

IBM Connect:Direct for Microsoft Windows  
6.1

*Documentation*



This edition applies to Version 5 Release 3 of IBM® Connect:Direct and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1993, 2018.**

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

---

# Contents

<b>Chapter 1. Release Notes.....</b>	<b>1</b>
Requirements.....	1
Features and Enhancements.....	2
Special Considerations.....	3
Known Restrictions.....	4
Restrictions for Connect:Direct for Microsoft Windows.....	4
Restrictions for Related Software.....	6
Installation Notes.....	6
Installation Notes for Connect:Direct Requester.....	6
Installation Notes for Connect:Direct File Agent.....	7
Connect:Direct Secure Plus for Microsoft Windows.....	7
Upgrading Guidelines.....	7
Upgrading Guidelines for Connect:Direct for Microsoft Windows.....	7
Upgrading Guidelines for Related Software.....	8
<b>Chapter 2. Getting Started Guide.....</b>	<b>9</b>
Prepare for the Installation of IBM Connect:Direct for Microsoft Windows.....	9
Requirements for Copying Files To and From Network Servers.....	9
Configure Microsoft Windows User Privileges.....	9
Customize a Connect:Direct Logon Account.....	10
Configure TCP/IP Connectivity.....	10
Connect:Direct for Microsoft Windows Installation Worksheet.....	10
Install and Configure Database Software.....	11
Automate the Connect:Direct for Microsoft Windows Installation.....	12
Installing in a Windows clustered environment.....	12
Install IBM Connect:Direct for Microsoft Windows.....	12
Installation Overview.....	12
Automate Installation.....	23
About Silent Installations.....	23
Customize Initialization Information for a Silent Installation.....	24
CD_SRVR.INI Parameter Values.....	24
Customize Connect:Direct for Microsoft Windows Configuration Information.....	29
Run a Silent Installation.....	29
Perform an Unattended Uninstallation.....	31
<b>Chapter 3. System Guide.....</b>	<b>33</b>
Configure the Local Node.....	33
Configuring the Connect:Direct Local Node.....	33
Adding or Modifying a Local Node Definition .....	33
Adding a User.....	34
Modifying a User ID.....	34
Deleting a User.....	35
Attaching to a Local Node.....	35
About Local Functional Authorities.....	35
Define Remote User Proxies .....	43
Creating or Modifying a Remote User Proxy .....	43
Deleting a Remote User Proxy .....	44
Define and Manage the IBM Connect:Direct Network.....	44
Define and Manage the Connect:Direct Network.....	44
About Defining the Network Map.....	45

Define Remote Node Information.....	45
Creating or Modifying a Remote Node Definition.....	46
Creating or Modifying a Communications Path Definition.....	50
Defining a New Mode for a Communications Path.....	51
Deleting a Network Map Entry.....	52
Validating a Network Map Entry.....	53
Viewing a Network Map as Text.....	53
Applying a Network Map.....	53
Printing and Viewing Node and Network Map Definitions.....	54
View the Sample Configuration Files.....	54
Customizing Configuration Files.....	55
Adding an Encrypted Password for a User Proxy.....	55
Validating Configuration Files for Use with Connect:Direct.....	55
Applying Updated Configuration Information.....	56
Stop IBM Connect:Direct for Microsoft Windows.....	57
Stopping Connect:Direct for Microsoft Windows.....	57
Stopping Connect:Direct for Microsoft Windows from Connect:Direct Requester.....	57
Stopping Connect:Direct for Microsoft Windows Using the Services Facility.....	57
Stopping Connect:Direct for Microsoft Windows Using the CLI.....	58
Stopping Connect:Direct for Microsoft Windows from the Admin Tool Utility.....	58
Create a Process.....	58
About Processes.....	58
Establishing Preferences.....	58
Creating a Process.....	61
Commands and Statements.....	62
Process or Command Options.....	69
Setting Security Options.....	69
Setting Control Functions for a Command or Process.....	70
Assigning Values to Symbolic Variables.....	71
Specifying Accounting Data.....	72
Add Comments.....	72
Validating Process Content.....	73
Saving a Process.....	73
Copying a Process.....	73
Changing a Submitted Process.....	74
Manage Processes Using a Work List.....	74
Manage Processes.....	77
SMTP Notification.....	77
Manage Processes.....	78
Understanding the TCQ.....	78
TCQ Logical Queues.....	78
View Processes in the TCQ.....	81
Creating a Process Monitor .....	81
Monitoring Processes Based on Selection Criteria.....	81
Opening a Process Monitor File.....	83
Saving a Process Monitor.....	83
The Process Monitor Output.....	83
Using the Output Display.....	85
Notification.....	85
NT Broadcast.....	85
Changing Process Notification.....	85
View Process Statistics.....	86
The Statistics Monitor Window.....	87
Delete Statistics Records.....	87
Control Statistics File Content.....	87
Creating a Statistics Report.....	88
Selecting Statistics Based on Process Name or Number.....	88
Statistics Report Output.....	90

Understand the Microsoft Windows Event Logging Facility.....	92
Control Event Log Content.....	93
Filter the Event Log.....	93
Viewing Messages.....	93
Use the Activity Log.....	94
Opening an Activity Log.....	94
Saving an Activity Log.....	94
Manage an IBM Connect:Direct Server.....	94
Manage a Connect:Direct Server.....	94
Starting the Admin Tool Utility.....	94
About the Toolbar.....	95
Starting and Stopping a Connect:Direct Server.....	95
Configuring a IBM Connect:Direct Server.....	95
Work with Active Directory.....	99
Adding an Active Directory Entry.....	99
Deleting an Active Directory Entry.....	99
Creating an Active Directory Report.....	99
Troubleshoot IBM Connect:Direct.....	100
Diagnose a Server Problem Using Traces.....	100
Defining a Trace.....	100
Stopping a Trace.....	101
Trace Startup Parameters.....	101
Recover from a Problem.....	103
Process Step Restart.....	103
Automatic Session Retry.....	103
Checkpoint/Restart.....	104
Restart During Run Task Operations.....	104
Troubleshoot Connect:Direct Problems.....	105
Worksheets.....	105
Network Map Communications Mode Object Worksheet.....	105
Network Map Communications Path Object Worksheet.....	105
Network Map Remote Node Object Worksheet.....	106
User Functional Authorization Worksheet.....	106
Remote User Proxy Worksheet.....	108
Change IBM Connect:Direct Settings.....	109
Change Connect:Direct for Microsoft Windows Settings.....	109
Sample Initialization Parameters Format.....	109
Changing Initialization Parameters.....	111
Specify an IP Address.....	129
Specify IP Addresses, Host Names, and Ports.....	129
Submit a Process Using the Command Line Interface.....	132
Submit Processes Using the CLI Command.....	132
Creating a Configuration File to Connect to a Server.....	132
Invoke the CLI.....	133
Terminating the CLI.....	134
CLI Commands.....	134
Command Syntax.....	134
Piping Conventions.....	135
Submit Process Command.....	136
Change Process Command.....	140
Delete Process Command.....	142
Select Process Command.....	143
Select Message Command.....	147
Select Statistics Command.....	148
Traceoff Command.....	155
Traceon Command.....	156
Help Command.....	158
Stop Connect:Direct.....	159

Modify Translation Tables.....	160
Translation Tables.....	160
Modify a Translation Table Using Connect:Direct Requester.....	160
Edit Connection Settings.....	161
About the Client Connection Utility.....	161
Start the Client Connection Utility.....	161
Add a Node.....	161
Deleting a Node.....	162
Adding a User with Client Connection Utility.....	162
Deleting a User with the Client Connection Utility.....	163
Updating a Node or User.....	163
Defining a Default Node or User.....	163
Importing Registry Settings.....	164
Exporting Registry Settings.....	164
Printing Registry Settings.....	164
Use IBM Connect:Direct for Microsoft Windows in a Test Mode.....	165
Use Connect:Direct in Test Mode.....	165
Preparing the NDMPXTBL Parameter Table.....	166
Sample Test Scenarios.....	168
Client API connections.....	169
Authenticating client connection.....	169
Implementing Client Authentication.....	169
Configuring Connect:Direct Windows for Authentication Management.....	170
Certificate Authentication for Client API Connections.....	170

## **Chapter 4. Using FASP with IBM Aspera High-Speed Add-on for Connect:Direct for Microsoft Windows (V4.8.0 or later) ..... 173**

Activating FASP.....	173
Licensed bandwidth for FASP transactions.....	173
Using Connect:Direct for Microsoft Windows with IBM Aspera High-Speed Add-on and Secure Proxy.....	174
Configuring FASP.....	175
FASP Process Language.....	179
FASP Messages.....	180
Monitoring FASP transactions.....	181
Known Limitations.....	181

## **Chapter 5. Secure Plus Option Implementation Guide..... 183**

Overview.....	183
About Connect:Direct Secure Plus.....	183
Security Concepts.....	183
Connect:Direct Secure Plus Tools.....	185
Plan the Connect:Direct Secure Plus Configuration.....	186
Summary of Processing Using Connect:Direct Secure Plus.....	187
IBM Connect:Direct Secure Plus for Microsoft Windows Documentation.....	188
Set Up Connect:Direct Secure Plus.....	189
Set Up Connect:Direct Secure Plus.....	189
Start Secure+ Admin Tool.....	189
Prepare to Set Up Connect:Direct Secure Plus.....	189
Populate the Connect:Direct Secure Plus Parameters File.....	190
Configure Nodes.....	190
Node Configuration Overview.....	190
Import Existing Certificates.....	191
Create CMS Key Store .....	192
Configure the Connect:Direct Secure Plus .Local Node Record.....	193
Configure Connect:Direct Secure Plus Remote Node Record.....	196
Validate the Configuration.....	198

Enable or Disable External Authentication for a Remote Node.....	199
Configure External Authentication in the .SEAServer Record.....	199
Automate Setup Using the CLI.....	200
Start and Set Up the Connect:Direct Secure Plus CLI.....	200
Use LCU Files to Encrypt Passwords for Use with the Connect:Direct Secure Plus CLI.....	201
Sample Scripts.....	201
Manage the Parameters File.....	202
Manage CMS Keystore .....	204
Update the .Local Node Record.....	205
Manage Remote Node Records.....	206
Update the .Client Node Record.....	210
Manage the External Authentication ServerRecord.....	211
Maintain the .Password File (Strong Password Encryption).....	211
Maintain Connect:Direct Secure Plus.....	212
Connect:Direct Secure Plus Node List.....	212
View Connect:Direct Secure Plus Parameters File Information.....	213
View Connect:Direct Secure Plus Node Record Change History.....	213
Disable Connect:Direct Secure Plus.....	214
Delete a Connect:Direct Secure Plus Remote Node Record.....	214
Resecure Connect:Direct Secure Plus Parameters and Access Files.....	214
View Statistics.....	215
View Statistics.....	215
Audits.....	217
Connect:Direct Secure Plus Parameters File Auditing.....	217
Connect:Direct Secure Plus Certificate Auditing.....	219
Troubleshoot Connect:Direct Secure Plus.....	220
Troubleshooting.....	220
Configuration Worksheets.....	223
Local Node Security Feature Definition Worksheet.....	223
Remote Node Security Feature Definition Worksheet.....	223
Certificate File Layout.....	224
Certificate File Layout.....	224
Validate the Configuration.....	225
Exchange Data and Verify Results.....	225
Automation Scripts.....	225
Configure Connect:Direct Secure Plus to Use the TLS Protocol.....	225
Use LCU to Configure Encrypted Passwords.....	227
Configure Encrypted Passwords Using the LCU.....	227
Create an LCU File.....	228

**Chapter 6. SDK Programmers Guide..... 231**

Overview.....	231
Connect:Direct for Microsoft Windows SDK Overview.....	231
Edit Connection Settings.....	232
Edit Connection Settings with the Client Connection Utility.....	232
Start the Client Connection Utility.....	233
Add and Delete Node Connection Definitions.....	233
Add a Node.....	233
Delete a Node.....	234
Add a User.....	234
Delete a User.....	235
Update Node Properties.....	235
Define a Default Node or Default User.....	236
Import Registry Settings.....	236
Export Registry Settings.....	237
Print Registry Settings Report.....	237
Apply the C API.....	237

The C Applications Programming Interface.....	237
Compile and Debug.....	238
Activate Tracing.....	238
Standard C API.....	238
View Sample Programs.....	239
Apply the C++ Class Interface.....	240
Compile and Debug.....	240
Manipulate Nodes.....	240
Create an Object to Connect to a Node.....	241
Manage Connections.....	242
View Information.....	242
Control the Return of Information.....	242
Execute Connect:Direct Commands.....	243
Manage Exception Conditions.....	245
Manage Administrative Functions.....	246
Multithreaded Access and Blocking.....	247
Objects On The Stack.....	247
Apply the ActiveX Control Interface.....	248
Submit Process.....	248
Display Select Statistics Results.....	249
Apply Automation Servers.....	255
Apply Automation Servers.....	255
Create Virtual Servers Using the Node Factory.....	255
Use Automation Objects.....	259
Enhance Security and Automate File Opening with User Exits.....	260
User Exits.....	260
Apply Enhanced Security.....	261
Apply Automated File Opening.....	262
Password Exit.....	263
Structure Types.....	266
Structure Types.....	266
NETMAP_DESC_STRUCT Structure.....	266
USER_STRUCT Structure.....	266
MESSAGE_STRUCT Structure.....	268
NETMAP_MODE_SNA Structure.....	269
NETMAP_MODE_TCP Structure.....	269
NETMAP_NODE_STRUCT Structure.....	270
NETMAP_PATH_STRUCT Structure.....	271
PROCESS_STRUCT Structure.....	272
NODE_STRUCT Structure.....	274
STATISTICS_STRUCT Structure.....	275
TRACE_STRUCT Structure.....	276
TRANSLATE_STRUCT Structure.....	277
Return Codes.....	277
C++ Class and the C API Functions Return Codes.....	277
<b>Chapter 7. .Net SDK User Guide.....</b>	<b>281</b>
Connect:Direct for Microsoft Windows .Net SDK Overview.....	281
Sample Programs.....	281
Add the .Net Class Interface.....	281
About Classes.....	282
Connect to a Connect:Direct for Microsoft Windows Node.....	282
Disconnect Nodes.....	282
Submit Processes.....	283
Manage Processes.....	283
Retrieve Statistics.....	284
Node Properties .....	285



Process Class.....	285
Process Properties.....	285
Statistic Class.....	286
Statistic Properties.....	287
<b>Notices.....</b>	<b>289</b>
Trademarks.....	290
Terms and conditions for product documentation.....	291



# Chapter 1. Connect:Direct for Microsoft Windows Release Notes

The IBM® Connect:Direct® for Microsoft Windows Release Notes document supplements Connect:Direct for Microsoft Windows documentation and the documentation for the following Connect:Direct for Microsoft Windows related software: Connect:Direct Requester, Sterling Connect:Direct File Agent, and Connect:Direct Secure Plus for Microsoft Windows. Release notes are updated with each release of the product and contain last-minute changes and product requirements. Read the document before installation.

## Requirements

Connect:Direct for Microsoft Windows has the following requirements.

### Hardware and Software

Connect:Direct for Microsoft Windows and related software require the following hardware and software.

Component	Hardware	Software
Connect:Direct for Microsoft Windows	<ul style="list-style-type: none"> <li>• 512 MB RAM (min.) 1 GB or more recommended</li> <li>• 1 GB disk space*</li> </ul>	64-bit Microsoft Windows operating system options: <ul style="list-style-type: none"> <li>• Microsoft Windows Server 2019 with Desktop Experience</li> <li>• Microsoft Windows Server 2016 with Desktop Experience</li> <li>• Microsoft Windows Server 2012 R2</li> <li>• Microsoft Windows 10</li> <li>• Clustered environment supported on:               <ul style="list-style-type: none"> <li>– Microsoft Windows Server 2012 R2</li> <li>– Microsoft Windows Server 2016</li> </ul> </li> </ul>
Microsoft TCP/IP Support		Microsoft TCP/IP WinSOCK interface installed as part of the Microsoft Windows system
Database Software	2 GB or more. The amount may vary depending on the product configuration and usage.	Install one of the following before you install Connect:Direct for Microsoft Windows: <p><b>Note:</b> PostgreSQL is the default database provided and installed with IBM Connect:Direct from Microsoft Windows release 6.1</p> <ul style="list-style-type: none"> <li>• Microsoft SQL Server 2016 (and future Fix Pack) are supported. This software is not provided. You can configure SQL Server during the IBM Connect:Direct installation.</li> </ul> <p><b>Note:</b> Local-domain access must be available to Microsoft SQL Server.</p> <p>For more information, see the <i>IBM Connect:Direct for Microsoft Windows Getting Started Guide</i>.</p>

Component	Hardware	Software
Connect:Direct File Agent	Same as requirements for Connect:Direct for Microsoft Windows	Same requirements as Connect:Direct for Microsoft Windows
IBM Connect:Direct Requester	Same as requirements for Connect:Direct for Microsoft Windows	Same requirements as Connect:Direct for Microsoft Windows
Connect:Direct for Microsoft Windows SDK	Same as requirements for Connect:Direct for Microsoft Windows	This software is required to build the samples, but is not required to run the samples.

\* When upgrading Connect:Direct for Windows through Control Center Director, an extra 3 GB is required for temporary storage.

## Virtualization and public cloud support

IBM cannot maintain all possible combinations of virtualized platforms and cloud environments. However, IBM generally supports all enterprise class virtualization mechanisms, such as VMware ESX, VMware ESXi, VMware vSphere, Citrix Xen Hypervisor, KVM (Kernel-based virtual machine), and Microsoft Hyper-V Server.

IBM investigates and troubleshoots a problem until it is determined that the problem is due to virtualization. The following guidelines apply:

- If a specific issue is happening because the system is virtualized and the problem cannot be reproduced on the non-virtualized environment, you can demonstrate the issue in a live meeting session. IBM can also require that further troubleshooting is done jointly on your test environment, as there is not all types and versions of VM software installed in-house.
- If the issue is not able to be reproduced in-house on a non-virtualized environment, and troubleshooting together on your environment indicates that the issue is with the VM software itself, you can open a support ticket with the VM software provider. IBM is happy to meet with the provider and you to share any information, which would help the provider further troubleshoot the issue on your behalf.
- If you chose to use virtualization, you must balance the virtualization benefits against its performance impacts. IBM does not provide advice that regards configuring, administering, or tuning virtualization platforms.

## Features and Enhancements

IBM Sterling Connect:Direct for Microsoft Windows version 6.1 and its related software have the following features and enhancements:

FixPack 2 (v6.1.0.2)

New Features and Enhancements
<p>To install this software, you should go to the <a href="#">Fix Central</a> website and install the latest available fix pack.</p> <ul style="list-style-type: none"> <li>• With this fix pack, support for configurable <b>Password Exit</b> feature is extended, which will eliminate the need to store the passwords. For more information on initialization parameters, refer to <a href="#">“Changing Initialization Parameters”</a> on page 111 and <a href="#">“CD_SRVR.INI Parameter Values”</a> on page 24.</li> </ul> <p><b>Note:</b> You are responsible for configuring the password vault software securely. For more information, refer to <a href="#">“Password Exit”</a> on page 263.</p> <ul style="list-style-type: none"> <li>• An option to enable and disable the feature from <a href="#">User Authority</a>, <a href="#">Group Authority</a> and <a href="#">Proxy</a> is also provided.</li> </ul>

### New Features and Enhancements

To install this software, you should go to the [Fix Central](#) website and install the latest available fix pack. IBM Connect:Direct for Windows introduces support for installing new Connect:Direct servers from [IBM Control Center Director](#). For more information see, [“IBM Sterling Control Center Director Support” on page 20](#).

### New Features and Enhancements

To install this software, you should go to the [Passport Advantage](#) website, and follow instructions described to complete the download.

- With this release Connect:Direct for Microsoft Windows introduces support to cache certificate validation responses from External Authentication Server when it interfaces External Authentication Server during an SSL/TLS session. This minimizes the overhead associated with requesting certificate validation from External Authentication Server, thus eliminating the need for Connect:Direct Secure Plus to query External Authentication Server each time. For related documents see. For related documents see:
  - [Parameters File>.SEAS Server](#)
  - [Enable Caching SEAS certificate validation response via Connect:Direct Secure Plus Admin Tool](#)
  - [Enable Caching SEAS certificate validation response via Connect:Direct Secure Plus CLI](#)
- Support for TLS v1.3 for Connect:Direct for Microsoft Windows introduced to secure communication sessions with traders partners. For more information see, [Secure Plus Option Implementation Guide](#).
- This release makes impersonation optional, allowing processes to be executed under the same Service account that Connect:Direct for Windows is running. To enable this feature, go to:
  - **User Proxies > Directories>** select **Allow process to run using Service Account**. In **User Function Authorities**, the setting is available on the **Main** panel.
  - **Functional Authorities Users Main panel** > select **Allow process to run using Service Account**

For more information, see [“Defining User Authority” on page 37](#).

**Note:** If impersonation is disabled, the account under which Connect:Direct runs (Connect:Direct’s Service Account) must have appropriate access to the source files and destination folders used when Connect:Direct transfers files.

- Local functional authority type template is now updated to include a new **Operator** user type. An **Operator** user has Read-only permissions to view configuration files and monitor file transfers but not modify, delete or submit a process. For more information see, [“About Local Functional Authorities” on page 35](#).

**Note:** To support this functionality both, IBM Connect:Direct Requester and Connect:Direct for Microsoft Windows server must be upgraded to fix pack v6.0.0.2 or above.

Default installation folder has changed:

```
C:\Program Files\IBM\Connect Direct v6.1.0
```

## Special Considerations

This section details special considerations to be aware of for your platform.

- Certain maintenance fixes should be applied to IBM Connect:Direct for z/OS to have the correct TLS protocol negotiation with Connect:Direct for Microsoft Windows. It is suggested the you upgrade IBM Connect:Direct for z/OS with these four HIPER fixes: UI14876, UI14924, UI16043, UI16936. For

additional information, see <http://www14.software.ibm.com/webapp/set2/psearch/search?domain=psp> and search for the FMID for CDZ/5.3 (HDGA520) and select Upgrade STRCD520, Subset HDGA520, then select Service Recommendations for the most current list of HIPER fixes.

- The database retry feature retries a connection for up to eight minutes before a failure is allowed to continue. When you use client applications like IBM Connect:Direct Requester, the application may appear to hang. If this occurs, check the event log for database errors indicating a retry is in progress. If the retry is unsuccessful, you may need to recycle the Connect:Direct for Microsoft Windows server.
- PostgreSQL

If you do not install PostgreSQL as your database and would like to install it at a later time, re-run **cdw\_install.exe**. You can also install the PostgreSQL feature.

## Known Restrictions

---

Connect:Direct for Microsoft Windows and its related software have the following known restrictions.

### Restrictions for Connect:Direct for Microsoft Windows

Connect:Direct for Microsoft Windows version 6.1 has the following restrictions:

- A password-less proxy can only be added or configured by the Windows Administrator or members of the local Administrators group.
- While importing `Initparm.cfg`, validation of Password Exit DLL file and Password Exit Hash is restricted to length check.
- In silent installation, validation of Password Exit Hash is limited to length check.
- There is no GUI support for creating ECDSA signed certificates.
- You can keep an earlier version of Connect:Direct for Microsoft Windows on the computer on which you are installing Connect:Direct for Microsoft Windows version 6.1.
- Connect:Direct for Microsoft Windows version 6.1 NT Broadcast do not send messages on 64-bit operating systems.
- Built-in variables should only be specified in a SUBMIT statement within a Process if the statement will be executed on a Connect:Direct for Microsoft Windows version 4.6 (or later) node or another IBM Connect:Direct version that supports built-in variables.
- Temporary addresses, which are a security feature of the IPv6 protocol, are generated automatically by the operating system and are used only for outbound connections. These addresses have a short life span and are replaced by other temporary outbound addresses. This feature of the IPv6 protocol causes problems with Netmap Checking. If the outgoing address of the PNODE randomly changes and netmap checking is enabled by the SNODE, the SNODE always refuses the connection because the IP address of the PNODE never matches the IP address configured for it.

You can work around the problem created by temporary addresses in two ways:

- On the PNODE, configure **outgoing.address** in the initialization parameters file using the IPv6 address for the PNODE server. This ensures that the IP address that the PNODE uses to create a connection to a remote node is always constant. Consider the following:
  - If a PNODE has several IP addresses configured, for example, two IPv6 addresses and two IPv4 addresses, configure the **outgoing.address** initialization parameter with one IPv6 address. This address can then be used to connect to an SNODE configured with either IPv6 or IPv4 addresses.
  - If a PNODE wants to use an IPv4 address to connect to an SNODE that has both IPv6 and IPv4 IP addresses, ensure that the **tcp.api.port** and **tcp.host.port** initialization parameters of the SNODE are configured with an IPv4 address and port.
- Disable temporary addresses for the PNODE. This is a configuration option in the Windows networking component. If the temporary addresses are not generated, connections to a remote that use the IPv6 protocol use the configured IPv6 address.

**Note:** To disable temporary addresses in Windows operating systems, see the *Microsoft Windows documentation*.

See *RFC 3041* for more information on IPv6 temporary addresses.

- If you modify user authorizations from the IBM Connect:Direct server and the IBM Connect:Direct Requester is attached, you must detach and reattach to the IBM Connect:Direct server. When you reattach to the IBM Connect:Direct server, IBM Connect:Direct Requester reads the updated user information.
- Connect:Direct for Microsoft Windows previously supported the DESKTOP(YES) parameter in the SYSOPTS statement of a IBM Connect:Direct Process. This parameter enabled user programs launched by the IBM Connect:Direct service to interact with the Windows desktop. Currently this parameter functions only on versions of Microsoft Windows prior to Windows Vista and Windows Server 2008. For security reasons, Microsoft has removed support for Interactive Services from those two operating systems. Microsoft blocks any attempt by a Windows service to interact with the desktop. IBM Connect:Direct administrators should begin to remove the DESKTOP(YES) parameter from all Connect:Direct for Microsoft Windows Process scripts. Alternatively, you can switch DESKTOP(YES) references to DESKTOP(NO).

To ease the transition of upgrading to Connect:Direct for Microsoft Windows, IBM Connect:Direct detects when a process using DESKTOP(YES) is submitted on a Windows system that does not support Interactive Services. When DESKTOP(YES) is detected, IBM Connect:Direct dynamically switches to DESKTOP(NO) and records the following warning in the statistics:

```
LPRS020I Invalid DESKTOP value specified.  
DESKTOP=YES is not supported on this version of Windows.  
The RUN TASK / JOB will continue with DESKTOP reset to NO.
```

After this warning is written to IBM Connect:Direct statistics, the Process is allowed to continue as if DESKTOP(NO) had been originally specified.

This transitional feature works only if the RUN TASK or RUN JOB is capable of running without desktop interaction. That is, if manually switching DESKTOP(YES) to DESKTOP(NO) would cause the IBM Connect:Direct Process to fail, then the dynamic switch to DESKTOP(NO) will not be an effective solution. If the program executed by the RUN TASK/JOB is unable to execute without user interaction then that program must be changed so that it no longer needs user interaction.

- Local transfers on the same computer, such as PNODE-SNODE could fail with error FASP041E: FASP initialization failed. FASP is not very well suited for this kind of a network connection. Use TCP/IP for local transfers instead.
- Connect:Direct for Microsoft Windows version 6.1 does not support Load Balanced environment and has not been tested or certified in a Load Balanced environment.

## Restrictions for Related Software

The related software has the following restrictions:

Related Software	Restriction
IBM Connect:Direct Requester	<p>The following functions are not supported for IBM Connect:Direct Requester connected to a Connect:Direct for UNIX or Connect:Direct for UNIX OpenVMS server:</p> <ul style="list-style-type: none"><li>• Server initialization parameters (initparms) maintenance</li><li>• Network map maintenance</li><li>• User authorization management</li><li>• Remote user proxies</li><li>• List all users</li><li>• New translation table</li><li>• Selecting statistics by copy file name (No statistics are available for OpenVMS servers.)</li></ul>
	<p>The following Trace facility options are not supported for IBM Connect:Direct Requester connected to Connect:Direct for UNIX for or IBM Connect:Direct OpenVMS servers:</p> <ul style="list-style-type: none"><li>• Ability to wrap file</li><li>• Ability to set maximum file size</li><li>• Ability to do MAIN trace</li><li>• Ability to trace by Process number, Process name, or destination (advanced options)</li></ul>

## Installation Notes

Before you install IBM Connect:Direct and its related applications, read all the information in this section and follow all the guidelines.

- Complete any worksheets before installing IBM Connect:Direct and its related software.
- Review your security configuration to ensure compatibility with IBM products.
- Verify that you have the current updates for Connect:Direct for Microsoft Windows, IBM Connect:Direct Requester, Connect:Direct for Microsoft Windows File Agent, and Connect:Direct Secure Plus for Microsoft Windows.
- Review the *IBM Connect:Direct for Microsoft Windows Getting Started Guide*.

### Installation Notes for Connect:Direct Requester

Before you install Connect:Direct Requester, complete the following pre-installation tasks:

- Define users for Microsoft Windows.
- Configure and test TCP/IP connectivity by configuring a valid IP address, IP subnet, and IP gateway, if necessary.

If the communications partner is on another subnet or network and a firewall is between the communications partner and the computer, verify that the Access Control Lists (ACLs) are correctly configured to allow access to and from the IP address and IP ports. Contact the security administrator for more information on configuring the ACLs.



## Installation Notes for Connect:Direct File Agent

Before you install Connect:Direct File Agent, read information in this section and follow the guidelines.

- Review the information on using and configuring Connect:Direct File Agent in *IBM Connect:Direct for Microsoft Windows Getting Started Guide*.
- When you install Connect:Direct File Agent initially or upgrade to a later version, it is installed as a service on Microsoft Windows using the Local System Account. If you change the user account for Connect:Direct File Agent after installation, each time you perform an upgrade, you must change from the Local System Account to a different user account for the service, if necessary.

## Connect:Direct Secure Plus for Microsoft Windows

Before you install Connect:Direct Secure Plus for Microsoft Windows, read all the information in this section and follow the guidelines.

- Print and review *IBM Connect:Direct Secure Plus for Microsoft Windows Implementation Guide*.
- To install Connect:Direct Secure Plus for Microsoft Windows at the same time that you install Connect:Direct for Microsoft Windows, follow the instructions in *IBM Connect:Direct for Microsoft Windows Getting Started Guide*.
- When you upgrade from a previous version of IBM Connect:Direct Secure Plus for Microsoft Windows, the parameters file is converted and can be used with the new version.

## Upgrading Guidelines

---

Observe the following guidelines for upgrading Connect:Direct for Microsoft Windows and its related software.

### Upgrading Guidelines for Connect:Direct for Microsoft Windows

Observe the following guidelines:

- Before you perform any upgrade procedure, create backup copies of the following Connect:Direct for Microsoft Windows installation files:
  - If you are upgrading from v6.0.0 to Release 6.1, the directory should be: `C:\Program Files\IBM\Connect Direct v6.0.0`
  - If you are upgrading from v4.8.0 or later to Release 6.1, the directory should be:  
`C:\Program Files (x86)\IBM\Connect Direct v4.8.0`
  - If you are upgrading from 4.7.0 or earlier to Release 6.1, the Directory should be:  
`C:\Program Files (x86)\Sterling Commerce\Connect Direct v4.n.n`, where n.n represents your current version number.
  - Registry location for 6.n.n registry location: Where **n.n** represents your version number.

#### 64-bit OS

```
HKEY_LOCAL_MACHINE\SOFTWARE\Sterling Commerce\Connect:Direct for Windows NT\v6.n.n
```

- Registry location for 4.7.0 and 4.8.0: Where **n.n** represents your version number.

#### 32-bit OS

```
HKEY_LOCAL_MACHINE\SOFTWARE\Sterling Commerce\Connect:Direct for Windows NT\v4.n.n
```

#### 64-bit OS

HKEY\_LOCAL\_MACHINE\SOFTWARE\WOW6432Node\Sterling Commerce\Connect:Direct for Windows NT  
\v4.n.n

- If you are upgrading from an earlier version of Connect:Direct for Microsoft Windows, the installation automatically copies the network map and user definitions.
- Microsoft Installation (MSI) does not recognize the Install Shield installations; therefore, Install Shield and MSI cannot point to the same installation folder or you may have problems uninstalling Connect:Direct for Microsoft Windows. If you upgrade Connect:Direct for Microsoft Windows, the installation copies the folders of the older version of Connect:Direct for Microsoft Windows to the new installation directory before the new version is installed. The new version overwrites existing files in the destination folder.
- The ODBC DSN is updated to include new DSN names to prevent previous uninstalls from removing them. DBWiz updates the configuration of SQL Server databases.

**Note:** In-place upgrades are not supported.

## Upgrading Guidelines for Related Software

Observe the following additional guidelines:

Related Software	Guideline
IBM Connect:Direct Requester	The installation has changed from a single-user installation to all users. Any user who is logged on can start IBM Connect:Direct Requester. Because the configuration is user specific, IBM Connect:Direct Requester automatically creates the base registry key if one does not exist.
	Microsoft Installation (MSI) does not recognize Install Shield installations; therefore, Install Shield and MSI cannot point to the same installation folder or you may have problems uninstalling Connect:Direct for Microsoft Windows. If you upgrade IBM Connect:Direct Requester, the installation copies the folders of the older version of IBM Connect:Direct Requester to the new installation directory before the new version is installed. The new version overwrites existing files in the destination folder.
Connect:Direct File Agent	When you upgrade Connect:Direct File Agent, it is installed as a service on Microsoft Windows using the Local System Account. If you change the user account for Connect:Direct File Agent after installation, each time you perform an upgrade, you must change from the Local System Account to a different user account for the service.

---

## Chapter 2. Getting Started Guide

### Prepare for the Installation of IBM Connect:Direct for Microsoft Windows

---

Before you install Connect:Direct for Microsoft Windows, make sure the following tasks are complete:

- Make sure system meets product hardware and software requirements
- Define users for Microsoft Windows
- Configure and test TCP/IP connectivity as needed
- Install and configure Microsoft SQL Server when not using the default PostgreSQL database support.
- If you plan to use NT Broadcast to send user notification messages, ensure that NT Broadcast is configured
- Complete installation worksheets

### Requirements for Copying Files To and From Network Servers

When you are ready to create Processes, be aware that Connect:Direct allows you to share information across computers. All of the data does not have to be on the server. Observe the following requirements to enable copying files among network servers:

- Files to copy must reside on a file server accessible by Connect:Direct for Microsoft Windows.
- You must provide a valid user ID and password for the file server where the files reside. Use a local node ID when you submit a Process on the local node and a remote node ID when you submit a Process on a remote node. Supply information as part of the process statement or the submit command or use the Login Connection Utility to provide this information.
- If a Process has multiple copy steps, the local node ID or remote node ID parameter must be appropriate for each file server from which, or to which you transfer a file. This method is necessary because the parameters apply to the Process as a whole and not to individual copy steps.
- Specify the Universal Naming Convention (UNC) form of the file name if the file is not on a drive directly connected to the same Microsoft Windows server as Connect:Direct. If the file is on the Microsoft Windows server where Connect:Direct is installed, you can specify the drive letter. The UNC name format is:

```
\\servername\sharename\filename
```

- The servername is the Microsoft Windows server where data resides.
- The sharename is the name under which the remote Microsoft Windows server shares the directory you want to access.
- The filename specifies the name of the file and any subdirectories.

### Configure Microsoft Windows User Privileges

Connect:Direct for Microsoft Windows must be installed by a Microsoft Windows administrator. However, ongoing administration requires that the administrator be a member of the Microsoft Windows Users group and is defined in the Connect:Direct User Authorities with administrative privileges based on the admin template.

After Connect:Direct for Microsoft Windows is installed, attach to Connect:Direct for Microsoft Windows as sysadmin and add a user as a Connect:Direct for Microsoft Windows administrator (for example,

cdadmin) with the user right, Log on locally. Then, delete sysadmin from the Connect:Direct for Microsoft Windows User Authorities.

Refer to Microsoft Windows system documentation for instructions on setting up an administrator account. Refer to the *IBM Connect:Direct for Microsoft Windows System Guide* for information on configuring user authorities.

## Customize a Connect:Direct Logon Account

### About this task

Connect:Direct for Microsoft Windows is installed under the local System account.

To create a custom service account, assign the account the following privileges:

- Log on as service—Set this privilege to allow a Microsoft Windows service to run in the context of the specified user instead of running in the context of the local system account.
- Replace a process level token—Turn on this privilege to allow Connect:Direct for Microsoft Windows to submit Processes on behalf of logged on users.
- Be a member of the Local Administrator Group—If you want to allow the node to update its entry in Active Directory, the account must also be a member of the Enterprise Admin group.
- Connect:Direct Secure Plus directory—Give the account full permissions to the Connect:Direct Secure Plus directory.

**Note:** These privileges are stored locally, even if the computer is a member of a domain. As a result, privileges cannot be set on the domain controller and granted to all computers on the domain.



**CAUTION:** Enabling the option, Allow service to interact with desktop when running Connect:Direct under the local System account, presents a security risk and may allow access to services that interact with the desktop.

After you create the account, you assign it as the account for Connect:Direct. To identify the custom logon account:

### Procedure

1. Select **Start > Settings > Control Panel > Administrative Tools > Services**.
2. Double-click the Connect:Direct server.
3. Click the **Log On** tab.
4. Select **This account to identify the custom logon account**.
5. Type the account name to use for logging onto Connect:Direct, or click **Browse** and double-click the user account.
6. Type the password in the **Password** and **Confirm password** fields.
7. Click **OK**.

## Configure TCP/IP Connectivity

To enable TCP/IP connectivity, configure each computer with a valid IP address, IP subnet, and IP gateway. If the communications partner is on another subnet or network and a firewall is between the communications partner and the Microsoft Windows computer, verify that the Access Control Lists (ACLs) are correctly configured. Contact your Firewall/Security Administrator for more information on configuring the ACLs for your firewall.

## Connect:Direct for Microsoft Windows Installation Worksheet

Complete this worksheet before you install Connect:Direct for Microsoft Windows.

Decision	Choices	Additional Information
What Connect:Direct software to install?	<ul style="list-style-type: none"> <li>• Connect:Direct Requester</li> <li>• Connect:Direct Server</li> <li>• Connect:Direct Secure Plus</li> </ul>	
Choose the installation directory (where X represents the drive letter.)	<ul style="list-style-type: none"> <li>• Accept the default installation directory - C:\Program Files\IBM\Connect Direct v6.1.0</li> <li>• Change the installation directory</li> </ul>	
Name the Connect:Direct node - The default local node name is the name of the Microsoft Windows computer.		<b>Note:</b> To change the local node name after the product is installed, you must reinstall Connect:Direct for Microsoft Windows.
Identify the database software	<ul style="list-style-type: none"> <li>• PostgreSQL</li> <li>• Microsoft SQL Server</li> </ul>	If security is implemented on Microsoft SQL Server—Microsoft Windows Authentication, create a valid user ID and password and a CREATE TABLE privilege within the database.
Identify TCP/IP communication information		Identify Node-to-Node IP Address, Node-to-Node Port, User Interface IP Address, User Interface Port
Identify additional components	<ul style="list-style-type: none"> <li>• Process Notification. Specify NT Broadcast or SMTP. If SMTP, specify Host Address, Host Port (default is 25), and Sender.</li> <li>• Load an Existing Network Map. Identify the fully qualified pathname for the MAP.CFG.</li> <li>• Load Existing User Authorities. Identify the Fully qualified pathname for the USER.CFG.</li> <li>• Load Existing Initialization Parameters. Identify the Fully qualified pathname for the Initparms.cfg.</li> </ul>	

## Install and Configure Database Software

The database logs Process statistics, internal messages, and the Process control queue. Determine which software to use as the database software.

- To use PostgreSQL, accept the default database option during the Connect:Direct for Microsoft Windows installation.
- To use SQL Server, install it on the local computer, a network drive, or remote computer accessible from the local Connect:Direct node.

## Automate the Connect:Direct for Microsoft Windows Installation

You can automate Connect:Direct for Microsoft Windows installations using an initialization (INI) file. Implement automated installations by specifying a path to the INI file from the command line of the Microsoft Windows setup. See [“About Silent Installations” on page 23](#) for information about automating installations.

## Installing in a Windows clustered environment

---

For instructions on installing Connect:Direct for Microsoft Windows in a Windows clustered environment, see [Deploying Connect:Direct in a Windows Failover Cluster](#)

**Note:** When installing in a clustered environment, you must deselect the Requester feature (do not install it at this time). Instead, do one of the following:

- Install the Connect:Direct Requester (Stand-Alone) using the CDRequester cdw\_install.exe. See [Installation Notes for Connect:Direct Requestor](#) for more information.
- Install the Requester feature on a different system outside the cluster environment.

## Install IBM Connect:Direct for Microsoft Windows

---

### Installation Overview

After you complete the installation worksheets, you are ready to install Connect:Direct for Microsoft Windows.

Install all components or selected components. Connect:Direct components include:

- Connect:Direct for Microsoft Windows server: Performs the functions issued from the user interface. If you want a dedicated server, install only this component. It also includes the Command Line Interface (CLI), a tool that allows you to issue commands to the server.
- Connect:Direct for Microsoft Windows Requester: A graphical user interface that makes it easy to configure the Connect:Direct environment, create Processes, and submit commands to the server.
- Connect:Direct File Agent: Scans watched directories for files. When a file is detected, Connect:Direct File Agent either submits a default Process to Connect:Direct or performs the actions specified by the rules for the file.
- Connect:Direct Secure Plus: Implements security into a Connect:Direct operation. It must be installed on both nodes in order to activate the security software.

Software Developer's Kit (SDK) allows programmers to utilize and integrate the functions of Connect:Direct for Microsoft Windows.

After you install Connect:Direct for Microsoft Windows, TCP/IP information, database information and notification method is defined in the initialization parameters. To change TCP/IP information, the notification type, or the Connect:Direct database, refer to the *IBM Connect:Direct for Microsoft Windows System Guide*.

### Connect:Direct for Microsoft Windows Installation Icons

You choose the installation type to meet your needs. For example, if you want to install a standalone server, install only the Connect:Direct for Microsoft Windows server software. If you want to use a computer as a server and a workstation, install both the server software and Connect:Direct Requester. To install Connect:Direct Secure Plus, use the custom installation to install both Connect:Direct for Microsoft Windows and Connect:Direct Secure Plus.

Additionally, if you have a large network of computers where you need to install Connect:Direct for Microsoft Windows, you can set up a silent installation. To configure a silent installation, install Connect:Direct for Microsoft Windows at one computer. Then use the initialization parameters file that

was created during the installation to create a silent installation. Send the modified INI file to each computer on which to perform the silent installation.

Icons are displayed by each component to illustrate the selected installation type. The icons and installation types are described below:



The component is installed to the local drive.



Selected features of the component are installed.



The component is not installed.



Adds an icon to the Connect:Direct program group of the selected computer. The component is only installed when the user clicks the program icon to run the program.



The component is installed on the network.

Connect:Direct for Microsoft Windows is installed in the directory called C:\Program Files\IBM\Connect Direct v6.1.0\component.

If you would like to install Connect:Direct for Microsoft Windows in another location, select the Custom setup during the installation and click **Change**. The remaining procedures use the default installation path.

## Install Connect:Direct for Microsoft Windows and Optionally Connect:Direct Secure Plus

### About this task

To install IBM Connect:Direct for Microsoft Windows and optionally Connect:Direct Secure Plus:

### Procedure

1. If you downloaded the software from IBM Passport Advantage, double-click **cdw\_install.exe** from the download folder.  
**Note:** Passport Advantage provides access to your IBM software purchases, so you can download products directly to the computers where you want to install them. For information on the how to download software using Passport Advantage see, <https://www.ibm.com/software/passportadvantage/index.html>.
2. On the Welcome dialog, click **Next**.
3. To install Connect:Direct Secure Plus, select **Custom** and click **Next**. The Custom Setup dialog box is displayed that shows the features available for installation.
  - To configure SQL Server, disable the PostgreSQL feature on this dialog.
  - To install Connect:Direct Secure Plus, click the icon next to application and identify the type of installation.
4. To change the default installation path, click **Change**. Select the installation location and click **OK**.
5. Click **Next**.
6. To configure the Connect:Direct, enable Custom and click **Next**.
7. The name of the local node is displayed. It is the hostname of the computer where you are installing the product. To change the local node name, type the alternate node name and click **Next**.

**Important:** Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$ . \_ - to ensure that the entries can be properly managed by Control Center, Sterling Connect:Direct Browser User Interface, or IBM Sterling Connect:Direct Application Interface for Java™ for Java (AIJ) programs. However, use of '@' characters in a node name is discouraged, as it will cause some clients to display proxy records incorrectly. If the '@' character is at the end of a node name, client display of a proxy record with this node name will fail.

8. Configure the database you selected as follows:

- If you accepted the default database, PostgreSQL, type the information provided by your system administrator and click **Next**.

**Note:** If you chose another database and decide to install PostgreSQL at a later time, run `cdw_install.exe`. You can also install the PostgreSQL feature using the "Add or Remove Programs" tool from the Windows Control Panel.

- To configure SQL Server, type the address of the remote SQL Server and identify the authentication method. Click **Next**.

9. To specify an optional user account to run the Connect:Direct service, type the service account name and password, and click **Next**.

When setting up the logon account, give the account the following privileges:

- Act as part of the operating system
- Log on as a service
- Log on locally
- Replace a process-level token
- Full permissions over the installation directory
- Make the user a member of the local Administrators Group

10. To configure TCP/IP as the connection method, provide the following information, and click **Next**:

- Node-to-Node IP Address
- Node-to-Node Port
- Application Interface IP address
- Application Interface Port

11. Control Center Director upgrades and applies maintenance to Connect:Direct through a Connect:Direct Agent instance. To configure the Agent installation information that Control Center Director will use to communicate with the Agent, provide the following information and click **Next**.

- **Install Agent Application Interface Port**

**Note:** Install Agent's Application Interface Address is the same as the Connect:Direct Server's Application Interface Address, specified in the step above.

- Control Center Director's **Event Repository URL** (Open Server Architecture - OSA URL)

12. To use Active Directory, select Register Client Connection Settings and click **Next**.

13. To configure notification support:

- To enable Microsoft Windows broadcasting, select NT Network Broadcast.
- To enable SMTP, select SMTP.

14. Click **Next**.

15. To use a predefined network map, user authorities or initialization parameters file, for example exported from a previous versions of the product, click **Browse** and select the configuration file to import. Click **Next**.

16. Click **Install**.

17. When the installation is finished, Connect:Direct Requester is automatically started. If you do not want it to start automatically, deselect this option.



18. Click **Finish**.

### **Connect:Direct Server and its Client connections**

For client-server connections between a Connect:Direct Server and its clients be aware of the following limitations.

The connections between a Connect:Direct Server and the following clients are not secure:

- Connect:Direct Requester
- Windows CLI
- User-defined Windows SDK Client
- Connect:Direct for UNIX CLI
- User-defined UNIX ndampi Client

Though passwords sent by any of these clients to the Connect:Direct Server are obfuscated, the session is still not encrypted.

**Note:** The connection with Sterling File Agent can be made secure. For more details, refer [File Agent version 1.4.0.1](#).

## **Install Connect:Direct Server and Requester**

### **Before you begin**

If you chose a database other than the included PostgreSQL, make sure you have installed your database before you begin this installation.

**Note:** The connections between some clients and a Connect:Direct Server are unsecure. Passwords sent by one of these clients to a Connect:Direct Server are obfuscated, but the session is not encrypted. The clients are: the CD Requester, the Windows CLI, any user-written Windows SDK client and FileAgent.

### **About this task**

To install IBM Connect:Direct Server and Requester:

### **Procedure**

1. If you downloaded the software from IBM Passport Advantage, double-click CDWindows \cdw\_install.exe from the download folder.  
Alternatively, double-click CDWindows\SetupTrace.cmd to start the installation with logging enabled. This is helpful to diagnose installation issues.

**Note:** For information on the how to download software using Passport Advantage see, <https://www.ibm.com/software/passportadvantage/index.html>.

2. On the Welcome dialog, click **Next**.
3. Select **Typical** and click **Next**.

**Note:** When installing in a clustered environment, you must deselect the Requester feature (do not install at this time). Instead, do one of the following:

- Install the Connect:Direct Requester (Stand-Alone) using the CDRequester cdw\_install.exe.
- Install the Requester feature on a different system outside the cluster environment.

4. To configure the IBM Connect:Direct Server, enable Default and click **Next**.
5. Select the database to use and click **Next**.
6. Click **Next**.
7. Configure the database you selected as follows:

- If you accepted the default database, PostgreSQL, type the information provided by your system administrator and click **Next**.  
**Note:** If you chose another database and decide to install PostgreSQL at a later time, run `cdw_install.exe`. You can also install the PostgreSQL feature using the "Add or Remove Programs" tool from the Windows Control Panel.
  - To configure SQL Server, type the address of the remote SQL Server and identify the authentication method. Click **Next**.
8. Click **Install**.
  9. When the installation is finished, Connect:Direct Requester is automatically started. If you do not want it to start automatically, deselect the option.
  10. Click **Finish**.

## Upgrade Connect:Direct for Microsoft Windows

### About this task

To upgrade Connect:Direct:

### Procedure

1. Double-click the `cdw_install.exe` file.
2. On the Welcome dialog, click **Next**.
3. To install Connect:Direct Server and Requester, select Typical. Click **Next**.
4. To install Connect:Direct Secure Plus and Connect:Direct for Microsoft Windows, select **Custom**. Click **Next**.
5. Select Upgrade and click **Next**.
6. To save the previous version of Connect:Direct, check Keep <version - nodename>." Click **Next**.
7. Click **Install**.
8. If you installed Connect:Direct Requester, it starts automatically. To disable the automatic start, deselect this option. Click **Finish**.

## Add Connect:Direct for Microsoft Windows Components

### Before you begin

Before you can add a component, Connect:Direct for Microsoft Windows must be installed.

### About this task

After you install the product, you can change the installed components including repair a component that is corrupt, remove a component, or install an additional component.

To add, remove, or repair a Connect:Direct component:

### Procedure

1. Start the Connect:Direct for Microsoft Windows installation.
2. On the Welcome dialog, click **Next**.
3. To install additional components:
  - a) Select Modify and click **Next**.
  - b) Select the component to add and enable **This feature will be installed on local hard drive**. Click **Next**.

- c) Click **Install**.
- d) Click **Finish**.

## Repair Connect:Direct for Microsoft Windows Components

### Before you begin

Before performing this procedure, Connect:Direct for Microsoft Windows must be installed.

### About this task

To repair Connect:Direct for Microsoft Windows components:

### Procedure

1. Start the Connect:Direct for Microsoft Windows installation.
2. On the Welcome dialog, click **Next**.
3. Select Repair and click **Next**
4. Click **Install** to repair the installation.
5. Click **Finish** to complete the modification.

## Remove Connect:Direct for Microsoft Windows Components

### Before you begin

Before performing this procedure, Connect:Direct for Microsoft Windows must be installed.

### About this task

To remove a Connect:Direct for Microsoft Windows installed component:

### Procedure

1. Start the Connect:Direct for Microsoft Windows installation.
2. On the Welcome dialog, click **Next**.
3. Select **Modify** to remove a Connect:Direct component and click Next.
4. Select the component to remove, select **This feature will not be available**, and click **Next**.
5. Click **Install**.
6. Click **Finish**.

## Install Connect:Direct Requester Only (Stand Alone)

### About this task

Computers other than the one on which IBM Connect:Direct is installed may require that Connect:Direct Requester be installed.

If you installed IBM Connect:Direct in a clustered environment, you cannot use Requester unless it is installed separately, as Stand-Alone or on a separate machine from IBM Connect:Direct. If you are installing in a clustered environment, you must deselect the Requester feature and use these instructions to install it separately.

To install Connect:Direct Requester (stand alone):

## Procedure

1. If you downloaded the software from IBM Passport Advantage, double-click the CDRequester \cdw\_install.exe file from the download folder.

**Note:** For information on the how to download software using Passport Advantage see, <https://www.ibm.com/software/passportadvantage/index.html>.

2. If the Autorun option is enabled for the CD drive, the installation automatically starts. If the Autorun option is disabled, start the installation from the Microsoft Windows Run dialog.
3. On the Welcome dialog, click **Next**.
4. Click **Next**.
5. On the Setup Type dialog, select Typical and click **Next**.
6. Click **Install**.

## Install Connect:Direct File Agent

### About this task

When you install Connect:Direct File Agent, Java Runtime Environment (JRE) is automatically installed.

To install Connect:Direct File Agent:

### Procedure

1. If you downloaded the software from IBM Passport Advantage, extract the installation files from the download folder.

**Note:** For information on the how to download software using Passport Advantage see, <https://www.ibm.com/software/passportadvantage/index.html>

2. From the Introduction dialog box, click **Next**.
3. To install Connect:Direct File Agent in a selected location, click **Choose** and select the location.
4. Click **Next**, then click **Install**.
5. When the installation is complete, click **Done**.

## Uninstall Connect:Direct for Microsoft Windows

### About this task

The Connect:Direct for Microsoft Windows Uninstall program removes the application, its components, Connect:Direct Requester, and Connect:Direct for Microsoft Windows server, program items, and most server and Registry settings.

To uninstall the Connect:Direct Server program and all of the server utilities:

### Procedure

1. Open the "Add or remove programs" tool from the Windows Control Panel.
2. Highlight Connect:Direct for Microsoft Windows and click Remove.
3. Click Yes to confirm the removal of this program.
4. Click Finish.

## Add Initialization Parameters to Support Firewall Navigation

### About this task

If you communicate with a trading partner using a firewall, set two initialization parameters to support it. You assign a specific TCP/IP source port number or a range of port numbers with a particular TCP/IP address or addresses for outgoing Connect:Direct sessions. Setting these parameters allows controlled access to a Connect:Direct server if it is behind a packet-filtering firewall without compromising security.

To add firewall support initialization parameters:

### Procedure

1. Coordinate IP addresses and associated port assignments with your firewall administrator.
2. Add the following parameter to the Local Node Characteristics section of the initialization parameters file:  
tcp.src.ports=(valid IP address/optional subnet mask | valid IP address pattern, associated port number | associated range of port numbers | associated port number, associated range of port numbers)
3. Apply the new initialization parameter.
4. A second parameter called tcp.src.ports.list.iterations is automatically added to the Local Node Characteristics section during installation and has a default value of 1. Refer to the following table for a description and valid values for these parameters:

Parameter Name	Parameter Definition	Valid Values
tcp.src.ports	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:  (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)  A wildcard character (*) is supported to define an IP address pattern. If the wildcard character is used, the optional mask is not valid.

Parameter Name	Parameter Definition	Valid Values
tcp.src.ports.list.iterations	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. 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.

5. Coordinate the specified port numbers with the firewall administrators.

## Firewall Configuration Example

The following example illustrates a configuration of the firewall navigation initialization parameter. An explanation follows the example. Although the tcp.src.ports parameter is shown as a multi-line parameter, Connect:Direct for Microsoft Windows only supports single-line parameters.

```
tcp.src.ports=      (199.2.4.*, 5000-5050),
                   (199.2.4.7, 1376),
                   (200.200.0.0/255.255.0.0, 2000-2100, 3000-3100),
                   (138.16.17.*, 2000-2050, 3000-3050, 4001, 4005)
```

In the example, if Connect:Direct initiates a session with a remote node at the IP address 199.2.4.7, it will use only port 1376. A session 199.2.4.6 (or any other address beginning with 199.2.4) will use any port in the range 5000 to 5050.

A session to 200.200.4.10 uses a port from within the two ranges specified (2000 to 2100 or 3000 to 3100). Additionally, because of the subnet mask specification, a session to any IP address beginning with 200.200 will choose a port from within the two ranges specified.

The port for a session to any address beginning with 138.16.17 is selected from one of the two ranges (2000 to 2050 or 3000 to 3050) or the two individual ports specified (4001 or 4005).

## IBM Sterling Control Center Director Support

Control Center Director installs, upgrades and applies maintenance to Connect:Direct through a Connect:Direct Agent instance.

After you have upgraded Connect:Direct for Windows to the required maintenance level complete the following procedures to ensure Connect:Direct for Windows servers are discovered dynamically by Control Center Director.

Control Center Dir uses Certificate-based authentication to authenticate itself to a Connect:Direct® server. For more information on how to configure Connect:Direct and Control Center Director for Certificate-Based Authentication see the following sections:

- [Enable Certificate-based authentication on Control Center Director](#)
- [Enable Client Authentication on the Connect:Direct Secure Plus](#)

### Known Restriction

- When Control Center Director upgrades or applies maintenance to Connect:Direct Windows, currently running Connect:Direct process are shut down immediately.
- Emergency restore feature is currently not available for Control Center Director v1.0.0.1 implementation of Connect:Direct for Windows.

## Configuring Connect:Direct for Windows for Server and Upgrade Management

Control Center Director upgrades and applies maintenance to Connect:Direct through a Connect:Direct Agent instance. Agent is included with the Connect:Direct software when it is at a required level of maintenance for Agent inclusion.

To successfully move to a Connect:Direct version that supports a Control Center Director deployment, there are a few scenarios to consider. Review the actions below in order to optimize your update experience:


- Go to [Fix Central](#) and download the required maintenance version of Connect:Direct software.
- Certificate-based authentication is available when you install and configure Connect:Direct Secure Plus for Windows.

You can supply all of the information needed to configure Connect:Direct Agent instance by setting the following properties when you install Connect:Direct Secure Plus for Windows package:

- Install Agent Application Interface Port
- Install Agent Event Repository URL

For the complete installation procedure see, [“Install Connect:Direct for Microsoft Windows and Optionally Connect:Direct Secure Plus”](#) on page 13.

Alternatively, set the following parameters (initparms) to configure Connect:Direct Agent instance for Connect:Direct for Windows. For more information on how to change initialization parameters (intiparms) see, [“Changing Initialization Parameters”](#) on page 111.

Parameter (initparm)	Description
agent.port	<p>Enter port details here to configure the Agent listening port that Control Center Director will use to communicate with the Agent.</p> <p>Default: 1365</p> <p>With the port configuration complete, Agent is now set to automatically listen for incoming connections from Control Center Director.</p> <p> <b>Attention:</b> With multiple Connect:Direct instances on the same system you’re likely to run into port conflict issues unless you allocate a unique Agent listening port per instance.</p> <p>It is also recommended that having upgraded an instance, its unique port number must be applied before upgrading the next instance. This prevents potential errors that you could encounter during an upgrade process due to port conflict.</p>
osa.rest.url	<p>Provide the Event Repository URL to configure the Control Center Director Open Server Architecture (OSA) URL, the target location where Agent posts all the events to Control Center Director.</p> <pre>osa.rest.url=https://&lt;ip/hostname;port&gt;/osa/events/post:</pre> <p>The default is None.</p>

Parameter (initparm)	Description
osa.disable	Enables Agent to post all events to Control Center Director except when set to <b>Y</b> . The default is <b>N</b> .
agent.enable	Use to enable the agent installation. The default is <b>y</b> . Other possible values are <b>n</b> and blank.

### Configuring Connect:Direct for Windows for License Governance

Set the following parameters (initparms) to automate license metrics collection from Connect:Direct for Windows.

Parameter (initparm)	Possible Values
license.edition	<ul style="list-style-type: none"> <li>• Premium</li> <li>• Standard</li> <li>• Solo</li> <li>• Default: Blank (undefined)</li> </ul>
license.type	<ul style="list-style-type: none"> <li>• Production</li> <li>• Non-Production</li> <li>• Default: Non-Production</li> </ul>
license.pvu	<p>A non-negative integer</p> <ul style="list-style-type: none"> <li>• The <code>license.pvu</code> parameter is only applicable for Connect:Direct Premium licenses</li> <li>• This value can be calculated using the IBM License Metric Tool (ILMT) or it can be looked up at the <a href="#">IBM Processor Value Unit licensing website</a>.</li> <li>• Default: 0</li> </ul>

**Note:** All three Initparms can be unset and a user does not have to supply a value.

**Note:** Ensure that you set the **Statistics** field to **All** when you define a User Authority for a Connect:Direct for Microsoft Windows user. Setting to **All** enables all process access license statistics. For more information see, “[Defining User Authority](#)” on page 37. This applies to the user that Control Center Director connects as using Certificate-Based Authentication.

Solo license edition type constraints:

- A warning message is logged if the number of Netmap entries in `netmap.cfg` exceeds 2.
- A warning message is logged when a transfer is initiated with third or later remote entry, in order of appearance.
- The number of concurrent sessions is restricted to 2 or fewer



## Configuring Connect:Direct for Windows for New Install Task

You can initiate fresh installation of Connect:Direct servers from Control Center Director. To automate new installation Sterling Connect:Direct for Microsoft Windows from Control Center Director, set the following parameters (initparms) in the ini file:

Parameter (initparm)	Definition	Possible Values
CD_AGENT_ENABLE	Use to enable the agent installation	<ul style="list-style-type: none"><li>• y (Default)</li><li>• n</li><li>• blank</li></ul>
CD_AGENT_OSA_DISABLE	Use to disable the agent installation	<ul style="list-style-type: none"><li>• y (Default)</li><li>• n</li><li>• blank</li></ul>
CD_AGENT_INSTALLATION_ID	Use to store initparms configuration	<ul style="list-style-type: none"><li>• blank (Default)</li><li>• any string (maximum length: 1023 bytes)</li></ul>
CD_TRUSTEDCERT_FILE	Specifies the trusted certificates to be imported	<ul style="list-style-type: none"><li>• File path</li><li>• blank (Default)</li></ul> <p><b>Note:</b> Trusted certificates are not imported if the parameter is not specified or left blank.</p>
CD_SECUREPLUS_FILE	Specifies the file containing additional SPCLI commands to configure on CD.	<ul style="list-style-type: none"><li>• File path</li><li>• blank (Default)</li></ul> <p><b>Note:</b> Splice command file will not be executed if the parameter is not specified or left blank.</p>
CD_AGENT_PORT	Port details to configure the Agent listening port that Control Center Director will use to communicate with the Agent. With the port configuration complete, Agent is now set to automatically listen for incoming connections from Control Center Director	1365 (Default)

For more information on how to install new Connect: Direct server for Windows from Control Center Dir , see [Installing new Connect:Direct server for Windows](#).

## Automate Installation

### About Silent Installations

You can automate Connect:Direct for Microsoft Windows installation and configuration for distribution throughout your enterprise by performing silent installations. Silent installations require no user

responses during the installation routine. Configuration information is supplied by the initialization file that you define. Before you perform a silent installation, install Connect:Direct for Microsoft Windows on a master node and configure a network map and user authorizations.

The Microsoft Installer Properties are the foundation of a silent installation. Properties can be set on the command line or by creating custom transforms. A transform is a collection of changes applied to an installation. Transforms alter the installation database and can be used to customize a base installation package. Applying a transform to a base installation package adds or replaces data in the installation database.

## Silent Installation Options

You can perform a default silent installation that installs Connect:Direct server and Connect:Direct Requester, or you can use transforms to modify the settings of an installation package. The transforms included with this product enable, disable, and remove a feature from the Connect:Direct for Microsoft Windows.msi package. To change the silent installation setting, use a transform to enable or disable the desired feature.

## Requirements for Silent Installation

For each node where a silent installation is performed, determine if the node uses special services, for example, Active Directory. For these nodes, a custom INI file is required. If no custom INI file is present, the default installation is performed.

## Customize Initialization Information for a Silent Installation

### About this task

The Connect:Direct server supports an INI file, which can specify the value of installation properties. If you plan to use the INI file to manage a silent installation, change the parameters of the INI file provided with Connect:Direct for Microsoft Windows to specify site-specific information.

The Connect:Direct for Microsoft Windows server initialization file (CD\_SRVR.INI) is located on the CD in the /advanced folder. The Connect:Direct Requester and SDK installations do not use initialization files. To set the installation directory for an installation, specify the INSTALLDIR property in the command line.

To define site-specific parameter values in an INI file for each node where you install Connect:Direct for Microsoft Windows using the silent installation, do the following:

### Procedure

1. Open the CD\_SRVR.INI file using any text editor, such as Microsoft Windows Notepad.
2. Define the site-specific initialization parameters for a node and save the file.
3. Repeat steps 1–2 for each server where Connect:Direct will be installed.

## CD\_SRVR.INI Parameter Values

Parameter Name	Parameter Definition	Values
CD_SETUP_TYPE	Specifies the type of installation. Default configures a new installation. Upgrade migrates an existing installation. Custom and Default values have the same effect in silent installs.	<u>Default</u>   Custom   Upgrade

Parameter Name	Parameter Definition	Values
CD_NODENAME	Local node name, from 1-16 characters, consisting of numbers, letters, @, #, \$, -, underscore (_), and period (.) with no spaces or apostrophes.	If this field is blank, the parameter defaults to the first 16 characters of the computer where the server is installed.
CD_UPGRADE_NODE	The version of software and node to upgrade. Valid only if CD_SETUP_TYPE=Upgrade.	Version number/node name. For example: CD_UPGRADE_NODE=v4.8.0\MYNODE. The default node is the first node of the most recent version installed.
CD_UPGRADE_KEEPSRC_FLAG	Prevents the current version from being deleted before installing the new version. Valid only if CD_SETUP_TYPE=Upgrade.	1=enabled. If the selected installation type is Upgrade/Migrate, the previously installed version is uninstalled by default.
CD_HOST_IP	The IP address or host name of the server used for node-to-node communication.	Any valid IP address or host name. If blank, the IP address is obtained from the destination computer's IP address in the TCP/IP stack.
CD_HOST_PORT	The port number that Connect:Direct for Microsoft Windows, or user-written API programs, will use to establish client sessions with this Connect:Direct server for outgoing communications.	A numeric value in the format nnnn, where nnnn is a positive integer no larger than four digits. The default is 1364.
CD_API_IP	The IP address or host name of the server used for API (client) connections.	Any valid IP address or host name. If blank, the IP address is obtained from the destination computer's IP address in the TCP/IP stack.
CD_API_PORT	The port number that Connect:Direct for Microsoft Windows or user-written API programs will use to establish client sessions with this Connect:Direct server. You must specify the port when it differs from the default.	A numeric value in the format nnnn, where nnnn is a positive integer no larger than four digits. The default is 1363.
CD_ACTIVEDIR_FLAG	Registers the client IP address to active directory.	1=enabled. Disabled by default.
CD_NOTIFY_TYPE	Defines the Process completion notification type.	<u>NT Broadcast</u>   SMTP
CD_NOTIFY_SMTP_HOST	If CD_NOTIFY_TYPE = SMTP, this field identifies the IP address of the SMTP host.	Host address in the format xxx.xxx.xxx
CD_NOTIFY_SMTP_PORT	If CD_NOTIFY_TYPE = SMTP, this field identifies the port used by the SMTP host.	Port number up to four characters. The default value is 25.

Parameter Name	Parameter Definition	Values
CD_NOTIFY_SMTP_SENDER	If CD_NOTIFY_TYPE = SMTP, this field identifies the e-mail address to identify the sender of the message	Valid e-mail address.
CD_NOTIFY_SMTP_AUTHENTICATE	If CD_NOTIFY_TYPE = SMTP, this field enables authentication. If this value is not set, the user ID and password fields are ignored.	1=enabled. Disabled by default.
CD_NOTIFY_SMTP_USERID	If CD_NOTIFY_TYPE = SMTP, this field identifies the user ID to use to authenticate the server.	A valid user ID.
CD_NOTIFY_SMTP_PWD	If CD_NOTIFY_TYPE = SMTP, this field identifies the user password to use to authenticate the server.	A valid user password.
CD_USERAUTH_FILE	The path name and file name of a valid user authority file.	Any valid user authority file. For example: C:\Configurations\MyUserAuth.cfg
CD_NETMAP_FILE	The path name and file name of a valid network map file.	Any valid network map file. For example: C:\Configurations\MyNetmap.cfg
CD_INITPARMS_FILE	The path name and file name of a valid initialization parameter file.	Any valid Microsoft Windows directory and initialization parameter file name. For example: C:\Configurations\MyInitparms.cfg
CD_SVC_ACCOUNT	Service user account. Installation fails if the user doesn't have the following privileges: act as part of the operating system, log on locally, log on as service, replace a process level token.	Domain\Username format. The default account is the local system account.
CD_SVC_ACCOUNT_PWD	Service user account password.	
CD_DATABASE_NAME	Specifies the name of the database.	The default database name is CDWINNT.
CD_DATABASE_TYPE	Configures the TCQ and Statistics databases. MSSQL must be installed and configured prior to installing IBM Connect:Direct. PostgreSQL is optionally installed and configured during the install.	<u>POSTGRESQL</u>   MSSQL
CD_POSTGRESQL_PORT	If CD_DATABASE_TYPE = POSTGRESQL, this parameter specifies the PostgreSQL server port number.	The default port number is 23610.
CD_POSTGRESQL_USERID	If CD_DATABASE_TYPE = POSTGRESQL, this parameter specifies the user ID of the PostgreSQL server administrator.	The default user ID is root.

Parameter Name	Parameter Definition	Values
CD_POSTGRESQL_PWD	If CD_DATABASE_TYPE = PostgreSQL, this parameter specifies the PostgreSQL server system administrator's password. The password is required.	
CD_SQL_SERVER	If CD_DATABASE_TYPE = MSSQL, this parameter specifies the name of the SQL server.	Valid SQL Server
CD_SQL_AUTHENTICATION	If CD_DATABASE_TYPE = MSSQL, this parameter specifies the SQL authentication.	Disabled by default.
CD_SQL_USERID	If CD_DATABASE_TYPE = MSSQL, this parameter specifies the user ID of the SQL server system administrator. If SQL authentication is enabled, the SQL user ID and password are required.	
CD_SQL_PWD	If CD_DATABASE_TYPE = MSSQL, this parameter specifies the SQL server system administrator's password.	Valid 1-30 character SQL Server System administrator password
CD_SPE_DISABLE_FLAG	Disables the Secure+ Strong password encryption feature.	1=disabled. Enabled by default.
CD_KEYSTORE_FILE	Specifies the file name for Secure+ KeyStore file.	The file name should not include a path. The default file name is cdkeystore.kdb.
CD_KEYSTORE_PWD	Specifies the password for Secure+ KeyStore file. The password is required when Secure+ is installed.	
CD_NETMAP_CHECK	Specifies the initialization parameter network map check settings.	<u>Y</u>   L   R   N
CD_NODE_CHECK	Specifies the initialization parameter node check settings.	A   <u>B</u>   C
CD_CLIENT_KEYCERT_FILE	Specifies the Secure+ Client Key Certificate file name. (PEM PKCS8).	
CD_CLIENT_KEYCERT_PWD	Specifies the password for the Secure+ Client Key Certificate file.	
CD_CLIENT_CIPHERSUITES	Enables the TSL or SSL cipher suites for the node record. Optional.	The default is (TLS_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA)
CD_ADMIN_USERID	Specifies the default Connect:Direct user authority.	Any valid user ID.

Parameter Name	Parameter Definition	Values
CD_TRUSTEDCERT_FILE	Specifies the trusted certificates to be imported	<ul style="list-style-type: none"> <li>File path</li> <li>blank (Default)</li> </ul> <p><b>Note:</b> Trusted certificates are not imported if the parameter is not specified or left blank.</p>
CD_SECUREPLUS_FILE	Specifies the file containing additional SPCLI commands to configure on CD.	<ul style="list-style-type: none"> <li>File path</li> <li>blank (Default)</li> </ul> <p><b>Note:</b> Splice command file will not be executed if the parameter is not specified or left blank.</p>
CD_AGENT_PORT	Specify the Agent port details here to configure the Agent listening port that Control Center Director will use to communicate with the Agent.	Default: 1365
CD_AGENT_ENABLE	Use to enable the agent installation	<ul style="list-style-type: none"> <li>y (Default)</li> <li>n</li> <li>blank</li> </ul>
CD_AGENT_OSA_DISABLE	Use to disable the agent installation	<ul style="list-style-type: none"> <li>y (Default)</li> <li>n</li> <li>blank</li> </ul>
CD_AGENT_INSTALLATION_ID	Use to store initparms configuration	<ul style="list-style-type: none"> <li>blank (Default)</li> <li>any string (maximum length: 1023 bytes)</li> </ul>
CD_OSA_REST_URL	Provide the Event Repository URL to configure the Control Center Director Open Server Architecture (OSA) URL, the target location where Agent posts all the events to Control Center Director.	None
CD_PSWDEXIT_DLL	This is the Password Exit DLL File to enable the Password Exit feature. If the value is blank, the feature is disabled.	<ul style="list-style-type: none"> <li>none (default)</li> <li>valid fully qualified DLL file</li> </ul>

Parameter Name	Parameter Definition	Values
CD_PSWDEXIT_DLLHASH	This is a SHA256 hash of the Password Exit DLL file. Before Connect:Direct Windows loads the your Password Exit DLL file, it will compute the SHA256 hash for the file. If the computed hash matches the configure Password Exit DLL Hash, then Connect:Direct Windows will load the dll and use it to obtain user passwords.  <b>Note:</b> This parameter is mandatory when CD_PSWDEXIT_DLL is enabled.	<ul style="list-style-type: none"> <li>• none (default)</li> <li>• valid SHA256 hash</li> </ul>
CD_PSWDEXIT_POLICYID	Optional It is sent to the Password Exit as a parameter.	<ul style="list-style-type: none"> <li>• none (default)</li> <li>• valid string</li> </ul>
CD_PSWDEXIT_APPLID	Optional It is sent to the Password Exit as a parameter.	<ul style="list-style-type: none"> <li>• none (default)</li> <li>• valid string</li> </ul>

## Customize Connect:Direct for Microsoft Windows Configuration Information

After you install Connect:Direct for Microsoft Windows on a master node, you can customize configuration information for distribution within your enterprise by using Connect:Direct Requester to configure the network map, user authorizations, and initialization parameters on the master node. You can then extract those files using the Configuration Utility. For more information about using the Configuration Utility, refer to "Defining and Managing the Connect:Direct Network" in Help or in the *IBM Connect:Direct for Microsoft Windows System Guide*.

To apply the customized configuration files during a silent installation, set parameter values in the CD\_SRVR.INI file as follows:

- To apply a customized network map file, set CD\_NETMAP\_FILE to the fully qualified path to a customized network map configuration file.
- To apply a customized user authorization file, set CD\_USERAUTH\_FILE to the fully qualified path to a customized User Authorization configuration file.
- To apply customized initialization parameters, set CD\_INITPARMS\_FILE to the fully qualified path to a customized initialization parameters file.

## Run a Silent Installation

### About this task

cdw\_install.exe installs the Microsoft Installer, Visual C++ Runtime Libraries and other prerequisites on a system when they do not already exist. To complete the installation, cdw\_install.exe reboots the system if required and resumes the installation after the computer has restarted.

cdw\_install.exe can accept a limited number of command-line parameters.

- You can pass parameters through cdw\_install.exe to the included .msi file (MsiExec) by using the /v option. After you specify this option, list any supported parameters that can be passed to Msiexec.exe.

- To prevent `cdw_install.exe` from displaying a progress bar when it launches, use the `/s` command-line parameter. For example, if you enter `cdw_install.exe /s`, `cdw_install.exe` launches, but the user interface is not displayed.
- If you use the `/v` option and a command contains a text with a quotation mark within existing quotes, type a backslash (`\`) before the text. For example, the command line contains the following: `/v"C:\My Files\SecurePlusEnable.mst"`. Because the path contains spaces, you must use quotes. However, because quotes are required around the complete argument, failure to use a backslash before internal quotes causes the statement to fail.
- Do not put a space between the command-line option (`/v`) and the arguments.
- To define multiple parameters with the `/v` option, separate them with a space.

To attach to the network and install Connect:Direct for Microsoft Windows features from the network location in a silent installation:

## Procedure

1. Click Start > Run. The Run dialog box is displayed.
2. In the Open field of the Run dialog box, type a command similar to following example:

```
cdw_install.exe /s /v/qn
```

This command installs all Connect:Direct for Microsoft Windows components including Connect:Direct Requester and Connect:Direct Secure Plus.

**Note:** `cdw_install.exe` installs Connect:Direct for Microsoft Windows in the default directory `C:\Program Files\IBM\Connect Direct v[current version]`. If you want to install Connect:Direct for Microsoft Windows in a different directory, use the `INSTALLDIR` option on the command line. Enclose the pathname in quotes and terminate the path with a backslash, as in the following example:

```
cdw_install.exe /v"INSTALLDIR=\"C:\Program Files\IBM\Connect Direct v[current version]\""
```

3. Computers in a Microsoft Windows Domain that use Active Directory can also automate installations using the software installation extension of the Group Policy Snap-In.

The Microsoft Windows installer uses an `/M` command line parameter to generate MIF files. In the following example, the `CDNT.mif` file is created to report success or failure:

```
cdw_install.exe /s /v/qn /M C:\Windows\CDNT.mif
```

## Suppressing an Automatic Reboot

### About this task

During an installation of Sterling Connect:Direct for Microsoft Windows (or its prerequisites), the Windows installer can determine when a reboot of the system is necessary. Commonly, a reboot is required because the installer is attempting to install a file that is currently being used. While you are prompted with a request to reboot during an interactive installation, the installer initiates the reboot automatically during a silent installation.

This section explains how to suppress most automatic reboots during an installation, allowing more control on scheduling a necessary reboot at your own convenience. It applies to new installations, as well as to upgrades from an older version or maintenance updates. Follow these steps to suppress most automatic reboots during the installation of IBM Sterling Connect:Direct for Microsoft Windows and detect when a reboot is necessary:



## Procedure

1. Close all other applications before starting the installation, especially any Connect:Direct application, such as Connect:Direct Requester or the Command Line Interface. There is no need to stop the Connect:Direct service, as it will be done by the installer automatically.
2. Specify REBOOT=ReallySuppress as part of code /v parameter on the installation command line, like:

```
cdw_install.exe /v"REBOOT=ReallySuppress ... " /w /clone_wait
```

**Note:** IBM Sterling Connect:Direct for Microsoft Windows must be installed by a Microsoft Windows administrator.

3. Wait for the installation to complete and check its return code. A return code of 3010 indicates that a reboot is required. A return code of 0 indicates success and no reboot is required. If a reboot is required, it is important to reboot the system as soon as possible and before using Sterling Connect:Direct for Microsoft Windows else the product may not function correctly.

## Example

Example batch file snippet for a new silent installation

```
call cdw_install.exe /v"REBOOT=ReallySuppress CD_SRVR_INI_FILE=C:\cd_srvr.ini /l*v
CDWinInst.log /qn" /s /w /clone_wait
if %ERRORLEVEL% equ 3010 (
    echo The installation requires a reboot. Please reboot the system as soon as possible.
) else if %ERRORLEVEL% equ 0 (
    echo The installation completed successfully. No reboot required.
) else (
    echo The installation has failed with RC=%ERRORLEVEL%
)
```

## Perform an Unattended Uninstallation

### Procedure

- Connect:Direct for Microsoft Windows provides for an unattended uninstallation. In the following example, the /x command parameter removes the package:

```
cdw_install.exe /x
```



---

# Chapter 3. System Guide

## Configure the Local Node

---

### Configuring the Connect:Direct Local Node

Before using IBM Connect:Direct for Microsoft Windows to transfer files, you have to configure the local node.

You can configure the local node using the Configuration Tool or with Connect:Direct Requester. This set of topics provides instructions for using Requester. Requester is a graphical user interface that enables you to define local nodes, user IDs, user authorities, and remote user proxies.

With the Connect:Direct Configuration Tool, you edit text files of user authorization and network map parameters, and you can export these files. However, using the Configuration Tool requires that you understand all the Connect:Direct keywords and their parameters. Connect:Direct Requester makes it easier to define user authorization parameters.

**Note:** The Configuration Tool also enables you to view or print a configuration. See [Print and View Node and Network Map Definitions](#) under [Define and Manage the Connect:Direct Network](#).

To configure the local node using Requester, you:

1. Add a node definition for each server at the local site.
2. Add a user ID for each person who accesses one of the local servers.
3. Define the functions each local user can perform (functional authorities).
4. If necessary, configure user proxies for remote user access to the local node.

#### Related concepts

[Remote User Proxy Worksheet](#)

[User Functional Authorization Worksheet](#)

#### Related tasks

[Printing and Viewing Node and Network Map Definitions](#)

## Adding or Modifying a Local Node Definition

### About this task

To use Connect:Direct for Microsoft Windows, you begin by configuring a local node for each server in the local network. You identified a local node when you first 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 a server:

### Procedure

1. Select **Node > Connection Settings > Insert Node**.
2. Type the node name in the **Name** field or select a node in **Active Directory Nodes** if Active Directory is enabled.
3. Optionally, select the default user ID to associate with the node in the **Default User ID** field.
4. Select the operating system used by 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.  
**Note:** See [“Specify IP Addresses, Host Names, and Ports”](#) on page 129.
6. Type the port number in the **Port** field.
7. If this is the default node, select **Set as the Default Node**.
8. Click OK.
9. To modify a local node definition for a server at your site, double-click the node on the **Nodes** tab of the Control Pad (left side of the Requester main window). Modify fields as required and click **OK**.

## Adding a User

### About this task

To add a new Connect:Direct user from Connect:Direct Requester:

### Procedure

1. If more than one local node is configured, on the **Nodes** tab of the Control Pad, highlight the local node to which the user is to be added.
2. Select Node > Connection Settings > Edit Userids.
3. Click **Insert**.
4. Type information in the following fields:
  - Name—Type the name of the new user. Either type the user name as defined in the Microsoft Windows setup, such as "cduser", or type a user name in the UPN format, such as "cduser@adtree.mycomp.com" for ActiveDirectory or "cduser@mydomain" for a domain.
  - Password—Type the password defined for the user.
  - Verify Password—Retype the password defined for the user.
5. Click **Remember Password** to automatically reload the password when you attach as this user.
6. Click **Set as the Default User** if you want the new user to be the default user for the node.
7. Click **OK**.
8. 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**.
9. Click **OK** on the **User Properties** window to save the settings, then click **Close**.



**Attention:** Changes made to node settings are not written to the Registry until you select OK.

## Modifying a User ID

### About this task

To modify a user ID from Connect:Direct Requester:

### Procedure

1. If more than one local node is configured, on the Nodes tab of the Control Pad, highlight the local node for which the user is to be modified.
2. Select Node > Connection Settings > Edit Userids.
3. Double-click the user ID to modify.
4. Modify fields as necessary.
5. Click **OK**.

## Deleting a User

### About this task

To delete a user from a node, from Connect:Direct Requester:

### Procedure

1. If more than one local node is configured, on the **Nodes** tab of the Control Pad, highlight the local node from which the user is to be deleted.
2. Select Node > Connection Settings > Edit Userids.
3. Select the user you want to delete.
4. Select Delete, then click **Confirm**.  
The user is deleted.

## Attaching to a Local Node

### About this task

After you configure the local node and define 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.

### Procedure

1. Select Node > Attach.
2. Select a user ID from the Userid list.
3. In the **Password** field, type the password that corresponds to the user ID.
4. Click **OK**.

## About Local Functional Authorities

After you define a user ID for each user with access to the local node, you can limit the tasks a user can perform by defining user authorities for each user ID. For example, you can permit a user to submit Processes but not monitor or delete them. Define user authority as a default administrator or a general user. Then define the directories where a user can perform functions. You can define authorities for remote users, and you can group users under group authorities. Connect:Direct

### Define a Functional Authority Type

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

<b>Authority</b>	<b>Default Administrator</b>	<b>Default General User</b>	<b>Default Operator User</b>
View Processes in the TCQ	yes	yes	yes
Issue the copy receive, copy send, run job, and run task Process statements	yes	yes	no
Issue the submit Process statement	all	yes	no

Authority	Default Administrator	Default General User	Default Operator User
Submit, change, and delete Processes for all users	yes	no	no
Monitor processes for all users	yes	yes	yes
Submit, change, Monitor, and delete your own Processes	yes	yes	no
Run programs	yes	yes	no
Access Process statistics	all	yes	all
Upload and download files from any directory	yes	yes	yes
Upload and download files to or from specific directories	no	no	no
Run programs from any directory	yes	yes	no
Run programs from specific directories	yes	no	no
Update the network map	yes	no	view
Update the translation table	yes	yes	view
Update local user authorities	yes	no	view
Update remote user secure point-of-entry proxies	yes	no	view
Stop Connect:Direct for Microsoft Windows	yes	no	no
Invoke the refresh initialization parameters options	yes	yes	view
Use the trace tool or issue traceon and traceoff commands	yes	no	no
Override execution priority, including Hold, Retain, and Plexclass status	all	yes	yes
User type can override the CRC status	on. <b>Note:</b> The CRC will be off if Secure+ is used.	off	off
Override Process options such as file attributes and remote node ID	all	yes	off

## Define Directories Where Users Can Perform Tasks

You then define directories where a user can perform tasks. If you do not specify a directory for a function, the user can perform it 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 perform operating system commands. These commands could include copy commands to copy files to any directory, bypassing the Upload and Download restrictions.

To prevent this, set directory restrictions for the Process and program features using a separate directory path from the Upload and Download directory path or disable the Run Job and Run Task for this user. Programs that be run are defined in the Process and Program directories.

## Define Remote User Proxies

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

## Defining User Authority

### About this task

Use this procedure to restrict the functions that a user can perform and the directories where a function can be performed.

To set user authorities:

### Procedure

1. Select Admin > Functional Authorities.  
The User Authorities view is the default view.
2. Choose one of the following types of users:
  - Click **New Admin** to create a new user authority with full privileges for Process controls and functions.
  - Click **New Genusr** to create a user authority with reduced privileges.
  - Click **New Operator** to create a user authority with view-only privileges.
3. Type a name, from 1 to 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 cduser@adtree.mycomp.com or cduser@mydomain. The UPN format allows you to identify both the user name and the domain.
4. Do one of the following:
  - To save the new user authority with the default privileges, click **OK**.
  - To modify the default user privileges, continue with the next step.
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**.

Field Name	Definition	Valid Values
Client Source Addresses	Use this parameter to list all of the IP addresses and/or host names that are valid for this user's API connection. If you specify values for this field, the IP address of this user's API connection is validated with the client.source_ip list. If the IP address does not match the one specified on the list, the connection is rejected.	A comma-separated list of client IP addresses or host names associated with client IP addresses.  The IP address of the client connection for this user must match the address configured in this field.  For example: nnn.nnn.nnn.nnn, localhost
Allow Client Certificate Authentication	Determines if the user can perform certificate authentication for client API connections.  Check Box selected—Enables client certificate authentication for the user  Check Box not selected—Disables client certificate authentication for the user	Selected   Not Selected
Allow No Password local Connections	Determines if the user can perform a local client API connection without a password.  Check Box selected—Enables local client API connection for the user  Check Box not selected—Disables local client API connection for the user	Selected   Not Selected
Allow Process to run using Service Account	Grants permission to run process using Service Account	Selected   Not Selected
Use Password Exit	Determines if user can obtain its password using the Password Exit.	Selected   Not Selected
Submit	Grants authority to submit Processes.	Yes   No
Monitor	Grants access to the Process Monitor function.  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.  Yes specifies that you can change your own Processes; All specifies that you can change anyone's Processes.	Yes   No   All



Field Name	Definition	Valid Values
Delete	Grants authority to delete Processes from the TCQ. 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. 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.
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 Directory Restrictions functions:

Field Name	Description
Upload	Specifies the directory that the user can copy files from and use as a source. Security in some Microsoft Windows systems prompts for administrative permissions confirmation when it writes to the Program Files subdirectories. If you specify a Program Files directory in the <b>Upload</b> field, the system may be unable to copy files to that location. To fix this problem: a. Specify an upload directory that is not in the Program Files directory. b. On the Connect:Direct for Microsoft Windows Server, use Microsoft Windows Control Panel to change User Account Control Settings to <i>Never Notify</i> . Reboot the server to enable the updates.
Download	Specifies the directory that the user can copy files to and use as a destination.

Field Name	Description
Process	Specifies the directory from which the user can submit a Process.  <b>Note:</b> Setting a Process directory restriction here only restricts submit statements within a Process. In other words, given an entry in this field, a user (or, in the case of a group functional authority, a group) can use Requester to submit a Process without restrictions on where the Process is submitted from, but a Submit Process statement within the Process will run only from the directory specified here.
Program	Specifies the directory from which the user can run a program.

8. To define administrative privileges, click the **Admin** tab.
9. To give a user access to an administrative function, change the value to Yes or select **View** to grant read-only access and click **OK**. Refer to the following table for Administrative functions:

Field Name	Definition	Valid Values
Netmap	Grants authority to update the network map.	Yes   No   View
Translation Table	Grants authority to update the translation tables.	Yes   No   View
User Authorities	Grants authority to update local user Connect:Direct functional authorities.	Yes   No   View
User Proxy	Grants authority to update user proxies.	Yes   No   View
Secure+	Grants authority to send Connect:Direct Secure Plus commands through the API.	Yes   No
Stop	Grants authority to stop Connect:Direct.	Yes   No
Initparms	Grants authority to refresh the Connect:Direct server initialization parameters.	Yes   No   View
Trace	Grants authority to access the Trace utility.	Yes   No

10. Click the **Override** tab to define override authority.
11. To grant access to the override function, set any of the override privileges to Yes. Refer to the following table for the override privilege functions:

Field Name	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

Field Name	Definition	Valid Values
CRC	Grants authority to override the CRC-enabled state in node and Process statements.	ON   OFF   Blank

12. Click **OK**.

## Modifying a User Authority

### About this task

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

### Procedure

1. Select Admin > Functional Authorities.
2. Double-click the user authority to edit.
3. Change the user authorities as necessary.
4. Click **OK**.

## Deleting a User Authority

### About this task

To delete a user authority:

### Procedure

1. Select Admin > Functional Authorities.
2. Select the user you want to delete and click **Delete**.

## Defining a Group Authority

### About this task

Use this procedure to create group authorities. Group authorities allow you to group users who exercise the same level of functional authorities.

To define a group authority:

### Procedure

1. Select Admin > Functional Authorities.
2. Click the **Group Authorities** tab.
3. To add a group authority, do one of the following:
  - Click **New Admin** to create a new group based on \*Admin settings as the default.
  - Click **New Genusr** to create a new group based on \*GENUSR settings as the default.
  - Click **New Operator** to create a new group based on \*Operator settings as the default.
4. Type the name, from 1 to 50 alphanumeric characters, for the new group. You can use spaces.
5. Define the group as either a local group or domain group by doing one of the following:
  - If the group is local, click **Local** and enter the name of the group as defined by the workstation administrator under My Computer/Manage/Local Users and Groups.

- If this is a domain group, click **Domain Name**, then enter the name of the group as defined by the domain administrator, and the name of the domain on which the group is defined.

**Note:** Local groups are groups defined locally on the server using the O/S utility for managing local users and groups. If your users are defined locally, use local groups. If your users are not defined locally but in a domain, use domain groups. If you have some users in local and some in domains, you can use both.

6. Check **User Password Exit** to allow user to obtain its password using the Password Exit.

**Note:** You can use this functionality only if initparms are configured to use Password Exit properly.

7. Do one of the following:

- To save the new group authority with default privileges, click **OK**.
- To modify the default group privileges, continue with the next step.

8. To restrict the control functions or statements users in the group are authorized to perform, change the value of the fields on the available tab to No to deny authority for that privilege and click **OK**.

**Note:** Refer to “Defining User Authority” on page 37 for definitions of specific user authorities.

9. Click **OK** to finish defining the group authority.

## Modifying a Group Authority

### About this task

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

### Procedure

1. Select Admin > Functional Authorities and click the **Group Authorities** tab.
2. Double-click the group authority to edit.
3. Change authorities for the group as necessary.
4. Click **OK**.

## Sequencing Group Authorities

### About this task

A user can be a member of multiple groups, local and domain. Since each group authority defined can contain different restrictions, the order of groups as they are checked for a match is important. More restrictive groups should be checked before less restrictive groups.

To change a group's position in the order of group functional authorities:

### Procedure

1. On the **Group Authorities** tab of the **Functional Authorities** window, select the group functional authority to move.
2. Move the group up in the order by clicking **Move Up**.
3. Move the group down in the order by clicking **Move Down**.
4. Click **OK**.

## Deleting a Group Authority

### About this task

To delete a group authority:

### Procedure

1. Select Admin > Functional Authorities and click the **Group Authorities** tab.
2. Select the group you want to delete and click **Delete**.

## Define 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 relationship enables users at remote nodes to submit work to their system that interacts with the Microsoft Windows node without defining user IDs and passwords for the specified Microsoft 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 Microsoft 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 Microsoft Windows accesses the proxy definition that corresponds to the specified user and logs on to the Microsoft 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

### About this task

To create or modify a proxy for a remote user:

### Procedure

1. Select Admin > User Proxies.
2. Do one of the following:
  - To add a new remote user proxy, click **Insert**.
  - To modify a proxy, double-click the Remote User name to modify.

3. Add or modify the following information:
  - a) For a new proxy, type the remote node user ID or select <ANY USER> to define a generic user ID.
  - 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 twice associated with the local user ID to which the proxy is mapped.

**Note:** You can enter a user ID in UPN format such as user@domain.com.
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**, **Allow Process to run using Service Account**, **Use Password Exit** or **Copy Receive** operations 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:
  - 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.
  - 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.
  - 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.
  - 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.
  - Specify whether the user can submit copy Process statements using the **Copy Send** field. If you specify USER, this setting defaults to the proxy entry's local user's functional authority setting.
  - Specify whether the user can receive copy Process statements using the **Copy Receive** field. If you specify USER, this setting defaults to the proxy entry's local user's functional authority setting.
7. Click **OK**.

## Deleting a Remote User Proxy

### About this task

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

### Procedure

1. Select Admin > User Proxies.
2. Select the remote user you want to delete.
3. Click **Delete**.
4. Click **Close**.

## Define and Manage the IBM Connect:Direct Network

---

### Define and Manage 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 user interface to create network map entries.
- The Configuration Tool enables you to extract a Connect:Direct for Microsoft Windows network map as a text file from the Microsoft Windows Registry. You can then update the network map and insert it into the Registry to apply it.

## Related concepts

[Network Map Communications Mode Object Worksheet](#)

[Network Map Communications Path Object Worksheet](#)

[Network Map Remote Node Object Worksheet](#)

[About the Client Connection Utility](#)

## About Defining the Network Map

You create a remote node entry in the network map for each remote node that the local node communicates with. 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.

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

## Define 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.
- Communications mode—Information relating to session characteristics for a protocol, such as TCP/IP information. A communications mode can be associated with one or more communications paths or remote node definitions.
- 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. Each communications path definition specifies a default communications mode.

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 in which 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.

## Related tasks

[Adding or Modifying Communications Mode Settings](#)

[Creating or Modifying a Remote Node Definition](#)

[Creating or Modifying a Communications Path Definition](#)

# Creating or Modifying a Remote Node Definition

## About this task

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:

## Procedure

1. Select Admin > Netmap to open the network map.
2. Do one of the following:
  - To add a node, select Netmap > Insert and type a node name in the **Name** field.  
**Important:** Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$ . \_ - to ensure that the entries can be properly managed by Control Center, Sterling Connect:Direct Browser User Interface, or IBM Sterling Connect:Direct Application Interface for Java for Java (AIJ) programs.
  - To modify an existing node, double-click the node name in the **Netmap** window.
3. Define information in the following fields for the remote node you are configuring on the **Main** tab. Refer to the following table for information on each field:

Field Name	Description	Valid Values
Name	The name of the remote Connect:Direct node. If you are modifying a node, this field cannot be edited.	A 1- to 16-character alphanumeric string
Operating System	The operating system for the remote node.	OpenVMS   OS/390   OS/400   Tandem   UNIX   VM   Windows
Max Pnode Sess	The maximum concurrent connections for all remote nodes when the local Connect:Direct node originates the Process. This field 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 Microsoft Windows, this field is limited to 1.
Max Snode Sess	The maximum concurrent connections, where the local Connect:Direct for Microsoft 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 Microsoft Windows, maximum SNODE sessions are limited to 2.



Field Name	Description	Valid Values
Default Class	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	The number of retry attempts if a short-term connection failure occurs. 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	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.
Long Term Retry Attempts	The number of retry attempts after all of the short-term retry attempts are used.	A numeric value from 0–255. The default is 0.
Long Term Retry Interval	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. The default is 00:03:00, or 3 minutes.

4. To configure TCP/IP settings, click the **TCP/IP** tab and set the TCP/IP attributes. Refer to the following table for definitions of the fields:

Field Name	Description	Valid Values
Host/IP Address	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.
Port/Service	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	Description	Valid Values
Alt Comm Outbound	<p>The 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.</p> <p>When the local node is the SNODE, the alternate addresses are used in the Netmap check.</p> <p><b>Note:</b> This parameter should not be used in a outbound Process if the SNODE is Connect:Direct/Plex.</p>	<p>Fully qualified host name or IP address and port number.</p> <p>A comma separates the list of alternate communication paths as shown in the following example:</p> <p>salmon;9400, 10.20.40.65;9500</p> <p>The list is processed from the top down.</p>
Alternate Comminfo	<p>Provides support for establishing netmap-checked sessions with high-availability (especially load balancing) systems with multiple IP addresses, such as Connect:Direct/Plex for z/OS. Use this parameter to list all IP addresses or host names that are part of the multiple IP address environment.</p> <p>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.</p> <p>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 configuration allows netmap checking to succeed when the remote node connects to the local node using any of the possible outgoing IP addresses specified.</p>	<p>hostname1/IP address, hostname2/IP address, hostname3/IP address</p> <p>host name—Host name associated with the IP address. For example:</p> <p>hops (where hops is a machine on the local domain)</p> <p>hops.domain.com (fully-qualified host name)</p> <p>nnn.nnn.nnn.nnn or nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn—IP address of a machine running Connect:Direct</p>

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

6. Perform the following actions as required:

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

- To add a new path to the network map, click **New**.
  - To delete a path, select the path in the **Available Paths** box and click **Delete**.
  - To select all available paths, click **Add All**.
  - To remove a selected path, select the path in the **Selected Paths** box and click the left arrow button.
  - To remove all selected paths, click **Remove All**.
7. To add a description of the node, click the **Description** tab. Connect:Direct does not use this information. Refer to the following table for a description of each field:

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

8. Click **OK**.

### Related concepts

[Define Remote Node Information](#)

## Adding or Modifying Communications Mode Settings

### About this task

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

### Procedure

1. Select Admin > Netmap to open the network map.
2. Do one of the following:
  - To modify an existing mode definition, double-click the mode.
  - 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 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	Description	Valid Values
Name	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- to 48-character alphanumeric string
Protocol	The type of communications protocol.	TCP/IP TCP/IP is the default value.

4. To add or update TCP/IP settings, click the **TCP/IP** tab. Refer to the following table for a description of each field:

Field Name	Description	Valid Values
Buffersize	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	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	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 <code>crc.override</code> parameter. When <code>crc.override</code> is enabled in the initialization parameter, only the nodes that require a different configuration need to be changed.	OFF   ON   blank

5. Click **OK**.

### Related concepts

[Define Remote Node Information](#)

## Creating or Modifying a Communications Path Definition

### About this task

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

### Procedure

- Do one of the following:
  - To modify a communications path definition, double-click the definition in the **CommPath** window.
  - To add a new communications path, right-click the **CommPath** box and click **Insert**.
- To define a communications path, type the name of the communications path in the **Name** field, and select the protocol to associate with it: TCP/IP.
- Define a new mode or assign an existing mode to the communications path.
- To associate an existing mode with the communications path, select the mode from the **Selected Mode** drop-down list.
- To define a new mode to associate with the communications path:

- Click **New**.
  - Type the name of the new mode in the **Name** field.
  - Select the protocol to associate with the mode: TCP/IP, and click **OK**.
6. 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	Description	Valid Values
Buffersize	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	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	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 <code>crc.override</code> parameter. When <code>crc.override</code> is enabled in the initialization parameter, only the nodes that require a different configuration need to be changed.	OFF   ON   blank

7. Click OK.

### Related concepts

[Define Remote Node Information](#)

[Specify IP Addresses, Host Names, and Ports](#)

## Defining a New Mode for a Communications Path

### About this task

To define a new mode for a communications path:

## Procedure

1. Select Admin > 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 UDT 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	Description	Valid Values
Buffersize	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	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	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 <code>crc.override</code> parameter. When <code>crc.override</code> is enabled in the initialization parameter, only the nodes that require a different configuration need to be changed.	OFF   ON   blank

8. Click **OK** twice to close Mode Properties and Path Properties.

## Deleting a Network Map Entry

### About this task

To delete a network map entry:

### Procedure

1. Select Admin > Netmap.
2. Right-click the network map entry you want to delete and click **Delete**.

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 the dialog box and click **Apply**.

## Validating a Network Map Entry

### About this task

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 message and explains why you received it.

To validate a network map entry:

### Procedure

1. Select Admin > Netmap.
2. Select the network map entry to validate.
3. Right-click the **Netmap for Nodename** window and click **Validate**.
4. After you validate the network map entry, close the **Netmap for Nodename** window.

## Viewing a Network Map as Text

### About this task

To view a network map entry as text:

### Procedure

1. Select Admin > Netmap.
2. Select the network map entry you want to view as text.
3. Right-click the **Netmap for Nodename** window to open the shortcut menu and click **Text View**.
4. Right-click the window again and select List View to view the network map entry in its original presentation.

## Applying a Network Map

### About this task

To apply a network map file to the node:

### Procedure

1. Select Admin > Netmap > Validate. If no errors are found, the output window contains the message Netmap validation Successful.
2. Select Netmap > Apply.
3. Select the node name and click **OK**.

# Printing and Viewing Node and Network Map Definitions

## About this task

After you set up a node and define a network map, 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 Microsoft Windows Registry where they are applied while the server is running, or you can apply them during a new Connect:Direct installation.

## Procedure

1. Click **Start** and point to **All Programs > IBM Connect:Direct > v6.1> CD Configuration Tool**.
2. Select **File > Extract**.
3. Click **OK**. The network map and User Authorization information for the node are extracted as cascading windows named Map and User.
4. To save the extracted user authorization information:
  - a) Click the **User** dialog box.
  - b) Select **File > 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) Select **File > 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**.

## Related concepts

[Configuring the Connect:Direct Local Node](#)

## View the Sample Configuration Files

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 Microsoft 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 *IBM Connect:Direct for Microsoft Windows Getting Started Guide* for 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, communications path, 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 Microsoft 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 Microsoft Windows Registry.



## Customizing Configuration Files

### About this task

You use the Configuration Utility to customize configuration files for your environment and prepare them for rollout. You can edit configuration information using this tool. You can also use it to add encrypted passwords to user proxies and validate the configuration files for use with Connect:Direct.

### Procedure

1. Select File > Open.
2. Select the drive and directory where the configuration file is located.
3. Select the file and click **Open**.

**Note:** 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. Select File > Save.

## Adding an Encrypted Password for a User Proxy

### About this task

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:

### Procedure

1. Open the User.cfg file.
2. Place the cursor on the line following the Proxy section header.
3. Select Tools > Password.
4. Type the user ID and password in the **Userid** and **Password** fields.
5. Retype 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

### About this task

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:

### Procedure

1. Open the configuration file to validate.

2. Select Tools > Validate. The file is validated.
  - If no errors are detected, a message indicating this is displayed. Click **OK** to return to the file.
  - 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

### About this task

You can update the network map and user authorizations, including proxies and group authorizations, by adding the updated configuration files to the Microsoft Windows Registry. After they are added, the settings are applied when the server is running.

**Restriction:** You cannot use CDConfig to change existing objects in the network map and user authorizations. It can only be used to create new netmap or user authorization objects; however, you can use CDConfig to change individual initialization parameters.

The CD Configuration Utility (CDConfig.exe) uses the following command-line parameters:

<i>Table 4. CDConfig.exe Command-Line Parameters</i>	
<b>Parameter</b>	<b>Description</b>
/q	Silently runs the utility while the file is extracted or inserted.
/i	Inserts the configuration file into the Registry (use with <i>/filename</i> ).
<i>/filename</i>	File to insert into the Registry.
<i>/pfilename</i>	Extracts initialization parameters.
<i>/mfilename</i>	Extracts netmap.
<i>/ufilename</i>	Extracts user configuration.

To create and apply user and netmap files in silent mode, type the following commands:

### EXTRACT CONFIGURATION

1. To extract initialization parameters to a Initparms.cfg file:

```
CDConfig.exe /pC:\MyDir\Initparms.cfg /q
```

2. To extract a netmap to a Map.cfg file:

```
CDConfig.exe /mC:\MyDir\Map.cfg /q
```

3. To extract user configuration to a User.cfg file:

```
CDConfig.exe /uC:\MyDir\User.cfg /q
```

### INSERT CONFIGURATION

1. To update (individual) initialization parameters from an Initparms.cfg file:

```
CDConfig.exe /i /fC:\MyDir\Initparms.cfg /q
```

2. To insert new netmap objects from a Map.cfg file:

```
CdConfig.exe /i /fC:\MyDir\Map.cfg /q
```

3. To insert new user authorization objects from a User.cfg file:

```
CdConfig.exe /i /fC:\MyDir\User.cfg /q
```

Review the CdConfig.log file to verify success.

## Stop IBM Connect:Direct for Microsoft Windows

---

### Stopping Connect:Direct for Microsoft Windows

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

- Issue the Stop command from Connect:Direct Requester or the command line interface (CLI).
- Use the Services facility on the Microsoft Windows Control Panel.
- Use the Admin Tool utility.

### Stopping Connect:Direct for Microsoft Windows from Connect:Direct Requester

#### About this task

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

#### Procedure

1. Select Admin > Netmap to open the node you want to stop.
2. Select Admin > Stop Node.
3. Select one of the stop server options:
  - 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.
  - Quiesce after active processes complete—Stops the server when all executing Processes are complete. No new Processes are started.
  - 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.
4. Click **OK** to stop the node. If the server is stopped, the icon in the Control Panel nodes view is unavailable.

### Stopping Connect:Direct for Microsoft Windows Using the Services Facility

#### About this task

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

#### Procedure

1. Click **Start > Settings > Control Panel > Administrative Tools > Services**.
2. Select the Connect:Direct node you want to stop.

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

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

## Stopping Connect:Direct for Microsoft Windows Using the CLI

### Procedure

- To stop Connect:Direct for Microsoft Windows from the CLI, issue the stop command.

### Related concepts

[Stop Connect:Direct](#)

## Stopping Connect:Direct for Microsoft Windows from the Admin Tool Utility

### About this task

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

### Procedure

1. Select the Connect:Direct for Microsoft Windows server to stop.
2. Click the red traffic light icon on the toolbar.

## Create a Process

---

### About 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
- Building the group of tasks that perform the work by adding commands
- Setting command options
- Validating Process content
- Saving the Process file

Before creating a Process, you can establish preferences related to Processes and other Connect:Direct Requester operations.

### Establishing Preferences

#### About this task

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:

## Procedure

1. From the Connect:Direct Requester Main Window, select Tools > Options.
2. To set general preferences, change the values in the fields on the **General** tab of the **Options** panel.

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 directly change the label field on the Program Definition Editor (PDE) and eliminate the need to edit the labels through the Process Properties page.
Activity log	<p>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.</p> <p>Check Enable the activity log in the Output window to display the activity log there.</p> <p>Check Write the activity output to a file and type the name of the file 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 startup.</p>

3. To set Process preferences, click the **Process Defaults** tab and change the values.

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.	<p>No—Execute a Process as soon as resources are available.</p> <p>Yes—Hold a Process.</p> <p>Call—Hold a Process until the remote node connects to the local node. At that time, the Process is released for execution.</p>
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.	<p>No—Do not retain Processes after execution.</p> <p>Yes—Retain Processes after execution.</p> <p>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.</p>
Plexclass	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. 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 OFF—Do not perform CRC checking. ON—Perform CRC checking. 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	The computer name of the user to receive Process completion messages.	A valid computer name.
Accounting Data	An arbitrary string 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.

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
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. 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.

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.

Field	Description
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 to associate with Connect:Direct.

File Extension	Description
.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 the values.

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 Filter Using Selection Criteria Pages 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 Refresh on open Autoscroll—Display the latest statistics

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

Field	Description	Valid Values
Monitor	Determine if all Processes will be monitored or only selected Processes, based on selection criteria.	All Processes 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 Refresh every _ seconds.	Refresh every _ seconds Refresh on open

9. Click **OK**.

## Creating a Process

### About this task

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

## Procedure

1. Select File > 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:
  - Select the node from the drop-down menu.
  - Type the name of a Connect:Direct node.
  - Specify an IP address and port, using the following format:

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

7. Click **OK**.
8. Add commands as necessary to the Process.
9. If desired, change the options for the Process.

### Related concepts

[Specify IP Addresses, Host Names, and Ports](#)

[Commands and Statements](#)

[Process or Command Options](#)

## Commands and Statements

You can add commands and conditional statements to a Process statement to perform various functions. When you add a command to a Process, you use the preferences you defined or you set unique values for each command.

The following table lists the commands you can insert in a Process.

Command	Description
Copy	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.
Run Task	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.



Command	Description
Run Job	<p>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.</p> <p>Use the Run Job command to perform the following types of tasks:</p> <ul style="list-style-type: none"> <li>• Submit jobs in an z/OS environment to the internal reader, a facility that transfers jobs to the job entry subsystem (JES).</li> <li>• Submit a job at the OpenVMS node in an OpenVMS environment.</li> <li>• Submit an OS/400 CL command to run as a separate job through the SBMJOB command.</li> <li>• Invoke a UNIX command shell and execute UNIX commands.</li> <li>• Start programs in the Microsoft Windows environment.</li> </ul>
Submit Process	<p>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.</p>

You can use the following conditional statements to perform tasks based on conditions you establish.

Statement	Description
If	<p>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.</p> <p>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.</p>
Else	<p>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.</p>
Endif (Eif)	<p>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.</p>
Goto	<p>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.</p>
Exit	<p>The Exit statement bypasses all remaining steps in a Process and ends the Process.</p>

#### **Related tasks**

[Creating a Process](#)

[Updating a Work List](#)

## **Add a Copy Statement to a Process**

### **About this task**

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 Microsoft 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 last checkpoint of the file that is not completely copied.

## Procedure

1. Open a Process file.
2. Select Process > Insert > Copy.
3. To identify the step within the Process, type a label of up to 8 alphanumeric characters in Copy Statement Label.
4. Select one of the following actions:
  - To copy a file to the SNODE, select Send.
  - To copy a file from the SNODE, select Receive.
5. Type the name of the source file in the Source Filename field.
6. Enter the name of the file in the Destination Filename field.
7. Select one of the following destination disposition options:
  - NEW—To create a new file at the destination
  - RPL—To replace the information in 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.
  - SHR—To replace the information in an existing file.
8. To enter compression and checkpoint restart options, click the Transfer tab and select from the following options:
  - To use compression, select one of the following compression types in the Compression window:

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 compare

Compression Option	Description	Valid Values
Extended	<p>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.</p> <p>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.</p> <p>Memory identifies how much virtual memory is allocated to maintain the internal compression state.</p>	<p>Comp Level—Select the level of compression from 1–9.</p> <p>Window—Select the window size level from 9–15. The Default is 13.</p> <p>Memory—Select a memory compression from 1–9. The Default is 4.</p>

- 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.
9. To override the preferences or provide additional parameters to describe the sending and receiving files, select one of the following:
- 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.
  - To define options for the receiving file, click the To <Operating system> tab, and type the sysopts parameters. Refer to the online help for more information.
- Note:** These tabs reflect the operating system of the sending and receiving file location.
10. Click the Comment tab and add an optional description of the statement.
11. Click OK.

## Adding a Submit Statement

### About this task

Use the Submit statement to execute tasks defined in the Process.

To add a Submit statement:

### Procedure

1. Open a Process file.
2. Select Process > Insert > Submit.
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 Microsoft Windows node, you can click the browse button to locate the Process file.
5. To override the Process name, type a 1- to 8-character 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 Track Execution in Output** Window to view activity during execution.
8. Continue defining the command.

## Adding a Run Task Statement

### About this task

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

To add a Run Task statement:

### Procedure

1. Open a Process file.
2. Select Process > Insert > Run Task.
3. To identify the step within the Process, type a label of up to 8 alphanumeric characters in Run Task Statement Label.
4. Select the node where the program or command will execute in the **Submit To** field.
5. Type one of the following, based on the node type, in the **Program** field:
  - Type Windows for a Microsoft 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 uppercase letters.
  - For HP NonStop, type the name of the object file.
6. Use the Optional Parameters or Commands as necessary, for the operating system you selected in Step 5. Refer to the Help for syntax.
7. Click the Comment tab and add a description of the command. This information is optional.
8. Click **OK**.

### Related concepts

[Process or Command Options](#)

## Adding a Run Job Command

### About this task

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.

**Restriction: You cannot execute IBM Connect:Direct HP NonStop commands using Run Job.**

### Procedure

1. Open a Process statement.

2. Select Process > Insert > Run Job.
3. Type the statement label in the **Run Job Statement Label** field.
4. Select the node where the job will execute.
5. Type the Filename based on the operating system used by the node. This field is valid only for the following operating systems:
  - For Microsoft 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 exist on the z/OS node where the job will execute. A data set containing JCL is limited to a record length of 80 bytes.
  - For i5 Series nodes, type i5 Series.
  - For UNIX, type dummy.
  - For OpenVMS, type PGM=VMS.
6. Type any Optional Parameters or Commands.
7. Click **OK**.

### Related concepts

[Process or Command Options](#)

## Adding an If Statement

### About this task

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.

### Procedure

1. Select Process > Insert > If.
2. Enter the statement label in the **If Statement Label** field.
3. Select a step label on which 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 condition.
16	Indicates a catastrophic error.

6. To add optional comments, click the Comments control tab.
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

### About this task

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:

### Procedure

1. Select Process > Insert > Else.
2. Optionally, add a comment to the Process.
3. Click **OK** to save. The Else statement is displayed in the **Process** window.

## Adding an Endif Statement

### About this task

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 use the Endif statement:

### Procedure

1. Select Process > Insert > End If.
2. Optionally, type a comment for the Process.
3. Click **OK** to save. The Endif statement is displayed in the **Process** window.

## Adding or Modifying a Goto Statement

### About this task

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:

### Procedure

1. Take one of the following actions:
  - To create a new Goto statement, select Process > 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 optional comments, click the **Comments** tab and type the comment.
3. Click **OK** to save the Process. The Goto statement is displayed in the **Process** window.

## Add an Exit Statement

### About this task

To add an Exit statement to a Process:

## Procedure

1. Select Process > Insert > Exit.
2. Optionally, click the **Comments** tab and add a comment to the Process.
3. Click **OK** to save the Process and exit the Exit Statement dialog box. The Exit statement is displayed in the **Process** window.

## Process or Command Options

After you create a statement, you set Process options, including when the Process is submitted, how the 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 it.

You can use one or more of the following options in a Process or command:

- Control functions identify how tasks are managed and how resources are allocated by defining default options once. Then these values are used as the default values for each new command or Process you define.
- Security options identify user IDs and passwords needed to access the SNODE and the PNODE.
- Variable values assign values to all symbolic variables before execution. The values are then substituted during execution whenever the symbolic variable is encountered.
- Accounting data as a free-form, user-defined field 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.
- Comments about the statement explain the context of the statement. When you view a Process in text format, the comments appear before the associated statement.

### Related concepts

[Submit Process Command](#)

### Related tasks

[Creating a Process](#)

[Adding a Run Job Command](#)

[Adding a Run Task Statement](#)

## Setting Security Options

### About this task

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

To set security options in the Requester:

### Procedure

1. Double-click the Process to open it and click the **Security** tab.
2. To set security for the PNODE, do the following:
  - 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, do the following:
  - 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.

## Setting Control Functions for a Command or Process

### About this task

When you run a command or submit a Process, you can set many control functions to use as the default values for each new command or Process you define.

To set control functions for a Process from the Requester:

### Procedure

1. Double-click the Process to open it 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:
  - 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.
  - No—If you do not want to retain the Process after execution.
  - 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.

**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.
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. Type the user ID in the **Notify Userid** field.
11. Click **OK** to close the dialog box or click one of the other tabs to continue modifying Process options.

## Assigning Values to Symbolic Variables

### About this task

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

To assign values to symbolic variables in a Process, from the Requester:

### Procedure

1. Double-click the Process to open it 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 or built-in variable enclosed in quotation marks in the **Variable Value** field or edit the existing value.

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.  <b>Note:</b> 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 hundredths 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. 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 yyyyymmdd date format.
%SUBDATE2	Use this parameter to substitute the submitted date in the yyyyddmm date format.

Built-in Variable	Value
%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.  <b>Note:</b> 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. To save and add the variable to the Variable list, click **Add**.
7. Click **OK** to close the dialog box or click one of the other tabs to continue modifying Process options.

## Specifying Accounting Data

### About this task

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 center, department number, satellite location, or any other type of code or identification that would benefit the management of data tracking.

To specify accounting data:

### Procedure

1. Double-click the Process in the Requester 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 the **Process Defaults** page of the **Options** dialog.
5. Click **OK** to close the dialog, or click one of the other tabs to continue modifying Process options.

## Add Comments

### About this task

Use comments to explain the context of a statement. Adding comments is helpful to explain what the statement does. This information is optional. When you view a Process in text format, comments appear after the associated statement, except in the case of the process statement, when the comment appears before the statement.

To add comments to a statement in a Process, from the Requester:

### Procedure

1. In the Process window, select the statement about which you want to add comments.

2. Select Process > 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

### About this task

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 the content of a Process:

### Procedure

1. Open the Process file.
2. Select Process > 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.

## Saving a Process

### About this task

When you have finished creating or editing a Process, save the Process for future use. Processes are stored in the Process directory.

### Procedure

1. Select File > Save.
2. Type a name for the Process including the .CDP extension.

## Copying a Process

### About this task

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:

### Procedure

1. Open the Process file.
2. Select File > 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

### About this task

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

- SNODE
- Hold
- Execution
- Class
- Priority

To change Process options:

### Procedure

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**.

## Manage 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 a local or remote Process, submit a local or remote program, and submit a 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.

You can use Work Lists in the following ways:

- Submit selected or all items in a Work List to the TCQ.
- 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.
- Create, update, monitor, edit, or validate a Work List.
- Cancel Work List operations.
- Insert a task into a Work List

### Creating a Work List

#### About this task

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.

To create a work list:

#### Procedure

1. From the Connect:Direct Requester Main Window, select File > New > Work List.
2. Add Work List items as desired.
3. To establish an automatic status monitor for the Work List as it executes, perform the following actions:
  - a) Select WorkList > 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, click the **Variables** tab. Select one of the following actions:
    - To add a new variable, type a name and value in the appropriate boxes and click **Add**.
    - To modify a variable, double click the variable. Type a new name or value and click **Add**.
    - 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.
  6. Click **OK**.

## Updating a Work List

### About this task

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:

### Procedure

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

### Related concepts

[Commands and Statements](#)

## Submitting a Work List

### About this task

You can submit all or selected items of a Work List to execute.

To submit a work list:

### Procedure

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

- To submit all items in the Work List, select WorkList > Submit.
- Highlight the items to submit and select WorkList > Submit Selected.

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

### About this task

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 work list operations:

### Procedure

- Select Work List > Cancel Execution.

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

### About this task

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



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

To edit a work list in text format:

### Procedure

1. Open the Work List that you want to edit.
2. Select Work List > Edit/View Text.
3. Edit the text as necessary using the following keyword and syntax rules. The **Work List Edit/View Text** window enables you to see the full text of all tasks in a Work List. You can edit the text directly if necessary. Put a keyword on a line by itself.

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 May have AutoMonitor and Serial Execution May have variables
Local Program Task Syntax	Must have the RUN keyword Must have the FILE keyword Must have arguments or a working directory

Syntax Type	Restrictions
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

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

### About this task

To insert a task into a Work List:

### Procedure

1. Select Work List > 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 Work List window is displayed.

## Validating a Work List

### About this task

To validate a Work List:

### Procedure

1. Open the Work List that you want to validate.
2. Select the tasks you want to validate.
3. Select Work List > Validate. The validation information is displayed at the bottom of your screen.

## Manage Processes

---

### 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.

## Manage Processes

Connect:Direct for Microsoft 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).
- 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 Microsoft Windows.
- Microsoft 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.
- 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.

### Related concepts

[Transmission Control Queue Parameters](#)

## 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

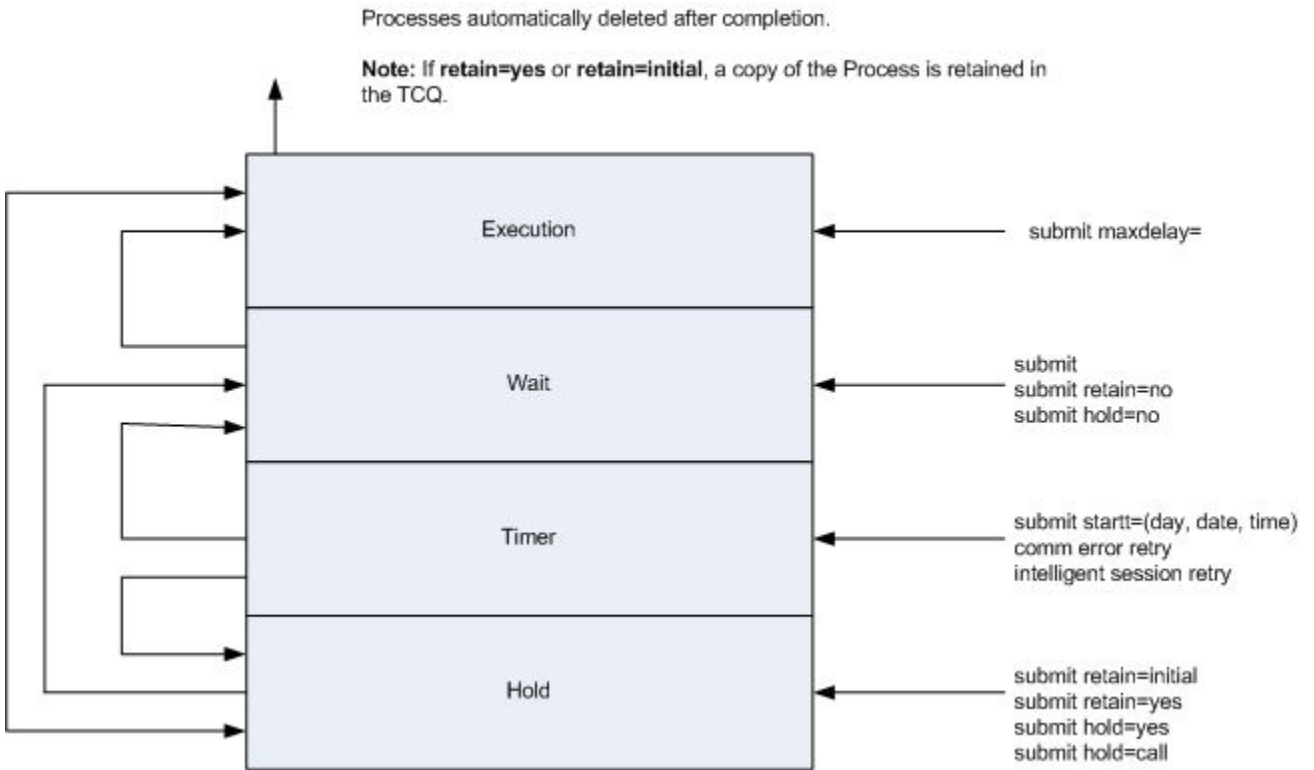
The initialization parameter **tcq.start** determines what the TCQ does with existing Processes. The default value is **tcq.start=w** (warm start), which specifies that all existing Processes in the TCQ are retained. A warm start restarts any Processes submitted with the **Process** statement parameter **retain=initial** as well as any Process that was executing in the TCQ when the server was brought down. 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:

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.

Status	Comment
WR	The Process is in retry status. The number of retries and intervals between retries is specified in the network map for the remote node.
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 or initialization parameters.
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.

## View 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

### About this task

To create a Process Monitor:

### Procedure

1. Select **File > 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.
  - 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**.

### Related tasks

[Monitoring Processes Based on Selection Criteria](#)

## Monitoring Processes Based on Selection Criteria

### About this task

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:

### Procedure

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.
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) Choose one of the following:
    - 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**.

### Related tasks

[Creating a Process Monitor](#)

## Opening a Process Monitor File

### About this task

To open a Process Monitor file:

### Procedure

1. Select Files > Open.
2. From Files of Type, select Process Monitors (\*.cdm) to display only Process Monitor files.
3. Locate and select the Process Monitor file to open.

## Saving a Process Monitor

### About this task

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.

### Procedure

1. Select File > 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.
Step Name	Process step name currently executing.
Status	Current status of the Process.
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.

<b>Field</b>	<b>Content</b>
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. 0—Success 4—Warning 8—Error 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.
Priority	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.
Source Disp 3	Source file disposition parameters.
Record Count	Number of data records read or written.
Xmit Bytes	Number of data bytes plus RU bytes sent
Xmit RUs	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

### About this task

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

- View details associated with a Process
- View the condition that caused a warning flag on a Process
- Change the order of fields displayed

To use the output display:

### Procedure

Do one of the following:

- To view the detail fields associated with a Process, double click 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, 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.

## Notification

When you install Connect:Direct for Microsoft 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 Microsoft Windows provides two notification methods:

- NT Broadcast—NT Broadcast notification is performed using the Microsoft Windows msg command.
- SMTP—E-Mail notification is performed using Simple Mail Transfer Protocol (SMTP) notification, a simple ASCII protocol.

## NT Broadcast

Connect:Direct for Microsoft Windows uses the msg 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 msg command. If the specified recipient cannot be located, or is not logged on, the notification fails and is not attempted again.

## Changing Process Notification

### About this task

To change the Process notification setup:

## Procedure

1. Validate that the Connect:Direct service has been stopped by selecting **Start > Settings > Control Panel > Administrative Tools > Services** and making sure the Connect:Direct service is stopped. If not, select the service and click **Action > Stop**.
2. Click **Start > Programs > IBM Connect:Direct > v6.1 > CD Change Notification Utility**.
3. In the **Node Name** field, type or select the local node to configure. The current notification for the node is displayed in the **Transport** field.
4. 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—SMTP server address, as the DNS name or IP address.
  - Host Port—Port to connect to the SMTP server. Default=25.
  - Sender—e-mail address uses for the sender.
  - Turn on Authentication to enable it. Provide a user ID and password to sign on to the SMTP server.
5. Click **Save**.
6. Click **OK**.

### Related concepts

[Specify IP Addresses, Host Names, and Ports](#)

## View 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 Microsoft Windows also uses the Microsoft 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 Microsoft Windows Event Logging facility. The event.log initialization parameter controls the statistics IDs that Connect:Direct sends to the Microsoft 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 HHMMSS.TTT.
SeqNum	Sequence number.
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.
SubmitterNode	Node that the Process was submitted on.
SubmitterUserID	User that submitted the Process.



Field	Description
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 Microsoft 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.
Step Name	Name of the Process step.

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

## Delete 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.

## Control 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 [“Statistics Parameters”](#) on page 126 for a description of the statistics parameters.

## Creating a Statistics Report

### About this task

To create a statistics report:

### Procedure

1. To open a new statistics monitor, select File > New > Statistics Monitor.
2. To modify an existing statistics monitor, select File > Open and select from the Statistics 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.
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 Statistics 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**.

### Related tasks

[Selecting Statistics Based on Process Name or Number](#)

## Selecting Statistics Based on Process Name or Number

### About this task

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 select statistics based on Process name or number:

### Procedure

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 the previous two steps until you have added all Process names to monitor.
5. To view Statistics based on Process number:
- a) Click the **New** icon.
  - b) Type the Process number and press **Enter**.
  - c) Repeat this step until you have added all Process numbers 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 the previous two steps until you have added all user IDs 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 the previous three steps until you have added all submitters 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:
    - 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.
9. 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	To limit the statistics based on error code values, select the condition code delimiters in the first drop-down box including: <ul style="list-style-type: none"> <li>• Equal to</li> <li>• Greater than or equal to</li> <li>• Greater than</li> <li>• Less than or equal to</li> <li>• Less than</li> <li>• Not equal to</li> </ul>

Option	Description
Conditional Code Range Error Codes	To limit the statistics to selected error codes, select the error code value from this drop-down box including: <ul style="list-style-type: none"> <li>• 0</li> <li>• 4</li> <li>• 8</li> <li>• 16</li> <li>• Any</li> </ul>

10. 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 to monitor are identified.
11. 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 selections, click Unselect All.
12. 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 the three previous steps until you have added all records you want to monitor.
13. Click **OK**.

#### Related tasks

[Creating a Statistics Report](#)

## 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.

Rec ID (Identifiers) are as follows:

<b>Record ID</b>	<b>Category</b>
CHGP	Event
COAC	Event
CRHT	Event
CSTP	Event
CTRC	Process
CTRM	Event
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

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:

- 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.

## Understand the Microsoft Windows Event Logging Facility

Connect:Direct for Microsoft Windows uses the Microsoft 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 Microsoft Windows subsystems. The Event Logging facility also allows for easy correlation of the various error messages that both Connect:Direct and Microsoft 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 Microsoft Windows:

Information	Connect:Direct Record IDs	Microsoft Windows Event Type
Server initialization/termination	NUIC and NUTC	Informational

Information	Connect:Direct Record IDs	Microsoft Windows Event Type
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

## Control 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 Microsoft 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
```

## Filter the Event Log

The Microsoft 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 Microsoft 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.

### Related concepts

[Select Message Command](#)

## Viewing Messages

### About this task

Use the Connect:Direct Message Lookup Utility to view the short and long text explanations for error messages.

### Procedure

1. Select **Start > All Programs > IBM Connect:Direct > v6.1 > CD Message Lookup**.
2. Enter the message ID in the **Message ID** field.
3. Click **Lookup** to display the short and long message text.

## Use the Activity Log

This Activity Log contains a list of Connect:Direct Requester 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.

## Saving an Activity Log

### About this task

The activity log information is lost when you close Connect:Direct Requester unless you set the activity log information to write to a file.

### Procedure

1. Select Tools > Options.
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) Select **Write the Activity Output to a File**.
  - b) Type the name of the file.
  - c) To create a new activity log every time you start Requester, select **Create New File at Startup**.
  - d) To append activity to the existing log file at start up, select **Append to Existing Activity File**.
4. Click **OK**.

## Manage an IBM Connect:Direct Server

---

### Manage 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 Microsoft Windows. If you want to change the server configuration or start or stop a Connect:Direct server, use the Admin Tool utility.

### Starting the Admin Tool Utility






1. Click **Start > All Programs**.
2. Click **IBM Connect:Direct > v6.1 > CD Admin Tool**.

The main window contains an icon representing the local computer and a traffic light icon for each Connect:Direct node 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 the process of starting or stopping is represented by a yellow light.



## About the Toolbar

The Admin Tool toolbar provides icons to perform frequently performed actions. The Admin Tool provides the following 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.
	Set general properties for a Connect:Direct for Microsoft Windows server. Stop the server before setting general properties.
	Display the About Admin Tool dialog box.
	Click a menu bar or toolbar item, to access a description of its function. This action does not work when a dialog box is open.

## Starting and Stopping a Connect:Direct Server

### About this task

Use the Admin Tool utility to start and stop a Connect:Direct server.

To start and stop a server:

### Procedure

1. Select the server to start or stop.
2. Select **Server > Start**.

If the server is already started, the server icon changes to yellow and then changes to green. If the server is unable to start, the light changes back to red.

## Configuring a IBM Connect:Direct Server

### About this task

To configure general, TCP/IP, and database properties for a server, use the Properties dialog box.



**CAUTION:** You must stop the Connect:Direct IBM Connect:Direct service before you configure properties. Stopping the service interrupts any Processes that are running.

To configure server properties:

### Procedure

1. Click the server to configure.
2. If the server is running, click the stop icon to stop the server.
3. Select **Server > Initialization Properties**.
4. Select one of the following Service Startup methods from the pull-down menu:
  - Automatic to start Connect:Direct IBM Connect:Direct every time the system starts.
  - Manual to start Connect:Direct IBM Connect:Direct manually.

- Disabled to prevent Connect:Direct IBM Connect:Direct from being started.
5. If necessary, change the settings on the General properties page. Refer to the following table for a description of the fields:

Field Name	Definition	Valid Values
Max API Connections	Specifies the maximum number of concurrent client API connections permitted for the local node. It defines the maximum number of connections by different userids. API connections by a single user ID has no limit.	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 Microsoft Windows server node is the originator of the Process.  This field 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 1–255
Max Snode Sessions	Specifies the maximum concurrent connections, where the local Connect:Direct server node is the partner node cooperating with a remote Connect:Direct node to execute a Process.  This field 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 1–255
TCQ Startup	Specifies how the Connect:Direct server program 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.
Max Total Sessions	Specifies the maximum concurrent connections in total, where the local Connect:Direct server node is Pnode or Snode.  This field is limited to the lesser of the sum/ values defined in the initialization parameters file and the network map definition for a given node.	A numeric value from 1–510.
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 API tab. If necessary change the default values of API fields. Refer to the following table for a description of the fields:

Field Name	Definition	Valid Values
Enable TCP/IP API Support	Enables TCP/IP API 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. API Additional Addresses and Ports	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.
API Port	Specifies the port number that the 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.
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 (IPv4) or nnnn:nnnn:nnnn:nnnn:nnnn :nnnn:nnnn:nnnn (IPv6), or the host name.  For example, 10.20.9.175;2363, fd00:0:0:2014::7; 2364  This is an optional field.
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 (IPv4) or nnnn:nnnn:nnnn:nnnn:nnnn :nnnn:nnnn:nnnn (IPv6), or the host name.  For example, 10.20.9.175:2364, mdallas;1364  This is an optional field.
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

7. Click the TCP/IP tab. If necessary change the default values of TCP/IP fields. Refer to the following table for a description of the fields:

Field Name	Definition	Valid Values
Enable TCP/IP Support	Enables TCP/IP support.	enabled   disabled

Field Name	Definition	Valid Values
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 (IPv6), or the host name.
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 (IPv4) or 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.
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. 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 **Database** tab.
9. Click **Modify Settings** to invoke the Database Wizard utility. Use the utility to configure the ODBC and database environments.
10. Select one of the following database types:
  - Microsoft SQL Server
  - PostgreSQL
11. If you selected Microsoft SQL server:
  - a) Identify the TCQ/Stats data source in the **TCQ/Stats Datasource** field.
  - b) Identify the data source for message in the **Message Datasource** field.
  - c) Identify the database name in the **Database Name** field.
  - d) To specify the SQL server, type or select the name of the SQL server or click (...) to select a network SQL server.
12. When you validate the information, click **Finish**.
13. Click **Yes** to build the database.
14. Click **OK**.

## Related concepts

[Specify IP Addresses, Host Names, and Ports](#)

## Work 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

### About this task

To add a node entry to Active Directory:

### Procedure

1. Start the Admin Tool utility.
2. Select Active Directory > Create.
3. Select the node to add and click **Add**.

## Deleting an Active Directory Entry

### About this task

To delete a node entry from Active Directory:

### Procedure

1. Start the Admin Tool utility.
2. Select Active Directory > Delete.
3. Select the node to delete and click **Remove**.

## Creating an Active Directory Report

### About this task

To create a report that identifies all nodes registered in Active Directory:

### Procedure

1. Start the Admin Tool utility.
2. Select Active Directory > Report.  
The Active Directory Report is displayed.

### Diagnose a Server Problem 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 the following levels:

- A basic trace to capture the function entry and exit only.
- An intermediate trace to capture the function argument and its start and stop.
- A full trace to capture internal control blocks and the function argument, start, and stop.

The trace timestamp is specified in microseconds to pinpoint server activity with greater precision.

You can create a trace using Requester or the CLI.

Refer to [“Traceoff Command” on page 155](#) and Traceon Command for instructions on using the CLI to define or turn off traces.

#### Related concepts

[Traceon Command](#)

[Traceoff Command](#)

### Defining a Trace

#### About this task

To define a Connect:Direct trace:

#### Procedure

1. Start Requester.
2. Select Admin > Tracing.
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. Generate a configuration report is selected by default. If you do not want to generate a configuration report, turn off this option.
6. 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.

7. Click **OK**.

## Stopping a Trace

### About this task

To stop a trace:

### Procedure

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**.

## Trace Startup Parameters

The following startup parameters for Connect:Direct traces are all optional:

Parameter	Description and Options
-ttyp=	<p>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 is valid.</p> <p>c – Starts a Command Manager trace. This trace affects all Command Managers started on behalf of users logging in to the Connect:Direct for Microsoft Windows server. It shows both the command processing flow and the interactions between the server executable and the Connect:Direct for Microsoft Windows.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>m – Starts a trace of the main-line logic that includes initialization and termination.</p> <p>a – Starts a trace of the Statistics subsystem. This trace shows all records sent to the Statistics thread.</p> <p>x – Starts all trace types. Specifying a value of x is the same as specifying cpsoma.</p>

Parameter	Description and Options
-tlvl=	Specifies the trace level. The values are: 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.
-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 Microsoft Windows server executable resides. Microsoft Windows Services treats a backslash (\) as an escape character, so type two backslashes for each backslash in the file path. Example: -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. The default is -tfsz=100K. bytes: You can specify the file size in bytes. bytesK: You can specify the file size by using the K (x1000) suffix. bytesM: 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 and Options
-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. pnum: Process number. (pnum1,pnum2,...,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. Arguments are: pnam: Process name. (pnam1,pnam2,...,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 sessionam with this local node. If you specify -tdest, Connect:Direct will only trace the specified remote node names. dest: Destination node name. (dest1,dest2,...,dest8): Trace of up to eight destination node names.



Parameter	Description and Options
-tlnode= or -tpnode=	Requests a trace of initiating node sessions only. This parameter modifies the effect of the -tpnam, -tpnum, and -tdest parameters by allowing only traces of Processes submitted on the local node.  <b>Note:</b> Connect:Direct for Microsoft Windows also supports -tpnode in place of -tlnode.
-trnode= or -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 allowing only traces of Processes submitted on remote Connect:Direct nodes.  <b>Note:</b> Connect:Direct for Microsoft Windows also supports -tsnode in place of -trnode.

## Recover from a Problem

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.

### Checkpoint Parameter for the Copy Statement

The checkpoint parameter in the Process copy statement specifies the byte interval at which checkpoints are taken.

**Note:** See the *IBM Connect:Direct Process Language Reference Guide* for use of the checkpoint parameter in the copy statement.

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 Microsoft Windows provides 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.

### Restart 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.

### Determine 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.

**Note:** For a full description of all run task statement parameters, see the *IBM Connect:Direct Process Language Reference Guide*.

## Troubleshoot Connect:Direct Problems

Before calling IBM 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

If you are troubleshooting a Connect:Direct for z/OS server, gather the following information:

- Network map parameter definitions
- Local Node source
- Remote Node source

## Worksheets

---

### 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.

**Note:** See [“Define and Manage the Connect:Direct Network”](#) on page 44 for field content.

Make a copy of this worksheet for each communications mode in the network.

Communications Mode	Information Needed
TCP/IP	Communications Mode Name Buffer Size Pacing Send Count Pacing Send Delay CRC

#### Related concepts

[Define and Manage the Connect:Direct Network](#)

### 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.

**Note:** See [“Define and Manage the Connect:Direct Network”](#) on page 44 for field content.

Make a copy of this worksheet for each communications path in the network.

Communications Path	Information Needed
TCP/IP	Communication Path Name Communications Mode

#### Related concepts

[Define and Manage the Connect:Direct Network](#)

## 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. See [“Define and Manage the Connect:Direct Network”](#) on page 44 for field content.

Make a copy of this worksheet for each remote node.

**Note:** The components below correspond to the tabs on the Netmap Node Properties dialog box.

Network Map Component	Information Needed
Main Options	Node Name Operating System Maximum Concurrent Local Node Session Maximum Concurrent Remote Node Session Default Class 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 Alt Comm Outbound (Alternate Outbound Addresses) Alternate Comminfo (Alternate Netmap-Checked Addresses)
Communications Path	Communications Path Names <b>Note:</b> You must designate at least one path.
Description Information	Contact Name Contact Phone Number Comment

### Related concepts

[Define and Manage the Connect:Direct Network](#)

## 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.

See [“Configuring the Connect:Direct Local Node”](#) on page 33 for field content. Make a copy of this worksheet for each Connect:Direct user.

<b>Type of Authorization Information</b>	<b>Information Needed</b>
Main Options	Local User ID Default User Type (choose one): <ul style="list-style-type: none"> <li>• General User</li> <li>• Administrator</li> <li>• Operator User</li> </ul>
Administrative	Indicate whether the administrator has the authority to update the following: <ul style="list-style-type: none"> <li>• User Authorizations</li> <li>• Upload</li> <li>• Remote User Proxies</li> <li>• Download</li> </ul>
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 Microsoft 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. <ul style="list-style-type: none"> <li>• Upload Directory</li> <li>• Download Directory</li> <li>• Process Directory</li> <li>• Program Directory</li> </ul>
Server Control	Grants authority to perform the following tasks in Connect:Direct: <ul style="list-style-type: none"> <li>• Stop (Connect:Direct)</li> <li>• Trace</li> <li>• Initparms (initialization parameters)</li> <li>• Update Network Map</li> <li>• Update Translation Table</li> <li>• Client Source IP Checking</li> <li>• Certificate Authentication for Client API Connections</li> </ul>
Statements Authorization	Grants authority to use the following statements in Connect:Direct Processes: <ul style="list-style-type: none"> <li>• Trace</li> <li>• Copy</li> <li>• Run Job</li> <li>• Run Task</li> <li>• Submit</li> </ul>
Process Control Submit	Grants authority to manipulate and observe Processes in the TCQ.

Type of Authorization Information	Information Needed
Overrides Authorization	Grants authority to override the following Microsoft Windows defaults: <ul style="list-style-type: none"> <li>• Execution Priority</li> <li>• ACL Updates</li> <li>• File Attributes</li> <li>• Remote Node ID</li> <li>• CRC</li> </ul>

**Related concepts**

[Configuring the Connect:Direct Local Node](#)

## Remote User Proxy Worksheet

The Remote User Proxy object defines a relationship between a remote node and a local user ID.

**Note:** See [“Configuring the Connect:Direct Local Node”](#) on page 33.

Make a copy of this worksheet for each Connect:Direct Remote User Proxy you want to define.

Type of Authorization Information	Information Needed
Main Options	Remote Node Name Remote User ID Local User ID Local User ID Password Allow the remote user to: <ul style="list-style-type: none"> <li>• Upload</li> <li>• Download</li> </ul>
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 to.  Upload Directory Download Directory Process Directory Program Directory

**Related concepts**

[Configuring the Connect:Direct Local Node](#)

## Change IBM Connect:Direct Settings

---

### Change Connect:Direct for Microsoft Windows Settings

When you install Connect:Direct for Microsoft Windows, initialization parameters are created in the Microsoft Windows registry and are used to determine settings at initialization. Initialization parameters (also called 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 Microsoft 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.
Statistics Information	Statistics settings determine the maximum age that statistics records are kept and what commands are logged in the statistics file.
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.
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.
License Parameters	The license information parameters determine the parameters used to automate license metrics collection from Connect:Direct for Windows.

### Sample Initialization Parameters Format

The following figures illustrate the format of the initialization parameters. The initialization parameters are listed in groups that are labeled in brackets.

The example below shows the miscellaneous parameters:

```

[Miscellaneous Commands]
server.path=C:\Program Files\IBM\Connect Direct v6.1.0\Server\
proc.prio.default=10
exec.prio.default=7
download.dir=C:\Program Files\IBM\Connect Direct v6.1.0\Server\Download\
upload.dir=C:\Program Files\IBM\Connect Direct v6.1.0\Server\Upload\
program.dir=C:\Program Files\IBM\Connect Direct v6.1.0\Server\Program\
restrict.cmd=N
security.exit=<None>
notify.level=A
file.exit=<None>
event.log=All
certificate.directory=C:\Program Files\IBM\Connect Direct v6.1.0\Server\Secure+\Certificates
s+cmd.enforce.secure.connection=Y
disable.proxy.password.security=N
password.exit.dll=<None>
password.exit.hash=<None>
password.exit.appl.id=<None>
password.exit.policy.id=<None>

```

The example below shows the local node parameters:

```

[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=<None>
contact.phone=<None>
descrip=<None>
name=CDPROD
sess.pnode.max=1
sess.snode.max=1
sess.total=2
sess.default=1
netmap.check=Y
node.check=B
proxy.attempt=N
protocol=1
tcp.api.port=cdprod;1363
tcp.host.port=cdprod;1364
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=0FF
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

```

The example below shows the parameters related to the Transmission Control Queue (TCQ):

```

[TCQ Information]
tcq.max.age=30
tcq.start=W
process.dir=C:\Program Files\IBM\Connect Direct V6.1.0\Server\Process\
runtask.restart=N
conn.retry.exhaust.action=hold

```

The example below shows the Global Copy parameters:



```
[Global Copy Parameters]
ckpt.interval=10240K
xlate.dir=C:\Program Files\IBM\Connect Direct V6.1.0\Server\Xlate\
xlate.send=XLATESND.CDX
xlate.recv=XLATERCV.CDX
disable.cache=N
continue.on.exception=N
ecz.cmprlevel=1
ecz.windowsize=15
ecz.memlevel=4
strip.blanks=I
record.wrap=N
retry.msgids=<None>
retry.codes=<None>
```

The example below shows the parameters related to statistical information:

```
[Statistics Information]
stat.max.age=7
log.select=N
log.commands=Y
stat.sort=Y
```

The example below shows the parameters related to license information:

```
[License Information]
license.edition=Standard
license.type=Production
license.pvu=0
```

## Changing Initialization Parameters

### About this task

You can change the Connect:Direct for Microsoft Windows initialization settings by editing the initialization parameters.

### Procedure

1. Click **Start > All Programs > IBM Connect:Direct > v6.1 > CD Requester**.
2. In Connect:Direct Requester, select **Admin > 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 equal 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 **Initparms > Apply** to update and save changes.
7. Close the Initparms dialog box by clicking the X in the upper-right-hand corner.


### 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	Description	Valid Values
certificate.directory	Default certificate directory for Connect:Direct Secure Plus commands issued from the Connect:Direct client API. If the directory is not configured, the default directory created during installation is used.	Directory path name
s +cmd.enforce.secure.connection	Determines if Connect:Direct Secure Plus commands are accepted from the Connect:Direct client API on unsecure connections.	y   n y=default. Commands from unsecure connections are not accepted. n=commands from unsecure connections are accepted
server.path	Fully-qualified path to all Connect:Direct files. Terminate the path name with a trailing backslash (\).  You cannot change the value of this parameter. If you want to change the local node name, you must reinstall IBM Connect:Direct for Microsoft Windows.	Valid, fully-qualified path name.
proc.prio.default	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 to 15, where 15 is the highest priority.  The default is 10.
exec.prio.default	The priority of the Process. The execution priority parameter is used to influence the Microsoft 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.  <b>Note:</b> Scheduling Processes with a high execution priority value can affect the execution of other applications on the system.	A numeric value from 1 to 15, where 15 is the highest priority.  The default is 7.  These values are mapped to Microsoft Windows Process priority classes and values.
download.dir	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	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

Parameter Name	Description	Valid Values
program.dir	The default working directory for a program 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	<p>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, it prevents the use of the special characters: “&amp;”, “ ”, and “&gt;”.</p> <p>To enable running of cmd tasks and the use of special characters, specify N.</p> <p>Only use restrict.cmd=Y when the controlling group or user functional authority record includes a directory restriction.</p>	<p>Y   N</p> <p>The default is N.</p>
security.exit	Specifies whether a security exit is implemented as a user exit during Process execution. See the <i>IBM Connect:Direct for Microsoft Windows SDK Programmer Guide</i> for details as presented in the sample user exit program userexit_samp.c.	Valid, fully-qualified path name to a user-defined DLL. The default is <NONE>.
notify.level	<p>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.</p> <p>Specify a value of W for a warning-level return code greater than 0. Specify a value of E for an error-level return code greater than 4.</p>	<p>A   W   E</p> <p>The default is A.</p>
file.exit	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>.

Parameter Name	Description	Valid Values
event.log	The statistics IDs the system sends to the Microsoft 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 View Process Statistics for a list of statistics IDs.	statistics ID   All   (list)  The default is All.
disable.proxy.password.security	Determines whether the Connect:Direct server returns the password in a proxy definition to a Connect:Direct client.	N   Y  N=default. The server returns a dummy password. This is the most secure setting.  Y=the server returns the real local password again.   <b>CAUTION:</b> This initialization parameter is security-related. It is highly recommended to only set disable.proxy.password.security=Y during the time while duplicating the proxy and return to disable.proxy.password.security=N afterward.
password.exit.dll	This is the Password Exit DLL File to enable the Password Exit feature. If the value is blank, the feature is disabled.	<ul style="list-style-type: none"> <li>• none (default)</li> <li>• valid fully qualified DLL file</li> </ul>
password.exit.dll.hash.	This is a SHA256 hash of the Password Exit DLL file. Before Connect:Direct Windows loads the your Password Exit DLL file, it will compute the SHA256 hash for the file. If the computed hash matches the configure Password Exit DLL Hash, then Connect:Direct Windows will load the dll and use it to obtain user passwords.  <b>Note:</b> This parameter is mandatory when CD_PSWDEXIT_DLL is enabled.	<ul style="list-style-type: none"> <li>• none (default)</li> <li>• valid SHA256 hash</li> </ul>
password.exit.policy.Id	It is sent to the Password Exit as a parameter.	<ul style="list-style-type: none"> <li>• none (default)</li> <li>• valid string</li> </ul>
password.exit.application.Id	It is sent to the Password Exit as a parameter.	<ul style="list-style-type: none"> <li>• none (default)</li> <li>• valid string</li> </ul>

## 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:

<b>Parameter Name</b>	<b>Description</b>	<b>Valid Values</b>
max.api.connects	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 to 255. The default is 10.
conn.retry.stwait	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.
conn.retry.stattempts	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	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	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	The name of a contact.	Any name description.
contact.phone	A phone number to use to contact the contact name.	Any valid phone number.
descrip	The description of the local node.	Any valid text string.
name	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 Microsoft Windows.	A 1- to 16-character alphanumeric string specified during installation.

Parameter Name	Description	Valid Values
sess.pnode.max	<p>The maximum concurrent connections for all remote nodes where the local server node is the originator of the Connect:Direct Process.</p> <p>This field is limited to the lesser of the values defined in the initialization parameters file and the network map definition for a given node.</p>	A numeric value from 1 to 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 one PNODE session.
sess.snode.max	<p>The maximum concurrent connections for all remote nodes where the local node is the partner node of a Process.</p> <p>This field is limited to the lesser of the values defined in the initialization parameters file and the network map definition for a given node.</p>	A numeric value from 1 to 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 two SNODE sessions.
sess.total	<p>The maximum number of total concurrent connections for all remote nodes.</p> <p>This field is limited to the lesser of the values defined in the initialization parameters file and the network map definition for a given node.</p>	A numeric value from 1 to 510. The default value is the sum of the sess.pnode.max and the sess.snode.max parameter values. The workstation version of the product is limited to a maximum of 3 sessions.
sess.default	<p>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.</p>	<p>A numeric value from 1 to the value specified for sess.pnode.max.</p> <p>The default is 1.</p>
netmap.check	<p>The level of network map checking that occurs for each node that you communicate with.</p> <ul style="list-style-type: none"> <li>• Y—Checks the network map for all nodes that Connect:Direct will communicate with to validate the node name and the IP address.</li> <li>• L—Checks the network map only for nodes that the local Connect:Direct will initiate sessions with.</li> <li>• R—Checks the network map only for remote nodes that will communicate with this node.</li> <li>• N—Does not validate any session establishment requests in the network map.</li> </ul>	<p>Y   L   R   N</p> <p>The default is Y.</p>

Parameter Name	Description	Valid Values
node.check	The level of node checking. B—Checks both the node address and the node name. C—Checks the node name only.	B  C The default is B.
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   N The default is N.
protocol	The communications protocol or protocols to be used by the local node. If more than one, separate entries with commas.	TCP TCP—Specifies TCP/IP. This is the default.
tcp.api.port	One or more IP addresses (or host name) and optional port numbers used to establish client sessions with this Connect:Direct node.	[IP address   hostname][;port number] The default port number is 1363. See <a href="#">“IP Addresses” on page 129</a> .
tcp.host.port	One or more 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 Microsoft Windows node. One or more IP addresses (or host names) and/or port numbers that remote Connect:Direct nodes will connect to for Process execution with this local Connect:Direct for Microsoft Windows node. You can specify IP address/hostname, IP address/hostname and port, or just a port.	[IP address   hostname][;port number] The default port number is 1364. See <a href="#">“IP Addresses” on page 129</a> .

Parameter Name	Description	Valid Values
outgoing.address	<p>A virtual IP address for adjacent nodes in a cluster (in a high-availability environment) to use for netmap checking by the remote node.</p> <p>In a Connect:Direct for Microsoft Windows high-availability cluster, each instance of Connect:Direct for Microsoft Windows has a separate IP address and a virtual IP address assigned to the cluster. After all of the Connect:Direct for Microsoft 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.</p> <p>Be sure remote nodes specify this virtual IP address in their netmap entries for each node in the high-availability cluster.</p>	<p>IP address   hostname</p> <p>See <a href="#">“IP Addresses” on page 129.</a></p>
tcp.src.ports	<p>An IP address or multiple addresses and the ports permitted for the address when going through a packet-filtering firewall.</p>	<p>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:</p> <p>(199.2.4.*, 1024),  (fd00:0:0:2015:&gt;::*, 2000-3000),  (199.2.4.0/255.255.255.0,  4000-5000),(fd00:0:0:2015::0/48,  6000, 7000)</p> <p>See <a href="#">“IP Addresses” on page 129.</a></p> <p>A wildcard character (*) is supported to define an IP address pattern. If the wildcard character is used, the optional mask is not valid.</p>



Parameter Name	Description	Valid Values
udp.src.ports	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:  (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)  See “IP Addresses” on page 129.  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	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.  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.
udp.src.ports.list.iterations	The number of times that Connect:Direct scans the list of available ports defined in udp.src.ports to attempt a connection before going into a retry state.  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	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 to 65536.  The default is 65535.
pacing.send.delay	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).  The default is 00.00.00.

<b>Parameter Name</b>	<b>Description</b>	<b>Valid Values</b>
pacng.send.count	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 to 32768. The default is 0.
tcp.crc	Globally turns on or off the CRC function for TCP/IP processes.	ON   OFF The default is OFF.
udp.crc	Globally turns on or off the CRC function for UDP processes.	ON   OFF The default is OFF.
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   N The default is N.
udp.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   N The default is N.
tcp.max.time.to.wait	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.
udp.max.time.to.wait	The maximum time to wait for each pending UDP 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	Description	Valid Values
tcp.window.size	<p>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 Microsoft 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.</p> <p>In order for this parameter to take effect, ensure that the following criteria are met:</p> <ul style="list-style-type: none"> <li>• RFC1323 support is enabled on the system.</li> <li>• The system's maximum TCP window size is larger than the configured value.</li> </ul> <p>The formula for determining the optimum window size is as follows:  BDP (bytes) = 125 x &lt;bandwidth (Mb/s)&gt; x &lt;roundtrip delay in ms&gt;</p> <p><b>Note:</b> This system configuration must be done on any Microsoft Windows system involved in the transfer, regardless of whether it is a PNODE or an SNODE.</p>	<p>A numeric value from 0 to 2147483646.</p> <p>The default is 0.</p>
runstep.max.time.to.wait	<p>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."</p>	<p>A 24-hour time value formatted as hh:mm:ss. The maximum value is 23:59:59. The default is 00:00:00.</p>

Parameter Name	Description	Valid Values
active.directory.enabled	Specifies whether Connect:Direct for Microsoft 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   N The default is N.
quiesce.resume	Specifies whether testing mode is enabled for Connect:Direct for Microsoft 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 Microsoft Windows is installed.  See <a href="#">“Use Connect:Direct in Test Mode” on page 165.</a>	Y   N The default is N.
tcp.api.inactivity.timeout	The number of seconds of session inactivity to wait before exiting a session. This helps prevent maximum connections (api.max.connects) being reached because of abrupt disconnections that do not free up resources in an orderly manner.  If you are using IBM Control Center to monitor your Connect:Direct for Microsoft Windows server, set this value to at least twice the value of the Monitor Rest Time setting in Control Center.	A numeric value from 0 to 32767. The default is 0.

## 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.

See [Manage Processes](#).

The following table identifies the TCQ information parameters:

Parameter Name	Description	Valid Values
tcq.max.age	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 to 999. The default is 30 days.

Parameter Name	Description	Valid Values
tcq.start	Specifies how to start the Connect:Direct node with respect to the TCQ.	W   C w (warm)—Retain all existing Processes in the TCQ at startup. This is the default. c (cold)—Delete all existing Processes in the TCQ at startup.
process.dir	The default directory a Process runs from if a submit statement does not specify a fully-qualified path.	Valid, fully qualified path name. The default is X:\installation directory\PROCESS.
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   N The default is N.
conn.retry.exhaust.action	Action to take after the specified short and long-term retries have been used.	Hold   Delete <ul style="list-style-type: none"> <li>Hold - Places Processes in the hold queue in "Held in Error" status after all retry attempts are used.</li> <li>Delete - Causes the Processes to be deleted from the TCQ.</li> </ul> The default is Hold.

### Related concepts

[Manage Processes](#)

## 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 *IBM Connect:Direct Process Language Reference Guide* for a description of the copy Process statement.

Parameter Name	Description	Valid Values
ckpt.interval	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   bytesK   bytesM The default is 10240K bytes.
xlate.dir	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	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	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 Microsoft Windows file cache.	Y   N The default is N.
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.  Y—Attempt to continue processing. N—Go into HOLD status.	Y   N The default is N.
ecz.cmprlevel	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.	1   2   3   4   5   6   7   8   9 The default is 1.
ecz.windowsize	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   13   14   15 The default is 15.
ecz.memlevel	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   4   5   6   7   8   9 The default is 4.
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 Microsoft Windows text file or ignored (I). The strip.blanks parameter is ignored when datatype(binary) is specified.	Y   N   I The default is I.

Parameter Name	Description	Valid Values
record.wrap	<p><b>Note:</b> This parameter is needed only in certain circumstances because it restructures the data.</p> <p>Influences the way that a sending copy step works when a logical record size (LRECL) is specified for the remote platform.</p> <p>If record.wrap is set to N, and a record length greater than LRECL is encountered in the source file, Connect:Direct for Microsoft Windows reports an error. This is the desired behavior in most cases.</p> <p>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.</p>	<p>Y   N</p> <p>The default is N.</p>
retry.msgids	<p>The message IDs to use to support a file allocation retry attempt.</p> <p>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.</p> <p>You can perform retry attempts based on codes only, message IDs only, or a combination of the two.</p> <p>When a retry condition is detected, the session is terminated cleanly and the Process is placed in the Timer queue.</p>	<p>Any of the valid file allocation retry messages.</p>

Parameter Name	Description	Valid Values
retry.codes	<p>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.</p> <p>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.</p> <p>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.</p> <p>You can perform retry attempts based on codes only, IDs only, or a combination of the two.</p> <p>When a retry condition is detected, the session is terminated cleanly and the Process is placed in the Timer queue.</p>	Any valid error code

## 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	Description	Valid Values
stat.max.age	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 to 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   N The default is N.
stat.sort	Determines whether statistics are sorted by timestamp date.	Y   N The default is Y.



Parameter Name	Description	Valid Values
log.commands	Specifies whether Connect:Direct logs all commands issued from Connect:Direct for Microsoft 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.	Y   N The default is Y.

### Related concepts

[Control Statistics File Content](#)

## Install Agent Parameters

The install Agent parameters determine the Agent installation configuration parameters.

The following table lists the Install Agent commands parameters:

Parameter Name	Description	Valid Values
agent.port	Port details to configure the Agent listening port that Control Center Director will use to communicate with the Agent.  With the port configuration complete, Agent is now set to automatically listen for incoming connections from Control Center Director.	Default: 1365
osa.rest.url	Provide the Event Repository URL to configure the Control Center Director Open Server Architecture (OSA) URL, the target location where Agent posts all the events to Control Center Director.	Valid, OSA URL.  <code>osa.rest.url=https://&lt;ip/hostname&gt;:port&gt;/osa/events/post:</code>  The default is None.
osa.disable	Enables Agent to post all events to Control Center Director except when set to <b>Y</b> .	Default: <b>N</b>

## License Information Parameters

The license information parameters determine the parameters used to automate license metrics collection from Connect:Direct for Windows.

The following table lists the miscellaneous commands parameters:

Parameter Name	Valid Values
license.edition	<ul style="list-style-type: none"> <li>• Premium</li> <li>• Standard</li> <li>• Solo</li> <li>• Default: Blank (undefined)</li> </ul>
license.type	<ul style="list-style-type: none"> <li>• Production</li> <li>• Non-Production</li> <li>• Default: Non-Production</li> </ul>
license.pvu	<p>A non-negative integer</p> <ul style="list-style-type: none"> <li>• The license.pvu parameter is only applicable for Connect:Direct Premium licenses</li> <li>• This value can be calculated using the IBM License Metric Tool (ILMT) or it can be looked up at the <a href="#">IBM Processor Value Unit licensing website</a>.</li> <li>• Default: 0</li> </ul>

## New Install Task Parameters

Following parameters (initparms) are used to automate new installation of Connect:Direct server for Windows from [IBM Sterling Control Center Director](#).

Parameter (initparm)	Definition	Possible Values
CD_AGENT_ENABLE	Use to enable the agent installation	<ul style="list-style-type: none"> <li>• y (Default)</li> <li>• n</li> <li>• blank</li> </ul>
CD_AGENT_OSA_DISABLE	Use to disable the agent installation	<ul style="list-style-type: none"> <li>• y (Default)</li> <li>• n</li> <li>• blank</li> </ul>
CD_AGENT_INSTALLATION_ID	Use to store initparms configuration	<ul style="list-style-type: none"> <li>• blank (Default)</li> <li>• any string (maximum length: 1023 bytes)</li> </ul>
CD_TRUSTEDCERT_FILE	Specifies the trusted certificates to be imported	<ul style="list-style-type: none"> <li>• File path</li> <li>• blank (Default)</li> </ul> <p><b>Note:</b> Trusted certificates are not imported if the parameter is not specified or left blank.</p>

Table 5. Initialization Parameters (continued)

Parameter (initparm)	Definition	Possible Values
CD_SECUREPLUS_FILE	Specifies the file containing additional SPCLI commands to configure on CD.	<ul style="list-style-type: none"> <li>File path</li> <li>blank (Default)</li> </ul> <p><b>Note:</b> Splice command file will not be executed if the parameter is not specified or left blank.</p>
CD_AGENT_PORT	Port details to configure the Agent listening port that Control Center Director will use to communicate with the Agent. With the port configuration complete, Agent is now set to automatically listen for incoming connections from Control Center Director	1365 (Default)

## Specify an IP Address

### Specify IP Addresses, Host Names, and Ports

Connect:Direct for Microsoft 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

#### Related tasks

[Creating or Modifying a Communications Path Definition](#)

[Creating a Process](#)

[Changing Process Notification](#)

[Configuring a IBM Connect:Direct Server](#)

### IP Addresses

Connect:Direct for Microsoft Windows accepts both IPv4 and IPv6 addresses. Wherever an IP address is specified in Connect:Direct for Microsoft Windows, you can use either IPv4 or an IPv6 addresses.

#### IPv4 Addresses

IPv4 supports 2<sup>32</sup> addresses written as 4 groups of dot-separated 3 decimal numbers (0 through 9), for example, 10.23.107.5.

## IPv6

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::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
```

or

```
0000:0000:0000:0000:0000:ffff:0102:0304
```

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 Microsoft Windows does a DNS lookup to get the IP address from the operating system. The first IP address returned in the DNS lookup is used regardless of whether it is in IPv4 or IPv6 format.



**Warning:** If `tcp.api.port` or `tcp.host.port` are defined with a host name, the binding IP address is determined by DNS lookup. As a result, either an IPv4 or IPv6 address is returned depending on your DNS configuration.

A host name (net, host, gateway, or domain name) is a text string of up to 24 characters comprised of the alphabet (a-z and 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 semicolon (;), for example, `10.23.107.5;1364`. This convention is specific to Connect:Direct for Microsoft 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 semicolon (;), 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.

## Use 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 use a specific port, multiple ports, or a range of ports.

Connect:Direct for Microsoft 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.*, 1000), (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.\*, 1000)—Any IPv4 address that falls in the range from 199.2.4.0 through 199.2.4.255 and uses only port 1000.

(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 and uses 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 and uses 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 and uses 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.

**Restriction:** Masks in hexadecimal format are not supported in Connect:Direct for Microsoft Windows.

## Submit a Process Using the Command Line Interface

---

### Submit Processes Using the CLI Command

The Command Line Interface (CLI) provides another tool to submit Process statements and commands to the Connect:Direct server.

See [“Creating a Process” on page 61](#) 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 a Process
- Delete a Process
- Retrieve Process status information
- Retrieve Process statistics records

### Creating a Configuration File to Connect to a Server

#### About this task

If you want to connect to a Connect:Direct server using the CLI, use the Logon Connection utility.

This utility can be used to run batch-oriented jobs. It enables you to identify the parameters necessary to connect to a server. It then saves this information to a configuration file. If you do not identify a directory, the file is saved in the root directory.

After you create the configuration file, invoke the CLI and include the -f parameter, to identify the location and file name for the configuration file. Then define a user ID and password in the Microsoft Windows setup utility.

To create the configuration file:

## Procedure

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 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.

## Invoke the CLI

### About this task

To run the command line client:

### Procedure

1. From a command prompt, navigate to the Common Utilities directory where the CLI is installed or define the CLI location in the user's path.
2. To start the CLI, type the following command, including any of the parameters you wish to specify:

```
direct -nNodeName -uUserId -pPassword -mScrollLineCount -l -x -fLogonFile-zfilename
```

Below is an explanation of the parameters:

Parameters	Description
<i>-fLogonFile</i>	LCU file to use to automatically connect to a Connect:Direct server.
<i>-nNodeName</i>	Node name or IP address.
<i>-uUserId</i>	User ID to use to connect to the node.
<i>-pPassword</i>	Password to use connect to the Connect:Direct node.
<i>-l</i>	By default, the CLI limits output to 128 characters per line. Include this switch to display longer lines, such as file names or sysopts strings.
<i>-mScrollLineCount</i>	The number of lines to display before pausing the statistics and Process output. Value is calculated automatically if parameter is not defined.  Do not use this option if the output is piped to a file because it will be paused with no evidence on the screen.

Parameters	Description
-x	Echoes the command input on the display. Writes the command immediately before the output.
-zfilename	Copies command output to the specified file as well as displaying the output.
-?	Provides help for the command line interface.

## Terminating the CLI

### Procedure

- To terminate the CLI, use the quit command followed by a semicolon as shown in the following example:

```
quit;
```

## CLI Commands

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 pro, proc, procs, p	Deletes the specified Process from the TCQ.
select process	sel pro, proc, procs, p	Retrieves status information about the specified Process.
select message	sel mes, msg, m	Retrieves the text explanation of any message Connect:Direct issues.
select statistics	sel sta, stat, stats, st	Retrieves statistics records for the specified Process.
traceoff	tof, troff	Disables the trace facility.
traceton	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.

## Command Syntax

The following information describes the general conventions used in the command syntax:



- All commands must be followed by a semicolon.
- 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 type of field and data.
- 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 Microsoft 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.

## Submit Process Command

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.

Parameter	Description
file=filename	The name of the Process file, up to 256 characters. If you specify the file parameter, you must specify it 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	<p>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.</p> <p>The values for the execution priority range from 1 to 15 and are mapped to Microsoft Windows Process priority classes and values. The highest priority is 15.</p> <p>Only Connect:Direct for Microsoft Windows supports the execution priority option.</p>
hold=yes   no   call	<p>The TCQ hold status of the Process.</p> <p>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.</p> <p>no—Does not place the Process in the Hold queue. The Process executes as soon as resources are available.</p> <p>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.</p>
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 to 48 alphanumeric characters. You must specify both the ID and the password.

Parameter	Description
maxdelay=unlimited   hh:mm:ss   0	<p>Causes the command processor to wait until the submitted Process completes execution or the specified time interval expires.</p> <p>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.</p> <p>unlimited—Specifies that the submit command processor is to wait until the Process completes execution.</p> <p>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).</p> <p>0—Specifies that the submitted Process must begin execution immediately. If the submitted Process cannot begin execution immediately, the submit operation fails.</p>
newname=new process name	<p>Specifies a new Process name, 1 to 8 alphanumeric characters long, to override the name within the submitted Process.</p>
notify=userid	<p>The user to receive Process completion messages.</p>
sacct="snode accounting data"	<p>Accounting data, from 1 to 256 characters, for the SNODE. Enclose the string in double quotation marks.</p>
retain=yes   no   initial	<p>Determines whether a copy of the Process is retained in the TCQ for re-execution after the Process executes.</p> <p>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.</p> <p>no—Specifies that the Process is deleted after execution.</p> <p>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.</p> <p>If startt is set, you must set retain=yes to execute the Process at regular intervals based on the value of startt.</p> <p>If retain=initial, do not use the startt parameter. This causes the submit command to fail.</p>

Parameter	Description
<p>snode=[nodename]   [hostname   IPaddress ; portnumber   servicename]</p>	<p>Identifies the SNODE. The SNODE name is a 1- to 16-character alphanumeric 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.</p> <p>nodename—Identifies the remote node object in the Connect:Direct network map.</p> <p>hostname   IPaddress ; portnumber   servicename—Specifies an IP address for the SNODE. This is used for TCP/IP connectivity only.</p> <p>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 <i>Specifying IP Addresses, Host Names, and Ports</i>.</p> <p>If you specify an IP address, you must also specify netmap.check=r or netmap.check=n in your initialization parameters.</p> <p>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.</p>
<p>snodeid=(id [,pswd [,newpswd]])</p>	<p>The security user IDs and security passwords on the SNODE. The subparameters can contain one or more alphanumeric characters.</p> <p>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.</p> <p>If you specify the password, you must also specify the ID. If you specify a new password, you must also specify the existing password.</p>
<p>prty=nn</p>	<p>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.</p> <p>Values range from 0 to 15, where 15 is the highest priority.</p>

Parameter	Description
startt=([date   day   daily] [,time])	<p>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, daily, and time are positional parameters. If you do not specify date or day, type a comma before the time.</p> <p>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.</p> <p>day—Specifies the day of the week. Values are today, tomorrow, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday.</p> <p>daily—Runs the Process daily.</p> <p>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.</p> <p>If the time 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), and the Process submits at 5:30 PM, Connect:Direct schedules the Process to run the next day.</p> <p>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.</p> <p>Do not set the startt parameter if retain=initial. This causes the submit command to fail.</p>
&symbolic name n="variable string n"	<p>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.</p>

## 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);
```

## Related concepts

[Process or Command Options](#)

## Change Process Command

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 performs the following functions:

- 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 to 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 to 999999.
snode=snode name   generic   (list)	Searches for a Process or Processes by the SNODE (partner) name. The SNODE name can contain 1 to 16 alphanumeric characters. You can use the IP address of the SNODE as your SNODE name.

Parameter	Description
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 to 15 and are mapped to Microsoft Windows Process priority classes and values. The highest priority is 15.  Only Connect:Direct for Microsoft Windows supports the execution priority option.
hold=yes   no   call	Moves the Process to the Hold or Wait queue.  yes—Places the Process in the Hold queue in HO (Held by Operator) status until another change process command explicitly releases it.  no—Places the Process in the Wait queue in WC (Waiting for Connection) status. The Process executes as soon as resources are available.  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.
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.  Values range from 0 to 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;
```

## Delete Process Command

Use the delete process command to remove a Process 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 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 to 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 deleted Processes are managed:

Parameter	Description
force=yes   no	<p>Forcibly terminate an executing Process. Use this parameter if a Process is in the executing state and is waiting for unavailable resources.</p> <p>yes—Forcibly and immediately terminates the Process or Processes.</p> <p>no—Notifies the partner node of the intent to terminate and terminates the Processes. This is the default.</p>



Parameter	Description
hold=yes   no	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;
```

## Select Process Command

Use the select process command 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   no]

### 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=all   exec   hold   wait   timer	Identifies queue names to select. all—Selects all queues. This is the default. exec—Selects Processes from the Execution queue. hold—Selects Processes from the Hold queue. timer—Selects Processes from the Timer queue. wait—Selects Processes from the Wait queue.
snode=snode name   generic   (list)	Identifies the SNODE name, from 1 to 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. ex—Selects Processes from the Execution queue. hc—Selects Processes submitted with hold=call. he—Selects Processes held due to a connection error. hi—Selects Processes submitted with hold=yes. ho—Selects Processes held by a change process command issued with hold=yes. hr—Selects Processes submitted with retain=yes. hs—Selects Processes suspended by a delete process command issued with hold=yes. 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. re—Selects Processes that are waiting for restart after session failure. wa—Selects Processes that are eligible for execution but not yet selected or running. wc—Selects Processes that are ready for execution and awaiting an available connection to the SNODE. 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   no	Specifies the type of report generated for the selected Process or Processes. The default is no.  yes—Generates a detailed report  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;
```

## Select Message Command

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 Connect:Direct message LCCC00I:

```
select message msgid=LCCC00I;
```

### Related concepts

[Filter the Event Log](#)

## Select Statistics Command

Issue the select statistics command 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 the select statistics command, 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.

### Format

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)]
	[snode=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)]
	<b>Note:</b> To use a wildcard within the submitter parameter, you must have administrator access.
/* Display option */	
[detail=yes   no]	

### Parameters

Provide one or more of the following parameters to determine what statistics are returned:

Parameter	Description
ccode=(operator, code)	<p>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.</p> <p>Following are the valid completion code operators:</p> <p>eq   =   == Equal (This is the default.)</p> <p>ge   &gt;=   =&gt; Greater than or equal</p> <p>gt   &gt; Greater than</p> <p>le   &lt;=   =&lt; Less than or equal</p> <p>lt   &lt; Less than</p> <p>ne   != Not equal</p> <p>Following are the valid completion codes:</p> <p>0—Successful execution of the Process.</p> <p>4—A warning level error was encountered. The statement probably completed normally but you should verify the execution results.</p> <p>8—An error occurred during Process execution.</p> <p>16—A Severe error occurred during Process execution.</p>
dfile=destination filename   (list)	<p>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.</p> <p>This parameter is not supported by Connect:Direct for UNIX.</p>
pname=Process name   generic   (list)	<p>Selects Process statistics by Process name, a generic name, or a list of names. The name can be 1 to 8 alphanumeric characters long.</p>
pnumber=Process number   (list)	<p>Selects statistics by Process number or a list of Process numbers. The Process number is assigned when the Process is submitted.</p>
reccat=caev   capr   (caev , capr)	<p>Selects statistics based on whether the record category is related to events or to a Process.</p> <p>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.</p> <p>caev—Specifies that the retrieved statistics file records should include those related to Connect:Direct events, such as a Connect:Direct shutdown.</p> <p>capr—Specifies that the retrieved statistics file records should include those related to one or more Connect:Direct Processes.</p>

Parameter	Description
recids=record id   (list)	<p>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:</p> <p>AUPR—Authorization file processing</p> <p>CHGP—Change Process command issued</p> <p>COAC—Communication activated</p> <p>CMLT—CMGR listen thread terminated</p> <p>CRHT—Connect:Direct copyright</p> <p>CSTP—Child Process stopped</p> <p>CTRC—Copy control record written</p> <p>CTRM—Child Process terminated</p> <p>CUKN—Child process unknown status</p> <p>CXIT—Child process exited</p> <p>DELP—Delete Process command issued</p> <p>FLSP—Flush Process command issued</p> <p>FMRV—Formatted Header (FMH) received</p> <p>FMSD—Formatted Header (FMH) sent</p> <p>GPRC—Get Process issued</p> <p>IFED—If statement ended</p> <p>IPPR—Initialization parameter processing</p> <p>LIOK—Listen okay</p> <p>LSST—The record ID of a step on the local node</p> <p>NAUH—Node Authorization check issued</p> <p>NMOP—Network map file opened</p> <p>NMPR—The network map is updated through Sterling Connect:Direct Browser User Interface, IBM Control Center, or KQV Interface.</p> <p>NUIC—Connect:Direct initialization complete</p> <p>NUIS—Connect:Direct start initialization</p> <p>NUT1—Connect:Direct phase one termination complete status</p> <p>NUT2—Connect:Direct phase two termination complete status</p> <p>NUTC—Connect:Direct termination complete</p> <p>NUTR—Connect:Direct termination requested</p> <p>PERR—Process error was detected</p> <p>PFLS—Process was flushed</p> <p>PMED—Process Manager ended</p>



Parameter	Description
recids=record id   (list) (continued)	PMIP—Process Manager Initprocs thread initialized 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

Parameter	Description
	<p>SIGC—System error</p> <p>SMED—Session Manager ended</p> <p>SMST—Session Manager started</p> <p>SNMP—SNMP</p> <p>SSTR—Session start issued</p> <p>STOP—Stop Connect:Direct command issued</p> <p>SUBP—Submit command issued</p> <p>TCPI—TCP started</p> <p>TRAC—Trace command issued</p> <p>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</p> <p>UNKN—Unknown command issued</p> <p>USEC—User Security check issued</p> <p>xxxx—Record types identified by the first four characters of the message ID</p>
snode=snode name   generic   (list)	<p>Selects statistics file records by SNODE name, a generic node name, or a list of node names. The SNODE name can be 1 to 16 alphanumeric characters long.</p>
sfile=filename  (list)	<p>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.</p> <p>This parameter is not supported by Connect:Direct for UNIX.</p>
startt=([date   day] [, time])	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

Parameter	Description
stopt=([date   day] [, time])	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
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	<p>Specifies the type of report generated for the selected Processes. The default is no.</p> <p>yes—Generates a detailed report.</p> <p>no—Generates a short report.</p>

## Example

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);
```

## Traceoff Command

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.

### Format

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)]

**Note:** The dest, pnode, pname, and pnum parameters are valid for smgr and comm traces only.

Specify 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.

Parameter	Description
type=cmgr   pmgr   smgr   comm   (list)	<p>Disables traces by type.</p> <p>cmgr—Traces events relating to the interaction of the server with connected workstations and with the server console.</p> <p>pmgr—Traces events relating to the manipulation of Connect:Direct Processes.</p> <p>smgr—Traces events relating to the execution of Connect:Direct Processes and the server's interaction with other Connect:Direct nodes.</p> <p>comm—Traces only interactions with external communications facilities invoked from Session Manager threads and used to communicate with other Connect:Direct nodes.</p>

### 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	<p>The PNODE or SNODE session managers.</p> <p>pnode—Disables the trace of all PNODE session managers.</p> <p>snode—Disables the trace of all SNODE session managers.</p>
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;
```

### Related concepts

[Diagnose a Server Problem Using Traces](#)

[Traceon Command](#)

## Traceon Command

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.

## Format

The format for the traceon command follows:

Command	Parameters
traceon	[file=filename]
	[filesize=nnK   nnM   0]
	[level=basic   intermediate   full]
	[type=cmgr   pmgr   smgr   comm   (list)]
	[wrap=yes   no]
	[dest=destination   (list)]
	[pnode   snode]
	[pname=process name   (list)]
	[pnum=process number   (list)]

**Note:** The dest, pnode, pname, and pnum parameters are valid for smgr and comm traces only.

## Parameters

Specify 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 to 256 characters. The default is CDTRACE.CDT in the Connect:Direct directory.
filesize=nnnK   nnnM   0	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   full	The level of trace detail.  basic—Produces a trace of function entry and exit.  intermediate—Produces a trace of function entry and exit, plus arguments to functions.  full—Produces a trace with function entry/exit, function argument display, plus display of selected internal control blocks.

Parameter	Description
type=cmgr   pmgr   smgr   comm   (list)	<p>The type of event or a list of events to be traced.</p> <p>cmgr—Traces events relating to the interaction of the server with connected workstations and the server console.</p> <p>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.</p> <p>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.</p> <p>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.</p>
wrap=yes   no	<p>Specifies whether you want your trace output to be a wraparound file.</p> <p>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.</p> <p>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.</p>

**Related concepts**

[Diagnose a Server Problem Using Traces](#)

[Traceoff Command](#)

**Help Command**

Use the help command to receive a list of the commands supported by the Connect:Direct CLI.

Any command with a -? or /? will display the syntax for the command's usage.



```
command /?;

or

command -?;
```

## Stop 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  quiesce]

### 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.
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;
```

### Related tasks

[Stopping Connect:Direct for Microsoft Windows Using the CLI](#)

## Modify Translation Tables

---

### 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 for use when sending or receiving data to or from a remote Connect:Direct node:

- 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.

### Modify a Translation Table Using Connect:Direct Requester

#### About this task

Translation is performed if a data type of text is specified in the sysopts parameter of a copy statement.

#### Procedure

1. Select Admin > Translation Table.
2. Type the name of the translation table file, or select one of the following default translation tables from the drop-down list:
  - XLATERCR.CDX
  - XLATESND.CDX
3. Click OK.

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. To view the shortcut menu, right-click in the Translation Table dialog box.
5. Select one of the following representations for the table element:
  - Hex
  - Dec
  - Oct
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 Apply to save your changes.
8. Click OK to save your changes to the selected Connect:Direct node.

## Edit Connection Settings

---

### About the Client Connection Utility

Before you begin using the SDK to create your own programs or use Connect:Direct for Microsoft Windows to transfer files, you can use the Client Connection Utility to create connection settings for each user.

To use Connect:Direct Requester, refer to [“Define and Manage the Connect:Direct Network”](#) on page 44 for instructions.

The Connect:Direct for Microsoft Windows client software uses the Microsoft Windows Registry to store its configuration information. The Client Connection Utility allows you to update the connection settings within the Registry.



**CAUTION:** Use the Client Connection Utility to update any Registry settings rather than editing them directly.

You can view, edit, and update Connect:Direct for Microsoft Windows connection settings in the Microsoft 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.

#### Related concepts

[Define and Manage the Connect:Direct Network](#)

### Start the Client Connection Utility

#### About this task

To start the Client Connection Utility:

#### Procedure

1. Click **Start > All Programs**.
2. Click **IBM Connect Direct > v6.1 > CD Client Connection Utility**. The **Client Connection Utility** main window is displayed.

### Add a Node

#### About this task

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).

## Procedure

1. Select File > New Node.
2. To add a node registered in the Active Directory, follow these steps:
  - a) Select Windows in the **Operating System** field.
  - b) Select the node to add from Active Directory Nodes.

**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, follow these steps:
  - 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 Set as the default node.
5. Click **OK** to save your settings and close Node Properties.
6. Select File > Save to save the new settings.



**Attention:** Changes made to the node settings are not written to the Registry until you select Save.

## Deleting a Node

### About this task

To delete a node:

### Procedure

1. In the Client Connection Utility main window, select the node you want to delete.
2. Select Edit > Delete.
3. Select File > Save to delete the node.



**Attention:** Changes made to the node settings are not written to the Registry until you select Save.

## Adding a User with Client Connection Utility

### About this task

To add a new Connect:Direct user from the Client Connection Utility:

### Procedure

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 in 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.domain.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 the node settings are not written to the Registry until you select Save.

## Deleting a User with the Client Connection Utility

### About this task

To delete a user from the node using the Client Connection Utility:

### Procedure

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 the node settings are not written to the Registry until you select Save.

## Updating a Node or User

### About this task

To update node or user properties:

### Procedure

1. Do one of the following:
  - To update a node, highlight the node you want to configure.
  - To update a user, highlight the user you want to configure.
2. Select **File > Properties**.
3. Select the fields you want to edit and make the appropriate changes.
4. Click **OK** to save your settings and return to Node Properties.
5. Select **File > Save** to save the settings.



**Attention:** Changes made to the node and user settings are not written to the Registry until you select **Save**.

## Defining a Default Node or User

### About this task

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.

## Procedure

1. Do one of the following:
  - To define a default node, highlight the node you want to designate as the default.
  - To define a default user, highlight the user you want to designate as the default.
2. Select File > Save to save the settings.



**Attention:** Changes made to the node and user settings are not written to the Registry until you select Save.

## Importing Registry Settings

### About this task

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.

### Procedure

1. Select the node to which to import the Registry settings.
2. Select File > Import.



**CAUTION:** 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**.
4. Select File > Save to save the settings.



**Attention:** Changes made to the node settings are not written to the Registry until you select Save.

## Exporting Registry Settings

### About this task

To export Registry settings:

### Procedure

1. From the Client Connection Utility main window, select the node containing the Registry settings you want to export.
2. Select File > Export.
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

### About this task

To print a report of Registry settings:

### Procedure

1. Select File > Print.

2. Configure the print settings and click **OK**. A report of all Registry settings is generated.
3. Select File > 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 without printing the report.

**Note:** Additional node detail is provided if the node has been used at least once by the client software.

## Use IBM Connect:Direct for Microsoft Windows in a Test Mode

---

### Use Connect:Direct in Test Mode

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.

You can enable test mode for production instances of Connect:Direct for Microsoft 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

### 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.

**Note:** You can modify the quiesce.resume initialization parameter while the server is active.

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 168](#) in this section.

When the testing mode is enabled, Connect:Direct for Microsoft 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 Microsoft 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

### About this task

You can create or update the parameter table file while the server is active. Any changes made to the file take effect for sessions that begin after the changes are made.

### Procedure

1. To modify the sample NDMPXTBL parameter table supplied with Connect:Direct for Microsoft Windows, open any text editor.
2. Change the table using the following commands.

**Note:** 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.  H—Places the Process in the Hold queue  R—Places the Process in the Timer queue in session retry  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.”



Command Code	Description	Subparameters/Examples
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 PACH*—Matches all Processes beginning with “ACH” 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 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. S*@TPM002—Matches all IDs from node TPM002 SACTQ0ACD@*—Matches ID ACTQ0ACD from all nodes S*@*—Matches all IDs from any node. This is another way to match all Processes.
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.  <b>Note:</b> 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 I NCD.BOSTON  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 X DALLASOPS@*  Excludes Processes for execution submitted by the ID SDALLASOPS from any node.
L	Last entry in table.	

3. When you update the parameter table, name it NDMPXTBL and save it to the Server directory of the installation.

## 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.

### Specify 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
```

### Specify 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
```

### Permit 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
```

### Stop 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
```

## Client API connections

---

### Authenticating client connection

Implementing strong security programs provides Connect:Direct for Microsoft Windows users an assurance that file transfer is closely guarded. Connect:Direct for Microsoft Windows uses several approaches to manage client API connections.

IBM Connect:Direct server uses the following client authentication methods to establish the identity of the requesting client and determines whether that client is authorized to connect to the Connect:Direct server using the supplied credentials:

- Username/password-based authentication
- Digital security certificate-based authentication
- Trusted Local Host Authentication (user name only)

However, passwords configured for the Windows System and digital certificates are set to expire at some interval and must be changed. Any time the password is changed on the Windows server, it must also be changed in the client configuration resulting in tedious password management routine in a large deployment.

To ease password management routines for local-host client connections, Connect:Direct for Microsoft Windows extends the client API authentication process to allow no-password access for local connections.

IBM recommends using password-based authentication or certificate-based authentication method to authenticate client connections. Connect:Direct for Microsoft Windows users should be aware of the following implication of using Trusted Local Host Authentication.



**Attention:**

- Trusted Local Host Authentication allows any program running on the same host as Connect:Direct to submit API commands to Connect:Direct without specifying a password.
- In some cases, for example when Connect:Direct is running on a multi-user system, Trusted Local Host Authentication may not provide adequate security. Enable Trusted Local Host Authentication when the Admin is certain that doing so does not create a security risk.

### Implementing Client Authentication

When connecting to a Connect:Direct server, a user or client application must supply the user ID of a valid user account that is configured in the functional authorities of a Connect:Direct node.

In addition, the application usually includes a means of authentication, such as a password or a security certificate.

There are two types of client authentication:

- LOCAL—Authenticating users or applications that are trying to connect from the same node that the connect:direct server is running on.
- HOST—Authenticating users or applications that are trying to connect from a node that has a different IPv4 or IPv6 address than the connect:direct server.

Connect:Direct for Windows takes the following steps to authenticate users:

1. When a user or application attempts to connect to a CDW server, the system checks to see if the user is configured in local functional authority.
2. If so, authentication occurs using any one of these assigned authentication method:

- a. **Password-based authentication** if the user has provided the username/password.

For example, File Agent is configured with a userid and password that allows it to connect to Connect:Direct for Windows via. an API connection. This user id and password must be a valid Windows credentials for the Connect:Direct for Windows server that File Agent connects to.

- b. **Certificate-based authentication**

Authenticates a client such as, IBM Control Center (ICC) using digital certificates (SSL). For more information on Client Authentication see, [“Certificate Authentication for Client API Connections” on page 170.](#)

To enable Certificate-based authentication, go to **IBM Connect:Direct Requester>Functional Authorities Users Main panel** > select **Allow client certificate authentication** check box. For more information see, [“Defining User Authority” on page 37.](#)

- c. **Trusted Local Host Authentication** (no password)

Authenticates with a user name configured in the functional authorities but no password.

To enable local authentication without a password, go to **IBM Connect:Direct Requester>Functional Authorities Users Main panel** > select **Allow no-password local connection** check box. For more information see, [“Defining User Authority” on page 37.](#)

## Configuring Connect:Direct Windows for Authentication Management

Connect:Direct for Windows users can use local functional authority templates to assign user authorities and restrict privileges. For more information on how to enable or disable an authentication method see, [“Defining User Authority” on page 37.](#)

## Certificate Authentication for Client API Connections

The API connection certificate authentication feature allows clients to connect to a Connect:Direct server by using only an SSL Certificate and an unreal user ID. You can configure this feature in the functional authorities of a Connect:Direct node. The API certificate authentication requires no user password since the user ID is unreal.

This feature improves password management in large deployments of Connect:Direct, as it removes the extra administrative steps that result from password usage.

### Note:

This feature is specific only to API connections. These connections must also be AIJ-based. When you use the authentication feature, ensure that the version of the AIJ is at least 1.1.00 Fix 000025. This version of the AIJ contains updates that allow blank passwords to be used. These AIJ version requirements also apply if you use the authentication feature in IBM Control Center. API connection certificate authentication is not supported for the Direct.exe CLI, IBM Connect:Direct Requester, or the Connect:Direct native C/C++/C# non Java APIs.

## Configuring API certificate authentication

Client Authentication must be enabled on the Connect:Direct Secure Plus .Client record. Client authentication is not enabled by default in Connect:Direct Secure Plus. During an API connection, a peer certificate is required from IBM Control Center or the AIJ client. That certificate must contain a common name field of an SSL certificate whose contents match a Connect:Direct functional authorities user record in the Connect:Direct node. You also must use a blank password in order for IBM Connect:Direct to trigger the API certificate authentication process.

A new functional authorities configuration parameter is added to Connect:Direct for Microsoft Windows. The parameter specifies whether a specific user can log in as a client via API certificate authentication, and it must be set to Yes when you configure API certificate authentication.



---

## Chapter 4. Using FASP with IBM Aspera High-Speed Add-on for Connect:Direct for Microsoft Windows (V4.8.0 or later)

IBM Aspera High-Speed Add-on for Connect:Direct for Microsoft Windows uses FASP (Fast and Secure Protocol) network transport to transfer files over high bandwidth and high latency network connections.

At low latency it performs similarly to TCP/IP. However, as latency and packet loss increase, unlike TCP/IP, its performance does not degrade, and FASP continues to take advantage of all the available bandwidth.

IBM Aspera High-Speed Add-on for Connect:Direct for Microsoft Windows supports interoperability with Connect:Direct for UNIX (V4.3.0 or later) and Secure Proxy (V3.4.3.0 or later).

**Note:** Secure+ is used to secure FASP transfers exactly the same way it is used for TCP/IP transfers.

### Related concepts

[“Using Connect:Direct for Microsoft Windows with IBM Aspera High-Speed Add-on and Secure Proxy” on page 174](#)

You can send files using IBM Aspera High-Speed Add-on through Secure Proxy using Connect:Direct for Microsoft Windows.

---

## Activating FASP

By default, IBM Aspera High-Speed Add-on for Connect:Direct is not enabled. To enable it, you must download a license key and install Connect:Direct for Microsoft Windows V4.8.0 or later.

### Before you begin

You must have Connect:Direct for Microsoft Windows V4.8.0 or later installed.

### Procedure

1. Download and install Connect:Direct for Microsoft Windows V4.8.0, or later from IBM Fix Central.
2. Download the IBM Aspera High-Speed Add-on for Connect:Direct license key for your Connect:Direct node from Passport Advantage.
3. Rename the file *aspera-license*.
4. Save the renamed file to the `<install_dir>\ConnectDirect\vx.x.x\Server` directory.

### What to do next

**Important:** The Connect:Direct install package includes the IBM Aspera High-Speed Add-on for Connect:Direct configuration file (*aspera.conf*). It contains the minimum necessary basic configuration statements to use FASP on Connect:Direct. It is always installed even if you do not purchase IBM Aspera High-Speed Add-on for Connect:Direct. Do NOT make any changes to this file.

---

## Licensed bandwidth for FASP transactions

The bandwidth available to a file transfer is limited by, among other things, the bandwidths specified in the sender's and receiver's Aspera license keys.

There are two types of available license keys:

- Datacenter licenses (available in 10gbps, 1gbps, 300mbps and 100mbps) - can send and receive files using FASP when connected to a node that has an Endpoint or DataCenter license.

- Endpoint license - can send and receive files using FASP when connected to a node that has a DataCenter license.

When both sender and receiver only have Endpoint licenses, file transfer over FASP is not supported. When either the sender or receiver has an Endpoint license and the other has a Datacenter license, the available bandwidth is limited to the value in the Datacenter license. When both sender and receiver have Datacenter licenses, the bandwidth is limited to the smaller of the two values in the Datacenter licenses.

## Using Connect:Direct for Microsoft Windows with IBM Aspera High-Speed Add-on and Secure Proxy

You can send files using IBM Aspera High-Speed Add-on through Secure Proxy using Connect:Direct for Microsoft Windows.

FASP is supported in Secure Proxy V3.4.3 or later. If you send a file from your local Connect:Direct for Microsoft Windows node configured for FASP, it passes through your Secure Proxy instance using FASP, and is sent to the remote node.

In addition to the FASP parameter values outlined in [Configuring FASP](#), the following parameter should be used when using Secure Proxy between Connect:Direct nodes:

```
fasp=(yes|no|ssp,yes|no|ssp)
```

The first parameter is the default for Connect:Direct as the PNODE. The second parameter is the default for Connect:Direct as the SNODE.

This parameter can now be used in the netmap local node record and remote node trading partner record in Connect:Direct for Windows.

The following table shows results when Connect:Direct FASP protocol is used between two Connect:Direct nodes with no Sterling Secure Proxy involved.

<b>PNODE fasp=</b>	<b>Protocol</b>	<b>SNODE fasp=</b>
N	TCP	N
N	TCP	Y
N	TCP	SSP
Y	TCP	N
Y	C:D FASP	Y
Y	TCP	SSP
SSP	TCP	N
SSP	TCP	Y
SSP	TCP	SSP

The following table shows results when Connect:Direct FASP protocol is used with two Connect:Direct nodes going through a single instance of Sterling Secure Proxy.

<b>PNODE fasp=</b>	<b>Protocol</b>	<b>SSP</b>	<b>Protocol</b>	<b>SNODE fasp=</b>
N	TCP	SSP	TCP	N
N	TCP	SSP	TCP	Y
N	TCP	SSP	TCP	SSP
Y	TCP	SSP	TCP	N
Y	C:D FASP	SSP	C:D FASP	Y



Y	C:D FASP	SSP	TCP	SSP
SSP	TCP	SSP	TCP	N
SSP	TCP	SSP	C:D FASP	Y
SSP	TCP	SSP	TCP	SSP

The following table shows results when Connect:Direct FASP protocol is used with two Connect:Direct nodes going through two instances of Sterling Secure Proxy.

<b>PNode fasp=</b>	<b>Protocol</b>	<b>SSP</b>	<b>Protocol</b>	<b>SSP</b>	<b>Protocol</b>	<b>SNode fasp=</b>
N	TCP	SSP	TCP	SSP	TCP	N
N	TCP	SSP	TCP	SSP	TCP	Y
N	TCP	SSP	TCP	SSP	TCP	SSP
Y	TCP	TCP	TCP	SSP	TCP	N
Y	C:D FASP	SSP	C:D FASP	SSP	C:D FASP	Y
Y	C:D FASP	SSP	C:D FASP	SSP	TCP	SSP
SSP	TCP	SSP	TCP	SSP	TCP	N
SSP	TCP	SSP	C:D FASP	SSP	C:D FASP	Y
SSP	TCP	SSP	C:D FASP	SSP	TCP	SSP

For more information on using Secure Proxy with FASP, see *Using FASP with Sterling Secure Proxy (V3.4.3 or later)*.

## Configuring FASP

### About this task

To enable IBM Aspera High-Speed Add-on for Connect:Direct for Microsoft Windows, you must update the local node (initialization parameters) with FASP parameters.

### Procedure

1. From the Admin Tool initialization parameters, click the **TCP/IP** tab.
2. In the Configure FASP section, click **Properties**.
3. Configure the values for the Local Node by completing the following fields:

<b>Field</b>	<b>Value</b>
FASP Listen Ports	Type the port numbers you want to use for FASP. Only valid for SNode configuration.
PNode FASP Flag	Use when this is PNode. Valid values are: <ul style="list-style-type: none"> <li>• If set to Yes, use FASP on PNode</li> <li>• If set to No, don't use FASP</li> <li>• If set to SSP , use FASP with SSP bridging</li> <li>• If set to Blank, use the default value of No.</li> </ul>
SNode FASP Flag	Use when this is SNode. Valid values are:

Field	Value
	<ul style="list-style-type: none"> <li>• If set to Yes, use FASP on SNode</li> <li>• If set to No, don't use FASP</li> <li>• If set to SSP , use FASP with SSP bridging</li> <li>• If set to Blank, use the default value of No.</li> </ul>
File Size Threshold	<p>Optional. Used to restrict small files from being sent using FASP.</p> <ul style="list-style-type: none"> <li>• If the file is greater than or equal to the stated value, the Connect:Direct server sends the file using FASP. Otherwise, it is sent using TCP/IP.</li> <li>• Default is 1GB.</li> <li>• You can use KB, MB, or GB designators. If no designator is included, the system uses bits.</li> <li>• This setting can be overridden by the remote node record or process parameters.</li> </ul>
Target Bandwidth	<p>Optional. Default is as stipulated in the FASP license key. Specifies how much bandwidth each transfer can use.</p> <ul style="list-style-type: none"> <li>• Default value can be changed, but cannot exceed the bandwidth specified in the license key.</li> <li>• You can use KB, MB, or GB designators. If no designator is included, the system uses bits per second.</li> <li>• This setting can be overridden by the remote node record or process parameters, but cannot exceed the bandwidth specified in the license key.</li> </ul>
Policy	<p>Optional. Specifies the fairness of each transfer. Default is <i>fair</i>.</p> <ul style="list-style-type: none"> <li>• This setting can be overridden by the remote node record or process parameters.</li> <li>• Valid values are: <ul style="list-style-type: none"> <li>– Fixed - FASP attempts to transfer at the specified target rate, regardless of the actual network capacity. This policy transfers at a constant rate and finishes in a guaranteed amount of time. This policy typically occupies a majority of the network's bandwidth, and is not recommended in most file transfer scenarios.</li> <li>– Fair - FASP monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When other types of traffic build up and congestion occurs, FASP shares bandwidth with other traffic fairly by transferring at an even rate.</li> </ul> </li> </ul>

Field	Value
	<p>This is the best option for most file transfer scenarios.</p> <ul style="list-style-type: none"> <li>– High - FASP monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, a FASP session with high policy transfers at a rate twice of a session with fair policy.</li> <li>– Low - Similar to Fair mode, the Low (or Trickle) policy uses the available bandwidth up to the maximum rate as set in the Aspera license file. When congestion occurs, the transfer rate is decreased all the way down to the minimum rate as set in the Aspera license file.</li> </ul>

4. (Optional) Using Connect:Direct Requester, select **Netmap** and specify the values for the remote node using the following chart. Configure the remote node if you need to override your local node settings. For example, if you want to exclude a trading partner from using FASP. You can also configure the remote node record later.

Field	Value
Pnode FASP flag	<p>Use when this is PNode. Valid values are:</p> <ul style="list-style-type: none"> <li>• If set to Yes, use FASP on PNode</li> <li>• If set to No, don't use FASP</li> <li>• If set to SSP , use FASP with SSP bridging</li> <li>• If set to Blank, use the default value of No.</li> </ul>
SNode FASP Flag	<p>Use when this is SNode. Valid values are:</p> <ul style="list-style-type: none"> <li>• If set to Yes, use FASP on SNode</li> <li>• If set to No, don't use FASP</li> <li>• If set to SSP , use FASP with SSP bridging</li> <li>• If set to Blank, use the default value of No.</li> </ul>
File Size Threshold	<p>Optional. Used to restrict small files from being sent using FASP.</p> <ul style="list-style-type: none"> <li>• If the file is greater than or equal to the stated value, the Connect:Direct server sends the file using FASP. Otherwise, it is sent using TCP/IP.</li> <li>• Default is 1GB.</li> <li>• You can use KB, MB, or GB designators. If no designator is included, the system uses bits.</li> <li>• This setting can be overridden by the process parameters.</li> </ul>
Target Bandwidth	<p>Optional. Default is as stipulated in the FASP license key. Specifies how much bandwidth each transfer can use.</p>

Field	Value
	<ul style="list-style-type: none"> <li>• Default value can be changed, but cannot exceed the bandwidth specified in the license key.</li> <li>• You can use KB, MB, or GB designators. If no designator is included, the system uses bits per second.</li> <li>• This setting can be overridden by the process parameters, but cannot exceed the bandwidth specified in the license key.</li> </ul>
Policy	<p>Optional. Specifies the fairness of each transfer. Default is <i>fair</i>.</p> <ul style="list-style-type: none"> <li>• This setting can be overridden by the process parameters.</li> <li>• Valid values are: <ul style="list-style-type: none"> <li>– Fixed - FASP attempts to transfer at the specified target rate, regardless of the actual network capacity. This policy transfers at a constant rate and finishes in a guaranteed amount of time. This policy typically occupies a majority of the network's bandwidth, and is not recommended in most file transfer scenarios.</li> <li>– Fair - FASP monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When other types of traffic build up and congestion occurs, FASP shares bandwidth with other traffic fairly by transferring at an even rate. This is the best option for most file transfer scenarios.</li> <li>– High - FASP monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, a FASP session with high policy transfers at a rate twice of a session with fair policy.</li> <li>– Low - Similar to Fair mode, the Low (or Trickle) policy uses the available bandwidth up to the maximum rate as set in the Aspera license file. When congestion occurs, the transfer rate is decreased all the way down to the minimum rate as set in the Aspera license file.</li> </ul> </li> </ul>

## FASP Process Language

---

Once the FASP parameters for both trading partners have been configured, you can override the default settings on a process by process basis to perform exception processing.

### Optional Parameters

FASP Parameters:

- FASP (Yes | No)
- FASP POLICY (Values are the same as the FASP Local and Remote node record parameters)
- FASP.FILESIZE.THRESHOLD (Values are the same as the FASP Local and Remote node record parameters)
- FASP.BANDWIDTH (Values are the same as the FASP Local and Remote node record parameters)

FASP Parameters are applicable in three different contexts:

- COPY statement - The four FASP parameters may be used individually or as a group within a COPY statement. This will set FASP values for the duration of the COPY statement and will not have any effect on statements within the submitted Process.
- PROCESS statement - The four FASP parameters may be used individually or as a group at the end of a PROCESS statement. This will set the FASP parameters for all of the COPY statements in the process
- SUBMIT command - The four FASP parameters may be set individually or as a group at the end of a SUBMIT command. This will set the FASP parameters for all COPY statements in the process being submitted. These settings will set FASP information for their relevant part of the scope, potentially overriding the Local Node settings, Remote Node settings and each other.

### Examples

Copy statement example:

```
step01 copy
from
(
file = \tmp\exampleout
pnode
)
ckpt = 2M
compress extended
to
(
file = \tmp\examplein
snode
disp = rpl
)
fasp=yes
fasp.policy=fair
fasp.bandwidth=500M
fasp.filesize.threshold=10G
```

Process statement example:

```
SAMPLE PROCESS    SNODE=WINVM-470
fasp=yes
fasp.policy=fair
fasp.bandwidth=500M
fasp.filesize.threshold=10G
step01 copy
from
(
file = \tmp\exampleout
pnode
)
ckpt = 2M
compress extended
to
(
file = \tmp\examplein
```

```
snode
disp = rpl
)
PEND
```

## Hierarchy Settings

The system uses the following hierarchy to process overrides:

1. Remote node record overrides Local node (initialization parameters) values.
2. Process parameters override remote node record.
3. Submit statement overrides the process parameters.
4. Each Copy statement overrides the effective settings of the session established by the node settings, Process, or Submit statements. The Copy statement override is effective only for the duration of the Copy step.

## FASP Messages

Use the following table to obtain FASP error message information.

**Note:** Long text message files for these message IDs can be viewed using the Connect:Direct Requester Message Lookup utility.

Non-Detailed Statistics Mode (Message ID only)	Detailed Statistics Mode
FASP001E	FASP001E: FASP server session creation failed.
FASP002E	FASP002E: FASP client session creation failed.
FASP003E	FASP003E: FASP could not be initialized.
FASP004E	FASP004E: Lock timeout.
FASP005E	FASP005E: Memory allocation failure.
FASP006E	FASP006E: Condition wait timed out.
FASP007E	FASP007E: No FASP listen ports available.
FASP008E	FASP008E: FASP disabled due to file size &FILESIZE < threshold &THRESHOLD
FASP009E	FASP009E: FASP session terminated unexpectedly.
FASP010E	FASP010E: SNODE refused FASP, FASP disabled.
FASP011E	FASP011E: FASP CRC verification failed.
FASP012E	FASP012E: FASP disabled due to conflict with UDT33.
FASP020E	FASP020E: Session Manager received invalid FASP control message.
FASP021E	FASP021E: FASP control message fragmented or invalid.
FASP022E	FASP022E: Session Manager failed to receive FASP control message.
FASP023E	FASP023E: The FASP control message to send exceeds the buffer size.
FASP024E	FASP024E: Session Manager failed to send FASP control message.

Non-Detailed Statistics Mode (Message ID only)	Detailed Statistics Mode
FASP030E	FASP030E: FASP license file not found.
FASP031E	FASP031E: FASP license file expired.
FASP032E	FASP032E: FASP license in error.
FASP033E	FASP033E: FASP license is malformed.
FASP034E	FASP034E: FASP license is malformed.
FASP035E	FASP035E: FASP License file at &LOCATION will expire in &VALUE day(s).
FASP040E	FASP040E: FASP initialization failed - remote &TYPE &NODE. Error=&ERROR.
FASP041E	FASP041E: FASP initialization failed - local &TYPE &NODE. Error=&ERROR.
FASP042E	FASP042E: FASP initialization failed.

## Monitoring FASP transactions

You can view the FASP parameters for a particular message in the Copy Termination Record (CTRC) using the Connect:Direct Requestor Statistics Details grid. For example, you can verify that FASP was used and which port number was used for the FASP transfer.

In the example below, note the following explanations:

- FASP - Y indicates that FASP was used for the copy step. N indicates FASP was not used. TCP/IP was used.
- FASP Listen Port - indicates which port number was used for FASP transfer. It is taken from Remote node FASP listen port settings.
- FASP Filesize Threshold - indicates the filesize threshold setting
- FASP Chunk Buffer size - size of the FASP buffers

**Note:** Some values might not be available. See “Known Limitations” on page 181 for more information.

FASP	Y
FASP Listen Port	20014
FASP Filesize Threshold	1073741824
FASP Chunk Buffer size	16777216

## Known Limitations

The following features cannot be used with FASP and Connect:Direct for Microsoft Windows:

- Firewall navigation source ports should not be used with FASP
- Fasp bandwidth and policy negotiated values can be found in the 'aspera-stream-transfer.log' file. Submit a process and let the session complete. In the aspera-stream-transfer.log file located in the <d\_dir>\Server folder, search for 'LOG FASP Session Params':

```
2016-07-01 07:46:44 [2470-00001a70] LOG LOG FASP Session Params
uuid=1ea5dc66-4cca-4b3c-bf27-cd82eba733a3 userid=0 user="-"
targetrate=1000000000 minrate=0 rate_policy=fair
cipher=none resume=0 create=0 ovr=0 times=0 precalc=no mf=0 mf_path=- mf_suffix=-
partial_file_suffix= files_encrypt=no files_decrypt=no file_csum=none dgram_sz=0 prepostcmd=-
tcp_mode=no rtt_auto=yes cookie="-" vl_proto_ver=1 peer_vl_proto_ver=1 vl_local=0
vlink_remote=0 vl_sess_id=3924 srcbase=- rd_sz=0 wr_sz=0 cluster_num_nodes=1 cluster_node_id=0
cluster_multi_session_threshold=-1 range=0-0 keepalive=no test_login=no proxy_ip=-
net_rc_alg=alg_queue exclude_older/newer_than=0/0
```



---

# Chapter 5. Secure Plus Option Implementation Guide

## Overview

---

### About Connect:Direct Secure Plus

IBM Connect:Direct Secure Plus for Microsoft Windows provides enhanced security for Connect:Direct. It is available as a separate component. Connect:Direct Secure Plus uses cryptography to secure data during transmission. You select the security protocol to use.

#### Secure Plus Microsoft Windows Video Tutorials

You can view video tutorials about the installation, configuration, troubleshooting, and other technical features of Connect:Direct Secure Plus for Microsoft Windows.

The Connect:Direct Secure Plus videos are useful for Connect:Direct administrators. These tutorials provide a quicker way to access information and remove the need to reference the IBM Connect:Direct Secure Plus documentation library.

Click the link below to access the Connect:Direct Secure Plus for Microsoft Windows video channel to view tutorials about the following topics:

- Installation
- Configuration
- Troubleshooting

The Connect:Direct Secure Plus Microsoft Windows video channel can be found at this link: [Connect:Direct Secure Plus for Microsoft Windows Video Channel](#).

### Security Concepts

Cryptography is the science of keeping messages private. A cryptographic system uses encryption keys between two trusted communication partners. These keys encrypt and decrypt information so that the information is known only to those who have the keys.

There are two kinds of cryptographic systems: symmetric-key and asymmetric-key. Symmetric-key (or secret-key) systems use the same secret key to encrypt and decrypt a message. Asymmetric-key (or public-key) systems use one key (public) to encrypt a message and a different key (private) to decrypt it. Symmetric-key systems are simpler and faster, but two parties must somehow exchange the key in a secure way because if the secret key is discovered by outside parties, security is compromised. Asymmetric-key systems, commonly known as public-key systems, avoid this problem because the public key may be freely exchanged, but the private key is never transmitted.

Cryptography provides information security as follows:

- Authentication verifies that the entity on the other end of a communications link is the intended recipient of a transmission.
- Non-repudiation provides undeniable proof of origin of transmitted data.
- Data integrity ensures that information is not altered during transmission.
- Data confidentiality ensures that data remains private during transmission.

Connect:Direct Secure Plus enables you to select the security protocol to use to secure data during electronic transmission: Transport Layer Security (TLS). Depending on the security needs of your environment, you can also validate certificates using the Sterling External Authentication Server application.

Connect:Direct Secure Plus provides alternative cryptographic solutions depending upon the protocol enabled. The following table identifies the protocols available in Connect:Direct Secure Plus and the encryption algorithms available for each protocol:

	Protocol	Encryption Algorithms			
		RC4	DES	Triple DES	AES
Connect:Direct Secure Plus V4.7 or later	SSL TLS The SSL3.0, TLS 1.0 and TLS 1.1 protocols are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.	x	x	x	x

## Transport Layer Security Protocol (TLS)

The TLS protocol provides three types of authentication:

- During the first type of authentication, called server authentication, the site initiating the session (PNODE) requests a certificate from its trading partner (SNODE) during the initial handshake. The SNODE returns its ID certificate (read from its KeyStore) and the PNODE authenticates it using one or more trusted root certificates stored in its KeyStore. Root certificates are signed by a trusted source—either a public certificate authority, such as Thawte, or by the trading partner acting as its own CA. If the ID certificate from the SNODE cannot be validated using any root certificate found in the KeyStore, or if the root certificate has expired, the PNODE terminates the session. IBM Connect:Direct writes entries to the statistics logs of both nodes and the session is aborted.
- The second type of authentication, called client authentication, is optional. If this option is enabled in the SNODE's IBM Connect:Direct parameters file definition for the PNODE, the SNODE will request a certificate from the PNODE and authenticate it using the information in its KeyStore. If this authentication fails, the SNODE terminates the session and IBM Connect:Direct writes information about the failure to the statistics log of both nodes.
- The third type of authentication is also optional and consists of validating the certificate common name. This authentication is enabled when the security administrator specifies the common name (CN) expected to be contained in the ID certificate to be validated in its IBM Connect:Direct Parameters file.
  - During the first type of authentication, the PNODE compares the common name it has specified for the SNODE in its IBM Connect:Direct Parameters file with the common name contained in the certificate sent by the SNODE. If the compare fails, that is, the information is not identical, the PNODE terminates the session, and IBM Connect:Direct writes information about the failure to the statistics logs of both nodes.
  - During the second type of authentication, the SNODE compares the common name it has specified for the PNODE in its IBM Connect:Direct Parameters file with the common name contained in the certificate sent by the PNODE. If the compare fails, that is, the information is not identical, the SNODE terminates the session, and IBM Connect:Direct writes information about the failure to the statistics logs of both nodes.

## Related tasks

[Enable or Disable External Authentication for a Remote Node](#)

[Configure External Authentication in the .SEAServer Record](#)

## NIST SP800-131a and Suite B support

Connect:Direct supports a new standard from The National Institute of Standards and Technology (NIST), SP800-131a to extend the current FIPS standards, as well as Suite B cryptographic algorithms as specified by the National Institute of Standards and Technology (NIST).

The government of the United States of America produces technical advice on IT systems and security, including data encryption and has issued Special Publication SP800-131a that requires agencies from the United States of America to transition the currently-in-use cryptographic algorithms and key lengths to new, higher levels to strengthen security.

Applications must use strengthened security by defining specific algorithms that can be used and what their minimum strengths are. These standards specifies the cryptographic algorithms and key lengths that are required in order to remain compliant with NIST security standards.

To comply with the new requirements, IBM products with cryptographic functionality must:

- Enable TLS 1.2 and be prepared to disable protocols less than TLS 1.2
- Cryptographic keys adhere to a minimum key strength of 112 bits
- Digital signatures are a minimum of SHA-2

The following is included in Secure Plus for NIST SP800-131a and Suite B support:

- Support TLS 1.1 and 1.2 with SHA-2 cipher suites
- Support for SP800-131a transition and strict modes
- Support for NSA Suite B 128 and 192 bit cipher suites and modes
- Support for IBM CMS Keystore
- Support migrating existing Secure+ certificates to the IBM CMS Keystore
- Support for JRE 1.7 SR1 iKeyman/iKeycmd utilities for certificate management.

For more information on NIST security standards, see <http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf>.

For more information on Suite B security standards, see [http://www.nsa.gov/ia/programs/suiteb\\_cryptography/index.shtml](http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml)

## Connect:Direct Secure Plus Tools

Connect:Direct Secure Plus consists of five components: the Administration Tool (Secure+ Admin Tool), the parameters file, the access file, the Strong Password Encryption parameters file, and the Command Line Interface (Secure+ CLI). The following sections describe these components and their function.

**Note:** Only one instance of the Secure+ Admin Tool or the Secure+ CLI may be used at a time because they access the same configuration file. Do not open these tools at the same time or multiple copies of the same tool at the same time (two instances of Secure+ Admin or two instances of Secure+ CLI). Only the user who accessed the configuration file first will be able to save updates.

### Administration Tool

The Secure+ Admin Tool enables you to configure and maintain the Connect:Direct Secure Plus environment. The Admin Tool is the only interface for creating and maintaining the Connect:Direct Secure Plus parameters file; operating system utilities and editing tools do not work.

## Parameters File

The Connect:Direct Secure Plus parameters file contains information that determines the protocol and encryption method used during security-enabled Connect:Direct operations. To configure Connect:Direct Secure Plus, each site must have a parameters file that contains one local node record and at least one remote node record. The local node record defines the most commonly used security and protocol settings for the node at the site. Each remote node record defines the specific security and protocol used by a trading partner. You create a remote node record in the Connect:Direct Secure Plus parameters file for each Connect:Direct node that you communicate with.

At installation, a record named .SEAServer is created in the parameters file, which enables Connect:Direct to interface with External Authentication Server during TLS sessions. External authentication is configured in this record and enabled/disabled in the local and remote node records.

With v6.1, Connect:Direct Secure Plus support to cache certificate validation responses from External Authentication Server when it interfaces External Authentication Server during a TLS session. This minimizes the overhead associated with requesting certificate validation from External Authentication Server, thus eliminating the need for Connect:Direct Secure Plus to query External Authentication Server each time. External Authentication Server response caching feature is disabled by default. To enable it see, [“Configure External Authentication in the .SEAServer Record” on page 199](#) and [“Manage the External Authentication ServerRecord” on page 211](#).

For additional security, the parameters file is stored in an encrypted format. The information used for encrypting and decrypting the parameters file (and private keys) is stored in the Connect:Direct Secure Plus access file.

## Access File

The Connect:Direct Secure Plus access file is generated automatically when you create the Connect:Direct parameters file for the first time. You type a passphrase when you first initialize Connect:Direct. This passphrase encrypts and decrypts the private keys in the parameters file. Your administrator must secure the access file. This file can be secured with any available file access restriction tools. Availability of the access file to unauthorized personnel can compromise the security of data exchange.

## Strong Password Encryption Parameters File

Strong password encryption protects Connect:Direct passwords at rest on the Connect:Direct server using strong encryption. Strong password encryption parameters are stored in the parameters file (`<CD installation directory>\Server\Secure+\Nodes\.Password`). This feature is enabled by default.

## Connect:Direct Command Line Interface

The Java-based Connect:Direct Command Line Interface (Secure+ CLI) is provided to enable you to create customized scripts that automate implementing Connect:Direct Secure Plus. Sample Microsoft Windows scripts are provided as models for your customized scripts. You can save these scripts with another name, modify them to reflect your environment, and distribute them throughout your enterprise. For more information about using the Secure+ CLI, commands and parameter descriptions, and the scripts, see [“Start and Set Up the Connect:Direct Secure Plus CLI” on page 200](#).

## Plan the Connect:Direct Secure Plus Configuration

Before you configure the Connect:Direct environment for secure operations, first plan how you will use Connect:Direct Secure Plus. Configure the Connect:Direct Secure Plus environment, based on company needs or preferences.

## General Planning for Connect:Direct Secure Plus

Since all remote nodes are automatically enabled with the protocol defined in the local node record, determine which protocol will be used by most trading partners. Then configure the local node with this

protocol. If a trading partner is not using the protocol defined in the local node record, you must configure the remote node record for that trading partner with the protocol.

Connect:Direct Secure Plus uses two files to initiate a TLS session: a trusted root certificate file and a key certificate file.

**Note:** Connect:Direct Secure Plus does not support server gated crypto (SGC) certificates.

- The trusted root certificate file verifies the identity of trusted sources who issue certificates. To use Connect:Direct Secure Plus communication with a trading partner, exchange trusted root file with the trading partner. The trading partner must identify the trusted root file used to validate trusted sources in a certificate when it configures its Connect:Direct Secure Plus parameters file.
- A key certificate file is required at all sending locations and describes the identity of the secure node. This file includes information about the certificate being exchanged and the private key that defines the server.

When a trading partner attempts to establish communications with a Connect:Direct node, the node sends the public key certificate to the trading partner to verify its identity. The location of the key certificate file is configured in the Connect:Direct Secure Plus parameters file. The private key in the key certificate file is never sent (disclosed) by Connect:Direct.

The following list summarizes the tasks to configure Connect:Direct Secure Plus:

- Populate the parameters file at your site by importing the Connect:Direct network map. This task creates a local node record and remote node records.
- Enable the TLS protocol in the local node record. Enabling the local node record configures remote nodes to default to the settings in the local node record. To enable TLS, activate the following options:
  - Identify the trusted root certificate file that authenticates the trusted authorities.
  - Identify the key certificate file.

If you identify the trusted root file and the key certificate file to use for secure communications in the local node record, the trusted root file must define the identity of all CAs for all trading partners, and the root certificate associated with the key certificate file must include certificate and private key information for all certificates.

- Identify a cipher suite to use to encrypt data in both the sending and receiving node. After secure communication is established, Connect:Direct Secure Plus determines what cipher has been defined at both the sending and the receiving node and uses this cipher to encrypt data before sending it. If more than one cipher is enabled, the preferences defined in the server parameters file determine the cipher suite used for the SSL protocol and the preferences defined in the client parameters file determine the cipher suite used for the TLS protocol.
- If you want to enable a second level of security, activate client authentication.
- If you want to enable common name checking, you must enable this feature in the remote node record.
- For remote nodes that are using the protocol defined in the local node record, configure the remote nodes to implement any of the following additional security features:
  - Activate client authentication.
  - Configure the remote node record of trading partners with the same cipher suites enabled by the trading partner because trading partners must use the same cipher suite to enable data encryption.
- If a trading partner uses a protocol that is different from the protocol defined in the local node record, define the protocol in the remote node record. The remote node record must identify the same protocol as that used by the trading partner. Otherwise, Connect:Direct Secure Plus fails.
- If a trading partner does not use Connect:Direct Secure Plus, disable it in that remote node record.

## Summary of Processing Using Connect:Direct Secure Plus

After you configure Connect:Direct Secure Plus, you are ready to exchange data securely with other security-enabled Connect:Direct nodes. Your node must also be defined in the parameters file of trading

partners. Data is securely exchanged between two nodes using the protocol defined in the parameters file.

## TLS Connect:Direct Secure Plus Data Exchange

Data exchange consists of three steps: authentication, sending data, and receiving data. The TLS protocol data exchange process is described in the following sections. The primary node initiates the data transmission, and the secondary node receives the data. The following description of processing depicts the PNODE as sending data and the SNODE as receiving data.

### Authentication

The following steps occur during authentication:

1. The PNODE sends a control block to the SNODE. The SNODE confirms that it has a record defined in the Connect:Direct Secure Plus parameters file for the PNODE and determines the cipher suite to use for secure communication. If the SNODE finds a record for the PNODE and a common cipher suite can be negotiated, the session continues.
2. The SNODE sends its certificate back to the PNODE. Information for creating an encryption key is included. If client authentication is enabled, the SNODE also requests a certificate from the PNODE.
3. The PNODE verifies that the certificate of the SNODE is in its parameters file and generates a session key. If requested, it sends a client certificate to the SNODE for verification.
4. The SNODE confirms that a secure environment is established and returns a secure channel message.
5. The PNODE authenticates the SNODE and establishes communications.

### Sending Customer Data

After communication is authenticated, the PNODE begins transmitting data.

- Information for encrypting data is exchanged in the control blocks.
- If data compression is enabled, the PNODE compresses the data.
- The PNODE encrypts the data with a cipher suite recognized by both communications nodes.

### Receiving Customer Data

The SNODE receives the data.

- The SNODE decrypts the data using a cipher suite available for both the PNODE and the SNODE.
- If the data is compressed, the receiving node decompresses it.

## IBM Connect:Direct Secure Plus for Microsoft Windows Documentation

The *IBM Connect:Direct Secure Plus for Microsoft Windows Implementation Guide* describes how to implement point-to-point security into Connect:Direct operations with Connect:Direct Secure Plus. This document includes information to plan, install, configure, and use Connect:Direct.

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

### Task Overview

The following table directs you to the information required to perform the tasks documented in the Connect:Direct documentation:

Task	For More Information see
Understanding Connect:Direct Secure Plus	<a href="#">“About Connect:Direct Secure Plus” on page 183</a>

Task	For More Information see
Setting up Connect:Direct Secure Plus	<a href="#">“Set Up Connect:Direct Secure Plus” on page 189</a>
Automating the Set up of Connect:Direct Secure Plus	<a href="#">“Start and Set Up the Connect:Direct Secure Plus CLI” on page 200</a>
Maintenance tasks such as viewing all nodes and their attributes	<a href="#">“Connect:Direct Secure Plus Node List” on page 212</a>
Viewing Connect:Direct Secure Plus statistics	<a href="#">“View Statistics” on page 215</a>
Understanding error messages and resolving errors	<a href="#">“Troubleshooting” on page 220</a>

## Set Up Connect:Direct Secure Plus

---

### Set Up Connect:Direct Secure Plus

Before you begin using Connect:Direct Secure Plus, you must configure nodes for secure operations.

You can install Connect:Direct Secure Plus using the Connect:Direct for Microsoft Windows installation script. For more information on installing Connect:Direct Secure Plus, see the *IBM Connect:Direct for Microsoft Windows Getting Started Guide*.



**CAUTION:** After Connect:Direct Secure Plus is installed, the system administrator is responsible for securing access to the Connect:Direct Secure Plus Administration Tool, Connect:Direct Secure Plus CLI, and parameters files. The Connect:Direct Secure Plus administrator and Connect:Direct server need full access to the Connect:Direct Secure Plus directory. No other users require access.

### Start Secure+ Admin Tool

#### About this task

Use the Secure+ Admin Tool to set up and maintain a Connect:Direct Secure Plus operation.

You can also use the Connect:Direct Secure Plus Command Line Interface (Secure+ CLI) to set up and manage Connect:Direct Secure Plus operations. See [“Start and Set Up the Connect:Direct Secure Plus CLI” on page 200](#).

**Note:** The parameters file is not dynamically updated. When multiple users update the parameters file, each user must close and reopen the file to display new records added by all sources.

#### Procedure

1. Click **Start > All Programs**.
2. Select **IBM Connect Direct v6.1 > CD Secure+ Admin Tool**. The Secure+ Admin Tool Main Window is displayed.

The Secure+ Admin Tool starts and opens the Connect:Direct Secure Plus parameters file for the associated Connect:Direct node.

## Prepare to Set Up Connect:Direct Secure Plus

---

Before you configure the Connect:Direct Secure Plus environment, perform the following setup procedures.

- Complete a worksheet for the local node record and a worksheet for each trading partner for whom you plan to enable Connect:Direct Secure Plus. Use the Local Node Security Feature Definition Worksheet to record the settings you plan to enable for the local node. For each trading partner, complete a Remote

Node Security Feature Definition Worksheet and record the settings to enable Connect:Direct Secure Plus for the trading partner.

- A keystore is used to help create and manage certificates using the IBM utility, iKeyman. You can use iKeyman to:
  - Create and manage key databases.
  - Create self-signed digital certificates for testing.
  - Add certificate authority (CA) and intermediate certificates.
  - Transfer certificates between key databases.
  - Create certificate requests and receive a digital certificate issued by a CA in response to a request.
  - For additional information on how to use iKeyman, see [http://www-01.ibm.com/support/knowledgecenter/SSYKE2\\_6.0.0/com.ibm.java.security.component.60.doc/security-component/ikeyman\\_overview.html?lang=en](http://www-01.ibm.com/support/knowledgecenter/SSYKE2_6.0.0/com.ibm.java.security.component.60.doc/security-component/ikeyman_overview.html?lang=en).
- Populate the Connect:Direct Secure Plus parameters file to include a record for each node running Connect:Direct Secure Plus. To communicate with a node running Connect:Direct Secure Plus, the node must have a record in the Connect:Direct network map and the Connect:Direct Secure Plus parameters file.

## Populate the Connect:Direct Secure Plus Parameters File

### About this task

To communicate with a trading partner using Connect:Direct Secure Plus, you define a node record for that partner in both the Connect:Direct network map and the Connect:Direct Secure Plus parameters file. To set up the Connect:Direct Secure Plus environment, you can populate the Connect:Direct Secure Plus parameters file from entries defined in an existing network map.

When you populate the parameters file from the network map, a record is automatically created in the parameters file for each node entry in the network map. Initially, Connect:Direct Secure Plus is disabled for each of the records created. You configure the local node record to activate Connect:Direct Secure Plus for all nodes in the parameters file.

### Procedure

1. From the Secure+ Admin Tool Main Window, click **File > Sync with Netmap**. The **Available Netmaps** dialog box is displayed.
2. Select the netmap to open and click **Sync**. The **Select Netmap Entries to Add** dialog box is displayed.
3. Click **Add All**. The **Select Parameters File Entries to Delete** dialog box is displayed.
4. Click **Skip** to close the parameters file without deleting any entries.

The Connect:Direct Secure Plus parameters file is populated and the Secure+ Admin Tool Main Window displays remote node records in the parameters file including the records you added from the network map.

## Configure Nodes

---

### Node Configuration Overview

When you import the network map records into the Connect:Direct Secure Plus parameters file, Connect:Direct Secure Plus parameters are disabled and you will need to configure the .Local node record.

To configure the nodes, complete the following procedures:

- Import existing Certificates.



- Configure or create a new CMS Key Store through the Key Management menu on the Secure+ Admin Tool.
- Configure the Connect:Direct Secure Plus .Local node record.  
Define the security options for the local node. Because TLS provide the strongest authentication with easy-to-maintain keys, configure the local node for one of these protocols. Determine which protocol is used by most trading partners and configure the local node with this protocol.
- Disable remote nodes that do not use Connect:Direct Secure Plus
- Customize a remote node for the following configurations:
  - To use a unique certificate file to authenticate a trading partner
  - To use a different self-signed certificate for client or server authentication
  - To identify a unique cipher suite used by a trading partner
  - To activate common name validation
  - To activate client authentication
  - To activate external authentication
- Configure all remote nodes that use a protocol that is not defined in the local node  
When you configure the local node, all remote nodes are automatically configured to the protocol defined in the local node. If a trading partner uses a different protocol, you must turn on the protocol in the remote node record. For example, if you activate the TLS protocol in the .Local node record and a trading partner uses the SSL protocol, configure the SSL protocol in the remote node record for the trading partner.
- If you want to use External Authentication Server to validate certificates:
  - Update the .SEAServer record with the External Authentication Server host name and port
  - Enable TLS
  - Enable external authentication
  - Specify the certificate validation definition to use
- If you want to prevent non-secure API connections from communicating with a Connect:Direct Secure Plus enabled server:
  - Define a remote node called .Client
  - Enable TLS
  - Disable override

## Import Existing Certificates

### About this task

Before performing your .Local node configuration, you need to import existing certificates.

To import existing certificates:

### Procedure

1. Import existing certificates, either keycerts or trusted root files from trading partners into the Key Store. On the Secure+ Admin Tool main window, from the Key Management menu, select **Configure Key Store**. The Key Store Manager window appears.
2. Verify the CMS Key Store path. If incorrect, click **browse** to locate the Key Store path. The Browse CMS KeyStore File window appears.
3. The default Key Store name is: cdkeystore.kdb To locate the default Key Store path, navigate to the Key Store file.

```
Windows path: <cdinstalldir>\Server\Secure+\Certificates\cdkeystore.kdb
Unix path: <cdinstalldir>/ndm/secure+/certificates/cdkeystore.kdb
```

4. Click **Import**. On the Import PEM KeyStore File window, navigate to and select the certificate file you want to use and click **OK**.
5. If a key certificate file is being imported, the password must be entered. The KeyStore Password window appears. Type your password and click OK.
6. The PEM Certificate Viewer displays to allow a review of the certificate file. Verify the certificate is valid and click the **Import** button. Import Results window displays with status of imported certificate. Click **Close**.
7. The certificate is imported and given a Label based on the certificate Common Name, (CN=). Note the serial number to identify the correct certificate after import.  
**Note:** A common name is used for Label and identification which means that multiple certificates can have the same common name and therefore, can be overwritten depending on the setting of the Default Mode. Additionally, the Default Mode of Import is Add or Replace Certificates.
8. Click **OK** to create the new CMS KeyStore file. Key Store Manager will display contents of the new keystore.

## Create CMS Key Store

### About this task

Before performing your .Local node configuration, you may need to create a new CMS Key Store file.

To create a new CMS Key Store file:

### Procedure

1. On the Key Store Manager window, click **New**. The Create new CMS KeyStore File dialog box appears.
2. Enter the Directory location (you can also Browse to the location desired), the KeyStore file name, and the password for the new KeyStore file. You can also choose to Populate with standard certificate authorities. This will import all standard public CA Root certificates into the new KeyStore file.
3. Click **OK** to create the new CMS KeyStore file. Key Store Manager will display contents of the new keystore.
4. Click **Import**. On the Import PEM KeyStore File window, navigate to and select the certificate file you want to use and click **OK**.
5. If a key certificate file is being imported, the password must be entered. The KeyStore Password window appears. Type your password and click OK.
6. The PEM Certificate Viewer displays to allow a review of the certificate file. Verify the certificate is valid and click the **Import** button. Import Results window displays with status of imported certificate. Click **Close**.
7. The certificate is imported and given a Label based on the certificate Common Name, (CN=). Note the serial number to identify the correct certificate after import.  
**Note:** A common name is used for Label and identification therefore multiple certificates can have the same common name and therefore, can be overwritten depending on the setting of the Default Mode. Additionally, the Default Mode of Import is Add or Replace Certificates.

# Configure the Connect:Direct Secure Plus .Local Node Record

## About this task

Before you can configure the .Local node record, you must either import your existing certificates or create and configure a CMS Key Store. For additional information, see Import Existing Certificates or Create CMS Key Store in the documentation library.

It is recommended that you configure the .Local node record with the protocol used by most of your trading partners. Because remote node records can use the attributes defined in the .Local node record, defining the .Local node record with the most commonly used protocol saves time. After you define the protocol in the .Local node record, all remote nodes default to that protocol. Also, identify the trusted root file to be used to authenticate trading partners.

To configure the local node, refer to the Local Node Security Feature Definition Worksheet that you completed for the .Local node record security settings and complete the following procedure:

## Procedure

1. From the Secure+ Admin Tool Main Window, double-click the .Local record. The Edit Record dialog box displays the Security Options tab, the node name, and the type of node.
2. Set the Security Options for the local or remote node entry you are configuring and if necessary, modify the time-out value in **Authentication Timeout**.

Refer to the following table for an explanation of the Security Options boxes:

**Note:** The SSL3.0, TLS 1.0 and TLS 1.1 protocols are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2. If deprecated protocols are required, TLS 1.3 should not be enabled in the trading partner's configuration, otherwise the handshake may fail. Deprecated protocols should be exclusively configured per node. The Secure+ feature continues to support SSL 3.0, TLS 1.0 and TLS 1.1.

Field Name	Field Definition	Valid Values
Node Name	Specifies the node record name.	.Local This is not an editable field.
Base Record	Specifies the name of the base record. If an alias record is selected, the base record name is displayed in this box.	Name of the local Connect:Direct node.
Type	Specifies the current record type.	Local for a local record and Remote for a remote record. This is not an editable field.
Disable Secure+	Disables Connect:Direct Secure Plus.	Default value is Disable Secure+. Note: If this option is selected, override is enabled, and no remote node definition exists for the remote node in the Connect:Direct Secure Plus parameters file, Connect:Direct Secure Plus is bypassed.

<b>Field Name</b>	<b>Field Definition</b>	<b>Valid Values</b>
Enable SSL 3.0 Protocol	Enables SSL protocol to ensure that data is securely transmitted.  The SSL3.0 is deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.	The default value is Disable Secure +.
Enable TLS 1.0 Protocol	Enables TLS protocol to ensure that data is securely transmitted.  The TLS 1.0 is deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.	The default value is Disable Secure +.
Enable TLS 1.1 Protocol	Enables TLS protocol to ensure that data is securely transmitted.  The TLS1.1 is deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.	The default value is Disable Secure +.
Enable TLS 1.2 Protocol	Enables TLS protocol to ensure that data is securely transmitted.	The default value is Disable Secure +.
Enable TLS 1.3 Protocol	Enables TLS protocol to ensure that data is securely transmitted.	The default value is Disable Secure +.
Disable	Disables the ability to override values in the .Local node record with values in the remote node record.	The default value is Disable.
FIPS 140-2	Enables FIPS 140-2 security.	The default value is Disable.
SP800-131A Transition	Enables NIST SP800-131a security in transition mode.	The default value is Disable.
SP800-131A	Enables NIST SP800-131a security mode.	The default value is Disable.
Suite B 128 bit	Enables Suite B 128 bit security.	The default value is Disable.
Suite B 192 bit	Enables Suite B 192 bit security.	The default value is Disable.

Field Name	Field Definition	Valid Values
Node or Copy Statement Override	<p>There are several types of overrides. For both PNODE and SNODE, this parameter indicates whether Remote Node record parameters will override the .Local Node record parameters or not.</p> <p>If it is set to No, or if set to Yes and there is no correlating Remote Node record for a given session, then:</p> <ul style="list-style-type: none"> <li>• For PNODE, this parameter indicates whether process overrides, which may optionally be specified in Process, Submit, and Copy statements, will be allowed.</li> <li>• For SNODE, this parameter indicates whether: <ul style="list-style-type: none"> <li>– The Secure+ protocol specified by the PNODE will be allowed to override that specified by the SNODE.</li> <li>– To allow unsecured incoming sessions to proceed.</li> </ul> </li> </ul>	The default value is No.
Authentication Timeout	<p>Specifies maximum time, in seconds, that the system waits to receive the Connect:Direct Secure Plus blocks exchanged during the Connect:Direct Secure Plus authentication process.</p> <p>If you specify a value of 0, Connect:Direct waits indefinitely to receive the next message.</p> <p>Specify a time to prevent malicious entry from taking as much time as necessary to attack the authentication process.</p>	<p>A numeric value equal to or greater than 0, ranging from 0 to 3600.</p> <p>The default is 120 seconds.</p>

3. Click the **TLS Options** tab. The **TLS Options** dialog box is displayed.
4. Select an existing Key Certificate from the key store. To select a Key Certificate from the keystore, click **Browse** next to **Key Certificate Label**. The **CMS KeyStore Certificate Viewer** appears.

**Note:** You must add or import the key certificate into your key store prior to configuring your node. For additional information, see Import Existing Certificates or Create CMS Key Store in the documentation library. For additional information on how to use iKeyman, see [http://www-01.ibm.com/support/knowledgecenter/SSYKE2\\_6.0.0/com.ibm.java.security.component.60.doc/security-component/ikeyman\\_overview.html?lang=en](http://www-01.ibm.com/support/knowledgecenter/SSYKE2_6.0.0/com.ibm.java.security.component.60.doc/security-component/ikeyman_overview.html?lang=en).

5. In the Key Certificates area, select the key certificate you want to use and click **OK** box.
6. Click the **External Authentication** tab. The **External Authentication** dialog box is displayed.
7. Choose one of the following options:

- To enable external authentication on the remote node, click **Yes** in the **Enable External Authentication** box.
  - To disable external authentication on the remote node, click **No**.
8. Type the Certificate Validation Definition character string defined in External Authentication Server.
  9. Click **OK** to close the **Edit Record** dialog box and update the parameters file.

## Configure Connect:Direct Secure Plus Remote Node Record

### About this task

Before you can configure the .Remote node record, you must either import your existing certificates or create and configure a CMS Key Store. For additional information, see [Import Existing Certificates or Create CMS Key Store](#) in the documentation library.

Configure the Remote node record with the protocol used by most of your trading partners. Because remote node records can use the attributes defined in the Remote node record, defining the Remote node record with the most commonly used protocol saves time. After you define the protocol in the Remote node record, all remote nodes default to that protocol. Also, identify the trusted root file to be used to authenticate trading partners.

To configure the local node, refer to the [Local Node Security Feature Definition Worksheet](#) that you completed for the Remote node record security settings and complete the following procedure:

### Procedure

1. From the Secure+ Admin Tool Main Window, double-click the .Remote record. The Edit Record dialog box displays the Security Options tab, the node name, and the type of node.
2. Set the Security Options for the local or remote node entry you are configuring and if necessary, modify the time-out value in **Authentication Timeout**.

Refer to the following table for an explanation of the Security Options boxes:

**Note:** The SSL3.0, TLS 1.0 and TLS 1.1 protocols are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2. If deprecated protocols are required, TLS 1.3 should not be enabled in the trading partner's configuration, otherwise the handshake may fail. Deprecated protocols should be exclusively configured per node. The Secure+ feature continues to support SSL 3.0, TLS 1.0 and TLS 1.1.

Field Name	Field Definition	Valid Values
Node Name	Specifies the node record name. <b>Important:</b> Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$ . _ - to ensure that the entries can be properly managed by Control Center, Sterling Connect:Direct Browser User Interface, or IBM Sterling Connect:Direct Application Interface for Java for Java (AIJ) programs.	.Remote This is not an editable field.
Base Record	Specifies the name of the base record. If an alias record is selected, the base record name is displayed in this box.	Name of the local Connect:Direct node.

Field Name	Field Definition	Valid Values
Type	Specifies the current record type.	Local for a local record and Remote for a remote record.  This is not an editable field.
Disable Secure+	Disables Connect:Direct Secure Plus.	Default value is Disable Secure+.  Note: If this option is selected, override is enabled, and no remote node definition exists for the remote node in the Connect:Direct Secure Plus parameters file, Connect:Direct Secure Plus is bypassed.
Enable SSL 3.0 Protocol	Enables SSL protocol to ensure that data is securely transmitted.  The SSL3.0 is deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.	The default value is Disable Secure+.
Enable TLS 1.0 Protocol	Enables TLS protocol to ensure that data is securely transmitted.  TLS1.0 is deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.	The default value is Disable Secure+.
Enable TLS 1.1 Protocol	Enables TLS protocol to ensure that data is securely transmitted.  The TLS1.1 is deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.	The default value is Disable Secure+.
Enable TLS 1.2 Protocol	Enables TLS protocol to ensure that data is securely transmitted.	The default value is Disable Secure+.
Enable TLS 1.3 Protocol	Enables TLS protocol to ensure that data is securely transmitted.	The default value is Disable Secure+.
Disable	Disables the ability to override values in the .Remote node record with values in the remote node record.	The default value is Disable.
FIPS 140-2	Enables FIPS 140-2 security.	The default value is Disable.
SP800-131A Transition	Enables NIST SP800-131a security in transition mode.	The default value is Disable.
SP800-131A	Enables NIST SP800-131a security mode.	The default value is Disable.
Suite B 128 bit	Enables Suite B 128 bit security.	The default value is Disable.
Suite B 192 bit	Enables Suite B 192 bit security.	The default value is Disable.

Field Name	Field Definition	Valid Values
Node or Copy Statement Override	<p>For PNODE, this parameter indicates whether process overrides, which may optionally be specified in Process, Submit, and Copy statements, will be allowed.</p> <p>For SNODE, this parameter indicates whether the Secure+ protocol specified by the PNODE will be allowed to override that specified by the SNODE.</p>	The default value is No.
Authentication Timeout	<p>Specifies maximum time, in seconds, that the system waits to receive the Connect:Direct Secure Plus blocks exchanged during the Connect:Direct authentication process.</p> <p>If you specify a value of 0, Connect:Direct waits indefinitely to receive the next message.</p> <p>Specify a time to prevent malicious entry from taking as much time as necessary to attack the authentication process.</p>	<p>A numeric value equal to or greater than 0, ranging from 0 to 3600.</p> <p>The default is 120 seconds.</p>

3. Click the **TLS Options** tab. The **TLS Options** dialog box is displayed.
4. Select an existing Key Certificate from the key store. To select a Key Certificate from the keystore, click **Browse** next to **Key Certificate Label**. The **CMS KeyStore Certificate Viewer** appears.
 

**Note:** You must add or import the key certificate into your key store prior to configuring your node. For additional information, see Import Existing Certificates or Create CMS Key Store in the documentation library. For additional information on how to use iKeyman, see [http://www-01.ibm.com/support/knowledgecenter/SSYKE2\\_6.0.0/com.ibm.java.security.component.60.doc/security-component/ikeyman\\_overview.html?lang=en](http://www-01.ibm.com/support/knowledgecenter/SSYKE2_6.0.0/com.ibm.java.security.component.60.doc/security-component/ikeyman_overview.html?lang=en).
5. In the Key Certificates area, select the key certificate you want to use and click **OK** box.
6. Click the **External Authentication** tab. The **External Authentication** dialog box is displayed.
7. Choose one of the following options:
  - To enable external authentication on the remote node, click **Yes** in the **Enable External Authentication** box.
  - To disable external authentication on the remote node, click **No**.
8. Type the Certificate Validation Definition character string defined in External Authentication Server.
9. Click **OK** to close the **Edit Record** dialog box and update the parameters file.

## Validate the Configuration

### About this task

Perform this procedure to ensure that the nodes have been properly configured. The validation process checks each node to ensure that all necessary options have been defined and keys have been exchanged.

To validate the parameters file:



## Procedure

1. In the Secure+ Admin Tool, click **File > Validate Secure+**. The **Secure+ Admin Tool - Validation Results** window is displayed.

If the parameters file is not correctly configured, warning and error messages are displayed.

2. Read each warning message. To correct each warning or error reported, go back to the parameters file and make changes as needed.

**Note:** Warning messages do not always mean that the parameters file is incorrectly configured. Some warning messages are informational only.

3. Click **Close** to close the **Validation Results** window.

## Enable or Disable External Authentication for a Remote Node

### About this task

On a node-by-node basis, you can specify whether a remote node uses external authentication or if that remote node defaults to the external authentication setting in the .Local node record.

Complete the following procedure to configure a remote node for external authentication:

### Procedure

1. If necessary, open the remote node record. The **Edit Record** dialog box is displayed.
2. Click the **External Authentication** tab.
3. Choose one of the following options:
  - To enable external authentication on the remote node, click **Yes** in the **Enable External Authentication** box.
  - To disable external authentication on the remote node, click **No**.
  - To default to the external authentication setting defined in the .Local node record, click **Default to Local Node**.

**Note:** If external authentication is enabled in the .Local node record, it is automatically enabled in all remote node records.

4. Type the Certificate Validation Definition character string defined in Sterling External Authentication Server.
5. Click **OK** to close the **Edit Record** dialog box and update the parameters file.

### Related concepts

[Transport Layer Security Protocol \(TLS\)](#)

## Configure External Authentication in the .SEAServer Record

### About this task

At installation, a record named .SEAServer is created in the parameters file, which enables Connect:Direct Secure Plus to interface with External Authentication Server during TLS sessions to validate certificates. External Authentication Server properties are configured in this record and enabled/disabled in the local and remote node records.

Complete the following procedure to configure the server properties that will allow Connect:Direct for Microsoft Windows to interface with External Authentication Server:

**Note:** The values specified for this procedure must match the values specified in External Authentication Server.

## Procedure

1. Double-click the record called **.SEAServer**.
2. Type the Host Name for External Authentication Server.
3. Type the Port Number where External Authentication Server is listening. The default is 61366.
4. To enable caching SEAS certificate validation response, select **Enable Caching**.  
When enabled, Connect:Direct Secure Plus can reuse previously fetched certificate validity responses from External Authentication Server that is, cache the responses to ease the certificate validation process when Connect:Direct interfaces with External Authentication Server during TLS sessions.
5. Type the **Cache Validity per certificate in hours**. Default is 24 hours. Range: 1-720 hours.
6. **Cache grace validity time per certificate when SEAS is unavailable in hours**  
Type the number of hours when the local cache entry of certificate expires and External Authentication Server is unavailable such that Connect:Direct Secure Plus can accept it from its cache. Default is 0 hours which means cache grace validity time does not apply. Range: 0-720 hours.  
**Note: Cache grace validity time per certificate when SEAS is unavailable in hours** should always be greater than or equal to **Cache Validity per certificate in hours**.
7. Click **OK** to update the record.

### Related concepts

[Transport Layer Security Protocol \(TLS\)](#)

## Automate Setup Using the CLI

### Start and Set Up the Connect:Direct Secure Plus CLI

The following sections describe the commands and parameters used to start and set up the command line environment.

#### Start the Connect:Direct Secure Plus CLI

To start the Connect:Direct Secure Plus Command Line Interface:

1. Change to the following directory: C:\Program Files\IBM\Connect Direct v4.6.00\Server\Secure+.
2. Enter the following command:

```
spcli.cmd
```

#### Control the Display of Commands

Set the following parameters to define how error messages are captured:

Parameter	Definition	Values
-li	Switch to enable the display of commands to the terminal.	y   n
-lo	Switch to enable the display of output and error messages to the terminal.	y   n
-le	Switch to enable the display of errors to STDERR.	y   n
-e	Switch to tell the Connect:Direct Secure Plus CLI to exit when the return code is higher than the specified number. If you do not include this parameter, Connect:Direct Secure Plus CLI runs even after an error occurs.	0   4   8   16
-p	Full path of the default parameters file directory. The file in this directory is opened automatically.	

Parameter	Definition	Values
-h	Switch to display the usage of the Connect:Direct Secure Plus CLI.	

## Control Help

The Help command determines what help information is displayed. You can list all the Connect:Direct Secure Plus CLI commands and display help for individual commands.

Command	Description
help	Displays all the Connect:Direct Secure Plus CLI commands.
help <command>	Displays help for the specified command.

## Specify Delimiter Characters

Define the following commands to determine how error messages are captured:

Command	Definition	Values
Set begdelim= enddelim=	Defines beginning and ending character to use to enclose keywords that use blanks and other special characters.	Any character The default value is " (double quotes).

## Use LCU Files to Encrypt Passwords for Use with the Connect:Direct Secure Plus CLI

The Connect:Direct Secure Plus CLI displays passwords in plain text. If you are required to use encrypted passwords, use the Local Connection Utility (LCU) to create an LCU file that contains non-encrypted information for encrypting the password.

For more information on creating and using LCU files, see [“Configure Encrypted Passwords Using the LCU”](#) on page 227.

## Sample Scripts

The following script is provided as a model for creating custom scripts to define your Connect:Direct Secure Plus environment and automate the implementation of it. To prevent any loss of data, you cannot run the script, but you can save it with a different name and modify it to suit your needs. The sample script is available in Model Automation Scripts. The script is designed to assist you as follows:

### **spcust\_sample1.sh**

An example of configuring IBM Connect:Direct to use the TLS protocol with the Secure+ CLI. The example demonstrates the configuration of IBM Connect:Direct with the trusted root file, key certificates, and ciphers.

## Manage the Parameters File

The commands in the following table describe how to maintain the Secure+ parameters file from the command line interface.

Command	Description	Parameter	Values
<b>Init Parmfile</b>	Creates the Secure+ parameters file. Must be initialized before you can define nodes.	localnode=Name of the local node where the Secure+ parameters file will be created.	local node name
		path=Location where the Secure+ parameters file will be created.	directory location For example, <i>d_dir</i> \Server\Secure+\Nodes
		passphrase=Arbitrary set of characters that encrypts the Secure+ parameters file.	a string at least 32 characters long
<b>Open Parmfile</b>	Opens a Secure+ parameters file so that you can configure it.	path=Location where the Secure+ parameters file will be created.	directory location For example, <i>d_dir</i> \Server\Secure+\Nodes
<b>Close Parmfile</b>	Closes the Secure+ parameters file. After this command is issued, no more updates can be performed on the Secure+ parameters file.	None	None
<b>Refresh Parmfile</b>	Refreshes the Secure+ parameters file. This will close the current parameters file and reopen it, bringing in any changes since last opened.	None	None
<b>Validate Parmfile</b>	Validates the Secure+ parameters file and ensures that it is a valid file.	None	None
<b>Rekey Parmfile</b>	Recreates the Secure+ parameters file if it becomes corrupted.	passphrase=Arbitrary set of characters that encrypts the Secure+ parameters file.	passphrase, up to 32 characters long

Command	Description	Parameter	Values
<b>Sync Netmap</b>	Imports remote node records defined in the IBM Connect:Direct network map.	path=Location and name of the network map file.	location of network map file
		name=Name of the node in the network map. Use wildcard characters to resync more than one node at a time.	node name or wildcard Wildcard values are:  Asterisk (*)—any number of characters. Example: kps.* syncs up all nodes with a name that starts with kps.  Question mark (?)—a single character. Example: k?s.* syncs up kas.* and kbs.*

## Displaying Information

The following commands are available to display information:

Command	Description	Parameter
display info	Displays information about when the parameters file was last updated.	None
display all	Displays all nodes in the parameters file.	None
display localnode	Displays the values defined in the .Local node record.	None
display remotenode	Displays the values defined in remote node records.	name—The name of the node to display information about.  Use wildcard characters to display information about a group of remote node records. The options are:  Asterisk (*)—Indicates any number of characters. For example, kps.* displays all nodes with a name that starts with kps.  Question mark (?)—Indicates a single character. For example: k?s.* displays kas.* and kbs.*
display client	Displays the values defined in the .Client node record.	None
display seaserver	Displays the values defined in the .SEAServer record.	None

## Manage CMS Keystore

The commands in the following table describe how to create and maintain the CMS keystore file from the command line interface.

Command	Description	Parameter	Values
create keystore	Will create a new CMS Key Store file.	File=While a default keystore file is created at installation and can be used, you may need to create a new CMS KeyStore File.	<path to CMS KeyStore file (*.kdb)> Default path is in: d_dir\Server\Secure+\Certificates \ndm\secure+\certificates \cdkeystore.kdb
		Passphrase=The password for the new KeyStore file.	A string with a minimum of three characters and a maximum of eighty characters.  *This password must be retained; it will be required to administer the Secure+ KeyStore.
		PopulateRoots=Populate with standard certificate authorities. This will import all standard public CA Root certificates into the new KeyStore file.	y   <u>n</u>
update keystore	Updates the CMS KeyStore	File=Path to existing CMS KeyStore and filename.	<path to CMS KeyStore file (*.kdb)> Default path is in: d_dir\Server\Secure+\Certificates \ndm\secure+\certificates \cdkeystore.kdb
		Passphrase=The password for the KeyStore file.	The retained password which was given at the creation of the keystore.
import keycert	Imports existing keycerts into the keystore file.	File=Existing key certificate file.  *This file contains the private key*	Full path and filename to key certificate file to be imported.
		Passphrase=Password of key certificate file to be imported.	Pre-defined password of key certificate file.
		Label=(optional) Name of imported key certificate file.	A string of characters which can be an alias name but if it is not defined, the Common Name of the certificate will be the label used.
		SyncNodes=Update node/certificate references	y   <u>n</u>
		ImportMode=Type of import to be used.	Add   Replace   <u>AddOrReplace</u>

Command	Description	Parameter	Values
import trustedcert	Imports public certificate files from trading partners.	File=Trusted public file from trading partner.	Full path and filename to trusted certificate file to be imported.
		ImportMode=Type of import to be used.	Add   Replace   <u>AddOrReplace</u>
delete keystoreentry	Deletes certificates from CMS keystore.	File=Can be either key certificate file or trusted public trading partner file.	Full path and filename to certificate file.
		Label=Specified label of imported certificate file.	Label which was defined at time of import of the certificate file.
		DeleteChain=Defines whether to delete the entire chain, if it exists.	y   <u>n</u>
		SyncNodes=Reset node/certificate references	y   <u>n</u>

## Update the .Local Node Record

The **update localnode** command configures the protocol for the .Local node record. The command has the following parameters:

Command	Parameter	Values
update localnode	protocol=Specifies a comma delimited list of Protocols to use in the .Local node record.	<u>Disable</u>   TLS 1.2,TLS 1.3 TLS1.0, TLS1.1, and SSL3.0 are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2. (See Display Protocols)
	SecurityMode	<u>Disable</u>   FIPS140-2   SP800-131A_TRANSITION   SP800-131A_STRICT   SUITE_B-128   SUITE_B-192 (See Display SecurityModes)
	override=Identifies if values in the remote node can override values defined in the .Local node record.	y   <u>n</u>
	AuthTimeout=Specifies the maximum time, in seconds, that the system waits to receive the IBM Connect:Direct control blocks exchanged during the IBM Connect:Direct authentication process.	0–3600 The default is <b>120</b> seconds.

Command	Parameter	Values
	KeyCertLabel=Identifies the label of the key certificate.	keycert label   null <b>Note:</b> If no keycert label is specified, the following should be noted: Pnode sessions will fail if the remote node requires client authentication. Snode sessions will fail.
	EncryptData=If no is specified, Encrypt Only Control Block Information; data is sent unencrypted. Default is Yes - data and control block information are encrypted.	y   n
	ClientAuth = Enables client authentication in a .Client node record.	y   n
	CipherSuites= Specifies the cipher suites enabled. <b>Note:</b> Only certain cipher suites are supported in FIPS-mode. For a list of the FIPS-approved cipher suites, see <i>Special Considerations</i> in the <i>IBM Connect:Direct for UNIX Release Notes</i> .	comma delimited list of cipher suites   all   null all—Enables all ciphers. null—Clears any existing values from the node definition.
	SeaEnable=Enables certificate validation by Sterling External Authentication Server	y   n
	SeaCertValDef=Character string defined in Sterling External Authentication Server (SEAS).	character string   null null—Clears any existing values from the node definition.

## Manage Remote Node Records

This section contains the commands and parameters used to create, update, display, and delete remote node records.

**Important:** Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$ . \_ - to ensure that the entries can be properly managed by Control Center, Sterling Connect:Direct Browser User Interface, or IBM Sterling Connect:Direct Application Interface for Java for Java (AIJ) programs.

### Create a Remote Node Record

The **create remotenode** command creates a remote node record and configures the protocol settings. The command has the following parameters:

Command	Parameter	Values
create remotenode	model=Name of an existing node to use as a model to copy from.	name of a valid remote node
	Name=Identifies name of the remote node record.	name



Command	Parameter	Values
	protocol=Specifies a comma delimited list of Protocols to use in the remote node record.	Disable   TLS1.2,TLS 1.3   <u>DefaultToLN</u> TLS1.0, TLS1.1, SSL3.0 are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2. (See Display Protocols)
	SecurityMode	Disable   FIPS140-2   SP800-131A_TRANSITION   SP800-131A_STRICT   SUITE_B-128   SUITE_B-192   <u>DefaultToLN</u> (See Display SecurityModes)
	override=Identifies if values in the copy statement can override values defined in the remote node record.	y   n   <u>DefaultToLN</u>
	AuthTimeout=Specifies the maximum time, in seconds, that the system waits to receive the IBM Connect:Direct control blocks exchanged during the IBM Connect:Direct authentication process.	0–3600 The default is <b>120</b> seconds.
	KeyCertLabel=Identifies the label of the key certificate.	keycert label   null
	EncryptData=If no is specified, Encrypt Only Control Block Information; data is sent unencrypted. Default is Yes - data and control block information are encrypted.	y   n   <u>DefaulttoLN</u>
	ClientAuth = Enables client authentication with a remote trading partner.	y   n   <u>DefaultToLN</u>
	CertCommonName=The certificate common name defined in the certificate.	name   null null—Clears any existing values from the node definition.
	CipherSuites= Specifies the cipher suites enabled.	comma delimited list of cipher suites   All   null
	SeaCertValDef=Character string defined in Sterling External Authentication Server (SEAS).	character string   null null—Clears any existing values from the node definition.

## Update the Remote Node Record

The **update remotenode** command creates a remote node record and configures the protocol settings. The command has the following parameters:

Command	Parameter	Values
update remotenode	Name=Specifies name for the remote node record.	remote node name   wildcard Use wildcard characters to update a group of remote node records. The options are: Asterisk (*)—Any number of characters. Example: kps.* displays remote nodes with a name that starts with kps. Question mark (?)—Single character. Example: k?s.* displays kas.* and kbs.*
	protocol=Specifies a comma delimited list of Protocols to use in the remote node record.	Disable   TLS 1.2, TLS 1.3   <u>DefaultToLN</u> TLS1.0, TLS1.1, and SSL3.0 are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2. (See Display Protocols)
	SecurityMode	Disable   FIPS140-2   SP800-131A_TRANSITION   SP800-131A_STRICT   SUITE_B-128   SUITE_B-192   <u>DefaultToLN</u>
	override=Identifies if values in the copy statement can override values defined in the remote node record.	y   n   <u>DefaultToLN</u>
	AuthTimeout=Specifies the maximum time, in seconds, that the system waits to receive the IBM Connect:Direct control blocks exchanged during the IBM Connect:Direct authentication process.	0–3600 The default is <b>120</b> seconds.
	KeyCertLabel=Identifies the label of the key certificate.	keycert label   null
	EncryptData=If no is specified, Encrypt Only Control Block Information; data is sent unencrypted. Default is Yes - data and control block information are encrypted.	y   n   <u>DefaulttoLN</u>
	ClientAuth = Enables client authentication with a remote trading partner.	y   n   <u>DefaultToLN</u>
	CertCommonName=The certificate common name defined in the certificate.	name   null null—Clears any existing values from the node definition.

Command	Parameter	Values
	CipherSuites= Specifies the cipher suites enabled.  <b>Note:</b> Only certain cipher suites are supported in FIPS-mode. For a list of the FIPS-approved cipher suites, see <i>Special Considerations</i> in the <i>IBM Connect:Direct for UNIX Release Notes</i> .	comma delimited list of cipher suites   All   null
	SeaEnable=Enables certificate validation by Sterling External Authentication Server.	y   n   <u>DefaultToLN</u> DefaultToLN—Defaults to the setting specified in the .Local node record
	SeaCertValDef=Character string defined in Sterling External Authentication Server (SEAS).	character string   null null—Clears any existing values from the node definition.

### Display a Remote Node Record

The display remotenode command displays information about one or more remote node records. The command has the following parameter:

Parameter	Values
name=Name of the remote node record to display information about.	node name   wildcard value  To display information about more than one remote node record, use wildcard characters.  Use wildcard characters to display information about a group of remote node records. The options are:  Asterisk (*)—Any number of characters. Example: kps.* displays remote nodes with a name that starts with kps.  Question mark (?)—A single character. Example: k?s.* displays kas.* and kbs.*

### Manage Remote Node Records

Create Alias

The **create alias** command will create an alias record for an existing node record in the Secure+ parmfile. The command has the following parameter:

Command	Parameter	Value
create alias	name=The alias name to be used.	An alias name for an existing node name record.
	basename=The name of the existing node record.	The existing node name

### Delete a Remote Node Record

The delete remotenode command deletes one or more remote node records. The command has the following parameter:

Parameter	Values
<p>name=Name of the remote node record to display information about.</p> <p>Use wildcard characters to delete a group of remote node records.</p>	<p>remote node name   wildcard value</p> <p>To display information about more than one remote node record, use wildcard characters.</p> <p>Use wildcard characters to display information about a group of remote node records. The options are:</p> <p>Asterisk (*)—Any number of characters. Example: kps.* displays remote nodes with a name that starts with kps.</p> <p>Question mark (?)—A single character. Example: k?s.* displays kas.* and kbs.*.</p>

## Update the .Client Node Record

The **update client** command creates a .Client node record and configures the protocol settings. The command has the following parameters:

Command	Parameter	Values
update client	protocol=Specifies a comma delimited list of Protocols to use in the .Client record	Disable   TLS1.2,TLS 1.3   <u>DefaultToLN</u> TLS1.0, TLS1.1, and SSL3.0 are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2. (See Display Protocols)
	SecurityMode	Disable   FIPS140-2   SP800-131A_TRANSITION   SP800-131A_STRICT   SUITE_B-128   SUITE_B-192   <u>DefaultToLN</u> (See Display SecurityModes)
	override=Enforces secure connection between a Connect:Direct client and the Connect:Direct server	y   n   <u>DefaultToLN</u>
	AuthTimeout=Specifies the maximum time, in seconds, that the system waits to receive the IBM Connect:Direct control blocks exchanged during the IBM Connect:Direct authentication process.	0–3600 The default is <b>120</b> seconds.
	KeyCertLabel=Identifies the label of the key certificate	keycert label   null
	EncryptData=If no is specified, Encrypt Only Control Block Information; data is sent unencrypted. Default is Yes - data and control block information are encrypted.	y   n   <u>DefaulttoLN</u>
	CipherSuites= Specifies the cipher suites enabled.	comma delimited list of cipher suites   All   null

## Manage the External Authentication ServerRecord

This section contains the commands and parameters used to update and display the .SEAServer record.

### Update the External Authentication Server Record

The **update seaserver** command configures properties for Sterling External Authentication Server (SEAS) in the .SEAServer record that is created at installation. The command has the following parameters:

Command	Parameter	Values
update seaserver	Protocol=Specifies a comma delimited list of Protocols to use in the .SEAServer record.	Disable   TLS1.2,TLS 1.3   <u>DefaultToLN</u> TLS1.0, TLS1.1, SSL3.0 are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.  (See Display Protocols)
	SeaHost=External authentication host name defined in SEAS.	host name   null null—Clears any existing values from the node definition
	AuthTimeout=Specifies the maximum time, in seconds, that the system waits to receive the IBM Connect:Direct control blocks exchanged during the IBM Connect:Direct authentication process.	0–3600 The default is <b>120</b> seconds.
	SeaPort=External authentication server port number (listening) defined in SEAS.	port number   <u>61366</u>
	SeaCacheEnable=Enable caching External Authentication Server certificate validation response.	Y N The default is <b>N</b> .
	SeaCacheValidityTime=Time duration during which the local cache entry is valid for certificates.	The default is <b>24</b> hours. Range: 1 to 720 hours
	SeaGraceValidityTime=Number of hours when the local cache entry of certificate expires and External Authentication Server is unavailable such that Connect:Direct Secure Plus can accept it from its cache.	The default is <b>0</b> hours which means cache grace validity time does not apply. Range: 0 to 720 hours

### Display the External Authentication Server Record

The display SEAServer command displays information about the .SEAServer record.

## Maintain the .Password File (Strong Password Encryption)

This section contains the commands and parameters used to update and display the .Password file through the CLI.

## Update the .Password File

The **Update Password** command enables or disables strong password encryption. The update goes into effect after you start the Connect:Direct server. The command has one parameter, SpeEnable, which can be set to Y or N to enable or disable strong password encryption. Following is an example:

```
Update Password
    SpeEnable=<Y>
;
```

If you enable or disable strong password encryption, the server displays the following warning:

```
The Connect:Direct Server must be restarted to update Strong
Password Encryption.
```

## Display the .Password File

The **Display Password** command displays the strong password encryption setting.

## Maintain Connect:Direct Secure Plus

---

### Connect:Direct Secure Plus Node List

After you set up node records in Connect:Direct Secure Plus, you can view all of the nodes and their attributes from the Secure+ Admin Tool Main Window. From the **Node Name** list, you can view the record of an individual node by double-clicking on a node name.

The following table shows all the fields in the Node Name List:

Field Name	Description	Valid Values
<b>Node Name</b>	Displays the node record name.	.Local remote node name .client
<b>Type</b>	Displays the current record type.	L—Local record R—Remote record
<b>Secure+</b>	Displays the status of	N—Disabled TLS—TLS protocol SSL—SSL protocol TLS1.0, TLS1.1, and SSL3.0 are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2. *—Default to local node
<b>Override</b>	Displays the status of override. Enable override in the local node to allow remote node records to override the settings in the local node record.	Y—Enabled N—Disabled *—Default to local node

Field Name	Description	Valid Values
<b>CipherSuites</b>	Displays the TLS cipher suites that are enabled for the node record.	Varies, based on the cipher suites enabled.
<b>ClientAuth</b>	Displays the status of client authentication. If the TLS protocol is used, enabling client authentication means the SNODE verifies the identity of the PNODE.	Y—enabled N—Disabled *—Default to local node
<b>LimExpr</b>	Identifies if the Limited Export version is being used by a remote node.	Y—Enabled N—Disabled *—Default to local node
<b>AutoUpdate</b>	Indicates if the option to automatically update key values during communication is enabled.	Y—enabled N—disable *—default to local node
<b>Base Record</b>	Displays the name of the base record for the alias records.	

## View Connect:Direct Secure Plus Parameters File Information

### About this task

To view information about the parameters file:

### Procedure

1. Open the Secure+ Admin Tool.
2. Click **File > Info**. The following fields are displayed in the **File Information** dialog box.

Field Name	Description
<b>Current File</b>	The name of the parameters file opened.
<b>Number of Records</b>	The number of nodes defined in the parameters file.
<b>Number of Updates</b>	How many times the parameters file has been updated.
<b>Last 3 Updates</b>	The name of the last three nodes updated.

## View Connect:Direct Secure Plus Node Record Change History

To view the history of changes to a node record:

### Procedure

1. From the Secure+ Admin Tool Main Window, double-click the node record name.
2. Click the **Security Options** tab.

The history of changes is displayed in the **Update History** field.

## Disable Connect:Direct Secure Plus

### About this task

You can use this procedure to disable all nodes in a configuration or one remote node.

### Procedure

1. Do one of the following:
  - To disable all nodes in a configuration, open the local node record.
  - To disable one node, open the remote node record for that node.
2. Click the **Security Options** tab.
3. Click **Disable Secure+**.
4. Click **OK** to update the node record.

**Note:** In order to continue Connect:Direct operations with Connect:Direct Secure Plus disabled, both trading partners must disable Connect:Direct Secure Plus.

## Delete a Connect:Direct Secure Plus Remote Node Record

### About this task

If a remote node record is no longer defined in the network map, you can remove it from the parameters file. The following procedure deletes nodes that are defined in the Connect:Direct Secure Plus parameters file but not in the selected network map.



**CAUTION:** Do not delete the remote node record that is named for the Connect:Direct node. It is the base record for the .Local node record. You cannot delete the .Local node record.

### Procedure

1. From the Secure+ Admin Tool Main Window, click **File > Sync with Netmap**.
2. Click the network map to use from the pulldown list.
3. Click **OK**.
4. Click **Skip** to move through the **Select Netmap Entries** to the **Add** dialog box.
5. To delete node records, do one of the following:
  - To delete selected node records, highlight the remote nodes to delete and click **Delete Selection**.
  - To delete all remote node records not found in the network map, click **Delete All**.

## Resecure Connect:Direct Secure Plus Parameters and Access Files

### About this task

Routinely, or if your access file is compromised, perform the following steps to resecure Connect:Direct Secure Plus:



**CAUTION:** Do not type a new passphrase if an error occurs. If an error occurs while you are resecuring the files, restore the node records from the ACFSave directory. This directory is created after the Rekey Secure+ feature is executed.

### Procedure

1. From the Secure+ Admin Tool Main Window, click **File > Rekey Secure+**. The **Rekey Secure+** dialog box is displayed.



2. Type an alphanumeric string of at least 32 characters in the **Passphrase** field. uses the passphrase to re-encrypt the Connect:Direct Secure Plus parameters and access files. You do not have to remember this passphrase value.
3. Click **OK** to accept the new passphrase. Connect:Direct Secure Plus decrypts and re-encrypts the parameters file and access file.

## View Statistics

### View Statistics

Connect:Direct logs statistics for Connect:Direct Process activity. The Connect:Direct statistics include Connect:Direct Secure Plus information for a Process. Connect:Direct information is included in the Process statistics information only when you attach to a Connect:Direct server.

#### Connect:Direct Requester Select Statistics

Use the Select Statistics function of Connect:Direct Requester to view the information about a Connect:Direct Process, including statistics information about a particular Process. If Connect:Direct Secure Plus is enabled, view Connect:Direct Secure Plus by scrolling to the bottom of the **Statistics Detail** dialog box, in the Session Start Record (SSTR) and Copy Termination Record (CTRC).

For more information on using Connect:Direct Requester to view statistics information, see the Connect:Direct for Microsoft Windows Help or the *IBM Connect:Direct for Microsoft Windows System Guide*.

The Connect:Direct Secure Plus fields and valid values available using the Select Statistics function of the Connect:Direct Requester are shown in the following table:

Field Name	Description	Valid Values
<b>Secure+ Enabled</b>	Specifies whether Connect:Direct Secure Plus is enabled.	Y   N
<b>Secure+ Protocol</b>	Which protocol is enabled.	TLS 1.2   TLS 1.3  TLS1.0, TLS1.1, and SSL3.0 are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.
<b>Cipher Suite</b>	Displays the cipher suite used during a session. cipher suite name, for example:	SSL_RSA_EXPORT_WITH_RC4_40_MD5
<b>PNode Cipher List</b>	Specifies the encryption algorithms available for the PNODE during the session.	
<b>PNode Cipher</b>	Specifies the preferred data encryption as specified in the Connect:Direct Secure Plus parameters file of the PNODE.	Y   N
<b>SNode Cipher List</b>	Specifies the encryption algorithms available for the SNODE during the session as specified in the Connect:Direct Secure Plus parameters file of the SNODE.	

Field Name	Description	Valid Values
<b>SNode Cipher</b>	Specifies the preferred data encryption algorithm as defined in the Connect:Direct Secure Plus parameters file of the SNODE.	Y   N

## Connect:Direct CLI Select Statistics Detail

When you use the CLI Select Statistics function to view the information about a Connect:Direct Process, you see statistics information about a particular Process. The Connect:Direct fields are shown in bold in the following samples.

### Session Start (SSTR) Record

The following sample Session Start Record (SSTR) displays the output of an SSL session:

```

Record Id          => SSTR
Process Name       =>
Process Number    => 0
Submitter Id      =>
Start Time        => 15:23:20
Stop Time         => 15:23:21
SNODE             => JKTIB8100
Completion Code   => 0
Message Id        => LSMI004I
Message Text      => PNODE session started - remote node &NODE
Secure+ Protocol  => SSL 3.0
SSL Cipher Suites => SSL_RSA_WITH_RC4_128_MD5
-----

```

### Copy Termination (CTRC) Record

The Copy Termination Record (CTRC) sample below uses the SSL protocol:

```

Record Id          => CTRC
Process Name       => XX
Process Number    => 195
Submitter Id      => user1
Start Time        => 15:23:47
Stop Time         => 15:26:32
SNODE             => DLAS8100
Completion Code   => 0
Message Id        => SCPA000I
Message Text      => Copy operation successful.
COPY DETAILS: Ckpt=> Y Lkfl=> N Rstr=> N XLat=> N Scmp=> N Ecmp=> N
From node         => S
Src File          => D:\long path
Dest File         => D:\long path
Src CCode         => 0
Src Msgid        => SCPA000I
Bytes Read        => 23592960
Records Read     => 1024
Bytes Sent       => 23791420
RUs Sent         => 30721
Secure+ Protocol  => SSL 3.0
SSL Cipher Suites => SSL_RSA_WITH_RC4_128_MD5
-----

```

## Connect:Direct CLI Select Process Detail

When you use the CLI select process command to view information about a Connect:Direct Process, you see statistics about a Process. If Connect:Direct Secure Plus is not enabled, no Connect:Direct Secure Plus information is displayed:

```

Process Name      => XX                Class           => 32
Process Number   => 197                Priority         => 10
Submitter Node   => DALLAS            PNode           => DALLAS
Submitter        => user1          SNode           => DALLAS
Retain Process   => N
Submit Time      => 15:55:55        ScheduleTime    =>
Submit Date      => 10/19/2009     ScheduleDate    =>
Queue            => EXEC
Process Status   => EX
Message Text     =>
Function         => COPY
Step Name        => TWO
Type             => Send
File Bytes       => 3202560        File Recs       => 0
Xmit Bytes       => 3247926        Xmit Buffers    => 0
Signature Enabled => Y
-----

```

## Audits

### Connect:Direct Secure Plus Parameters File Auditing

The Secure+ Admin Tool and the Connect:Direct Secure Plus Command Line Interface log changes made to the parameters file.

The following events are logged:

- Application Startup
- Init Parmfile
- Open Parmfile
- Sync Netmap
- Rekey Parmfile
- Create Node
- Update Node
- Delete Node

The parameters file logging feature has the following operational characteristics:

- The logging feature is always enabled and cannot be disabled.
- If errors occur when the log is being updated, the application terminates.
- Each log entry contains a timestamp, user ID, and a description of the action/event.
- When an existing node is updated, any changed fields are reported.
- When a node is created or deleted, the values of all non-empty fields are reported.
- Any commands that modify a node are logged.

**Note:** The certificates used by Connect:Direct Secure Plus are individual files that can be stored anywhere on the system. As a result, the logging feature cannot detect when existing certificate files are modified. Connect:Direct Secure Plus only stores the certificate path name and detects changes to this field only.

### Accessing Parameters File Audit Logs

The parameters file audit logs are stored in a dedicated directory, ..\secure+\log. The log file naming convention is SP[YYYY][MM][DD].001 (using local time), and the contents of a log file are limited to a single calendar date. You can view these log files using any text editor. Log files are not deleted by Connect:Direct Secure Plus.

## Parameters File Audit Log Entries

Each audit log has the following header:

```
[YYYYMMDD] [HH:MM:SS:mmm] [userid]
```

When a parameter file is created or opened, an ID is generated that associates the change with the node being updated, as shown in the following:

```
[YYYYMMDD] [HH:MM:SS:mmm] [userid] [ParmFileID]
```

The following fields may appear in a create, update, or delete audit record.

Field Name	Description
Name	Name of the node
BaseRecord	Name of the base record
Type	Record type of local, remote, or alias
Protocol	Enables Connect:Direct Secure Plus protocol
Override	Enables overriding the current node
AuthTimeOut	Authentication timeout
SslTlsTrustedRootCertFile	Pathname to trusted roots file
SslTlsCertFile	Pathname to key certificate file
SslTlsCertPassphrase	Key certificate passphrase (masked)
SslTlsEnableClientAuth	Enable client authentication
SslTlsCertCommonName	Common name of the remote certificate to verify
SslTlsEnableCipher	List of SSL/TLS cipher suites
SslTlsSeaEnable	Enable external authentication
SslTlsSeaCacheEnable	Enable caching External Authentication Server certificate validation response.
SeaCacheValidityTime	Time duration during which the local cache entry is valid for certificates
SeaGraceValidityTime	Number of hours when the local cache entry of certificate expires and External Authentication Server is unavailable such that Connect:Direct Secure Plus can accept it from its cache.
SeaCertValDef	External authentication validation definition
SeaHost	External authentication host name
SeaPort	External Authentication port number

## Parameters File Audit Log Error Reporting

Errors are reported for the following logging functions: open log, write log, and lock log. If an error occurs during one of these functions, an error message is displayed and the application is terminated. The lock function times out after 30 seconds. Typically, Secure+ Admin Tool or the CLI hold the lock for less than one second per update.

## Connect:Direct Secure Plus Certificate Auditing

In a TLS session, audit information about the identity certificate and its signing certificate is logged in the statistics log in the Session Start (SSTR) and Copy Termination (CTRC) records. The audit information is included in the response data from a Select Statistics command in the SSTR and CTRC records.

In an TLS session, the PNODE (client) always logs the audit information. The SNODE (server) only logs the information when client authentication is enabled. For logging to occur, the session handshake must succeed and progress to the point of logging the SSTR and CTRC records.

### Certificate Audit Log Entries

The audit consists of the subject name and serial number of the identity and its signing certificate. The identity certificate also contains an issuer attribute, which is identical to the signing certificate subject name. Although many signing certificates may exist between the identity and final root certificate, the audit includes only the last two certificates in a chain: an intermediate certificate and an end certificate.

In the SSTR and CTRC records, the CERT contains the common name and serial number of the key certificate, and the CERI contains the common name of the issuer and the serial number of an intermediate or root CA. They may also contain the certificate serial number, for example:

```
CERT=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/CN=Test ID/SN=99c0ce01382e6c83) |
CERI=(/C=US/ST=MA/L=Marshfield/O=test.org/CN=root CA/SN=da870666bbfb5538)
```

Connect:Direct Secure Plus certificate audits may contain the following fields:

Field Name	Abbreviation	Max Lengths (RFC 2459)
Common Name	CN	64
Country	C	2
Locality	L	128
State	ST	128
Organization	O	64
Organization Unit	OU	64
Email Address	emailAddress	128
Serial Number	SN	128 (estimated)

### Accessing Certificate Audit Logs

Certificate audit information located in the SSTR and CTRC records cannot be accessed directly using Connect:Direct Requester or Connect:Direct Browser User Interface. To access certificate information, you can issue a query directly to the database or use an SDK-based or JAI-based program to issue a Select Statistics command. The response to the Select Statistics command contains the AuditInfo field of the statistics records, including the SSTR and CTRC records. This field contains certificate audit information.

The following example was generated using a database query.

```
'2009-05-21 14:50:27', 2, 'SSTR', 'CAEV', '', 0, '2009-05-21 14:50:26', '2009-05-21
14:50:27', '', '', 'JLYON-XP.4500', 0,
```

```
'MSGI=LSMI004I|SBST=(&NODE=JLYON-XP.4500)|PNOD=JLYON-XP.4500|CSPE=Y|CSPP=TLsv1|CSPS=
TLS_RSA_WITH_AES_256_CBC_SHA|
CERT=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/
CN=Example Test ID/SN=a9febbeb4f59d446)|
CERI=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/CN=Example
IntermediateCA/SN=a69634a8a7830268)|STSD=2|TZDI=-14400|
'2009-05-21 14:50:28', 2, 'CTRC', 'CAPR', 'SAMPLE', 1, '2009-05-21 14:50:27',
'2009-05-21 14:50:28', 'JLYON-XP.4500', 'jlyon', 'JLYON-XP.4500', 0,
'MSGI=SCPA000I|LCCD=0|MSG=SCPA000I|OCCD=0|MSG=SCPA000I|PNAM=SAMPLE|PNUM=1|SNAM=STE
P1|SBND=JLYON-XP.4500|SBID=jlyon|PNOD=JLYON-XP.4500|SNOD=JLYON-XP.4500|LNOD=P|FROM=P
XLAT=N|ECZI=N|ECMP=N|SCMP=N|OERR=N|CKPT=Y|LKFL=N|RSTR=N|RUSZ=65535|PACC=|SACC=|PPMN
=|SFIL=C:\Program Files\IBM\Connect Direct
v4.6.00\Server\Process\Sample.html|SDS1=|SDS2=|SDS3=
|SFSZ=0|SBYR=861|SRCR=1|SBYX=863|SRUX=1|SNVL=-1|SVOL=|DFIL=C:\Program Files\IBM\
Connect Direct v4.5.00\Server\Process\Verify.html|PPMN=|DDS1=R|DDS2=|
DDS3=
|DBYW=861|DRCW=1|DBYX=863|DRUX=1|DNVL=0|DVOL=|CSPE=Y|CSPP=TLsv1|CSPS=TLS_RSA_WITH_AE
S_256_CBC_SHA|CERT=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/CN=Example Test
ID/SN=a9febbeb4f59d446)|CERI=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/
CN=Example Intermediate CA/SN=a69634a8a7830268)
|PCRC=N|ETMC=60|ETMK=10|ETMU=0|STSD=2|TZDI=-14400|'
```

## Certificate Audit Log Error Reporting

If an error occurs when the subject name is extracted from the identity (CERT) or issuer's (CERI) certificates, the following message ID is logged:

```
CERT=(MSGI=CSPA310E)|CERI=(MSGI=CSPA310E)
```

Only the message ID is displayed with the CERT or CERI tokens; the standard Connect:Direct error function is not used. After the error occurs, the session continues.

## Troubleshoot Connect:Direct Secure Plus

### Troubleshooting

Use the following table to help troubleshoot problems with Connect:Direct Secure Plus.

Problem	Possible Cause	Solution
Connect:Direct Secure Plus features are enabled in the parameters file, but the statistics record indicates that the functions are disabled.	The Connect:Direct network maps do not contain entries for the PNODE and SNODE.	Verify that the netmap entries for both the PNODE and the SNODE exist.
Running a Process with a remote node fails with an authentication error.	Unique public/private key pairs are generated for the remote node record and the .Local node record is set to Enable Override=N.	Change the .Local node record to Enable Override=Y.
The Connect:Direct Secure Plus parameter, ENCRYPT.DATA, specified from the COPY statement causes the copy step to fail with error message CSPA080E.	The algorithm name used in the COPY statement is not in the supported algorithm list for both nodes.	Verify that the algorithm name in the COPY statement is in the supported algorithm list for both nodes.

<b>Problem</b>	<b>Possible Cause</b>	<b>Solution</b>
Connect:Direct Secure Plus is installed, but error message CSPA001E occurs on non-Connect:Direct Secure Plus transfers.	Remote node records do not exist.	<ul style="list-style-type: none"> <li>• A remote node record must exist for every node in the netmap. Use the Sync with Netmap feature to create any missing nodes.</li> <li>• Disable Connect:Direct Secure Plus by clicking Disable Secure + in the .Local node record.</li> </ul>
Signature verification fails with error message CSPA002E.	Configuration settings missing or incorrect.	<ul style="list-style-type: none"> <li>• If this is a non-secure node, make sure the remote node record has Disable Secure+ selected.</li> <li>• Check the Connect:Direct Secure Plus settings for the node.</li> </ul>
Strong authentication fails with the error, CSPA010E.	<ul style="list-style-type: none"> <li>• The time allowed for strong authentication expired.</li> <li>• A security attack in progress.</li> </ul>	<ul style="list-style-type: none"> <li>• Increase the timeout value.</li> <li>• Execute standard operating procedure for investigating security violation.</li> </ul>
Connect:Direct Secure Plus session fails with the error, CSPA011E.	An illegal attempt to override Connect:Direct Secure Plus parameters.	<ul style="list-style-type: none"> <li>• Turn on Enable Override in the remote node record to allow the COPY statement to override the node settings.</li> <li>• Check the COPY statement and remove the override statements.</li> </ul>
Connect:Direct Secure Plus session fails with the error, CSPA014E.	Connect:Direct Secure Plus cannot read the remote node definition.	Check the remote node definition settings.
Connect:Direct Secure Plus session fails with the error, CSPA016E.	Connect:Direct Secure Plus is not enabled in the local node definition.	Make sure Connect:Direct Secure Plus is enabled for the local node.
Connect:Direct Secure Plus session fails with the error, CSPA019E.	Error generating digital signature.	<ul style="list-style-type: none"> <li>• Resubmit the Process.</li> <li>• Call IBM Support.</li> </ul>
Connect:Direct Secure Plus session fails with the error, CSPA077E.	The COPY statement requested Connect:Direct Secure Plus parameters but Connect:Direct Secure Plus is not configured.	Remove the SECURE= parameter from the COPY statement.
Connect:Direct Secure Plus session fails with the error, CSPA079E.	Invalid encryption algorithm identified in COPY statement.	Change the ENC.DATA parameter and specify one of the following values: Y, N, IDEACBC128, TDESCBC112, or DESCBC56 and resubmit the Process.

<b>Problem</b>	<b>Possible Cause</b>	<b>Solution</b>
Connect:Direct Secure Plus session fails with the error, CSPA080E.	No common algorithms are available for both nodes.	Verify the algorithm list for both nodes contains at least one common algorithm name.
Connect:Direct Secure Plus session fails with the error, CSPA091E.	Session attempted but remote node is not configured.	Make sure both nodes are defined for the remote node record.
Connect:Direct Secure Plus session fails with the error, CSPA200E.	Both nodes are not configured for the same protocol.	<ul style="list-style-type: none"> <li>• Check the protocol setting at both sites and verify that the same protocol is configured at each site.</li> <li>• If necessary, edit the remote node record.</li> </ul>
Connect:Direct Secure Plus session fails with the error, CSPA202E.	TLS protocol handshake failed.	Edit the cipher suite list and add a cipher suite used by the trading partner.
Connect:Direct Secure Plus session fails with the error, CSPA203E or CSPA204E.	The TLS protocol could not validate the server's certificate.	Make sure the certificate information is typed into the node record.
Connect:Direct Secure Plus session fails with the error, CSPA205E.	A trading partner is not using TCP/IP for communication.	Make sure that both ends of the communication use TCP/IP.
Connect:Direct Secure Plus session fails with the error, CSPA206E.	The TLS protocol could not validate the server's certificate.	Make sure the certificate information is entered into the node record.
Connect:Direct Secure Plus session fails with the error, CSPA208E.	The common name in the certificate received does not match the Connect:Direct Secure Plus configuration.	Make sure the certificate common name is spelled correctly and uses the same case as that in the certificate.
Connect:Direct Secure Plus session fails with the error, CSPA209E.	The certificate has expired or is invalid.	Obtain a new certificate and reconfigure the node record.
Connect:Direct Secure Plus session fails with the error, CSPA211E.	The remote trading partner failed to send a certificate.	Notify the trading partner that a certificate is required.
Connect:Direct Secure Plus session fails with the error, CSPA280E.	The trusted root certificate could not be loaded.	Check the local node configuration and make sure the location of the trusted root certificate is correctly identified.
Connect:Direct Secure Plus session fails with the error, CSPA281E.	The trusted root certificate is empty.	Check the local node configuration and make sure the location of the trusted root certificate is correctly identified.
Connect:Direct Secure Plus session fails with the error, CSPA282E.	The user certificate file cannot be loaded.	Check the local node configuration and make sure the location of the user certificate file is correctly identified.



Problem	Possible Cause	Solution
Connect:Direct Secure Plus session fails with the error, CSPA303E.	The parameters files have not been initialized.	Run the Admin Tool to initialize the parameters files.
Connect:Direct Secure Plus session fails with the error, CSPA309E.	The SSL library failed during the handshake.	Examine all related errors to determine the cause of the failure.
Connect:Direct Secure Plus session fails with the error, CSPA311E.	Certificate validation failed.	Verify that the root certificate is properly configured. An alternate certificate may be required.

## Configuration Worksheets

---

### Local Node Security Feature Definition Worksheet

Record the security definitions for the local Connect:Direct node.

Local Node Name	
Configured Security Functions <ul style="list-style-type: none"> <li>• Enable TLS protocol (Yes   No)</li> <li>• Enable SSL protocol (Yes   No)</li> <li>• Authorization timeout</li> <li>• Key store location. The default is ...\\Secure+\\certificates\\cdkeystore.kdb.</li> <li>• Certificate label</li> <li>• Certificate passphrase</li> <li>• Cipher suite(s) enabled</li> </ul> TLS1.0, TLS1.1, and SSL3.0 are deprecated and should not be used. It is recommended that trading partners using deprecated protocols migrate to TLS 1.3 or TLS 1.2.	
External Authentication <ul style="list-style-type: none"> <li>• Enable external authentication (Yes   No)</li> <li>• Certificate validation definition</li> </ul>	

### Remote Node Security Feature Definition Worksheet

Make a copy of this worksheet for each remote node defined in the parameters file that you are configuring for Connect:Direct Secure Plus operations. Record the security feature definitions for a remote node record on this worksheet.

Remote Node Name	
------------------	--

<p>Security Options</p> <ul style="list-style-type: none"> <li>• Protocol defined in the .Local node record (TLS   SSL) <ul style="list-style-type: none"> <li>SSL protocol is deprecated but supported</li> </ul> </li> <li>• Is the remote node using the protocol defined in the .Local node record? ( Y   N) <ul style="list-style-type: none"> <li>– If you answered No to the question, identify the protocol to use for the remote node (TLS   SSL)</li> <li>– Enable TLS protocol (Y   N)</li> <li>– Enable SSL protocol (Y   N)</li> <li>– To use the same protocol defined in the local node, select Default to Local Node.</li> <li>– Enable override (Y   N)</li> <li>– Authorization timeout</li> </ul> </li> </ul>	
<p>TLS Protocol Functions</p> <ul style="list-style-type: none"> <li>• Key store location. The default is ...\\Secure+\\certificates\\cdkeystore.kdb.</li> <li>• Certificate label</li> <li>• Certificate passphrase</li> <li>• Cipher suite(s) enabled</li> <li>• Enable client authentication (Y   N   Default to local node)</li> <li>• Certificate common name</li> </ul> <p><b>Note:</b> If you want to add a second type of security, enable client authentication for the remote node. A third type of security that you can enable is certificate common name validation.</p>	
<p>External Authentication</p> <ul style="list-style-type: none"> <li>• Enable external authentication (Y   N   Default to local node)</li> <li>• Certificate validation definition</li> </ul>	

## Certificate File Layout

### Certificate File Layout

The TLS security protocols use a secure server RSA X.509V3 certificate to authenticate your site to any client that accesses the server and provides a way for the client to initiate a secure session. When you obtain a certificate from a certificate authority or create a self-signed certificate, it is stored in a key store.

When you obtain a key certificate from a certificate authority, you have to add it to a local key store file. To configure Connect:Direct Secure Plus, you have to import a key certificate from the key store. Add the certificate label and common name to the node record using the Secure Plus Admin Tool.

Use the IBM Key Management tool to add or delete certificate information in the key store. In simple configurations, only one key store is used, but the key store can contain multiple key certificates. The key store might also contain multiple trusted root and intermediate certificates. Each certificate has a unique label to differentiate them from one another. In more sophisticated configurations, you can associate individual key certificate labels with one or more node records.

When you use a certificate signing request (CSR) tool, such as iKeyman, you do not need to change the contents of the key certificate. This is created for you by iKeyman.

## Certificate Format

A certificate is encoded as a general object with the identifier string CERTIFICATE or X.509 CERTIFICATE. The base64 data encodes a Bit Error Rate (BER)-encoded X.509 certificate. This is the same format used for PEM. Anyone who provides or understands PEM-format certificates can accommodate the certificate format. For example, VeriSign commonly fulfills certificate requests with certificates in this format, SSLeay supports them, and SSL servers understand them. Most browsers support this format for importing root CA certificates.

## Validate the Configuration

---

### About this task

Perform this procedure to ensure that the nodes have been properly configured. The validation process checks each node to ensure that all necessary options have been defined and keys have been exchanged.

To validate the parameters file:

### Procedure

1. In the Secure+ Admin Tool, click **File > Validate Secure+**. The **Secure+ Admin Tool - Validation Results** window is displayed.

If the parameters file is not correctly configured, warning and error messages are displayed.

2. Read each warning message. To correct each warning or error reported, go back to the parameters file and make changes as needed.

**Note:** Warning messages do not always mean that the parameters file is incorrectly configured. Some warning messages are informational only.

3. Click **Close** to close the **Validation Results** window.

## Exchange Data and Verify Results

---

To exchange data and verify the results, submit the sample Process that is provided with Connect:Direct.

To verify the success of the sample Process and review the Connect:Direct Secure Plus statistics for the session, refer to *Connect:Direct for Microsoft Windows Getting Started Guide*.

## Automation Scripts

---

### Configure Connect:Direct Secure Plus to Use the TLS Protocol

The spcust\_sample1 script demonstrates using the CLI to import certificates to configure Connect:Direct Secure Plus and TLS protocol.

```
@echo off
REM
REM spcust_sample1.sh contains an example of configuring
REM Secure+ to use SSL or TLS protocols with the Secure+ CLI.
REM The example demonstrates the configuration of Secure+
REM with the trusted root and key certificates and ciphers
REM
REM
REM Variables
REM
REM The return code.
REM spcli.sh returns the highest return code of the commands
```

```

REM it executed. Possible return codes and their meanings are
REM     0  success
REM     4  warning
REM     8  error
REM    16  fatal error

set cdInstallDir=C:\Program Files\IBM\Connect Direct v6.1
set spDir=%cdInstallDir%\Server\Secure+

pushd "%spDir%"

REM
REM Main script
REM

echo.
echo This script has been prevented from running because it will alter
echo The configuration of Secure+. Before removing this warning and its
echo exit call, please modify the script so that it carries out only
echo desired modifications to the configuration of Secure+.
echo.
goto :EOFc

all :initCustom

call :invokeCLI

call :terminateCustom

REM End of main script
goto :EOF

REM
REM Functions
REM

```

```

REM
REM Custom initialization logic written by customer.
REM

:initCustom

REM Customer adds custom initialization code here.

echo Init custom...
echo.

REM del /F "%spDir%\Nodes"

REM End of initCustom
goto :EOF

REM
REM Invoke CLI to configure Secure+.
REM

:invokeCLI
set tempFile=clicmds.txt

echo ; >>%tempFile%
echo display info >>%tempFile%
echo ; >>%tempFile%
echo ; >>%tempFile%
echo ; -- Synch with netmap >>%tempFile%
echo ; >>%tempFile%
echo sync netmap >>%tempFile%
echo path=v6.1\JLYON-LT >>%tempFile%
echo name=* >>%tempFile%
echo ; >>%tempFile%
echo ; >>%tempFile%
echo ; -- Import keycert and trusted cert files. >>%tempFile%
echo ; >>%tempFile%
echo import keycert >>%tempFile%
echo file="%spDir%\Certificates\keycert.txt" >>%tempFile%
echo passphrase=password >>%tempFile%
echo Label="My KeyCert"; >>%tempFile%

```

```

echo ; >>%tempFile%
echo import trustedcert >>%tempFile%
echo file="%spDir%\Certificates\trusted.txt" >>%tempFile%
echo ; >>%tempFile%
echo update localnode >>%tempFile%
echo override=n >>%tempFile%
echo protocol=(tls1.2,tls1.3) >>%tempFile%
echo securitymode=FIPS140-2 >>%tempFile%
echokeycertlabel="My KeyCert" >>%tempFile%
echociphersuites=(TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_GCM_SHA256)
>>%tempFile%
echo ; >>%tempFile%
echo ; >>%tempFile%
echo ; -- Display localnode >>%tempFile%
echo ; >>%tempFile%
echo display localnode >>%tempFile%
echo ; >>%tempFile%
echo validate parmfile >>%tempFile%
echo ; >>%tempFile%
echo quit; >>%tempFile%

```

```

call "%spDir%\spcli.cmd" -e 8 -li y < %tempFile%
set RC=%ERRORLEVEL%
del %tempFile%

REM End of invokeCLI
goto :EOF

REM
REM Custom termination logic written by customer.
REM
:terminateCustom

REM Customer adds custom termination code here.
REM For example, E-mail standard out log for review.
REM Send error messages to system monitoring facility.
echo.
echo Custom Terminating with errorlevel of %RC%
echo.REM End of terminateCustom
goto :EOF

popd

```

## Use LCU to Configure Encrypted Passwords

### Configure Encrypted Passwords Using the LCU

The Connect:Direct Secure Plus CLI displays passwords in plain text. If you need to encrypt passwords for use with the Connect:Direct Secure Plus CLI, use the Local Connection Utility (LCU) to create an LCU file that contains non-encrypted information used to encrypt the password and the encrypted password, such as a keycert passphrase. You can then refer to this file when prompted for passwords.

#### LCU Files

The following example shows how to specify when an LCU file is used in place of a plain-text password:

```

C:\...\...\Connect Direct v4.6.00\Common Utilities>lcu -f C:\SomeDir\MyLCU.dat
*****
*          Connect:Direct Java Client Connection Utility          *
*                               Version 4.6.00 *
*-----*
* Copyright (c) 1983, 2011 *
* All Rights Reserved. *
*****

```

```

Node:
>JLYON-XP.4600
API Address: <Enter> = 'JLYON-XP'
>
API Port: <Enter> = '1363'
>
User Name:
>SomeValue
Password:
> Confirm Password:
>
Saving file: C:\SomeDir\MyLCU.dat
C:\...\...\Connect Direct v4.6.00\Server\Secure+>SPCLI
...
SPCLI> Create STSKeyPair
KeyPairFile=C:\SomeDir\StsKeyPairFile.dat
Passphrase=LCU:C:\SomeDir\MyLCU.dat;
SPCG670I rc=0 Create stskeypair command successful.
SPCLI> Update RemoteNode
Name=JLYON-XP.4600
StsAuthLocalKey=set
StsAuthKeyPairFile=C:\SomeDir\StsKeyPairFile.dat
StsAuthKeyPairFilePassphrase=LCU:C:\SomeDir\MyLCU.dat
SPCG470I rc=0 Update remote node "JLYON-XP.4600" command successful.

```

The use of the LCU syntax “LCU:” indicates that what follows is an LCU filename and not a passphrase. The pathname of the LCU file can be a relative path, a relative path to the bin directory, or a full path. If LCU:filename contains spaces, it must be enclosed in quotation marks: “LCU:filename”. The default name of the LCU file is cddef.bin. After the cddef.bin file is created, you can rename it as needed.

LCU files can be used to provide encrypted passwords for the following commands and parameters:

Command	Parameter
Update LocalNode	StsAuthKeyPairFilePassphrase StsSigKeyPairFilePassphrase SslTlsCertPassphrase
Create RemoteNode	StsAuthKeyPairFilePassphrase StsSigKeyPairFilePassphrase SslTlsCertPassphrase
Update RemoteNode	StsAuthKeyPairFilePassphrase StsSigKeyPairFilePassphrase SslTlsCertPassphrase
Create STSKeyPair	Passphrase
Update Client	SslTlsCertPassphrase
Update SEAServer	SslTlsCertPassphrase

## Create an LCU File

### About this task

To create an LCU file:

## Procedure

1. Type the following command to run the LCU utility:

```
lcu.bat
```

2. As you are prompted, enter values for the following parameters:
  - Node
  - API Address
  - API Port
  - User Name
  - Password
  - Confirm Password
3. The cddef.bin file is created.





---

# Chapter 6. SDK Programmers Guide

## Overview

---

### Connect:Direct for Microsoft Windows SDK Overview

Use the IBM Connect:Direct for Microsoft Windows Software Development Kit (SDK) to extend an application to include the automated file transfer capabilities of Connect:Direct for Microsoft Windows. SDK uses a 64-bit interface for C and C++ as well as an OLE automation server for Visual Basic applications. SDK also provides ActiveX controls for Submit Process and Select Statistics commands.

- C API functions—Standard and registry API functions. The standard functions allow you to connect to a Connect:Direct node, execute Connect:Direct commands, manage command response data, and retrieve error information. The Registry API functions store and retrieve client connection information to and from the Registry. The C API is implemented using the C++ Classes.
- C++ Class interface—Provides the foundation for the other Connect:Direct interfaces and provides Visual C++ programmers an object-oriented interface to Connect:Direct.
- ActiveX control interface—Uses the CDSubmit and CDStatistics functions to submit Processes to the server and display statistics from the statistics database.
- Direct Automation Servers—Provides an automation wrapper around the Connect:Direct SDK C++ classes. They provide direct automation support for languages like Visual Basic. The Connect:Direct Automation Servers provide the following primary classes that map directly to the CDNode, CDProcess, and CDStatistics classes in the SDK C++ classes:
- User exits—Provides a way to customize Connect:Direct operations. User exits are user-defined dynamic link libraries (DLLs) that are loaded and called when the user exit is enabled through an initialization parameter. Three user exits are provided: one for enhanced security, one for automated file opening and password exit.

Before you can use the SDK tools, you can run the Client Connection Utility to configure server access information, such as TCP/IP information. Alternatively, you can let your SDK application specify the access information. Some SDK languages also support the Logon Configuration Utility (LCU files).

### Distribute an Application

The following SDK files are required to be included when distributing an application developed with this SDK.

- For C++ applications:
  - CdCore.dll
- For C applications:
  - CdCore.dll
  - CdCapi.dll ("C" wrapper for cdcore.dll)
- For VB - Automation Server
  - CdCore.dll
  - CDAuto.dll
  - CdAuto.tbl
- For VB - Active X
  - CdCore.dll
  - CDStats.ocx

- CDSubmit.ocx

DLL files are loaded by using the following algorithm:

1. The directory containing the .exe that is loading the .dll
2. The current directory
3. The system directory (system32)
4. The Microsoft Windows directory
5. The directories list in the PATH environment variable.

Also, the OCX files must be registered in the following manner:

- regsvr32 "C:\Program Files\IBM\Connect Direct v6.0.0\SDK\CDSubmit.ocx"
- regsvr32 "C:\Program Files\IBM\Connect Direct v6.0.0\SDK\CDStats.ocx"

Or you may use the "/s" option to do so without bringing up a dialog box:

- regsvr32 /s "C:\Program Files\IBM\Connect Direct v6.0.0\SDK\CDSubmit.ocx"
- regsvr32 /s "C:\Program Files\IBM\Connect Direct v6.0.0\SDK\CDStats.ocx"

In addition, when using the automation server, you also need to register CDAuto.dll. For example:

```
regsvr32 "C:\Program Files\IBM\Connect Direct v6.0.0\SDK\CDAuto.dll"
```

If you are using the automation server, you must also register your Type Library files (.TLB) using regtlib.exe. Regtlib.exe is distributed with Visual Studio 6 and above and has updates available in the service packs or in other Microsoft Windows Library updates.

**Note:** CDCoreD.dll and CDCapiD.dll are debug versions and do not need to be distributed with the application.

Applications may also require the Microsoft Visual Studio Redistributable Runtimes. Not every system has this installed by default.

For checking about required DLLs, Microsoft's Dependency Walker (depends.exe) is the tool to use. It lists in detail all DLLs required by an application. The tool is included in the Resource Kit, Microsoft Windows 2000 Support Tools, Visual Studio and other packages.

## Edit Connection Settings

---

### Edit Connection Settings with the Client Connection Utility

To use the SDK to create your own programs, you must create connection settings for each user.

Two methods are available to create local node definitions. You can use either Connect:Direct Requester or the Client Connection Utility. If you want to use Connect:Direct Requester, refer to the *IBM Connect:Direct for Microsoft Windows System Guide* for instructions.

The Connect:Direct for Microsoft 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 Registry settings for Connect:Direct API connections, rather than editing them directly.

You can view, edit, and update Connect:Direct for Microsoft Windows connection settings in the Windows Registry with the Client Connection Utility. The connection settings enable communication between the user interfaces and the Connect:Direct server. You can set up and update connection settings by:

- Adding a node
- Deleting a node
- Adding a user

- Deleting a user
- Updating node properties
- Defining 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.

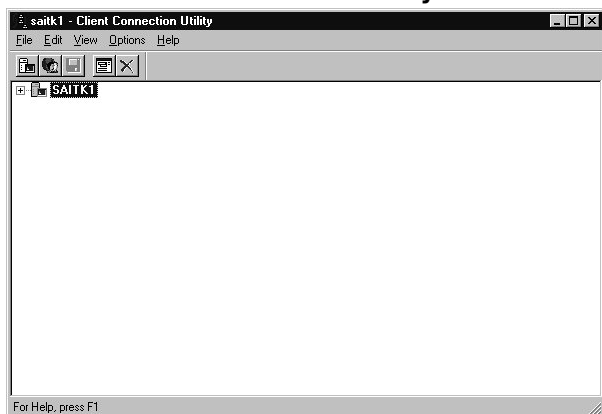
## Start the Client Connection Utility

### About this task

To start the Client Connection Utility:

### Procedure

1. Click **Start > All Programs > IBM Connect:Direct > v6.1**.
2. Select **CD Client Connection Utility**. The Client Connection Utility main window is displayed.



## Add and Delete Node Connection Definitions

Use the Client Connection Utility to add new nodes, look at node properties, and delete existing nodes.

The Connect:Direct Client Connection Utility enables you to add new nodes and identify their properties, 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.

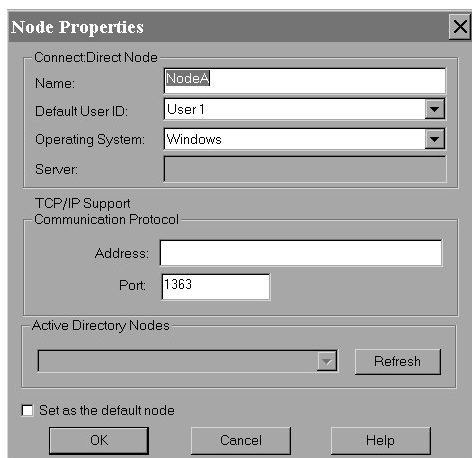
## Add a Node

### About this task

To add a Connect:Direct node:

### Procedure

1. Select **File > New Node**. The Node Properties dialog box displays:



2. To add a node that is registered in the Active Directory:
  - a) In **Operating System**, select **Windows**.
  - b) Select the node to add from **Active Directory Nodes**.  
The name, address, and port fields are automatically updated with information from the Active Directory list.
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 value in **Operating System**.
  - c) In **Address**, 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 **Set as the Default Node**.
5. Click **OK** to save your settings and close **Node Properties**.
6. Select **File > Save** to save the new settings.

**Note:** Changes made to node settings are not written to the Registry until you select **Save**.

## Delete a Node

### About this task

To delete a Connect:Direct node:

### Procedure

1. In the Client Connection Utility main window, select the node you want to delete.
2. Select **Edit > Delete**.
3. Click **Yes** to confirm the deletion.
4. Select **File > Save** 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.

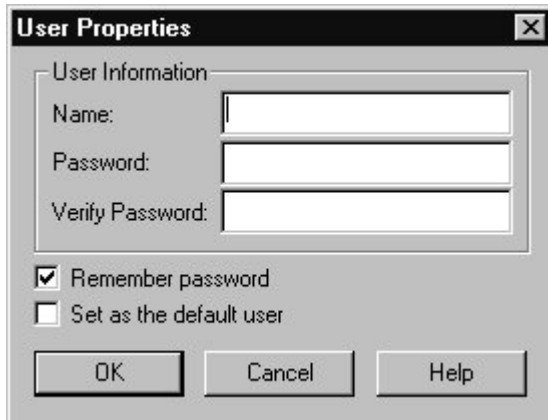
## Add a User

### About this task

To add a new Connect:Direct user:

## Procedure

1. In the Client Connection Utility main window, select the node where you want to add a new user.
2. Select **File > New User** to display the User Properties dialog box.



3. Type information into the following fields:
  - **Name**—type the name of the new user. Either type the user name as defined in the Microsoft Windows setup, such as lmore, or type a fully qualified user name in the UPN format, such as testuser@xxxxx.com
  - **Password**— type the password defined for the user.
  - **Verify Password**—retype the password defined for the user.
4. Click **Remember Password** to automatically reload the password when you attach as this user.
5. Click **Set as the Default User** if you want the new user to be the default user for the node.
6. Click **OK** to save the settings and close User Properties.
7. If the verification password you typed does not match the initial password, you receive a message indicating that the passwords do not match. Retype the verification password and click OK.
8. **Select File > Save** to save the settings.

**Note:** Changes made to node settings are not written to the Registry until you select **Save**.

## Delete a User

### Procedure

1. If the user names are not displayed, click the plus (+) sign next to the node containing the user you want to delete.
2. Select the user you want to delete.
3. Select **Edit > Delete**.
4. Click **Yes** to confirm the deletion.
5. Select **File > Save** to save the new configuration.

**Note:** Changes made to node settings are not written to the Registry until you select **Save**.

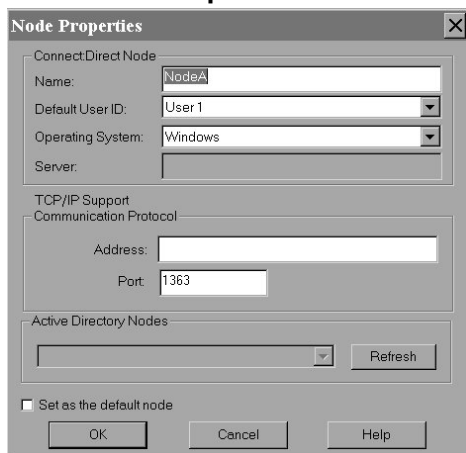
## Update Node Properties

### About this task

To update node and user properties:

## Procedure

1. Do one of the following:
  - To update a node, highlight the node you want to configure.
  - To update user properties, highlight the user you want to configure.
2. Select **File > Properties**.



3. Make the appropriate changes.
4. Click **OK** to save your settings and return to Node Properties.
5. Select **File > Save** to save the settings.

**Note:** Changes made to node settings are not written to the Registry until you select **Save**.

## Define a Default Node or Default User

### About this task

To define a default node or default user:

### Procedure

1. Take one of the following actions:
  - To define a default node, highlight the node.
  - To define a default user, highlight the user.
2. Select **Options > Set as Default** to set the default node or user.
3. Select **File > Save** to save the settings. The default node or user is displayed in the main Client Connection Utility window as bold text.

**Note:** Changes made to node settings are not written to the Registry until you select **Save**.

## Import Registry Settings

### About this task

To import registry settings from a file:

### Procedure

1. Select the node in which to import the Registry settings.
2. Select **File > Import**. A message displays informing you that all settings will be lost.

3. Click **Yes**. The Open dialog box displays.

**Note:** Importing a Registry settings file causes all current changes to the selected node to be lost if they have not been saved.

4. 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.

5. Select **File > Save** to save the settings.

**Note:** Changes made to node settings are not written to the Registry until you select **Save**.

## Export Registry Settings

### About this task

To export Registry settings to a file:

### Procedure

1. From the Client Connection Utility main window, select the node containing the Registry settings you want to export.
2. Click **File > Export**.
3. Name the exported Registry file with a REX extension and click **OK**. The Registry settings in the file can now be imported into another computer or node.

## Print Registry Settings Report

### About this task

To generate and print the registry settings report:

### Procedure

1. To preview the Registry settings report before printing it:
  - a) Select **File > Print Preview**.
  - b) Click **Zoom In** to enlarge the text and read the report.
2. To print the report:
  - a) Select **File > Print**.
  - b) If necessary, select a printer.
  - c) Click **OK**. A report of all Registry settings is generated.

**Note:** Additional node detail is provided if the node has been used at least once by the client software.

## Apply the C API

---

### The C Applications Programming Interface

The Connect:Direct C applications programming interface consists of Standard and Registry API functions. The Standard API functions connect to a Connect:Direct node, execute Connect:Direct commands, manage command response data, and retrieve error information. The Registry API functions store and retrieve client connection information to and from the Registry. The C API is implemented using the C++ Classes. This interface is used by C programmers.

## Compile and Debug

When you are ready to compile the program created with the API, include the CDCAPI.H header file. Including the CDCAPI.H file in your project automatically links a program with the appropriate import library. Debug configurations link with the CDCAPID.LIB and release configurations link with the CDCAPI.LIB.

The CDCAPI.LIB and CDCAPID.LIB files contain the following information:

- Name of the DLL to dynamically load at run time.
- Definitions of all exported functions. This is used by the linker to resolve all calls to the CDCAPI.DLL.

When the program runs or the DLL is loaded, the appropriate CDCAPI.DLL is loaded. The CDCAPI.DLL is dynamically loaded when a release configuration is executed, and the CDCAPID.DLL is dynamically loaded to support debug configurations.

The C APIs are based on the core C++ APIs. This required API layer is contained in CDCORE.DLL (or CDCORED.DLL if compiling for debug mode). The appropriate core DLL must be in your path for the C APIs to work properly.

## Activate Tracing

The Output window of the Microsoft Visual Studio displays trace messages.

The following table describes the tracing parameters. Use the trace parameters to activate tracing.

Parameter	Description
CdGetTraceFlags(unsigned int* pgrfTrace);	Retrieves the current trace settings for the Connect:Direct API.
CdSetTraceFlags(unsigned int grfTrace);	Sets new trace settings for the Connect:Direct API.
CdSetTraceFile(LPCTSTR pszFilename);	Provides a file name to the tracing facility. If a file is defined, trace messages are written to the Output window and specified file.

## Standard C API

### Overview

Use the Standard API functions to connect to a Connect:Direct node, execute Connect:Direct commands, manage command response data, and retrieve error information.

The C API is implemented using the C++ Classes. This interface is used by C programmers.

### Handles

Handles simplify object and memory management by referencing a particular object. Pass a handle to an API to uniquely identify an object. The Connect:Direct C API uses the following types of object handles to return node, Process, statistics, message, and trace information:

- Node Handles—Represent the Connect:Direct node that is the target of the operation. It is a virtual connection to a Connect:Direct node. The node handle is a special type of object handle; it holds information about the node but does not return data from the node.

A node handle is created by calling the CdConnect() function and passing it the node name, user ID, password, and protocol within a NODE\_STRUCT structure. After you finish with a node handle, you call the CdCloseHandle() to close it. Closing the handle releases the virtual connection and any internal resources associated with it. The node handle is no longer valid on subsequent operations.



**Note:** You are responsible for closing the node handle and for releasing any resources that you allocate.

- **Process Handles**—Handles returned from a submit command or from a Process object, which is created when a select process, change process, or delete process command is executed. The following example demonstrates the select process command returning a Process:

```
if (CdExecuteCommand (hNode, "SELECT PROCESS", &hProc))
{
    if (CdGetProcRec(hProc, &Proc))
    {
        printf("%d %s/n", Proc.ProcessNumber, Proc.ProcessName);
    }
}
```

- **Statistic Handles**—Statistics objects that are returned after a select statistics command is executed.
- **Message Handles**—Message objects that are returned when a select message command is executed.
- **Trace Handles**—Trace objects that are returned when a traceon or traceoff command is executed.

## Block the Calling Thread

**CdWaitOnProcess()**—Use this function to serialize Connect:Direct Process execution. This function blocks the calling thread until the specified Process is no longer in the TCQ. It takes a Process handle that contains references to the target Process object. Any Process object handle can enable you to specify Processes to wait on. Use this method to wait on a Process returned from a submit command and any Process returned by the select process command.

## Retrieve Error Text

- **CdGetErrorText()**—Call this function to translate return code values into messages that explain the error. This helps the user understand the error message and provides a method for logging meaningful trace messages within an application.
- **CdGetDetailedError()**—Use this function to retrieve messages one at a time until CD\_ENDOFDATA is returned. This call fills in the MESSAGE\_STRUCT structure with a detailed error message for node, parser, and connection errors. The messages are erased upon entry to any other API to prepare for other potential errors.

## Blocking

The C Application Programming Interface is synchronous; when an API that performs a complex function (such as the CdConnect() or CdExecuteCmd() functions) is called, the caller's thread is blocked until the request is completed or until a failure occurs. The caller's thread blocks while waiting for other threads to finish the request.

If the CdConnect() function is called from a Microsoft Windows application, it should not be called from the primary user interface (UI) thread. Calling the function from the UI thread causes the user interface of the program to run slowly.

## View Sample Programs

Sample programs are available for viewing.

Refer to the documentation CD directory, SDK\Samples for the C, C++, and Visual Basic sample code. The sample code contains the following:

- The CSample1.C sample program demonstrates how to connect to a node, execute a command, and view the data returned by the node.
- The CSample2.C sample program demonstrates a more complex transaction of connecting to a node, submitting a Process, waiting for completion, and requesting statistics for the Process.
- CPPSamp1

- CPPSamp2
- VBAuto
- VBStat
- VBSsubmit
- VBSsubmit2

## Apply the C++ Class Interface

---

### Compile and Debug

Include the CDSKD.H header file to use the C++ interface. CDSKD.H automatically links the program with the appropriate import library. Debug configurations link with the CDCORED.LIB, and the release configurations link with the CDCORE.LIB.

**Note:** You do not need to add the LIB to the LINK section of the project or makefile.

The CDCORED.lib and CDCORE.lib files contain the name of the DLL to dynamically load at run time and class definitions for the linker to resolve the Connect:Direct SDK symbols included in the CDSKD.H file. When a program executes or a DLL is loaded, the appropriate CDCORE.DLL is loaded. Applying.DLL is dynamically loaded when a debug configuration is executed and to support a release configuration.

### Manipulate Nodes

Component Group classes provide methods to make changes on a Connect:Direct node.

The Component Group classes represent Connect:Direct entities and provide methods to manipulate an object to generate changes on the Connect:Direct node. Use the following classes to manipulate nodes:

Class	Description
CDNode	Contains the high-level Connect:Direct functionality. It returns network map, initialization parameters, and translation table information as well as User and Proxy objects that maintain node information and execute command objects.
CDUser	Contains the user functional authority information. Use to add, delete, and update functional authorities on the Connect:Direct node, including Network map Access Flags, Command Access Flags, Control Flags, Process Statement Flags, and default directories.
CDProxy	Contains the Connect:Direct proxy information. Use to add, delete, and update proxy information on the Connect:Direct node. The remote user proxy contains information for operations initiated from a remote Connect:Direct node and defines relationships between a remote node and local user IDs.
CDTranslationTable	Contains and maintains the translation table information that translates data being sent to other nodes and provides methods for setting and retrieving translation information.
CDTrace	Holds the trace criteria. It contains all the fields returned from the node with the TRACEON command, with no parameters and provides access methods for all of the Trace fields.
CDNetmapNode	Contains the network map node information.
CDNetmapDesc	Contains the description for a network map node.
CDNetmapPath	Contains the network map path information.

Class	Description
CDNetmapMode	Contains the network map mode information.

When using the C++ Class interface, no sequence must be followed when using the C++ classes. All objects are self-contained and are not dependent on any other classes when fully constructed. Each object's constructor is different and some of the objects require another object to be built successfully.

The first and most important class is the CDNode class. This class is the first one to use when interacting with any Connect:Direct node.

While the only prerequisite for constructing a class is the creation of the objects needed by the constructor, the following example shows a possible sample execution sequence:

```

CDNode creation
    CDSelectProcCommand creation
        CProcIterator creation
            (Use the data)
        CProcIterator destruction
    CDSelectProcCommand destruction
CDNode destruction

```

The Connect:Direct CDNode class serves as the virtual Connect:Direct node. It enables you to manipulate and send commands to the actual Connect:Direct node. You manipulate this object through the use of the CDNode methods and issue commands to the node using Command objects. Calling these methods and using the objects sends KQV streams to the physical Connect:Direct node. See the *C++ API Reference Guide* for more information.

## Create an Object to Connect to a Node

The name of the Connect:Direct node and the connection information is set at object creation time using the CDNode constructor. If a parameter is not supplied (NULL pointer), the default value for that parameter is read from the Registry. During construction, the CDNode object attempts to connect to the physical Connect:Direct node using the protocol information contained in the Registry. If the connection fails, the CDConnectionException is returned. If the connection is successful but the logon is denied by the server, a CDLogonException is returned.

The CDNode object creates and removes the connection to the Connect:Direct node as needed. Connections are shared and reused as different requests are made. The following section of the class definition displays the methods to construct a CDNode object and methods to retrieve node information:

```

// Constructor for CDNode
CDNode(LPCTSTR szName=NULL, LPCTSTR szUserid=NULL, LPCTSTR szPassword=NULL,
       int nProtocol=CD_PROTOCOL_TCPIP);
CDNode(LPCTSTR szFilename);
CDNode(const CDNode &Node);
~CDNode();
//Node Information Methods
const CString GetName() const;
LPCTSTR GetCDName() const;
LPCTSTR GetUserid() const;
LPCTSTR GetServer() const;
int GetProtocol();

```

The following two examples illustrate two different methods for creating a CDNode object. The first method creates the CDNode object locally on the stack. The second example creates a dynamic allocation of a CDNode object from the stack. Both methods then execute a SELECT PROCESS command using the CDNode object.

```

{
    CDNode MyNode("MYNODE", "MYUSERID", "MYPASSWORD");
}

```

```

CDSelectProcCmd cmd;
//Execute the "SELECT PROCESS" command
CDProcIterator it = cmd.Execute(MyNode);
}
}

CDNode *pNode = new CDNode("MYNODE", "MYUSERID", "MYPASSWORD");
CDSelectProcCmd cmd;
//Execute the "SELECT PROCESS" command
CDProcIterator it = cmd.Execute(pNode);
delete pNode;
}

```

## Manage Connections

Use the CDNode class to manage Connect:Direct connections. The CDNode class creates and deletes connections to the Connect:Direct node as needed and deletes the connections if they are idle for a specified period of time.

The connections are stored in an array and are created and assigned by the CDNode object when a command requests a connection to the physical node. Connections are reused when they are idle and are deleted if they remain idle for an extended period of time. Because each connection consumes resources on both the client and the server, use them as efficiently as possible. The DisconnectAll member function is used to disconnect all connections to all nodes.

## View Information

Record Group classes allow you to view information about processes, statistics, messages, and users.

Use the following classes to obtain information:

Class	Description
CDProcess	<p>Contains all of the Process criteria information returned from a SUBMIT or SELECT PROCESS command after a Process is submitted. You can submit a Process for execution using one of the following methods:</p> <p>Create a CDSubmitCmd object and initialize the parameters. Next, call the CDSubmitCmd::Execute() method and specify the CDNode object to run on. Call the CDNode::Submit() method and specify the text of the Process. This method internally creates the CDSubmitCmd object and calls the Execute() method.</p>
CDStatistic	<p>Provides two methods for holding statistics information.</p> <p>GetAuditField() Method—Because audit data is optional, and different records have different KQV keys, use a single method to access the data. To retrieve a value, call GetAuditField(), passing the KQV key for the desired field.</p> <p>The GetAuditMap() function retrieves all audit fields defined in the current record. An MFC CMapStringToString object maps from KQV keywords to the corresponding values. This method enables you to view each association in the map to determine what audit fields are available and to ask the map for the value of the given field.</p>
CDMessage	<p>Holds information about a specific message that is retrieved from the Connect:Direct node.</p>
CDUser	<p>Holds the user functional authority information to add, delete, and update functional authority information on the Connect:Direct node.</p>

## Control the Return of Information

Use iterators to enumerate through multiple returned objects.

Commands and methods store multiple items in an iterator. The iterator provides methods to enumerate through each returned object.

## Iterators

Commands that retrieve a single record from the server block the calling thread in the `Execute()` method until the data arrives. The data is then put into a record object and returned. Other commands, like `select statistics`, can potentially return hundreds of records. If the `Execute()` method blocks until all records are returned, it can take longer to receive any feedback. If the records are all returned in one large block instead of being consumed one at a time, the computer slows down.

To solve these problems, commands that potentially retrieve multiple records return an iterator object as soon as the first record arrives. As data is returned, a background thread automatically appends to the iterator. The iterator has a connection to the server and the command object is not involved. This method allows you to process records as they arrive. The following example demonstrates the `select process` command returning a process iterator:

```
CDSelectProcCmd cmd;  
CDProcIterator it = cmd.Execute(node);
```

## Accessing Iterator Records

The iterator keeps an internal list of all records returned from the server. Use the following commands to control iterator records:

- `HasMore()`—Call this method to determine if any records are available in the list.  
**Note:** You must always call `HasMore()` before calling `GetNext()`. It is not legal to call `GetNext()` if there are no records.
- `GetNext()`—If `HasMore()` returns `TRUE`, obtain the next record in the list using this command. It removes the next record from the list and returns it.

When all records are received from the server, the server notifies the iterator that the command is complete. After all records are removed using `GetNext()`, `HasMore()` returns `FALSE`.

If the iterator's list is empty, but the server has not notified the iterator that the command is complete, the iterator cannot determine whether there are more records. In this case, `HasMore()` blocks until more records are received from the server or a completion notification is received. Only then can the iterator return `TRUE` or `FALSE`.

The following is an example of accessing statistics records using an iterator:

```
CDSelectStatCmd cmd;  
CDStatIterator it = node.Execute (cmd);  
while (it.HasMore()) {  
    CDStatistic stat = it.GetNext();  
    // use the statistics object }  
}
```

## Execute Connect:Direct Commands

Command Group classes execute `Connect:Direct` commands against `Connect:Direct` nodes.

Class	Description
CDCommand	<p>The base class for all Connect:Direct command objects. It wraps the parser within a class and enables methods for data manipulation. Each derived class provides an Execute() method to execute the command and return the resulting data or object.</p> <p>If the result is several items, the command object returns a iterator object that holds the data. The following CDCommand class definition shows the type of methods available in this class:</p> <pre data-bbox="607 449 1476 772"> Class CDCommand { public: // Constructor for CDCommand CDCommand(LPCTSTR pCommand=NULL); virtual ~CDCommand(); virtual void ClearParms(); void SetCommand(const CString&amp; strCmd); virtual CString GetCommand() const; virtual CString GetQOC() const; // Execute() methods are provided by each // derived command class. </pre>
CDSelectStatCmd	<p>Derived from the CDCommand base class, it enables you to set the SELECT STATISTICS parameters. When you call the Execute() method, an iterator data object is dynamically created and attached to the connection assigned by the CDNode object to execute the command.</p>
CDSelectProcCmd	<p>Derived from the CDCommand base class, it enables you to set the SELECT PROCESS parameters. When you call the Execute() method, the CDProcIterator object is created dynamically and attached to the connection assigned to execute the command.</p> <p>The following example demonstrates the CDSelectProcCmd class:</p> <pre data-bbox="607 1142 1476 1287"> CDSelectProcCmd cmd; CDProcIterator it = node.Execute(cmd); while (it.HasMore()) {     CDProcess proc = it.GetNext();     // use the process } </pre>
CDChangeProcCmd	<p>Derived from the CDCommand base class, it enables you to set the CHANGE PROCESS parameters. When the Execute() method is called, an iterator data object is dynamically created and attached to the connection assigned to execute the command. A CDProcIterator is attached to the iterator data and returned from the Execute() method.</p>
CDDeleteProcCmd	<p>Derived from the CDCommand base class, it enables you to set the DELETE PROCESS parameters. When the Execute() method is called, a CDProcData object is dynamically created and attached to the connection assigned to execute the command. A CDProcIterator is attached to the iterator data and returned from the Execute() method.</p>
CDSelectMsgCmd	<p>Derived from the CDCommand base class, it enables you to set the SELECT MESSAGE parameters. When you call the Execute() method, the command is executed and the resulting message text is stored in the internal CDMessage object</p>
CDStopCmd	<p>Derived from the CDCommand base class, it enables you to set the STOP parameter. When you call the Execute() method, the command is executed.</p>

Class	Description
CDSubmitCmd	Used for submitting a Process object for execution on a node. It enables you to set the options of the SUBMIT command and then execute the command on a node. When you call the Execute() method, a CDProcess object is dynamically created and attached to the connection assigned to execute the command. The following example demonstrates the CDSubmitCmd class:  <pre> . . . CDSubmitCmd cmd; cmd.SetFile ("myproc.cdp"); CDProcess proc = node.Execute(cmd); proc.WaitForCompletion(); . . . </pre>
CDTraceOnCmd	Derived from the CDCommand base class, it enables you to set and retrieve trace options from the Connect:Direct node. The TraceOnCmd class handles all the options available from the TRACEON command. The Execute() method returns a CDTrace object that contains the current trace state.
CDTraceOffCmd	Derived from the CDCommand base class, it enables you to clear trace options from the Connect:Direct node. The CDTraceOffCmd class handles all of the options available from the TRACEOFF command. You call methods to clear the desired trace parameters and then call the Execute() method. The Execute() method returns a CDTrace object that contains the current trace state.

## Manage Exception Conditions

Exception Group classes manage exception conditions. Connect:Direct generates Exception Group classes if an exception condition is encountered while a request is being processed. Following is an exception scenario where a message is pushed into the exception before the initial throw.

Function A calls Function B, and Function B calls Function C. Function C is a helper routine called by many routines so it does not include information specific to a task. Since the exception occurred in C, it throws the exception. A message describing the error is added and flagged as a technical message.

Function B traps the exception. A message describing the error is added and flagged as a user message. User messages are displayed in dialog boxes. For example, a user message reads: Communication with the server has been lost.

The CDMsgException class stores the messages as an array of strings. The messages are stored in a last-in first-out (LIFO) order because messages added later are more general as the exception moves up the call stack.

Following is a description of the Exception Group classes:

Class	Description
CDMsgException	The base exception class for all Connect:Direct exception objects. It provides a message stack for troubleshooting.
CDConnectionException	This exception is generated when communication with the node is lost or cannot be established.

Class	Description
CDCommandException	Generated when an object cannot be executed because parameters are invalid, including a submitted Process containing errors.
CDLogonException	Generated if the Connect:Direct node rejects the user ID and password supplied in the logon attempt. You can respond to this exception by prompting the user for the correct logon information.

## Manage Administrative Functions

Helper Group classes provide common functionality, such as dialog boxes and thread creation and termination.

### Manage Administrative Functions

Class	Description
CDLogonDlg	<p>The Connect:Direct common logon dialog box enables you to write your own logon applications. The CDLogon dialog box enables you to change the node, the user ID and password to connect to the Connect:Direct node as well as enable the Remember Password check box, click the Configure button to save new server logon information and change the title.</p> <p>Below are the components of the CDLogonDlg class:</p> <p>Node—Specifies the Connect:Direct node to which the user wants to logon.</p> <p>userid—Specifies the user ID for the Connect:Direct node.</p> <p>Password—Specifies the password defined for the user ID.</p> <p>Remember Password—Specifies whether the user wants the password to persist after the user logs off. If the check box is enabled, the password is retrieved to set the password field of the dialog box when the logon dialog is displayed. This prevents the user from having to re-type the password information for the session. Enabling the check box also specifies whether or not to write the password information as nonvolatile data. Nonvolatile keys persist after the user logs off. If the user does not enable the Remember Password check box, the password only persists until the user logs off.</p> <p>The Connect:Direct Logon dialog box does not perform the logon. It captures the entries and returns them to the calling program.</p> <p>Normally, the programmer creates a CDLogon dialog box, sets the parameters, and calls the DoModal() function to display and run the dialog box. If the user clicks the OK button, then the CDLogonDlg class returns IDOK and a logon is attempted using the supplied connection information. If the user clicks the Cancel button, the CDLogonDlg class returns IDCANCEL and the logon is cancelled.</p> <p>After a user successfully logs on to the Connect:Direct node, the connection information is written to the Registry under the HKEY_CURRENT_USER key.</p>
CDExceptionDlg	Displays the exception dialog box. The dialog box displays the information in the exception object
CDThread	Coordinates the clean termination of threads and provides a thread class that can unblock object
CDBeginThread	Creates a worker thread for use with API objects.



Class	Description
Return Values	A pointer to the newly created thread object.

## Create A Thread Example

The following example illustrates how to create a thread:

```

void SomeFunc()
{
    CDThread* pThread = CDBeginThread(ThreadFunc);
} void ThreadFunc(LPARAM lParam)
{
    CSomeCmd cmd(...);
    CDProcess proc = cmd.Execute(...);
    DWORD dwId = proc.GetId();
    SetDlgItemInt(IDC_SOMECONTROL, (int)dwId);
}

```

## Terminate A Thread

In the preceding sample code, the only blocking that takes place is in the `Execute()` function. `Execute()` blocks until the `Process` information returns from the server. To terminate the thread without waiting, call `CDThread::Exit`, which signals any blocking `CD` objects in the thread to stop blocking and throw a thread exit exception. In the previous example, if `CDThread::Exit` is called, an exception is thrown, and no return object is returned from the `Execute()` function.

**Note:** It is not possible for one thread to throw an exception in another. `CDThread::Exit` sets flags in the `CDThread` object that other `CD` objects use.

When `CDThread::Exit` is called, `CDThread::IsExiting` returns `TRUE`. You can use this method in loops to determine when to exit because `CD` objects only throw the exception when they are blocking.



**CAUTION:** Do not call the Win32 `TerminateThread`. `TerminateThread` does not give the thread a chance to shut down gracefully. Calling `TerminateThread` can corrupt the state of the `CD` objects. `CD` objects use critical sections and other resources that must be managed carefully.

## Catch the Exception

It is not necessary to catch the `CDThreadDeath` exception. If not caught, the exception unwinds the stack, destroying all objects on the stack, and the `CDThread` object itself handles the exception. To provide clean-up for heap allocated items, the exception can be caught. Rethrowing the exception is not required.

## Multithreaded Access and Blocking

Because the `Connect:Direct C++ Class API` uses multiple threads, the API objects are thread safe. The API objects provide efficient blocking for use in multithreaded programs.

## Objects On The Stack

Use the stack to ensure efficiency and reduce complexity.

C++ programs that make good use of exceptions move as much data from the heap to the stack as possible. This ensures that destructors run and memory is released when an exception occurs. It also reduces the complexity of the program by eliminating many pointers, reducing the chances of memory leaks, and letting the compiler ensure that objects are valid (as opposed to pointers that could be `NULL` or `bad`).

To ensure objects are used on the stack efficiently, most `CD` objects store their data externally. The following example is of an iterator object that holds 500 statistics records:

When the iterator is created, an iterator data object is also created to hold the records. The data object also has a reference count that indicates how many objects are using the data. When an object is copied, the new object (the copy) is linked to the data and the reference count of the data object is incremented. There are still only 500 records (not 1000), and the reference count is now 2.

When connected objects are destroyed, they decrement the reference count in the data object. When the reference count reaches 0, the data object is also destroyed. The following figure provides an example of the efficiency possible when shared data is copied:

```

1. void Func()
2. {
3.     Iterator itFinal = CreateIterator();
4. }
5.
6. Iterator CreateIterator()
7. {
8.     CSomeCmd cmd(...);
9.     Iterator itLocal = node.Execute(cmd);
10.    return itLocal;
11. }

```

On line 3 the sample code calls the CreateIterator() function. The CreateIterator() function returns an iterator, called itLocal. This iterator is created on line 9 and returned on line 10.

At line 11 the C++ compiler creates a temporary copy of itLocal before destroying it. As part of the copy, the iterator data reference count is incremented to 2. When itLocal is destroyed, the reference count drops to 1 so that the records are not deleted.

Next, the C++ compiler constructs itLocal on line 3 by passing the temporary to its copy constructor. The reference count is again incremented to 2 because both iterators are pointing to it. The temporary is then destroyed, reducing the reference count to 1.

The result is that an unlimited number of records are passed to the stack with little more than the copying of two pointers and some reference counting.

## Apply the ActiveX Control Interface

### Submit Process

The Connect:Direct CDSubmit control is a command line control that submits Processes to the server. Because submitting a Process can be a lengthy procedure, the Execute command returns immediately. When a Process is submitted and the server responds, or a time-out occurs, the client is notified through the SubmitStatus event. Additionally, the client can request notification when the Process has completed on the server. Properties for the CDSubmit control follow:

Property	Description
Node=nodename	The name of the node that you want to connect to. The node name must be valid in the Microsoft Windows system Registry.
User=userid	The user ID used to log on to the Connect:Direct node.
Password=password	The password used by the user ID to log on to the node.
Text=text	The text of the Process.

### Methods

Use the following methods to submit a process:

Method	Description
Execute(BOOL bWait)	Submits the Process to the server. An event is fired when the server responds to notify the client of the status of the submit. If bWait is TRUE, another event is fired when the Process completes on the server.
SetSymbolic(symbolic, value)	Sets the symbolic value for symbolic. Call for each symbolic in the Process.
ClearSymbolics	Clears all symbolics. Call before submitting a Process to clear the previous values.

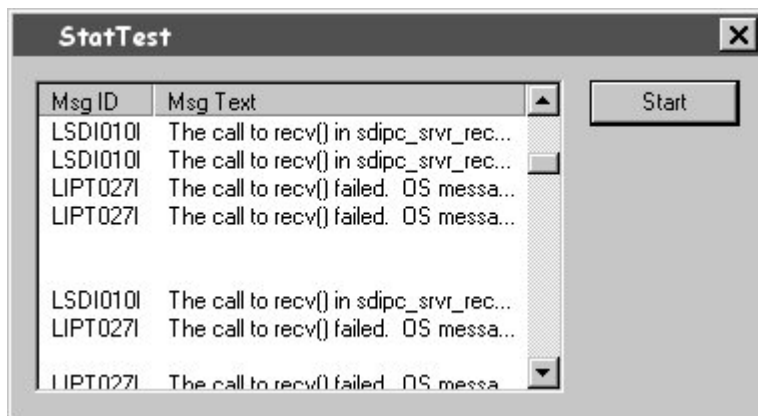
## Events

The following events are activated by the CDSubmit control:

Events	Description
Submitted	Describes whether the Process is accepted by the server.
Completed	The ProcessComplete event is sent when the Process is no longer in the server's queue. Because more resources are required to wait on a Process, this event is only fired if requested in the call to Execute.
Error	The standard error event. Possible codes are: CTL_E_PERMISSIONDENIED—cannot log onto the node. CTL_E_DEVICEUNAVAILABLE—cannot connect to the node. CTL_E_OUTOFMEMORY—out of memory. CTL_E_ILLEGALFUNCTIONCALL—an unknown error. The error message describes the error.

## Display Select Statistics Results

The CDStatistics control is a multi-column list that displays SELECT STATISTICS command results. The CDStatistics control properties determine the node that you are connected to, logon information, and selection criteria. The following figure shows the CDStatistics control where only the message ID and message text are selected.



## Properties

The following table lists the CDStatistics control properties:

Property	Description
ColCount=nnnnn	The number of columns to display. The range for the ColCount value is 1–32,000.
Col=nnnnn	The current column. The range for the Col value is 1–32,000.
ColWidth=nnnnn	The width of the current column (Col) in pixels. The range for the ColWidth value is 0–32,000.
Header	The column header text for the current column. Provide text for the value or leave it blank.
Row=nnnnn...	The current row. If set to 0, the current row is the header. The range for the Row value is 0–Infinity, where the number of rows is limited only by memory.
RowCount=positive integer	The number of rows in the list, not including the header. This field is read-only and is determined by the number of records returned by the server.
Node=node name	The name of the node to which you want to connect. The node name must be valid in the MicrosoftWindows NT system Registry.
User=userid	The user ID used to log on to the Connect:Direct node.
Password=password	The password defined to allow the user ID to log onto the node.
Field	The statistics structure field the current column is displaying. Valid values are Process Name, Process Number, Condition Code, Feedback, MsgId, MsgText, MsgData, LogDateTime, StartDateTime, StopDateTime, Submitter, SNode, RecCat, and RecId.
ccode=(operator, code)	Selects 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. Refer to dfile=destination filename   (list) below for valid operators and values.
dfile=destination filename   (list)	Searches all copy termination records (CAPR category, CTFC record ID) to find those with a destination file name matching the file name or list of file names specified.  This parameter is not supported in a UNIX environment.
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. Connect:Direct assigns the Process number when the Process is submitted.

Property	Description
reccat=caev   capr   (caev , capr)	<p>Selects statistics based on whether the record category is related to events or to a Connect:Direct Process.</p> <p>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.</p> <p>caev specifies that the retrieved statistics file records include those related to Connect:Direct events, such as a Connect:Direct shutdown.</p> <p>capr specifies that the retrieved statistics file records include those related to one or more Connect:Direct Processes.</p>
rnode=remote node name   generic   (list)	<p>Selects statistics file records by remote node name, a generic node name, or a list of node names. The range for the remote node name is 1–16 alphanumeric characters long.</p>
sfile=filename   (list)	<p>Searches all copy Process Termination records (CAPR category, CTCR record ID) to find those with a source file name matching the name or list of names you specify.</p>
startt=([date   day] [, time])	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

Property	Description
stopt=([date   day] [, time])	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
submitter=(node name, userid)   generic   (list)	<p>Selects statistics by the node name and user ID of the Process owner (submitter). You can also specify a generic name and user ID or a list of names and user IDs. The maximum combined length, including the node name and user ID, is 66 characters.</p> <p>Valid completion code operators for the ccode property are listed below:</p> <p>eq   =   == Equal (default)</p> <p>ge   &gt;=   =&gt; Greater than or equal</p> <p>gt   &gt; Greater than</p> <p>le   &lt;=   =&lt; Less than or equal</p> <p>lt   &lt; Less than</p> <p>ne   != Not equal</p> <p>Valid completion codes for the ccode property are listed below:</p> <p>0 – Successful execution of the Process.</p> <p>4 – A warning-level error was encountered. The statement probably completed normally, but verify the execution results.</p> <p>8 – An error occurred during Process execution.</p> <p>16 – A severe error occurred during Process execution.</p>

Property	Description
recids=record id   (list)	<p>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.</p> <p>AUPR – Authorization file processing</p> <p>CHGP – Change Process command issued</p> <p>COAC – Communication activated</p> <p>CMLT – CMGR listen thread terminated</p> <p>CRHT – Connect:Direct copyright</p> <p>CSTP – Child Process stopped</p> <p>CTRC – Copy control record written</p> <p>CTRM – Child Process terminated</p> <p>CUKN – Child Process unknown status</p> <p>CXIT – Child Process exited</p> <p>DELP – Delete Process command issued</p> <p>FLSP – Flush Process command issued</p> <p>FMRV – Formatted Header (FMH) received</p> <p>FMSD – Formatted Header (FMH) sent</p> <p>GPRC – Get Process issued</p> <p>IFED – If statement ended</p> <p>IPPR – Initialization parameter processing</p> <p>LIOK – Listen okay</p> <p>NAUH – Node Authorization check issued</p> <p>NMOP – Network map file opened</p> <p>NMPR – Network map processing</p> <p>NUIC – Connect:Direct Initialization complete</p> <p>NUIS – Connect:Direct start initialization</p> <p>NUT1 – Connect:Direct phase one termination complete status</p> <p>NUT2 – Connect:Direct phase two termination complete status</p> <p>NUTC – Connect:Direct termination complete</p> <p>NUTR – Connect:Direct termination requested</p> <p>NUTS – Connect:Direct termination started</p>

Property	Description
recids=record id   (list) (Continued)	PERR – Process error detected PFLS – Process flushed PMED – Process Manager ended PMIP – Process Manager Initprocs thread initialized PMMX – Process Manager Max Age thread initialized PMRC – Process Manager release cell thread initialized PMST – Process Manager started PPER – Pipe error PRED – Process ended PSAV – Process saved PSED – Process step detected PSTR – Process started RNCF – Remote server call failed RTED – Run Task command completed RJED – Run Job command completed RFIP – Refresh command issued SBED – Submit complete SELP – Select Process command issued SELS – Select Statistics command issued SEND – Session end issued SERR – System error SHUD – Connect:Direct shutdown SIGC – Signal caught SMED – Session Manager ended SMST – Session Manager started SNHI – APPC started SNMP – SNMP STOP – Stop Connect:Direct command issued SUBP – Submit command issued

Property	Description
recids=record id   (list) (Continued)	TCPI – TCP started TRAC – Trace command issued UNKN – Unknown command issued USEC – User Security check issued xxxx – Record types identified by the first four characters of the message ID



## Methods

The CDStatistics control provides the following methods:

Method	Description
BOOL Execute()	Executes the SELECT STATISTICS command and stores the returned records in the control. If the control was already retrieving records, the previous command is stopped and the old records are removed from the control.
Clear	Clears the existing records from the display. The Clear method does not stop retrieval.

## Events

The following events are controlled by CDStatistics.

Method	Description
Complete	Sent after all records are retrieved.
Error	The standard error event. Possible codes are: CTL_E_PERMISSIONDENIED—cannot log onto the node. CTL_E_DEVICEUNAVAILABLE—cannot connect to the node. CTL_E_OUTOFMEMORY—out of memory. CTL_E_ILLEGALFUNCTIONCALL—an unknown error.

## Apply Automation Servers

---

### Apply Automation Servers

The Connect:Direct Automation Servers provide an automation wrapper around the Connect:Direct SDK C++ classes.

The Automation Servers provide direct automation support for languages like Visual Basic. This section provides a reference for the automation objects and information about applying them.

### Create Virtual Servers Using the Node Factory

The node factory creates node objects, which act as virtual servers. Virtual servers represent a Connect:Direct server (a node). The Automation Server Node Factory provides the following properties:

Property	Description
Node Name	The name of the node to connect to. The node name is set using the Connect:Direct Client Connection Utility.
Userid	The user ID to use when connecting to the node.
Password	The password for the user ID to connect to the node.

The Connect:Direct Automation Server Node provides the following methods:

Method	Description
SelectStats(criteria)	Criteria specifies the complete SELECT STATISTICS string.
SelectProc(criteria)	Criteria specifies the complete SELECT PROCESS string.

Method	Description
Submit(text)	The text specifies the Process to SUBMIT.

## Identify Active Processes

The Process object represents a Process running on the node. The records are returned as Process objects, stored in a ProcCollection container. The Connect:Direct Automation Server Process object provides the following properties:

Property	Type	Description
ProcessName	String	The Process name.
ProcessNumber	Long	The Process number assigned by Connect:Direct when the Process is placed in the TCQ.
ConditionCode	Long	The return code.
Feedback	Long	Provides additional return code information.
MsgId	String	The message identifier.
MsgText	String	The message text field.
MsgData	String	Message substitution fields.
LogDateTime	Date	The logged time stamp.
SchedDateTime	Date	The date and time the Process is scheduled to be submitted.
SubmitNode	String	The name of the node from which the Process was submitted.
Submitter	String	The user ID of the person submitting the Process.
PNode	String	The primary or controlling node in the Process.
SNode	String	The secondary or partner node in the Process.
Status	String	The status of the Process in the queue.
Retain	String	Specifies whether the Process is to be retained in the TCQ for future submission.
Hold	String	The TCQ hold status of the Process.
Class	Long	The session class on which the Process is executing.
Priority	Long	The TCQ selection priority of the Process.
ExecPriority	Long	The operating system execution priority of the Process.
Queue	String	The logical queue where the Process is currently located (Execution, Hold, Wait, or Timer).
Step Name	String	The currently executing step of the Process.
LocalNode	String	Specifies whether the primary or secondary node is the local node and has primary control.
FromNode	String	Specifies whether the primary or secondary node is the source node in a copy.
SimpleCompress	Boolean	Specifies whether to perform repetitive character compression.

Property	Type	Description
ExtendedCompression	Boolean	Specifies whether to perform repetitive string compression.
Checkpoint	Boolean	Specifies the use of checkpointing in a copy step.
Restart	Boolean	Specifies whether the Process is restarted.
SourceFile	String	The name of the source file.
TotalBytes	Long	The number of data bytes read or written.
TotalRecs	Long	The number of data records read or written.
SentBytes	Long	The number of data bytes sent.
Sent RUs	Long	The number of RU bytes sent.
DestFile	String	The name of the destination file.

### Identify Statistic Records

The Statistic object represents the records in the statistics database. They are returned from a SELECT STATISTICS query. The Connect:Direct Automation Server Statistic object provides the following properties:

Property	Data Type	Description
ProcessName	String	The Process name.
ProcessNumber	Long	The Process number assigned by Connect:Direct when the Process is placed in the TCQ.
Feedback	Long	Provides additional return code information.
MsgId	String	Message identifier.
MsgText	String	Message text.
MsgData	String	Message substitution fields.
LogDateTime	Date	The logged time stamp.
StartDateTime	Date	The start time stamp.
StopDateTime	Date	The stop time stamp.
Submitter	String	The submitter's user ID.
SNode	String	The secondary node name.
RecCat	String	The record category.
RecId	String	The record identifier tag.

Property	Data Type	Description
GetAuditField	String	<p>Returns the audit field value.</p> <p>The GetAuditField() function supports the following audit information field names:</p> <ul style="list-style-type: none"> <li>"Step Name"</li> <li>"Primary Node Name"</li> <li>"Secondary Node Name"</li> <li>"Link Fail"</li> <li>"Translation"</li> <li>"Status"</li> <li>"Function"</li> <li>"Member Name"</li> <li>"Sysopts"</li> <li>"Bytes Read"</li> <li>"Records Read"</li> <li>"Bytes Sent"</li> <li>"RUs Sent"</li> <li>"Bytes Written"</li> <li>"Records Written"</li> <li>"Bytes Received"</li> <li>"RUs Received"</li> <li>"RU Size"</li> <li>"Local Condition Code"</li> <li>"Local Message ID"</li> <li>"Other Condition Code"</li> <li>"Other Message ID"</li> <li>"PNode Accounting Info"</li> <li>"SNode Accounting Info"</li> <li>"Local Node"</li> <li>"Retain"</li> <li>"Class"</li> <li>"Priority"</li> <li>"Execution"</li> <li>"Standard Compression"</li> <li>"Extended Compression"</li> <li>"Checkpoint"</li> <li>"Scheduled Date/Time"</li> <li>"Start Date/Time"</li> <li>"Stop Date/Time"</li> <li>"Submit Date/Time"</li> <li>"From Node"</li> <li>"Queue"</li> <li>"Restart"</li> <li>"Function"</li> </ul>

Property	Data Type	Description
GetAuditField (Continued)	String	<p>Returns the audit field value.</p> <p>The GetAuditField() function supports the following audit information field names:</p> <ul style="list-style-type: none"> <li>"Source File"</li> <li>"Source Disposition #1"</li> <li>"Source Disposition #2"</li> <li>"Source Disposition #3"</li> <li>"Destination File"</li> <li>"Destination Disposition #1"</li> <li>"Destination Disposition #2"</li> <li>"Destination Disposition #3"</li> <li>"Hold"</li> <li>"Substitution String"</li> <li>"Submitter Node"</li> </ul>

## Use Automation Objects

Create node objects, select processes, and select statistics using automation objects.

This topic explains how to use the node factory and nodes, select statistics, and select Processes. The Connect:Direct automation objects use late binding, so you must dimension your variables as type Object.

### Create Node Objects

The Connect:Direct node factory creates node objects. These node objects serve as virtual servers and represent a connection to a Connect:Direct server (node).

To obtain a connection (and therefore a node), you must use the node factory. Create the node factory using the ProgID CD.NodeFactory:

```
Dim factory as Object
Set factory = CreateObject ("CD.NodeFactory")
```

To determine the node you want to connect to, set the properties of the factory object. Next, call CreateNode to connect to the node. If the connection is successful, a node object returns. Otherwise, an error is thrown indicating the cause of the problem.

```
factory.NodeName = "CD.Node1"
factory.UserId = "user1"
factory.Password = "password"
{
Dim node as Object
Set node = factory.CreateNode()
```

The node name refers to the name used by the Client Connection Utility. You must set up the nodes that you want to connect to using the Client Connection Utility prior to using the Connect:Direct SDK.

### Node Usage

The node object represents the connection to a Connect:Direct node. Using the node enables you to select statistics or Processes.

## Select Processes

To select Processes, you must first format a select Process command and pass it to the SelectProc method. The records return as Process objects and are stored in the ProcCollection container. Because a background thread populates the collection, it is returned to the caller before it is completely filled. Therefore, the only access method available is using the For Each construct.

**Note:** The usual Count property is not available because the count is not known until all records are returned.

```
Dim procs as Object ; the process collection
Dim proc as Object ; each process record
Set procs = node.SelectProc ("SELECT PROCESS ")
For Each proc in procs
    Debug.Print proc.ProcessName
Next proc
```

## Select Statistics

To select statistics records, you must format a select statistics command and pass it on to the SelectStats method of the node. The records return as Statistic objects stored in a StatCollection container. Because a background thread populates the collection, it returns to the caller before it is completely filled. Therefore, the only access method available is using the For Each construct.

**Note:** The usual Count property is not available because the count is not known until all records are returned.

```
Dim stats as object ; the Statistics collection
Dim stat as Object ; each statistic record
Set stats = node.SelectStats ("SELECT STATISTICS")
For Each stat in stats
    Debug.Print stat.RecId
Next stat
```

Because the server can send records slowly, the interface can be jerky while reading records. Because records are read using a background thread, it useful to select the statistics before time-consuming tasks like constructing windows. This method enables the server to send records in background.

## Automation Class Errors

The automation classes use the standard Visual Basic error-handling mechanism. When an error is raised in an automation object, no real value is returned from the function. For example, if an error is raised in the node factory example in the Create an Object to connect to a Node topic (see related link below), the node does not have a value (it has the default value of nothing) because CreateNode has not returned anything.

When the Connect:Direct automation objects raise an error, they set the error number to a Connect:Direct SDK error value and store a description in the error text.

## Enhance Security and Automate File Opening with User Exits

---

### User Exits

You can customize Connect:Direct operations with user exits. User exits are user-defined dynamic link libraries (DLLs) that are loaded and called when the user exit is enabled through an initialization parameter. Two user exits are provided: one for enhanced security and one for automated file opening.

#### Related concepts

[Apply Automated File Opening](#)

## Apply Enhanced Security

### Apply Passticket Support

Use passtickets to implement enhanced security. A passticket is a one-time password generated on the primary node and passed to the secondary node within 10 minutes, where it is validated before further processing is performed. Connect:Direct passticket support is implemented by the user as a user exit called from the Connect:Direct session manager during Process execution. To enable the security exit, specify the name or path name of the security exit DLL in the value of the security.exit parameter.

See Changing Connect:Direct for Microsoft Windows Settings in the *IBM Connect:Direct for Microsoft Windows System Guide* or IBM Connect:Direct for Microsoft Windows Help for a description of the security.exit parameter. If the DLL is not in the search path of the server, then you must specify the fully qualified file name of the DLL.

The user's security exit must contain the GeneratePassticket() and ValidatePassticket() functions. The parameters for these functions are defined in the userexit.h header file. The userexit.h header file is in the Connect:Direct samples directory. If the security exit cannot be found or loaded, or if the addresses of the two required functions cannot be resolved successfully, an error message is generated and Process execution terminates.

- The passticket is only valid for 10 minutes after it is generated. As a result, the system clocks on the two nodes should be synchronized.
- When generating passtickets, Connect:Direct for Microsoft Windows fills in the GENMSG\_T structure fields and passes the structure to the security exit. The security exit should generate the passticket, fill in the GENMSG\_REPLY\_T structure fields, and return an appropriate return code to Connect:Direct.
- When validating a passticket, Connect:Direct for Microsoft Windows fills in the VALMSG\_T structure fields and passes the structure to the security exit. The security exit validates the passticket, fills in the VALMSG\_REPLY\_T structure fields, and returns an appropriate return code to Connect:Direct. If the passticket is successfully validated, Connect:Direct for Microsoft Windows continues as if the Process is using a remote user proxy. A proxy must be defined on the remote node for the effective ID being used on the SNODE for the Process.

### Security Exit Structure

Following is a list of the security exit structures:

- GENMSG\_T—Sends a message to the local node to allow the security exit to determine the user ID and security token (passticket) to use for remote node authentication. The GENMSG\_T contains:
  - Submitter ID
  - Local node ID and password
  - Remote node ID and password
  - Local node name
  - Remote node name
- GENMSG\_REPLY\_T—The user exit GeneratePassticket() function fills the GENMSG\_REPLY\_T structure. The GENMSG\_REPLY\_T contains:
  - Status value of GOOD\_RC (0) for success, or ERROR\_RC (8) for failure.
  - Status text message. If the status value is failure, then status text message is included in the error message.
  - ID to be used for security context on the remote node.
  - Passticket to use in conjunction with the ID for security on the remote node.

- VALMSG\_T—The message sent to the remote node to allow the security exit to validate the user ID and passticket. The VALMSG\_T contains:
  - Submitter ID
  - Local node ID and password
  - Remote node ID and password
  - Local node name
  - Remote node name
  - ID to be used for security checking from the local node
  - Passticket generated on the local node
- VALMSG\_REPLY\_T—The user ValidatePassticket0 function fills the VALMSG\_REPLY\_T structure. The VALMSG\_REPLY\_T contains:
  - GOOD\_RC (0) if the reply was a success or ERROR\_RC (8) for failure.
  - Status text message. If the status value is failure, the status text message is included in the error message.
  - ID to be used for security context the remote node side. This value may or may not be the same ID as in the generate message.
  - Passticket to use in conjunction with ID for security on the remote node.

## Security Exit Sample Code

The following header file and sample code files for passticket implementation are copied to *X:\installation directory\Server\samples* during the installation. You can use them as examples to follow in implementing your real-life security exit.

- userexit.h—Contains defined constants used for passtickets, the structures that are passed to the passticket functions, and the function prototypes.
- usersamp\_skel.c—Consists of the GeneratePassticket() and ValidatePassticket() functions. The GeneratePassticket() function replies with a hard-coded ticket, fills in the structure, and returns a valid return code. It demonstrates what should be input and output by the exit. The ValidatePassticket() function returns a good return code indicating that the passticket passed in is valid. There is no real checking done in this routine.
- userexit\_samp.c—Demonstrates a sample implementation of passticket support. It works if the same exit is on both sides. The GeneratePassticket() and ValidatePassticket() functions call the Passtk() function which performs the actual generation, or validation of the passticket.

The sample user exit can be compiled and linked into a DLL using Microsoft Visual C++. The userexit\_samp.sln and userexit\_skel.sln files can be found in the same samples directory where userexit\_samp.c and userexit\_skel.c is found. The exit was tested using Microsoft Visual Studio 2008.

### Related concepts

[User Exits](#)

## Apply Automated File Opening

Use the file open exit feature to override the values specified in the COPY statement. The file open exit is an initialization parameter (file.exit) that you can set to point to a user-written DLL. You can customize Connect:Direct COPY operations by defining values in the file open exit DLL that override the COPY statement parameters.

### Apply the File Open Exit

Connect:Direct file open support is implemented as a user exit called from the Connect:Direct session manager during Connect:Direct COPY statement execution. To enable the file open exit, change the value of the file.exit initialization parameter to the name or path name of the file open exit DLL.



Refer to Changing Connect:Direct for Microsoft Windows Settings in the *IBM Connect:Direct for Microsoft Windows System Guide* or IBM Connect:Direct for Microsoft Windows Help for a description of the file.exit parameter. If the DLL is not in the search path of the server, then you must specify the fully qualified file name of the DLL.

The user's file open exit must contain the FileOpen() function. The parameters for this function are File\_Open and File\_Open\_Reply. These parameters are pointers to corresponding structures in the userexit.h header file. The userexit.h header file is in the Connect:Direct samples directory.

## File Open Exit Structures

The file open exit contains the following types of structures:

- FILE\_OPEN: The FILE\_OPEN structure contains the information that implements the file open user exit. The FILE\_OPEN structure contains the following components:
  - int oflag—Flags that Connect:Direct uses to open the file.
  - int srcdstflag—Specifies whether the file is a source file (the file to read) or a destination file (the file to write to).
  - char user\_name[MAX\_USER\_NAME]—Specifies the name of the user that submitted the Process.
  - COPY\_T copy\_ctl—Points to the Connect:Direct Copy Control Block data structure that contains information concerning the COPY operation about to be performed.
  - COPY\_SYSOPTS\_T cp\_syopts—Points to the Sysopts data structure that contains a representation of all of the COPY operation sysopts that Connect:Direct supports. Refer to the *Connect:Direct Process Language Reference Guide* for more information about COPY sysopts.
- FILE\_OPEN\_REPLY: The FILE\_OPEN\_REPLY structure contains information that specifies whether the file exit operation succeeded. The FILE\_OPEN structure contains the following components:
  - HANDLE hFile—Contains a valid file handle if the file was opened successfully.
  - char filename[MAX\_FILE\_NAME\_LEN]—Contains the actual name of the file opened by the file open exit.

## Access Sample Code

The following header file and sample code files for file open exit implementation are copied to *X:\installation directory\Samples* during Connect:Direct for Microsoft Windows installation.

- userexit.h
- FileOpenDLL.CPP

### Related concepts

[User Exits](#)

## Password Exit

Connect:Direct for Microsoft Windows need the password for the UserId to impersonate, we must store the password until it is needed. This is a security concern. In an effort to eliminate having to store passwords, a configurable Password Exit feature is added to Connect:Direct for Microsoft Windows.



**Attention:** You are responsible for configuring the password vault software securely. In order to restrict access to the your Password Exit DLL, it is recommended to create a folder that only contains the Password Exit DLL. Your Password Exit DLL folder should have the following permissions:

- If Connect:Direct for Microsoft Windows service is configured to run using the 'Local System' account (which is the Connect:Direct for Microsoft Windows installation default):
  1. Add the group 'SYSTEM' to the folder's security.
  2. Set the folder Permissions for 'SYSTEM' to allow 'Read & execute'.

- If the Connect:Direct for Microsoft Windows service is configured to run using a user account:
  1. Add the user to the folder's security.
  2. Set the folder Permissions for the user to allow 'Read & execute'.

Permission to access this folder for other users should be restricted or removed.

## DLL Interface

Connect:Direct will load and unload DLL dynamically.

1. The Connect:Direct's Password Exit logic will call the 'GetPassword' method of the user supplied Password Exit DLL to obtain the password for the user. Following is the Typedef for GetPassword Function call:

```
typedef int (*PFNGETPASSWORD)(GET_PASSWORD_REQUEST_T*, GET_PASSWORD_REPLY_T*);
```

2. Connect:Direct calls the Windows API 'LoadLibrary' to load the user supplied Password Exit DLL and Windows API 'GetProcAddress' to obtain the GetPassword function's pointer in the user supplied Password Exit DLL. The request and reply structure are as follows:

**Get Password Request-** The request structure "GET\_PASSWORD\_REQUEST\_T" contains :

```
typedef struct get_password_request
{
    Int64 version;
    char applicationID[MAX_APPL_NAME];
    char policyID[MAX_POLICY_NAME];
    char userID[MAX_USER_NAME];
} GET_PASSWORD_REQUEST_T;
```

**Request structure fields contain :**

Parameter Name	Description	Valid Values
Version	64bit number	Value is always '1'. (Will increment if the request structure changes)
applicationID	Character string null terminated (Any valid ascii characters).	128 bytes in length maximum (127 bytes of data + null terminator). This is the password.exit.appl.id field in initparms.
policyID	Character string null terminated (Any valid ascii characters).	128 bytes in length maximum (127 bytes of data + null terminator). This is the password.exit.policy.id field in initparms.
userID	Character string null terminated (Any valid ascii characters).	128 bytes in length maximum (127 bytes of data + null terminator). This is the User Id for which we need the password.

**Get Password Reply-** The request structure GET\_PASSWORD\_REPLY\_T contains:

```
typedef struct get_password_reply
{
    Int64 version;
    Int64 status;
    char text_status[MAX_TXT_LEN];
    char userID[MAX_USER_NAME];
    char password[MAX_PASSWORD_LEN];
} GET_PASSWORD_REPLY_T;
```

**Reply structure fields contain:**

Parameter Name	Description	Valid Values
Version	64bit number	Value is always '1'. (Will increment if the request structure changes)
status	64bit number (Any valid numeric value) used as the return code returned by the Password Exit DLL.	A value of 0 (zero) indicates the Password Exit successfully supplied the password for the requested user.
text_status	<ul style="list-style-type: none"> <li>Character string null terminated (Any valid ascii characters)</li> <li>256 bytes in length maximum (255 bytes of data + null terminator)</li> <li>This is string which contains the result of the GetPassword request. It is intended to be a verbose success or error message returned by the Password Exit DLL.</li> </ul>	
userID	<ul style="list-style-type: none"> <li>Character string null terminated (Any valid ascii characters)</li> <li>256 bytes in length maximum (255 bytes of data + null terminator)</li> <li>This is the User Id for which the password request is received.</li> </ul>	
password	<ul style="list-style-type: none"> <li>Character string null terminated (Any valid ascii characters)</li> <li>256 bytes in length maximum (255 bytes of data + null terminator)</li> <li>This is the requested password for the User Id returned by the Password Exit DLL.</li> </ul>	

## Sample Password Exit DLL

A sample Password Exit DLL is provided in "Server/Samples/PasswordExit" directory and can be used to test the feature or as an example for writing your own. The sample simply reads a text file, gets the password for the user from the text file and returns it to the caller.

**Note:** You need Microsoft's Visual Studio 2019 to build the Password Exit DLL. The Password Exit DLL must be a 64bit DLL. The sample Password Exit DLL should not be used in a Production environment. It is only provided as an example.

Files in the PasswordExit directory are :

- **Visual Studio Files**

- **PasswordExitDLL.sln**– Visual Studio Solution file. Open this file in Visual Studio 2019 to build the Sample Password Exit DLL.
- **PasswordExitDLL.vcproj\***– Visual Studio project files referenced by the Visual Studio Solution file.

- **Sample Password Exit DLL source files**

- **Pswdexitlibrary.cpp**– Source file for the sample Password Exit DLL.
- **Pswdexlibrary.h**– Header file for the sample Password Exit DLL.
- **Dllmain.cpp**– dll entrypoint.

# Structure Types

## Structure Types

Following is a list of the common C and C++ Class interface structures, constants, and their descriptions.

- NETMAP\_DESC\_STRUCT Structure
- USER\_STRUCT Structure
- MESSAGE\_STRUCT Structure
- NETMAP\_MODE\_SNA Structure
- NETMAP\_MODE\_TCP Structure
- NETMAP\_NODE\_STRUCT Structure
- NETMAP\_PATH\_STRUCT Structure
- PROCESS\_STRUCT Structure
- NODE\_STRUCT Structure
- STATISTICS\_STRUCT Structure
- TRACE\_STRUCT Structure
- TRANSLATE\_STRUCT Structure

All of the common C and C++ Class API structures are contained within the CONNDIR.H header file.

## NETMAP\_DESC\_STRUCT Structure

The NETMAP\_DESC\_STRUCT structure contains the Netmap Node Description information. Use this structure to retrieve and set the Netmap Node Description information.

### Structure

```
struct Netmap_Desc_Struct
{
    TCHAR Name[MAX_NODE_NAME_LEN+1];
    TCHAR ContactPhone[MAX_PHONE_NUMBER+1];
    TCHAR ContactName[MAX_CONTACT_NAME+1];
    TCHAR Description[MAX_DESCRIPTION+1];
};
typedef struct Netmap_Desc_Struct NETMAP_DESC_STRUCT;
```

### Members

Member	Description
Name [MAX_NODE_NAME_LEN+1]	The node name.
ContactPhone [MAX_PHONE_NUMBER+1]	The phone number of the person responsible for this node.
ContactName [MAX_CONTACT_NAME+1]	The name of the person responsible for this node.
Description [MAX_DESCRIPTION+1]	Node description information.

## USER\_STRUCT Structure

The USER\_STRUCT structure contains the User Functional Authority information. Use this structure to retrieve and set user functional authorities.

## Structure

```

struct User_Struct
{
    TCHAR Name [MAX_OBJECT_NAME+1];
    TCHAR UpdateNetmap;
    TCHAR UpdateUser;
    TCHAR UpdateProxy;
    TCHAR ChangeProcess;
    TCHAR DeleteProcess;
    TCHAR SelectProcess;
    TCHAR SubmitProcess;
    TCHAR SelectStats;
    TCHAR SecureRead;
    TCHAR SecureWrite;
    TCHAR Stop;
    TCHAR Trace;
    TCHAR SelectNetmap;
    TCHAR SelectMessage;
    TCHAR Refresh;
    TCHAR ProcessCopy;
    TCHAR ProcessRunJob;
    TCHAR ProcessRunTask;
    TCHAR ProcessSubmit;
    TCHAR InheritRights;
    TCHAR TrusteeAssign;
    TCHAR UpdateACL;
    TCHAR FileAttributes;
    TCHAR SNodeId;
    TCHAR ExecutionPriority;
    TCHAR ProcessSend;
    TCHAR ProcessReceive;
    TCHAR UpdateTranslation;
    TCHAR DownloadDirectory[MAX_DIRECTORY_NAME+1];
    TCHAR UploadDirectory[MAX_DIRECTORY_NAME+1];
    TCHAR ProcessDirectory[MAX_DIRECTORY_NAME+1];
    TCHAR ProgramDirectory[MAX_DIRECTORY_NAME+1];
};
typedef struct User_Struct USER_STRUCT;

```

## Members

Member	Description
UpdateUser	Specifies permission to update other user functional authority.
UpdateProxy	Specifies permission to update proxy user information.
ChangeProcess	Gives a user permission to issue CHANGE PROCESS.
DeleteProcess	Gives a user permission to issue DELETE PROCESS.
SelectProcess	Gives a user permission to issue SELECT PROCESS.
SubmitProcess	Gives a user permission to issue SUBMIT PROCESS.
SelectStats	Gives a user permission to issue SELECT STATISTICS.
SecureRead	Gives a user permission to read Connect:DirectSecure Plus network map fields.
SecureWrite	Gives a user permission to modify Connect:Direct Secure Plus network map fields.
Stop	Gives a user permission to issue the STOP Connect:Direct server command.
Trace	Gives a user permission to start and stop Connect:Direct tracing.

Member	Description
SelectNetmap	Gives a user permission to get the network map objects from the Connect:Direct server.
SelectMessage	Gives a user permission to get Connect:Direct message information from the Connect:Direct server.
Refresh	Gives a user permission to execute the REFRESH INITPARMS commands.
ProcessCopy	Gives a user permission to issue a COPY command within a Process.
ProcessRunJob	Gives a user permission to issue a RUN JOB command within a Process.
ProcessRunTask	Gives a user permission to issue a RUN TASK command within a Process.
ProcessSubmit	Gives a user permission to issue a SUBMIT command within a Process.
Inherit Rights	The Inherit Rights flag.
TrusteeAssign	The Trustee Assign flag.
UpdateACL	The Update ACL flag.
FileAttributes	The File Attribute flag.
SNodeId	The Remote Node ID flag.
ExecutionPriority	Gives a user permission to change execution priority.
ProcessSend	The Process Send flag.
ProcessReceive	The Process Receive flag.
UpdateTranslation	Gives a user permission to update the translation table information.
DownloadDirectory [MAX_DIRECTORY_NAME+1]	The default download directory.
UploadDirectory [MAX_DIRECTORY_NAME+1]	The default upload directory.
ProcessDirectory [MAX_DIRECTORY_NAME+1]	The default Process file directory.
ProgramDirectory [MAX_DIRECTORY_NAME+1]	The default program file directory.

## MESSAGE\_STRUCT Structure

The MESSAGE\_STRUCT structure contains the Connect:Direct message information. Use this structure to retrieve the message information. It contains the unique message identifier.

### Structure

```

struct Message_Struct
{
    TCHAR MsgId[MAX_MESSAGE_ID+1];
}

```

```

int ConditionCode;
int Feedback;
TCHAR MsgText[MAX_MESSAGE_TEXT+1];
TCHAR MsgData[MAX_MESSAGE_DATA+1];
};
typedef struct Message_Struct MESSAGE_STRUCTURE;

```

## Members

Member	Description
MsgId [MAX_MESSAGE_ID+1]	The message identifier that uniquely identifies this message.
ConditionCode	The return code accompanying the message.
Feedback	Additional return code information.
MsgText [MAX_MESSAGE_TEXT+1]	The message text.
MsgData [MAX_MESSAGE_DATA+1]	Message substitution fields.

## NETMAP\_MODE\_SNA Structure

The NETMAP\_MODE\_SNA structure contains the Netmap SNA Mode information. This structure is part of the NETMAP\_MODE\_STRUCTURE for SNA modes.

### Structure

```

struct Netmap_Mode_Sna
{
    long lMaxRUSize;
    short MaxPacingSize;
    short MaxNetSessLimit;
};
typedef struct Netmap_Mode_Sna NETMAP_MODE_SNA;

```

## Members

Member	Description
lMaxRUSize	The maximum RU size.
MaxPacingSize	The maximum pacing size.
MaxNetSessLimit	The maximum net session limit.

## NETMAP\_MODE\_TCP Structure

The NETMAP\_MODE\_TCP structure contains the Netmap TCP/IP Mode information. This structure is part of the NETMAP\_MODE\_STRUCTURE for TCP/IP modes.

### Structure

```

struct Netmap_Mode_Tcp
{
    long lBufferSize;
    long lPacingSendCount;
    long lPacingSendDelay;
    char tcp_crc[4];
};

```

```
};
typedef struct Netmap_Mode_Tcp NETMAP_MODE_TCP;
```

## Members

Member	Description
lBufferSize	The buffer size.
lPacingSendCount	Pacing send count.
lPacingSendDelay	Pacing send delay.
char tcp_crc[4]	Whether TCP CRC checking is on.

## NETMAP\_NODE\_STRUCT Structure

The NETMAP\_NODE\_STRUCT structure contains the Netmap node information. Use this structure to retrieve and set the Netmap node information.

### Structure

```
struct Netmap_Node_Struct
{
    TCHAR Name[MAX_OBJECT_NAME_LEN+1];
    BOOL bDetail;
    int LongTermRetry;
    long lLongTermWait;
    int ShortTermRetry;
    long lShortTermWait;
    int MaxPNode;
    int MaxSNode;
    int DefaultClass;
    int RemoteOSType;
    TCHAR TcpModeName[MAX_OBJECT_NAME+1];
    TCHAR TcpAddress[MAX_TCP_ADDRESS+1];
    TCHAR SnaModeName[MAX_OBJECT_NAME+1];
    TCHAR SnaNetName[MAX_NET_NAME+1];
    TCHAR SnaPartnerName[MAX_PARTNER_NAME+1];
    TCHAR SnaTPName[MAX_TPNAME+1];
};
typedef struct Netmap_Node_Struct NETMAP_NODE_STRUCT;
```

## Members

Member	Description
Name [MAX_OBJECT_NAME_LEN+1]	The node name.
bDetail	Specifies detail-included flag.
LongTermRetry	Long-term retry interval.
lLongTermWait	Long-term wait interval.
ShortTermRetry	Short-term retry interval.
lShortTermWait	Short-term wait interval.
MaxPNode	The maximum number of local nodes.
MaxSNode	The maximum number of remote nodes.
DefaultClass	The default class.



Member	Description
RemoteOSType	Remote node operating system type.
TcpModeName [MAX_OBJECT_NAME+1]	The TCP/IP communications mode name.
TcpAddress [MAX_TCP_ADDRESS+1]	The node's TCP/IP address.
SnaModeName [MAX_OBJECT_NAME+1]	The SNA communications mode name.
SnaNetName [MAX_NET_NAME+1]	The SNA net name.
SnaPartnerName [MAX_PARTNER_NAME+1]	SNA partner name.
SnaTPName [MAX_TPNAME+1]	The TP name.

## NETMAP\_PATH\_STRUCT Structure

The NETMAP\_PATH\_STRUCT structure contains the Netmap path information. Use this structure to retrieve and set the Netmap path information.

### Structure

```

struct Netmap_Path_Struct
{
    TCHAR Name[MAX_OBJECT_NAME+1];
    BOOL bDetail;
    int Transport;
    int Adapter;
    BYTE Address[MAX_ADDRESS];
    char CustomQLLC[MAX_CUSTOM_ADDRESS+1];
    int Protocol;
    TCHAR SnaProfileName[MAX_PROFILE_NAME+1];
    TCHAR SnaLocalNetId[MAX_LOCALNETID+1];
    TCHAR SnaPUName[MAX_PUNAME+1];
    TCHAR SnaLUName[MAX_LUNAME+1];
    int SnaLULocAddr;
    int SnaLUSessLimit;
    int TCPMaxTimeToWait;
    int DialupHangon;
    char DialupEntry[MAX_DIALUP_ENTRY+1];
    char DialupUserId[MAX_OBJECT_NAME+1];
    char DialupPassword[MAX_OBJECT_NAME+1];
    TCHAR ModeName[MAX_OBJECT_NAME+1];
};
typedef struct Netmap_Path_Struct NETMAP_PATH_STRUCT;

```

### Members

Member	Description
Name [MAX_OBJECT_NAME+1]	The path name.
bDetail	The detail flag.
Transport	Transport type.
Adapter	Specifies the adapter.
Address [MAX_ADDRESS]	The adapter address.
CustomQLLC[MAX_CUSTOM_ADDRESS+1]	The custom or QLLC adapter address.
Protocol	The protocol type.
SnaProfileName[MAX_PROFILE_NAME+1]	The SNA profile name.
SnaLocalNetId [MAX_LOCALNETID+1]	The SNA local net ID.

Member	Description
SnaPUName [MAX_PUNAME+1]	The SNA PU name.
SnaLUName [MAX_LUNAME+1]	The SNA LU name.
SnaLULocAddr	The SNA LU local address.
SnaLUSessLimit	The SNA LU session limit.
TCPMaxTimeToWait	TCP maximum time to wait.
DialupHangon	Number of seconds to stay connected after dialup hangon completes.
DialupEntry[MAX_DIALUP_ENTRY+1]	Dialup entry name.
DialupUserid[MAX_OBJECT_NAME+1]	Dialup user ID.
DialupPassword[MAX_OBJECT_NAME+1]	Dialup password.
ModeName [MAX_OBJECT_NAME+1]	The mode name used by this path.

## PROCESS\_STRUCT Structure

The PROCESS\_STRUCT structure contains the Connect:Direct Process information. This structure is sent to the client from the Connect:Direct server upon accepting a Process for execution. It is also sent in response to a SELECT PROCESS command. It contains the Process name, Process number, and queue.

### Structure

```

struct Process_Struct
{
    TCHAR ProcessName[MAX_PROCESS_NAME+1];
    DWORD ProcessNumber;
    int ConditionCode;
    int Feedback;
    TCHAR MsgId[MAX_MESSAGE_ID+1];
    TCHAR MsgText[MAX_MESSAGE_TEXT+1];
    TCHAR MsgData[MAX_MESSAGE_DATA+1];
    time_t LogDateTime;
    time_t SchedDateTime;
    TCHAR SubmitNode[17];
    TCHAR Submitter[65];
    TCHAR PNode[17];
    TCHAR SNode[17];
    TCHAR Status[3];
    TCHAR Retain;
    TCHAR Hold;
    int Class;
    int Priority;
    int ExecPriority;
    TCHAR Queue[5];
    TCHAR Function[6];
    TCHAR StepName[9];
    TCHAR LocalNode;
    TCHAR FromNode;
    BOOL bStandardCompression;
    BOOL bExtendedCompression;
    BOOL bCheckpoint;
    BOOL bRestart;
    TCHAR SourceFile[MAX_FILENAME+1];
    TCHAR SourceDisp1;
    TCHAR SourceDisp2;
    TCHAR SourceDisp3;
    __int64 ByteCount;
    __int64 RecordCount;
    __int64 XmitBytes;
    long XmitRUs;
    TCHAR DestFile[MAX_FILENAME+1];
    TCHAR DestDisp1;

```

```

TCHAR DestDisp2;
TCHAR DestDisp3;
//SECURE_PLUS
BOOL bSecurePlusEnabled;
TCHAR EncAlgName[MAX_OBJECT_NAME];
BOOL bSignature;
};
typedef struct Process_Struct PROCESS_STRUCTURE;

```

## Members

Member	Description
ProcessName [MAX_PROCESS_NAME+1]	The Process name.
ProcessNumber	The Process number.
ConditionCode	The return code.
Feedback	Specifies additional return code information.
MsgId [MAX_MESSAGE_ID+1]	The message identifier field.
MsgData [MAX_MESSAGE_TEXT+1]	The message text field.
MsgData [MAX_MESSAGE_DATA+1]	The message substitution data.
LogDateTime	The logged time stamp.
SchedDateTime	The scheduled time stamp.
SubmitNode [17]	The submitter's node.
Submitter [65]	The submitter's user name.
PNode [17]	The primary node.
SNode [17]	The secondary node.
Status [3]	The current status.
Retain	The retain flag.
Hold	The hold flag.
Class	The class.
Priority	The current priority.
ExecPriority	The current execution priority.
Queue [5]	The current queue that contains this Process.
Function[6]	The function executing in the Process.
StepName [9]	The current step name.
LocalNode	The local node flag.
FromNode	The from node flag.
bStandardCompression	The standard compression indicator.
bExtendedCompression	The extended compression indicator.
bCheckpoint	The checkpointing enabled indicator.
bRestart	Restart indicator.
SourceFile [MAX_FILENAME+1]	The source file name.

Member	Description
SourceDisp1	The source displacement 1.
SourceDisp2	The source displacement 2.
SourceDisp3	The source displacement 3.
ByteCount	The total byte count.
RecordCount	The total record count.
XmitBytes	The sent byte count.
XmitRUs	The sent RU count.
DestFile[MAX_FILENAME+1]	The destination file name.
DestDisp1	The destination displacement 1.
DestDisp2	The destination displacement 2.
DestDisp3	The destination displacement 3.
bSecurePlusEnabled	The Secure+ enabled flag.
EncAlgName[MAX_OBJECT_NAME]	The effective encryption algorithm.
bSignature	Specifies the effective signature setting.

## NODE\_STRUCT Structure

The NODE\_STRUCT structure contains the Connect:Direct node information. This structure contains the node name, the login information, operating system information, and protocol information. This information is stored in the Registry and is sent to the client after successfully logging on.

### Structure

```

struct Node_Struct
{
    TCHAR Name[MAX_NODE_NAME_LEN+1];
    TCHAR CDName[MAX_NODE_NAME_LEN+1];
    TCHAR Server[MAX_OBJECT_NAME+1];
    long ApiVersion;
    long SecurePlusVersion;
    int CompLevel;
    int SelectedOSType;
    int OSType;
    int SubType;
    TCHAR Userid[MAX_OBJECT_NAME+1];
    TCHAR Password[MAX_OBJECT_NAME+1];
    BOOL bTemporary;
    BOOL bRememberPW;
    int Protocol;
    TCHAR TcpAddress[MAX_TCP_ADDRESS+1];
};
typedef struct Node_Struct NODE_STRUCT;

```

### Members

Member	Description
Name [MAX_NODE_NAME_LEN+1]	The Connect:Direct node alias name.
CDName [MAX_NODE_NAME_LEN+1]	The Connect:Direct node name.
Server [MAX_OBJECT_NAME+1]	The file server name.

Member	Description
ApiVersion	The API version.
SecurePlusVersion	The Secure+ version; value is 0 if Secure+ is not supported.
CompLevel	The KQV Communications Compatibility Level.
SelectedOSType	The user-selected operating system type.
OSType	The operating system type.
SubType	Specifies subtype information.
Userid [MAX_OBJECT_NAME+1]	The user name.
Password [MAX_OBJECT_NAME+1]	The user-defined password.
bTemporary	Specifies to hold the user information temporary.
bRememberPW	Specifies to save the password in the Registry.
Protocol	Protocol type.

## STATISTICS\_STRUCT Structure

The STATISTICS\_STRUCT structure contains the Connect:Direct statistics information for a Process. This structure is sent to the client as a result of a SELECT STATISTICS command.

### Structure

```

struct Statistic_Struct
{
    TCHAR ProcessName[MAX_PROCESS_NAME+1];
    DWORD ProcessNumber;
    int ConditionCode;
    int Feedback;
    TCHAR MsgId[MAX_MESSAGE_ID+1];
    TCHAR MsgText[MAX_MESSAGE_TEXT+1];
    TCHAR MsgData[MAX_MESSAGE_DATA+1];
    time_t LogDateTime;
    time_t StartDateTime;
    time_t StopDateTime;
    TCHAR Submitter[65];
    TCHAR SNode[17];
    TCHAR RecCat[5];
    TCHAR RecId[5];
};
typedef struct Statistic_Struct STATISTIC_STRUCT;

```

### Members

Member	Description
ProcessName [MAX_PROCESS_NAME+1]	The Process name.
ProcessNumber	The Process number.
ConditionCode	The return code.
Feedback	Additional return code information.
MsgId [MAX_MESSAGE_ID+1]	The message identifier field.
MsgText [MAX_MESSAGE_TEXT+1]	The message text field.

Member	Description
MsgData [MAX_MESSAGE_DATA+1]	Message substitution data.
LogDateTime	The logged time stamp.
StartDateTime	The start time stamp.
StopDateTime	The stop time stamp.
Submitter [65]	The submitter's user ID.
SNode [17]	The secondary node name.
RecCat [5]	The record category.
RecId [5]	The record identifier tag.

## TRACE\_STRUCT Structure

The TRACE\_STRUCT structure contains the trace information. Use this structure to retrieve the trace information.

### Structure

```

struct Trace_Struct
{
    TCHAR cMainLevel;
    TCHAR cCommLevel;
    TCHAR cCMgrLevel;
    TCHAR cPMgrLevel;
    TCHAR cSMgrLevel;
    TCHAR cStatLevel;
    TCHAR szFilesize[MAX_FILENAME+1];
    long cbFilesize;
    BOOL bWrap;
    BOOL bPNode;
    BOOL bSNode;
    int PNums[4];
    TCHAR PNames[4] [MAX_PROCESS_NAME+1];
    TCHAR DestNodes[4] [17];
};
typedef struct Trace_Struct TRACE_STRUCT;

```

### Members

Member	Description
cMainLevel	MAIN trace level.
cCommLevel	The COMM trace level.
cCMgrLevel	CMGR trace level.
cPMgrLevel	PMGR trace level.
cSMgrLevel	The SMGR trace level.
cStatLevel	STAT trace level.
szFilename[MAX_FILENAME+1]	The trace file name.
cbFilesize	The size of the trace file.
bWrap	Specifies whether to wrap when cbFile is reached.
bPNode	The PNODE trace flag.

Member	Description
bSNode	The SNode trace flag.
PNums[8]	Specifies an integer array of up to four Process numbers.
PNames[8] [MAX_PROCESS_NAME+1]	The string array of Process names.
DestNodes[8] [17]	The string array of destination node names.

## TRANSLATE\_STRUCT Structure

The TRANSLATE\_STRUCT structure contains the translation table information. Use this structure to retrieve and set the translation table information.

### Structure

```

struct Translate_Struct
{
    TCHAR Filename[MAX_OBJECT_NAME+1];
    BYTE Table[256];
    TCHAR MsgId[MAX_MESSAGE_ID+1];
    int ConditionCode;
    int Feedback;
    TCHAR MsgText[MAX_MESSAGE_TEXT+1];
    TCHAR MsgData[MAX_MESSAGE_DATA+1];
};
typedef struct Translate_Struct TRANSLATE_STRUCT;

```

### Members

Member	Description
FileName [MAX_OBJECT_NAME+1]	The name of the file where the translation information is stored.
Table [256]	The actual translation table information.
MsgId[MAX_MESSAGE_ID+1]	The message identifier that uniquely identifies a message.
ConditionCode	The return code that accompanies a message.
Feedback	Additional return code information.
MsgText[MAX_MESSAGE_TEXT+1]	The message text.
MsgData[MAX_MESSAGE_DATA+1]	The message substitution field.

## Return Codes

### C++ Class and the C API Functions Return Codes

#### CDAPI.H Return Code Values

This table describes the return code values defined in CDAPI.H.

Name	Description
CD_NO_ERROR	No error detected.

<b>Name</b>	<b>Description</b>
CD_ENDOFDATA	No more data available.
CD_PARM_ERROR	Invalid parameter detected.
CD_INITIALIZE_ERROR	Initialization failed or initialization has not been performed.
CD_CONNECT_ERROR	Error occurred during attach processing.
CD_CONNECT_CANCELLED	Attach operation cancelled by the user.
CD_CONNECTED_ERROR	Invalid Connect:Direct server name.
CD_DISCONNECT_ERROR	Connect:Direct server disconnected from the client.
CD_NODENAME_ERROR	The Name field not set and the default not found.
CD_USERID_ERROR	Invalid user ID specified.
CD_ADDRESS_ERROR	Invalid TCP/IP address.
CD_PROTOCOL_ERROR	Invalid or unsupported protocol specified.
CD_HANDLE_ERROR	Invalid handle.
CD_HANDLE_TYPE_ERROR	The wrong handle type specified.
CD_LOGON_ERROR	Error while logging on to the Connect:Direct server. The user ID or password may be invalid.
CD_DIALOG_ERROR	Dialog box not created correctly.
CD_CANCEL	An error occurred creating the dialog box or retrieving the entered information.
CD_BUSY_ERROR	Operation failed. Connection is currently busy.
CD_IDLE_ERROR	Operation failed. Connection is currently idle.
CD_KQV_ERROR	Invalid KQV stream detected.
CD_NOT_FOUND	Object not found.
CD_ALREADY_EXISTS	Object already exists.
CD_ALLOCATE_ERROR	Allocation error occurred.
CD_NODE_ERROR	Invalid network map node.
CD_PARSER_ERROR	Parser detected an error.
CD_ACCESS_DENIED	Object access denied.
CD_SEND_ERROR	Error while sending error.
CD_RECEIVE_ERROR	Error while receiving error.
CD_CONNECTION_ERROR	A connection error occurred.
CD_REGISTRY_ERROR	An error occurred while opening the Registry.
CD_TIMEOUT_ERROR	Time-out value was reached.
CD_BUFFER_ERROR	The buffer is not big enough to hold all of the items in the list.
CD_COMMAND_ERROR	The command was not recognized.
CD_PROCESS_ERROR	The Process status is HE, held in error.
CD_UNDEFINED_ERROR	An unknown exception.



Name	Description
CD_NOT_SUPPORTED	An unknown exception.



---

# Chapter 7. .Net SDK User Guide

---

## Connect:Direct for Microsoft Windows .Net SDK Overview

---

The IBM Connect:Direct for Microsoft Windows .Net SDK allows system programmers to extend the capabilities of the Connect:Direct for Microsoft Windows environment. It supports any version of the .Net framework from Microsoft using any .Net supported programming language, including C#, VB.Net and J#.

Connect:Direct for Microsoft Windows .Net SDK uses preconfigured connection settings. For information, see *Editing Connections Settings* in the *IBM Connect:Direct for Microsoft Windows SDK Programmer Guide*.

The following files are provided:

- ConnectDirectSdk.dll is a managed dll that interfaces the .Net managed program to the Connect:Direct for Microsoft Windows CdCore.dll. Copy this file to a folder in your executable path.
- CdCore.dll interfaces to the Connect:Direct for Microsoft Windows server. Copy this file to a folder in your executable path.
- ConnectDirectSdk.xml is the help file that provides autocompletion and parameter help.

---

## Sample Programs

---

Sample source code projects help you understand how to use the .Net SDK. To run the samples, place the CdCore.dll and ConnectDirectSdk.dll in your executable path. You can copy these files to the same directory as the sample executables.

The sample programs include:

Sample Type	Program Name	Description
VB.Net	VbDotNetSample1	Console program that connects to a node, submits a Process from a file, and displays statistics for the Process.  Change the \$todo tags in Module1.vb to valid variables for your Connect:Direct for Microsoft Windows node.
C#.Net	DotNetSample1 DotNetSample2	Change the \$todo tags in the SDKInterface.cs files for each sample to valid values for the Connect:Direct for Microsoft Windows node.  DotNetSample1 connects to a node, issues a select process, then displays the Process information returned.  DotNetSample2 – Connects to a node, submits a Process and displays the Process and statistics information.

---

## Add the .Net Class Interface

---

### About this task

To use the Connect:Direct .Net SDK, add the ConnectDirectSdk.dll as a reference in your Visual Studio project.

To add the dll:

## Procedure

1. Select **Project >Add Reference** from the menu.
2. Select Browse and search for the ConnectDirectSdk.dll file.  
Browse to the *C:\Program Files\IBM\Connect Direct v6.1.0\SDK.Net\Sdk\_Files\Release location*.
3. Highlight ConnectDirectSdk.dll and click **OK**.
4. Do one of the following to import the ConnectDirectSdk namespace:
  - In Vb.Net, add the following command to your source modules:

```
Imports ConnectDirectSdk
```

- In C#.Net, add the following to your source modules:

```
Using ConnectDirectSdk
```

## About Classes

---

Classes are provided to help configure your environment.

The Node class is the main interface to the Connect:Direct for Microsoft Windows server. It contains the high-level Connect:Direct functions. Use it to connect to a Node, submit Processes, and select statistics. Most access to the Connect:Direct for Microsoft Windows server is through the Node object. The Node object creates and removes the connection to the Connect:Direct for Microsoft Windows server. Connections are shared and reused as different requests are made.

The Process class allows you to retrieve information about Processes you submit or that are in the TCQ. It contains all of the criteria returned from a Submit or SelectProc method call.

The Statistic class allows you to retrieve statistic records from the TCQ. It represents a group of records in the statistics database. They are returned from a SelectStat method call.

## Connect to a Connect:Direct for Microsoft Windows Node

---

The Connect:Direct node name and connection information is set at object creation using the Node constructor. If a parameter is not supplied (NULL pointer), the default value is read from the Registry.

During construction, the Node object tries to connect to the physical Connect:Direct node, using the protocol information in the Registry. If the connection fails, an exception is generated

In the following constructor, stNode is required. stUser and stPass are optional. stPass is ignored if stUser is not provided.

```
Node(String stNode, String stUser, String stPass)
```

In the following constructor, stLcuFile is required. This is the file spec for an LCU file that contains the login information.

```
Node(String stLcuFile)
```

## Disconnect Nodes

---

Use the DisconnectAll method to disconnect from all Nodes.

```
bool DisconnectAll()
```

## Submit Processes

---

Use `Submit` and `SubmitFile` to submit Processes to a Node. These methods automatically create a Process object and associate it with the Node for the Submit.

Below is the standard `SubmitFile` method. `stFileName` is required and is the file specification of the Process to submit.

```
void SubmitFile(String stFileName)
```

The `SubmitFile` method allows more control of the submitted Process.

- `stFileName` is required and defines the requirements for the Process.
- `holdOverride` places the Process in the Hold queue.
- `startTime` specifies when to run the Process.
- `symbolics` define the substitution parameters to apply to the Process.

```
void SubmitFile(String stFileName, Hold holdOverride, String startTime, Dictionary<String, String> symbolics)
```

The `Submit` method is very similar to the `SubmitFile` method but instead of passing the file name of the Process to submit, you pass `stText` which is the text of a Process to submit.

```
void Submit(String stText, Hold holdOverride, String startTime, Dictionary<String, String> symbolics)
```

## Manage Processes

---

The Node object provides several methods to manage Processes. You can view, change and delete Processes, place a Process on Hold, or release it from Hold. Each method returns Process information in the `ProcessList` property of the Node class for each Process that was selected or changed.

The following `SelectProc` method allows you to retrieve a list of all Processes from the TCQ:

```
void SelectProc()
```

The following `SelectProc` method retrieves Processes from the TCQ whose Process name matches `stName`.

```
void SelectProc(String^ stName)
```

The following `SelectProc` method retrieves Processes from the TCQ whose Process number matches `nNumber`.

```
void SelectProc(int nNumber)
```

The following `SelectProc` method retrieves Processes from the TCQ whose Process name matches any name in the array `arrayNames`.

```
void SelectProc(array<String^>^ arrayNames)
```

The following SelectProc method retrieves Processes from the TCQ whose Process number matches a number in arrayNumbers.

```
void SelectProc(array<int>^ arrayNumbers)
```

The following HoldProc method places a Process in the TCQ on HOLD. pProcess is a Process object.

```
void HoldProc(Process^ pProcess)
```

The following ReleaseProc method releases a Process from the HOLD and allows it to run. pProcess is a process object.

```
void ReleaseProc(Process^ pProcess)
```

The following ReleaseProc method releases a Process that is on HOLD. nNumber is the Process number of the Process, stPNode is the primary node of the Process, and stUserid is the User ID of the Process.

```
void ReleaseProc(int nNumber, String^ stPNode, String^ stUserid)
```

The following DeleteProc method deletes a Process from the TCQ. pProcess is a Process object.

```
void DeleteProc(Process^ pProcess)
```

The following DeleteProc method deletes a Process from the TCQ. nNumber is the Process number to delete, stPNode is the primary node of the Process, and stUserid is the User ID of the Process.

```
void DeleteProc(int nNumber, String^ stPNode, String^ stUserid)
```

## Retrieve Statistics

---

Use SelectStat methods to retrieve statistics from the stats database. Statistics are returned in the StatsList property of the Node class.

The following SelectStat method retrieves all statistic records.

**Note:** The list could be large depending on how many days of records are kept in the database.

```
void SelectStat()
```

The following SelectStat method retrieves all statistic records for a specific Process. pProcess is the Process object to retrieve the stats for.

```
void SelectStat(Process^ pProcess)
```

The following SelectStat method retrieves all statistic records within a specified time range. dtBegin identifies the beginning time and dtEnd is the ending time. The time is in the format MM/DD/YYYY hh:mm:ss AM|PM.

```
void SelectStat(String^ dtBegin, String^ dtEnd)
```

## Node Properties

---

Following are the node properties returned:

- ApiVersion - API version of the node as a long
- CDName - Connect:Direct node name sent to the client after successfully logging on
- Name - Alias node name passed in the constructor
- OSSubType - Operating system sub-type (additional information) of the node
- OSType - Operating system type of the node
- ProcessEntry - Process from a Submit call
- ProcessList - Array of Processes
- SecurePlusSupported - Indicates if the node supports IBM Connect:Direct Secure Plus
- SecurePlusVersion - Connect:Direct Secure Plus version as a long
- Server - File server name where the Connect:Direct node is running
- StatsList - Array of Stat messages from a SelectStat call
- Userid - User ID used to log in to the node

## Process Class

---

The Process class contains Process criteria returned from a SUBMIT or SELECT PROCESS method. Processes are submitted using the Node.Submit or Node.SubmitFile method.

### Method to Wait for Process Completion

The following WaitForCompletion method blocks the current thread until the Process exits all queues on the Connect:Direct server, including error queues. It waits indefinitely.

```
void WaitForCompletion()
```

The following WaitForCompletion method blocks the current thread until the Process exits all queues on the Connect:Direct server, including error queues, or until the timeout period expires. timeout is in milliseconds.

```
void WaitForCompletion(long timeout)
```

## Process Properties

---

Following is a list of the Process properties:

- ByteCount - Returns the Bytes read from the file as a long
- Checkpoint - Returns the Checkpointing Enabled flag
- Class - Returns the session class property as a String
- ConditionCode - Returns the Return Code as an int
- DestDisp1 - Returns the Destination file disposition parameter 1 as a char
- DestDisp2 - Returns the Destination file disposition parameter 2 as a char
- DestDisp3 - Returns the Destination file disposition parameter 3 as a char
- DestFile - Returns the Destination File Name as a string

- ExecPriority - Returns the Current Execution Priority as a String
- ExtendedCompression - Returns the Extended Compression flag
- Feedback - Returns the Additional Return Code Information as an int
- FromNode - Returns the From Node flag
- Function - Returns the Current Function Executing as a string
- Hold - Returns the Hold flag as a char
- LocalNode - Returns the Local Node indicator flag
- LogDateTime - Returns the Logged Timestamp
- MsgData - Returns the Message Substitution Data as a string
- MsgId - Returns the Message Identifier field as a string
- MsgText - Returns the Message Text field as a string
- Name - Returns the Process Name as a string
- Number - Returns the Process Number as an int
- PNode - Returns the Primary Node Name as a string
- Priority - Returns the Current Priority as in int
- Queue - Returns the Process Queue as a string
- RecordCount - Returns the Records read/written as a long
- Restart - Returns the Restart flag
- Retain - Returns the Retain flag as a char
- SchedDateTime - Returns the Scheduled Timestamp
- SecureEnabled - Returns the Connect:Direct Secure Plus enabled flag
- SecureProtocol - Returns the Connect:Direct Secure Plus Protocol as a string
- Signature - Returns the Connect:Direct Secure Plus effective Signature setting
- SNode - Returns the Secondary Node Name as a string
- SourceDisp1 Returns the Source Disposition 1 as a char
- SourceDisp2 - Returns the Source Disposition 2 as a char
- SourceDisp3 - Returns the Source Disposition 3 as a char
- SourceFile - Returns the Source File Name as a String
- SSLCipherSuite - Returns the Connect:Direct Secure Plus SSL Cipher Suite as a string
- StandardCompression - Returns the Standard Compression flag
- Status - Returns the Current Status as a string
- StepName - Returns the Current Stepname as a string
- SubmitNode - Returns the Submitter Node Name as a String
- Submitter - Returns the Submitter User ID as a string
- XmitBytes - Returns the Bytes sent/received count as a long
- XmitRUs - Returns the RUs sent/received as a long

## Statistic Class

---

The Statistic class represents a group of records in the statistics database. They are returned by a SelectStat method call.

### Audit Information

The GetAuditField method returns the value of the field requested from the Stats Audit Information. Audit data in Stats records is optional and Stat records can have different audit fields available. stField is the



name of the audit field you request information for; stValue is the value of the field requested. This method returns TRUE if the audit field is found and FALSE if not.

```
BOOL GetAuditField(String^ stField, String^% stValue)
```

## Statistic Properties

---

Following are the statistics properties:

- ConditionCode - Returns the Return Code
- Feedback - Returns Additional Return Code information
- LogDateTime - Returns the Logged Timestamp
- MsgData - Returns the Message Substitution Data as a string
- MsgId - Returns the Message Identifier field as a string
- MsgText - Returns the Message Text field as a string
- ProcessName - Returns the Name of the process
- ProcessNumber - Returns the Process number
- RecCat - Returns the Record Category
- RecId - Returns the Record Identifier tag
- SNode - Returns the Secondary Node Name
- StartDateTime - Returns the Start Timestamp
- StopDateTime - Returns the Stop Timestamp
- Submitter - Returns the User Id of the submitter



## Notices

---

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

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

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

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

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

#### COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown in the next column.

© 2015.

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2015.

## Trademarks

---

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

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

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise®, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Commerce™, Information Broker®, and Integrator® are trademarks, Inc., an IBM Company.

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

## Terms and conditions for product documentation

---

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

### **Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

### **Personal use**

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

### **Commercial use**

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

### **Rights**

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

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

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

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,

INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.





Part Number:  
Product Number: 5655-X01

(1P) P/N: