run Job command completed |
| | RNCF—Remote Connect:Direct server call failed |
| | RSST—The record ID of a step on the remote node |
| | RTED—Run Task command completed |
| | SBED—Submit complete |
| | SELP—Select Process command issued |
| | SELS—Select Statistics command issued |
| | SEND—Session end issued |
| | SERR—System error |
| | SFSZ—Size of the file submitted |
| | SHUD—Connect:Direct shutdown |
| | SIGC—System error |
| | SMED—Session Manager ended |
| | SMST—Session Manager started |
| | SNMP—SNMP |
| | SSTR—Session start issued |
| | STOP—Stop Connect:Direct command issued |
| | SUBP—Submit command issued |
| | TCPI—TCP started |
| | TRAC—Trace command issued |
| | TZDI—Time zone of the local node represented as the difference in seconds between the time at the local node and the Coordinated Universal Time |
| | UNKN—Unknown command issued |
| | USEC—User Security check issued |
| | xxxx—Record types identified by the first four characters of the message ID |

| Parameter | Description |
|---|---|
| snode=*snode name* \| *generic* \| (*list*) | Selects statistics file records by SNODE name, a generic node name, or a list of node names. The SNODE name can be 1–16 alphanumeric characters long. |
| sfile=*filename*\| (*list*) | Enables you to search all copy termination records (CAPR category, CTRC record ID) to find those with a destination file name matching the file name or list of the file names specified.<br><br>This parameter is not supported by Connect:Direct for UNIX. |
| startt=([*date* \| *day*] [, *time*]) | Selects statistics starting with records logged since the specified date, day, or time. The date, day, and time are positional parameters. If you do not specify a date or day, type a comma before the time.<br><br>*date*—Specifies the day (dd), month (mm), and year (yy), which you can code as mm/dd/yyyy or mm-dd-yyyy. If you only specify date, the time defaults to 00:00:00. The current date is the default.<br><br>*day*—Specifies the day of the week. Values are today, yesterday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. If you specify a day of the week, Connect:Direct uses the previous matching day.<br><br>*time*—Specifies the time of day coded as hh:mm:ss[am \| pm] where hh is hours, mm is minutes, and ss is seconds. You can specify the hour in either 12- or 24-hour format. If you use the 12-hour format, then you must specify am or pm. The default format is the 24-hour format. The default value is 00:00:00, which indicates midnight. If you specify only the day value, the time defaults to 00:00:00. |
| stopt=([*date* \| *day*] [, *time*]) | Retrieves statistics including records logged up to and including the specified date, day, or time. The date, day, and time are positional parameters. If you do not specify a date or a day, type a comma before the time.<br><br>*date*—Specifies the day (dd), month (mm), and year (yy), that you can code as mm/dd/yyyy or mm-dd-yyyy. If you only specify date, the time defaults to 00:00:00. The current date is the default.<br><br>*day*—Specifies the day of the week. Values are today, yesterday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. If you specify a day of the week, Connect:Direct uses the previous matching day.<br><br>*time*—Specifies the time of day coded as hh:mm:ss[am \| pm] where hh is hours, mm is minutes, and ss is seconds. You can specify the hour in either 12- or 24-hour format. If you use the 12-hour format, then you must specify am or pm. The default is the 24-hour format. The default value is 00:00:00, which indicates midnight. If you specify only the day value, the time defaults to 00:00:00. |
| submitter=(*node name, userid*) \| *generic* \| (*list*) | Selects statistics by the node name and userid of the Process owner (submitter). You can also specify a generic name and userid or a list of names and userids. The maximum combined length, including the node name and userid, is 66 characters for this parameter. |

## Display Parameter

Provide this optional parameter if you want to generate a report of the statistics that are generated:

| Parameter | Description |
| --- | --- |
| detail=*yes* \| *no* | Specifies the type of report generated for the selected Processes. The default is no.<br><br>**yes**—Generates a detailed report<br>**no**—Generates a short report |

## Examples

The following example shows the report information returned when specifying **detail**=yes.

```
===================================================================
              SELECT STATISTICS
===================================================================
PROCESS RECORD    Record Id=>SUBP
Process Name      =>SAMPLE     Stat Log Time      =>09:54:33
Process Number    =>38         Stat Log Time      =>07/30/2003

Submitter Id      =>user1
Snode             =>CSGPUBS
Completion Code   =>0
Message Id        =>
Short Text        =>
-------------------------------------------------------
PROCESS RECORD    Record Id=>SUBp
Process Name      =>SAMPLE     Stat Log Time      =>09:54:33
Process Number    =>39         Stat Log Time      =>07/30/2003

Submitter Id      =>user1
Snode             =>CSGPUBS
Completion Code   =>0
Message Id        =>
Short Text        =>
-------------------------------------------------------
```

The following example shows the report information returned when specifying **detail**=no or omitting the **detail** parameter.

```
===================================================================
              SELECT STATISTICS
===================================================================
P RECID LOG TIME            PNAME PNUMBER STEPNAME CCOD FDBK MSGID
P RECID LOG TIME            MESSAGE TEXT
-----------------------------------------------------------------
P  SUBP 07/30/2003 09:54:33  SAMPLE  38    0  0
P  SUBP  07/30/2003 09:54:35  SAMPLE   39     0  0
```

The following command generates statistics output for Process number 7. The output consists of all records for that Process starting with those generated on July 11, 2003.

```
select statistics pnumber=7 startt=(07/11/2003);
```

# Using Traceoff

The Connect:Direct server provides a comprehensive trace facility that assists in the diagnosis of problems relating to any facet of the operation of the server. Use the **traceoff** command to disable a trace started with the **traceon** command.

The format for the **traceoff** command follows.

| Command | Parameter |
| --- | --- |
| traceoff | [file=*filename*] |
| | [**type=cmgr** | **pmgr** | **smgr** | **comm** | (*list*)] |
| | †[**dest=***destination* | (*list*)] |
| | †[**pnode** | **snode**] |
| | †[**pname=***process name* | (*list*)] |
| | †[**pnum=***process number* | (*list*)] |

† Valid for **smgr** and **comm** traces only.

Provide one or more of the following parameters to identify the trace to turn off including the output file of the trace, the type of trace, the destination, the node, and the Process name or number.

| Parameter | Description |
| --- | --- |
| file=*filename* | The name of the trace output file. The default is CDTRACE.CDT. |
| type=cmgr | pmgr | smgr | comm | (*list*) | Disables traces by type.<br>**cmgr**—Traces events relating to the interaction of the server with connected workstations and with the server console.<br>**pmgr**—Traces events relating to the manipulation of Connect:Direct Processes.<br>**smgr**—Traces events relating to the execution of Connect:Direct Processes and the server's interaction with other Connect:Direct nodes.<br>**comm**—Traces only interactions with external communications facilities invoked from Session Manager threads and used to communicate with other Connect:Direct nodes. |

## Additional Session Manager and Communications Trace Parameters

The following parameters are valid for SMGR and COMM trace types only. The **dest, pnode, snode**, **pname,** and **pnum** parameters are mutually exclusive.

| Parameter | Description |
|---|---|
| dest=*destination* \| (*list*) | The destination node name of the Process you want to stop tracing or a list of up to four node names. |
| pnode \| snode | The PNODE or SNODE session managers.<br>**pnode**—Disables the trace of all PNODE session managers.<br>**snode**—Disables the trace of all SNODE session managers. |
| pname=*process name* \| (*list*) | The name of the Process or up to four names you want to stop tracing. |
| pnum=*process number* \| (*list*) | The Process number, up to four Processes, you want to stop tracing. |

### Example

The following command turns off the full SMGR trace for the Process named **payroll**.

```
traceoff type=smgr pname=payroll;
```

# Using Traceon

Use the **traceon** command to enable the trace facility. The trace facility enables you to capture information to assist in the diagnosis of Connect:Direct problems.

The format for the **traceon** command follows:

| Command | Parameters |
|---|---|
| traceon | [**file**=*filename*] |
| | [**filesize**=*nn***K** \| *nn***M** \| **0**] |
| | [**level**=**basic** \| **intermediate** \| **full**] |
| | [**type**=**cmgr** \| **pmgr** \| **smgr** \| **comm** \| (*list*)] |
| | [**wrap**=**yes** \| **no**] |

† Valid for **smgr** and **comm** traces only.

| Command | Parameters |
|---|---|
| | †[**dest=***destination* \| (*list*)] |
| | †[**pnode** \| **snode**] |
| | †[**pname=***process name* \| (*list*)] |
| | †[**pnum=***process number* \| (*list*)] |

† Valid for **smgr** and **comm** traces only.

## Parameters

Define one or more of the following parameters to define the traces:

| Parameter | Description |
|---|---|
| file=*filename* | The name of the output file if you want to write the trace information to a file. The size of the name can range from 1–256 characters. The default is CDTRACE.CDT in the Connect:Direct directory. |
| filesize=*nnn*K \| *nnn*M \| <u>0</u> | The maximum file size as a number of kilobytes (K) or megabytes (M). A value of 0 indicates that the file can grow indefinitely. |
| level=basic \| intermediate \| <u>full</u> | The level of trace detail.<br><br>**basic**—Produces a trace of function entry and exit.<br><br>**intermediate**—Produces a trace of function entry and exit, plus arguments to functions.<br><br>**full**—Produces a trace with function entry/exit, function argument display, plus display of selected internal control blocks. |
| type=cmgr \| pmgr \| smgr \| comm \| (*list*) | The type of event or a list of events to be traced.<br><br>**cmgr**—Traces events relating to the interaction of the server with connected workstations and the server console.<br><br>**pmgr**—Traces events relating to the manipulation of Connect:Direct Processes. This trace type provides information about the submission, update, deletion, selection for processing, and termination of Connect:Direct Processes.<br><br>**smgr**—Traces events relating to the execution of Connect:Direct Processes and the server's interaction with other Connect:Direct nodes. Traces can be started for specific Process numbers or names, for specific destinations, or for all PNODE or SNODE Session Managers.<br><br>**comm**—Traces only interactions with external communications facilities invoked from Session Manager threads and used to communicate with other Connect:Direct nodes. You can start traces for specific Process numbers or names, for specific destinations, or for all PNODE or SNODE Session Managers. |

| Parameter | Description |
|---|---|
| wrap=yes \| <u>no</u> | Specifies whether you want your trace output to be a wraparound file. |
| | **no**—Requests a linear file that grows until either the user–specified space restriction is reached or you run out of disk space. Linear traces are useful when you can qualify the trace criteria sufficiently or when it is important to not miss any trace events. |
| | **yes**—Requests a wraparound file that, when the user–specified space restriction is reached, wraps back to the beginning and overwrites the oldest records. Wraparound traces are useful when the problem you are trying to trace occurs intermittently or is very difficult to reproduce. |

# Stopping Connect:Direct

Use the **stop** command to initiate an orderly Connect:Direct server shutdown sequence or forcibly terminate the software. Connect:Direct will not run new Processes and will not establish new connections with remote systems. You can issue commands and users can sign on until the server terminates. You must identify the type of stop or an error message is generated.

## Format

The format for the **stop** command follows.

| Command | Parameters |
|---|---|
| stop | [force \| immediate \| step\| <u>quiesce</u>] |

## Parameters

Choose one of the following options listed in order of severity:

| Parameter | Description |
|---|---|
| force | Forcibly terminates the Connect:Direct server with no attempt to terminate executing Processes in an orderly fashion or write statistics to disk. Connect:Direct releases all server resources, including connections, LU 6.2 sessions, memory, and semaphores. It retains all active Processes in the TCQ and restarts them when you restart the Connect:Direct server. |

| Parameter | Description |
|-----------|-------------|
| immediate | Begins an immediate, but orderly, shutdown of all activity and terminates Connect:Direct. The software terminates connections, writes statistics records, closes files, and shuts down. It retains all active Processes in the TCQ and restarts them when you restart the Connect:Direct server. |
| step | Shuts down Connect:Direct after all currently executing Process steps complete. The software writes statistics records, closes files, and shuts down. To restart Processes at the next step the next time you start Connect:Direct, set the tcq.start initialization parameter to require a warm start. |
| quiesce | Runs all executing Processes to completion. Connect:Direct starts no new Processes. |

### Example

The following command forcibly terminates Connect:Direct and returns control to the operating system.

```
stop force;
```

## Getting Help

Use the **help** command to receive a list of the commands supported by the Connect:Direct CLI.

Any command with a **–?** or **/?** will print out the syntax for the commands usage.

```
command /?;
or
command -?;
```

# Modifying Translation Tables

## About Translation Tables

Connect:Direct translates data from one character set to a different character set, such as from ASCII to EBCDIC, using character translation tables. These translation tables provide greater flexibility in the translation of data when copying data to or from a Connect:Direct node.

Default translation tables are defined in the initialization parameters for the Connect:Direct node. You also have the ability to specify a different translation table in the sysopts parameter with each Copy statement. Connect:Direct provides two standard translation tables:

✦ XLATERCV translates data from EBCDIC to ASCII.

✦ XLATESND translates data from ASCII to EBCDIC.

Translation is only performed when the data type is text. No translation is done if the data type is binary. This chapter explains how to modify the translation tables provided with Connect:Direct for Windows.

## Editing the Translation Tables

The Connect:Direct installation creates the following default translation tables for use when sending or receiving data to or from a remote Connect:Direct node.

✦ XLATERCV.CDX—Translates data from EBCDIC to ASCII

✦ XLATESND.CDX—translates data from ASCII to EBCDIC

Translation is performed if a data type of text is specified in the **sysopts** parameter of a **copy** statement.

To edit the translation tables from Connect:Direct Requester:

1. Select **Translation Table** from the **Admin** menu. The **Edit Installed Translation Table** dialog box is displayed.

2. Type the name of the translation table file, or select one from the drop-down list. The translation table names are XLATERCR.CDX and XLATESND.CDX

3. Click **OK** to open the selected translation table. The translation table view is displayed.

   Each cell stores the character value for the target character set. The source character set is used as an index into the table. For example, an ASCII blank (Hex 0) would fetch the byte at offset Hex 0 in the translation table. If the byte at location Hex 0 contains Hex code 40, that would translate to an EBCDIC code indicating a blank character.

4. Right-click anywhere within the **Translation Table** dialog box to view the following shortcut menu.

5. Click either **Hex**, **Dec**, or **Oct** to indicate whether the table elements are in a hexadecimal, decimal, or octal representation. The translation table refreshes to reflect your selection.

6. Select the elements in the table you want to modify and type in the new values.

7. Right-click to see the shortcut menu and click **Appl**y to save your changes. The following dialog box is displayed:

8. Click **OK** to save your changes to the selected Connect:Direct node.

# Appendix F

# Editing Connection Settings

Before you begin using the SDK to create your own programs or use Connect:Direct for Windows to transfer files, you must create connection settings for each user. Two methods are available to create local node definitions: Connect:Direct Requester or the Client Connection Utility. If you want to use Connect:Direct Requester, refer to Chapter 3, *Defining and Managing the Connect:Direct Network* for instructions. This section provides instructions on using the Client Connection Utility.

## About the Client Connection Utility

The Connect:Direct for Windows client software uses the Microsoft Windows Registry to store its configuration information. The Connect:Direct Client Connection Utility allows you to update the connection settings within the Registry.

---

*Caution:*  Use the Connect:Direct Client Connection Utility to update any Registry settings rather than editing them directly.

---

You can view, edit, and update Connect:Direct for Windows connection settings in the Windows Registry with the Client Connection Utility. These settings enable communication between the user interfaces and the Connect:Direct server. You can set up and update connection settings in the following ways:

✦  Add and delete a node

✦  Add and delete a user

✦  Configure node and user properties

✦  Define a default node or user

To facilitate updating connection settings on multiple servers, you can import and export connection settings using the Client Connection Utility. After you configure the connection for a server, you can export the server's settings for use on other servers. You can then import the settings into the target server's Registry. You can also print connection settings.

---

## Starting the Client Connection Utility

To start the Client Connection Utility:

1.  Click **Start**, and then point to **Programs**.

2.  Click **Sterling Commerce Connect Direct v4.4.00 > CD Client Connection Utility**. The **Client Connection Utility** main window is displayed.

## Understanding the Tool Bar

The Client Connection Utility toolbar provides the following icons to perform frequently used actions:

| Select | To |
| --- | --- |
| | Create a new node. |
| | Create a new user. |
| | Save Registry settings. |
| | Configure highlighted node or user settings. |
| | Delete a node or user. |

# Adding and Deleting a Node Connection Definition

The Client Connection Utility lets you add new Connect:Direct nodes and identify the properties of the nodes, such as node name, TCP/IP address, and port number. These properties establish a node so you can access it from Connect:Direct Requester or the Command Line Interface (CLI).

You can also use the Client Connection Utility to delete existing nodes.

## Adding a Node

To add a new Connect:Direct node:

1.  From the **File** menu, select **New Node**. The **Node Properties** dialog box is displayed.

2. To add a node registered in the Active Directory:

   a. Select Windows in the **Operating System** field.

   b. Select the node to add from the **Active Directory Nodes** drop-down box.

   The name, address, and port fields are automatically updated with information from the Active Directory list.

   > **Note:** Click **Refresh** to update the address and port stored on the local computer with the values from the Active Directory listing.

3. To add a node that is not registered in the Active Directory:

   a. In the **Name** field, type the name of the Connect:Direct node you want to add.

   b. If necessary, change the operating system value in the **Operating System** field.

   c. In the **Address** field, type the TCP/IP address of the new node.

   d. The **Port** field automatically defaults to 1363; if necessary, type in a different port number.

4. To specify the new node as the default node, click the **Set as the default node** check box.

5. Click **OK** to save your settings and close the **Node Properties** dialog box.

6. From the **File** menu, select **Save** to save the new settings.

   > **Note:** Changes made to the node settings are *not* written to the Registry until you select **Save**.

## Deleting a Node

To delete a node:

1. In the Client Connection Utility main window, select the node you want to delete.

2. Select **Delete** from the **Edit** menu.

3. Select **Save** from the **File** menu to delete the node.

   > **Note:** Changes made to the node settings are *not* written to the Registry until you select **Save**.

The node is no longer displayed in the Client Connection Utility window.

# Adding and Deleting a User Connection Definition

Use the Connect:Direct Client Connection Utility to set up new users for a node by specifying user properties, such as the user name and password. These properties establish users of the node who can access the server through the Connect:Direct Requester or the CLI.

## Adding a User

To add a new Connect:Direct user:

1. In the **Client Connection Utility** main window, select the node where you want to add a new user.

2. From the **File** menu, select **New User**. The **User Properties** dialog box is displayed.

3. Type information into the following fields:

    ◆ **Name**—Type the name of the new user. Either type the user name as defined in the Windows setup, such as lmore or type a fully qualified user name in the UPN format, such as lmore@adtree.stercomm.com

    ◆ **Password**—Type the password defined for the user.

    ◆ **Verify Password**—Retype the password defined for the user.

4. Click the **Remember password** check box to automatically reload the password when you attach as this user.

5. Click the **Set as the default user** check box if you want the new user to be the default user for the node.

6. Click **OK**.

7. If the verification password you typed does not match the initial password, you receive a message indicating the passwords do not match when you click **OK**. Retype the verification password and click **OK**.

8. From the **File** menu, select **Save** to save the settings.

---

**Note:**   Changes made to node settings are *not* written to the Registry until you select **Save**.

---

## Deleting a User

To delete a user from the node:

1. Click the plus (+) sign next to the node containing the user you want to delete.

2. Select the user you want to delete.

3. From the **Edit** menu, select **Delete**.

4. From the **File** menu, select **Save** to delete the user.

---

**Note:**   Changes made to node settings are *not* written to the Registry until you select **Save**.

---

The user is not displayed under the node that you deleted it from.

# Updating Node and User Properties

The Connect:Direct Client Connection Utility lets you update node and user properties to maintain accurate information.

To update node properties:

1. Do one of the following:

    ◆ To update a node, highlight the node you want to configure.

    ◆ To update a user properties, highlight the user you want to configure.

2. From the **File** menu, select **Properties**. The **Node Properties** dialog box is displayed:

3. Select the fields you want to edit and make the appropriate changes.

4. Click **OK** to save your settings and return to the **Node Properties** dialog box.

5. From the **File** menu, select **Save** to save the settings.

> **Note:** Changes made to node settings are *not* written to the Registry until you select **Save**.

# Defining a Default Node or Default User

The Client Connection Utility allows you to define a default node or default user. The default node and user will be used by the Connect:Direct Requester and the CLI.

To define a default node or user:

1. Do one of the following:

    ◆ To define a default node, highlight the node.

    ◆ To define a default user, highlight the user.

2. From the **Options** menu, select **Set as Default** to set the default node or user.

3. From the **File** menu, select **Save** to save the settings.

> **Note:** Changes made to node settings are *not* written to the Registry until you select **Save**.

The default node or user is displayed in the main **Client Connection Utility** window as bold text.

# Importing and Exporting Registry Settings

The Connect:Direct Client Connection Utility allows you to import and export connection settings to a file. These settings can be saved and used on another computer or node.

## Importing Registry Settings

To import Registry settings:

1. Select the node in which to import the Registry settings.

2. From the **File** menu, select **Import**. The **Open** dialog box is displayed.

   **Note:**   Importing a Registry settings file causes all current changes to the selected node to be lost if they have not been saved.

3. Select the Registry settings file you want to import (.REX extension) and click **OK.** The imported Registry settings are applied to the node you selected.

4. From the **File** menu, select **Save** to save the settings.

   **Note:**   Changes made to node settings are *not* written to the Registry until you select **Save**.

## Exporting Registry Settings

To export Registry settings:

1. From the **Client Connection Utility** main window, select the node containing the Registry settings you want to export.

2. From the **File** menu, select **Export**. The **Save As** dialog box is displayed.

3. Name the exported Registry file with a REX extension and click **OK**. The Registry settings in the file can now be imported to another computer or node.

# Printing Registry Settings

The Connect:Direct Client Connection Utility allows you to generate a report of Registry settings.

To print the Registry settings report:

1. From the **File** menu, select **Print**. The **Print** dialog box is displayed.

2. Configure the print settings and click **OK**. A report of all Registry settings is generated.

3. From the **File** menu, select **Print Preview** to preview the Registry settings report.

4. Click **Zoom In** to enlarge the text and read the report.

5.  Click **Print** to print the report, or click **Close** to close the dialog box without printing the
    report.

---

> **Note:**   Additional node detail is provided if the node has been used at least once by the client software.

---

# Using Connect:Direct for Windows in Test Mode

You can enable test mode for production instances of Connect:Direct for Windows to perform the following functions:

✦ Test new applications and customer connections

✦ Prevent future production work from executing until testing is complete after you have terminated all active production work using the Flush Process command

✦ Resume regular production work after testing

✦ Control individual file transfers by application

✦ Enable and disable individual nodes and applications

While testing is being conducted, only Processes, particularly file transfers, involved with the testing activity are executed. No production data is transferred to applications being tested while at the same time no test data is transferred to production applications.

## Processing Flow of the Test Mode

You enable the testing mode using the **quiesce.resume** initialization parameter and specify which Connect:Direct Processes to run and not run by storing your preferences as text records in a parameter table named NDMPXTBL. A sample parameters file, NDMPXTBL.sample, is located in the /Server/samples directory. You can specify the following criteria that are used to find matches for one or more Processes to include (using the "I" command code) or exclude ("X" command code) from execution:

✦ A partial or full Process name

✦ A partial or full remote node name

✦ A partial or full Connect:Direct submitter ID and submitter node combination

In addition to telling Connect:Direct which Processes to run, you tell the system what to do with the Processes which do not get executed. You can specify the following dispositions for Processes not permitted to run:

✦ Place the Process in the Hold queue

✦ Place the Process in the Timer queue for session retry

✦ Flush the Process from the queue

For more information on how the testing mode can be used, see *Sample Test Scenarios* on page 194.

When the testing mode is enabled, Connect:Direct for Windows performs a syntax check on the parameter table and fails initialization if the table is invalid. If the table is valid, Connect:Direct for Windows scans it looking for a pattern that matches the Process that is about to execute. If a match is found, the Process is permitted to execute if the "I" (Include) command code is in effect. If command code "X" (Exclude) is in effect, the process is not permitted to execute. If a match is not found in the table, the opposite processing occurs from the case where a match is found, that is, if no match is found and command code "I" is in effect, the Process is not permitted to execute, whereas if command code "X" is in effect, the Process is permitted to execute.

If a Process is not permitted to execute, the disposition specified in the NDMPXTBL parameter table to either hold, retry, or flush the Process is implemented and a non-zero return code is returned. When a Process is prevented from executing in testing mode, appropriate messages are issued and can be viewed in the statistics log.

---

**Note:**   For Processes initiated on remote nodes, the testing mode functions in the same manner as it does for Processes submitted on the local Connect:Direct node except that the remote node is the PNODE (Process owner) for that Process, and the local node is the SNODE (secondary node). The NDMPXTBL Parameter Table is searched for a matching entry, and the remotely-initiated Process is either permitted to execute or is excluded from execution. Because the local node is the SNODE for this type of transfer, it cannot enforce the Process disposition setting in the NDMPXTBL parameter table. The remote PNODE determines how the Process is handled. Typically, the remote node places the Process in the Hold queue with a status of "HE" (Held in Error).

---

## Preparing the NDMPXTBL Parameter Table

You can use any text editor to modify the sample NDMPXTBL parameter table supplied with Connect:Direct for Windows. When you update the parameter table, name it NDMPXTBL and save it to the Server directory of the installation. The parameter table file can be created or updated while the server is active, and any changes made to the file take effect for sessions that begin after the changes are made. Similarly, the **quiesce.resume** initialization parameter can be modified while the server is active. For more information on the quiesce.resume initialization parameter, see *About the Local Node Parameters* on page 137.

Each table entry or record consists of a single-character command code in column one. Most command codes have a parameter which begins in column two and varies according to the command code function.

| Command Code | Description | Subparameters/ Examples |
|---|---|---|
| * | Comment line. | * Only run the following Processes. |
| E | Enables execution of Processes based on table entries. Either "E" or "D" must be the first non-comment entry in the table. | The second column in this entry must contain one of the following values which indicates the disposition of a PNODE Process if it is not allowed to run.<br><br>H—Places the Process in the Hold queue<br><br>R—Places the Process in the Timer queue in session retry<br><br>F—Flushes the Process from the queue |
| D | Disables the execution of all Processes regardless of the contents of the parameter table and fails Process execution with a non-zero (error) return code and message LPRX003E. Either "E" or "D" must be the first non-comment entry in the table | The parameter for command code "E" can also be specified in column two. This is a convenience to make it easier to change from "E" to "D" and vice versa without having to change column two to a blank for command code "D." |
| P | Matches Processes based on a full or partial Process name. Supports the wild card trailing asterisk (*). Can be used to enable or disable Process execution for a particular application by using naming conventions to match an application. | PCOPY—Matches a single Process<br><br>PACH*—Matches all Processes beginning with "ACH"<br><br>P*—Matches all Processes |
| N | Matches Processes based on a full or partial remote node name. Supports the wild card trailing asterisk (*). | NCD.NODE1—Matches a single remote node name<br><br>NCD.NODEA*—Matches all remote node names beginning with "CD.NODEA" N*—Matches all remote node names |
| S | Matches Processes based on a full or wild card Connect:Direct submitter ID and submitter node combination. The format is <id>@<node>. | SACTQ0ACD@TPM002—Matches a specific ID and node combination.<br><br>S*@TPM002—Matches all IDs from node TPM002<br><br>SACTQ0ACD@*—Matches ID ACTQ0ACD from all nodes<br><br>S*@*—Matches all IDs from any node. This is another way to match all Processes. |

| Command Code | Description | Subparameters/ Examples |
|---|---|---|
| I | Includes Processes for execution that match the patterns in the table which follow this command code. Either "I" or "X" must be the second non-comment entry in the table. Processes which do not match a pattern in the table **are not** executed.<br><br>**Note:** To choose which command code to use to select Processes, determine which group is smaller and use the corresponding command Code. For example, if the number of Processes to be executed is smaller than the number of Processes to exclude from execution, specify "I" as the command code and add patterns to match that group of Processes. | ER<br>I<br>NCD.BOSTON<br><br>Includes Processes for execution on the CD.BOSTON node only. Processes destined for all other remote nodes are placed in the Timer queue in session retry |
| X | Excludes from execution those Processes that match the patterns in the table which follow this command code. Either "X" or "I" must be the second non-comment entry in the table. Processes which do not match a pattern in the table *are* executed. | EH<br>X<br>DALLASOPS@*<br><br>Excludes Processes for execution submitted by the ID SDALLASOPS from any node |
| L | Last entry in table. | |

## Sample Test Scenarios

The following examples show different applications of the test mode using the NDMPXTBL parameter table to define which Connect:Direct Processes to run and not run.

### Specifying Which Processes Run

In this example, Connect:Direct executes all Processes that start with ACH or are named DITEST01 or DITEST02. All other Processes are placed in the Hold queue.

```
* Enable processing. Only permit processes matching one of the patterns
* to execute. Hold processes that don't execute.
EH
I
PACH*
PDITEST01
PDITEST02
L
```

## Specifying Which Processes to Exclude

In this example, Connect:Direct does not execute any Process that starts with ACH or is named DITEST01 or DITEST02. All other Processes are executed.

```
* Exclude matching processes.  Permit all others to execute.
EH
X
PACH*
PDITEST01
PDITEST02
L
```

## Permitting Process Execution by Secondary Node and Submitter User ID/Node

In this example, Connect:Direct executes all Processes that match one of the following criteria:

✦  The specific secondary node (SNODE) name is DI.NODE1

✦  An SNODE whose name starts with DI0017

✦  Any Connect:Direct submitter ID from node DI0049

✦  The specific Connect:Direct submitter ID SACHAPP from any node

All Processes not matching one of the above criteria are flushed from the queue.

```
* Only permit matching processes to execute.  Flush those that do not.
EF
I
NDI.NODE1
NDI0017*
S*@DI0049
SACHAPP@*
L
```

## Stopping the Test Mode

In this example, no Processes will not be executed, and a non-zero return code will be displayed, which signifies an error along with message ID LPRX003E. The remainder of the table is ignored (including the "F" code to flush Processes from the queue), and all Processes are placed in the Hold queue.

To resume testing, change the "D" command code to an "E."

```
* Execute no processes at all.  Put them in the hold queue and return.
DF
I
PACH*
PDITEST01
PDITEST02
L
```

# Glossary

## A

### Adjacent Node

An entry in the network map that defines a Connect:Direct node with which the local
Connect:Direct node can communicate. The adjacent node is also referred to as a remote node.

### Advanced Program-To-Program Communication (APPC)

The general facility characterizing the LU 6.2 architecture and its various implementations in
products.

### Application Program Interface (API)

A Connect:Direct component that accepts commands and places them in an executable format.

## C

### Checkpoint Restart

Feature that eliminates the need to re-transmit an entire file in the event of a transmission failure. If
a copy procedure is interrupted, Connect:Direct restarts that copy at the last checkpoint.

### Commands

Initiate and monitor activity within the Connect:Direct system.

### Configuration Registry

A database of configuration information central to Windows operation. The Registry is the storage
location for configuration settings for Windows NT applications. The Registry centralizes all
NT settings and provides security and control over system, security, and user account settings.

# D

## Domain Name Service (DNS)

A distributed database that provides a hierarchical naming system for identifying Internet hosts.

# L

## Local Node Record

The base record in a parameters file that defines the Connect:Direct server. It includes the most commonly used settings at a site and is the central node through which all communication is filtered. Depending upon how each remote node record is configured, trading partner node records can use settings that are defined in the local node record.

## Logical Unit (LU) 6.2

A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation.

# N

## Network Map (Netmap)

The file that identifies all valid Connect:Direct nodes in a network including a local node record and a remote node record for each trading partner. The network map also defines the rules or protocols used by each node when communicating with the local Connect:Direct node.

## Node

Any site in a network from which information distribution can be initiated.

## New Technology File System (NTFS)

A system that uses fixed disks to organize data. NTFS treats files as objects with user and system defined attributes.

# P

## Primary Node (PNODE)

The node that submits the Connect:Direct Process to the secondary node (SNODE). In every communication, you must have a PNODE and an SNODE.

## Process

A series of statements that initiate Connect:Direct activity, such as copying files and running jobs.

## Process Statements

Instructions for transferring files, running operating system jobs, executing programs, or submitting other Connect:Direct Processes. They are used to build a Connect:Direct Process.

# R

## Registry

See Configuration Registry.

## Remote Node

An entry in the network map that defines a Connect:Direct node with which the local Connect:Direct node can communicate. The remote node is also referred to as an adjacent node.

## Retry Interval

The interval at which retries are performed as a part of the checkpoint-restart feature.

# S

## Secondary Node (SNODE)

The Connect:Direct node that interacts with the primary node (PNODE) during Connect:Direct Process execution and is the noncontrolling node. Every Process has one secondary node and one primary node.

## Statistics File

Holds Connect:Direct statistics records that document the history of a Process.

## Statistics Facility

Records Connect:Direct activities.

## Systems Network Architecture (SNA)

The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

# T

## Transmission Control Queue (TCQ)

Holds information about Connect:Direct Processes that are currently executing or scheduled to execute in the future.

# Index

## A

Access Control List (ACL)  26

Accounting Data
  Process preference  53
  specifying in a command  67

Active Directory
  add an entry  116
  adding a node  115
  create a report  116
  delete a node  116

active.directory.enabled
  parameter  144

Adapter Address
  APPC
    attribute setting  41

Adapter Number
  APPC
    attribute setting  41

Additional API IP Addresses and Ports
  TCP/IP  113

Additional Host IP Address
  TCP/IP  113

Admin Tool Utility
  start  109
  stop Connect:Direct  47
  toolbar, about  109

Alarm traps
  defined  104

All Off
  turn off tracing  118

alt.comm.outbound
  remote connection parameter  35

Alternate Comminfo  35

API IP Address
  TCP/IP  112

API Port
  TCP/IP  112

APPC
  attribute
    adapter address  41
    adapter number  41
    add settings  38
    LU name  41
    Max Pacing, specify  38
    Max RU Size setting  38
    mode override  36
    partner LU name  36
    SNA network name  36, 41
    update settings  38
    VTAM APPLID  36
  protocol, specify  37

authorities
  access to Process statistics  24
  access to trace utility  25
  change Process  24
  define user  22, 23
  delete  27
  delete Process  24
  download access to directory  25
  local user, update  25
  modify  26
  monitoring Process  24
  override privileges
    ACL update  26
    CRC  26
    define  25
    execution priority  26
    file attributes  26
    remote node ID  26
  Process copy statements  24
  Process from directory, submit  25
  Process receive copy statements  24
  Process statements  24
  Process within Process  24
  program run access from directory  25
  refresh of initialization parameters  25
  restrict control functions  24
  run task Process statements  24
  stop execution  25
  submit Process  24
  update to netmap  25

# D

*Connect:Direct for Windows System Guide*

*Connect:Direct for Windows System Guide*

# T

# X