Sterling Connect:Direct for UNIX

Administration Guide

Version 4.1

Last updated: 17 November 2015

Sterling Connect:Direct for UNIX

Administration Guide

Version 4.1

Last updated: 17 November 2015

Note
Before using this information and the product it supports, read the information in "Notices" on page 59.

This edition applies to version 4.1 of IBM Sterling Connect:Direct and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Maintaining configuration files

Configuration files

Configuration files define the operating environment for IBM® Sterling Connect:Direct®. The following configuration files are created during the customization procedure:

- Initialization parameters file
- Client configuration parameters file
- Network map file
- Two access files: userfile.cfg and sysacl.cfg

After the initial customization, you can modify these files, if necessary.

A configuration file is a text file composed of records. A record is a single logical line. A logical line is one or more physical lines that can be continued with the backslash (\) character. In the sample format below, physical lines 4 and 5 illustrate a logical line. Line 4 ends with a backslash (\) character, to indicate that the line is continued on the next physical line. Line 1 of the sample begins with a pound (#) sign. The pound sign indicates this line contains a comment.

A record consists of a record name and one or more parameter pairs. A parameter pair is a parameter name and parameter value. Line 2 contains the record name, **ndm.path**. Line 2 also contains the parameter pair, path and /ndm/users/c, where the parameter name is path and the parameter value is /ndm/users/c. The parameter pair is bound by colons (:) and separated by an equal sign (=) in the following format. The following example displays a complete record, where **ndm.path** is the record name, **path** is the parameter name, and /ndm/users/c is the parameter value:

```
ndm.path:path=/ndm/users/c:
```

Record names and parameter names are not case sensitive. Parameter values are case sensitive.

Lines 7 through 23 illustrate a longer logical record. Line 7 contains the record name **local.node** followed by an optional colon (:) and a backslash (\) character. All lines between 7 and 23 end with a backslash (\) character. Line 23 does not contain a backslash (\) character, to indicate the end of the record.

Sample format of a configuration file

The following table displays a portion of the initialization parameters file to illustrate the format of Sterling Connect:Direct configuration files:

Line	Contents	Notes
1	#Miscellaneous Parameters	# indicates a comment
2	ndm.path:path=/ndm/users/c:	record name=**ndm.path**, parameter=**path**, value= /ndm/users/c

Line	Contents	Notes
3	`proc.prio:default=8:`	record name=**proc.prio**, parameter=**default**, value= 8
6	`#Local Sterling Connect:Direct connection information`	# indicates a comment
7	`local.node:\`	record name=**local.node**
13	.	
. . .	.	
21	.	
22	`:tcp.api=rusty;3191:\`	parameter= **tcp.api**, value= `rusty;3191`
23	`:tcp.api.bufsize=32768:`	parameter= **tcp.api.bufsize**, value= 32768

Configuration files allow duplicate but not identical records, in some cases. For example, you can define more than one remote node information (**rnode.listen**) record in the initialization parameters file.

Modifying configuration files

Before you begin

You can modify Sterling Connect:Direct configuration files using any text editor or create a new configuration file using the cdcust command provided with Sterling Connect:Direct for UNIX.

- Modifying configuration files with a text editor—You can modify Sterling Connect:Direct configuration files with any text editor, such as vi editor.
- Creating configuration files with **cdcust**—Type the following command to start the customization procedure, where *d_dir* is the Sterling Connect:Direct for UNIX path name:

```
$ d_dir/etc/cdcust
```

Chapter 2. Maintaining the initailization parameters

Initialization parameters file

Initialization parameters determine various Sterling Connect:Direct settings that control system operation. The initialization parameters file is created when you install Sterling Connect:Direct for UNIX and can be updated as needed.

You can modify Sterling Connect:Direct initialization parameters file using any text editor. Before changing a value in the file, first shut down the Sterling Connect:Direct server. After you change a value and save the file, restart the server. Restarting the server validates the new values and generates an error message if a value is invalid.

You can also use the Sterling Connect:Direct Browser User Interface to perform some of the procedures related to the initialization parameters file. To learn more about the Sterling Connect:Direct Browser User Interface, see the documentation related to that product in the IBM Documentation Library. If you use Sterling Connect:Direct Browser User Interface to update parameters in the Local Node Connection Record, you do not have to stop and restart the server.

Contents of the initialization parameters file

The initialization parameters file resides in *d_dir*/ndm/cfg/*cd_node*/initparm.cfg, where *d_dir* is the destination directory where Sterling Connect:Direct for UNIX is installed and *cd_node* is the node name.

The initialization parameters file contains records. Each record includes parameters to define the attributes of the record. The records are summarized as follows:

- Miscellaneous parameters—Provide miscellaneous information including the name of the Sterling Connect:Direct for UNIX node; the location of Sterling Connect:Direct for UNIX, the Pluggable Authentication Modules (PAM) service configuration file, and the shared work area for SNODE work files; the default Process priority; and whether commands with special characters are restricted in the run directory.
- Remote node connection information—The **rnode.listen** record includes parameters to monitor inbound connections.
- Transmission Control Queue (TCQ) information—The **tcq** record defines how long a Process is held in error before being deleted.
- Global copy parameters—The **copy.parms** record defines default parameters used by the Copy operation including checkpoint parameters, file size limitations, translation table information, exception handling, CRC checking, file allocation retry parameters, and compression options.
- Global run task parameters—The **runtask.parms** record defines a parameter to define the restart option.
- Statistics file information—The **stats** record includes parameters to define default statistics file information including file size limitations, the type of information to write to the statistics file, and how long to maintain statistics files before archiving them.
- Server authentication information—The **authentication** record parameters to authenticate the server.

- User exit parameters—The **user.exits** record defines the programs used during a user exit procedure.
- Firewall navigation information—The **firewall.parms** record defines the ports or range of ports to use for outbound sessions when a server operates behind a firewall.

Sample initialization parameters file

The following example shows how some of these parameters are specified:

```
# Miscellaneous Parameters
ndm.path:path=/sci/users/mscarbro/cd4000:\
        :snode.work.path=/sci/users/mscarbro/cd4000/shared:

ndm.node:name=mws_joshua_4000:
ndm.pam:service=cdlogin:
ndm.quiesce:quiesce.resume=n

proc.prio:default=10:
restrict:cmd=y

# TCQ information
tcq:\
 :max.age=8:

# Global copy parameters.
copy.parms:\
 :ckpt.interval=2M:\
 :ulimit=N:\
 :xlate.dir=/sci/users/mscarbro/cd4000/ndm/xlate:\
 :xlate.send=def_send.xlt:\
 :xlate.recv=def_recv.xlt:\
 :continue.on.exception=y:

# Global runtask parameters.
runtask.parms:\
 :restart=y:

# Stat file info.
stats:\
 :file.size=1048576:\
 :log.commands=n:\
 :log.select=n:


# Authenticator
authentication:\
 :server.program=/sci/users/mscarbro/cd4000/ndm/bin/ndmauths:\
 :server.keyfile=/sci/users/mscarbro/cd4000/ndm/security/keys.server:

# user exit information
user.exits:\
 :security.exit.program=:\
 :file.open.exit.program=:\
 :stats.exit.program=:

# Remote CDU nodes
rnode.listen:\
 :recid=rt.sles96440:\
 :comm.info=0.0.0.0;9974:\
 :comm.transport=udt33:

# Secure+ parameters
secure+:\
 :certificate.directory=/home/nis02/jlyon/certs: \
 :s+cmd.enforce.secure.connection=n:
```

Updating miscellaneous records

You can update various parameters in the miscellaneous records that Sterling Connect:Direct uses. Required parameters are displayed in bold.

Path record

The ndm.path record identifies the path to Sterling Connect:Direct files. The following table describes the parameter available for this record:

Parameter	Description	Value
path	The path to all Sterling Connect:Direct subdirectories and files.	path specification

SNODE work path parameter

The **snode.work.path** parameter is part of the **ndm.path** record and identifies the path to the shared work area for SNODE work files on a cluster file system (not an NFS). This optional parameter provides a means to share SNODE work files among nodes in a load balancing environment. SNODE return code files (steprc files) and **copy** checkpoint information are created in this area when the **snode.work.path** parameter is specified. The following table describes the **snode.work.path** parameter:

Parameter	Description	Value
snode.work.path	The path to the shared work area for SNODE work files. **Note:** Specify the same path for all nodes in a cluster.	path specification

Node name record

The **ndm.node** record identifies the name of the Sterling Connect:Direct node. The following table describes the parameter available for this record:

Parameter	Description	Value
name	The name of the node.	The maximum length is 16 bytes. If a node name is longer, the name will be truncated.

PAM service record

The **ndm.pam** record identifies the PAM service configuration file used to authenticate the user authority for Sterling Connect:Direct Processes. If the service initialization parameter is defined and if PAM is installed on the Sterling Connect:Direct server, PAM is used to authenticate users for service-providing applications.

The service name required is typically defined in the `/etc/pam.conf` file for AIX, Solaris and HP operating systems, or defined and named by a file in the `/etc/pam.d` directory for Linux operating systems. Your system might also have a man page for PAM that provides further details.

The following table describes the parameter available for this record:

Parameter	Description	Value
service	PAM service configuration file name	File name

Quiesce/resume record

The **ndm.quiesce** record specifies whether Sterling Connect:Direct is operating in a "test" mode. Use this record in conjunction with the NDMPXTBL table to enable

the test mode. If you enable the **quiesce.resume** parameter, you must have an NDMPXTBL parameter table updated for your environment in the installation ndm/cfg/<*nodename*> directory. For more information on the test mode and the NDMPXTBL table, see Processing Flow of the Test Mode.

The following table describes the parameter available for this record:

Parameter	Description	Value
quiesce.resume	Enables/disables the test mode for Sterling Connect:Direct.	y | n y—Enables the test mode. n—Disables the test mode. The default is n.

Priority record

The **proc.prio** record identifies the default value of the Process priority. The following table describes the parameter available for this record:

Parameter	Description	Value
default	The default value of the Process priority.	1–15. The default value is 10. 15 is the highest priority.

Restrict record

If a run directory restriction is defined in the user configuration file (userfile.cfg), the restrict record determines if commands containing certain special characters are allowed. For more information on the userfile.cfg file, see Local User Information Record Format and Remote User Information Record. The following parameter is available for this record:

Parameter	**Description**	**Values**
cmd	Determines if commands with certain special characters are allowed.	y | n y—Restricts the ability to use commands with any of the following special characters: ; & ‘ | n—Does not restrict allowed commands.

Remote node connection record

The **rnode.listen** record contains parameters used by the local node to monitor inbound connection requests. You can modify the IP address and port number in the **rnode.listen** record while the server is running. However, you must recycle the server before the change is active. The following table describes the remote node connection parameters:

Parameter	**Description**	**Values**
recid	A unique identifier of an **rnode.listen** record.	A text string

Parameter	Description	Values
comm.info	The information needed to monitor connection requests from remote nodes using TCP/IP or LU6.2. This parameter is required. • For TCP/IP connections, specify the host name or the IP address and port number. If specifying an IP address and port, separate parameters with a semicolon (;). Separate multiple addresses/host names with a comma, for example: 10.23.107.5;1364, fe00:0:0:2014::7;1364, msdallas-dt;1364 For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports. • For LU6.2 connections, identify the profile name to identify the name of an SNA configuration profile for the remote connection. Sterling Connect:Direct generates the default name, hostl1, during the customization procedure. For AIX SNA, hostl1 refers to the side information profile name. For HP SNA, SunLink SNA, and Brixton SNA, hostl1 refers to the SNA profile file located in the same directory as the configuration files.	For TCP/IP connections, specify the host name or the IP address and port number: 10.23.107.5;1364 Separate multiple IP/host addresses with a comma (,): fe00:0:0:2014::7;1364, msdallas-dt;1364 A space can be added after the comma for readability. Set the IP address to monitor a specific adapter or to 0.0.0.0, to monitor all adapters. The default port is 1364. For LU6.2 connections, specify a profile name, up to 8 characters.
comm.transport	The transport protocol for the remote node.	tcp | lu62 | blklu62 | udt33 tcp—For TCP/IP connections lu62—For AIX SNA LU6.2 connections blklu62—For other LU6.2 connections udt33—For UDT connections

Transmission Control Queue record

The tcq record provides information pertaining to the Transmission Control Queue (TCQ). The following parameter is available for this record:

Parameter	Description	Value
max.age	The maximum number of days a Process with Held-in-Error status remains in the TCQ before it is automatically deleted.	A 3-digit decimal number. Sterling Connect:Direct does not automatically delete Processes when max.age=0. The default is 8 days.

Sterling Connect:Direct Secure Plus record

The Sterling Connect:Direct Secure Plus record (Secure+ record) provides information pertaining to remote configuration of Sterling Connect:Direct Secure Plus from the Sterling Connect:Direct client API. This record is not included in the initparm.cfg file by default. You must manually add the Secure+ record to the initparm.cfg file. The following parameters are available for this record:

Parameter	Description	Value
certificate.directory	Specifies a default certificate directory for Secure+ commands issued from the Sterling Connect:Direct client API. If the certificate directory is not configured, the default directory created during installation is used.	Directory path name
s+cmd.enforce.secure. connection	Specifies whether Secure+ commands are accepted from the Sterling Connect:Direct client API on unsecure connections.	y | n y—Commands from unsecure connections are not accepted. The default is y. n—Commands from unsecure connections are accepted

Global Copy record

The Global Copy record called **copy.parms** provides default information for the Sterling Connect:Direct copy operation. The ecz parameters are only used when extended compression is defined in a Process. The following parameters are available for this record:

Record	Description	Value
ckpt.interval	The default number of bytes transmitted in a copy operation before a checkpoint is taken. Following is a list of the maximum number of digits for each byte interval: no—No checkpointing nnnnnnnn—Up to an 8-digit decimal nnnnnnnnK—Up to an 8-digit decimal, where K denotes 1024 bytes nnnnnnnnM—Up to an 7-digit decimal, where M denotes 1048576 bytes nnnnG—Up to an 4-digit decimal, where G denotes 1073741824 bytes	The maximum possible value is 1 terabyte (TB). The normal value is 2MB.
ulimit	The action taken when the limit on a user output file size is exceeded during a copy operation.	n—Ignores the limit. n is the default value. y—Recognizes the user file size limit. If this limit is exceeded during a copy operation, the operation fails.
xlate.dir	The name of the directory containing the translation tables.	Any valid directory. The default path is *d_dir*/ndm/xlate.
xlate.send	The default translation table used when sending data to a remote node.	Any valid directory. The default file name is def_send.xlt.
xlate.recv	The name of the default translation table used when copying data from a remote node.	The default file name is def_recv.xlt in the directory defined in the xlate.dir parameter.

Record	Description	Value
continue.on.exception	The method to use to handle an exception condition in a Process. If a step fails due to a STOP IMMEDIATE or FLUSH exception issued on the remote node, the Process is placed in the Hold HE queue, regardless of the value of this parameter.	y—Continues Processing with the next step. n—Places a Process in the Hold queue with a value of HE. The default is y.
ecz.compression.level	Sets the compression level.	1–9. The default is 1. 1—The fastest but offers the least degree of compression. 9—Provides the greatest degree of compression but is the slowest.
ecz.memory.level	How much virtual memory to allocate to maintaining the internal compression state.	1–9. The default is 4. 1—Uses the least memory and 9 uses the most memory.
ecz.windowsize	The size of the compression window and history buffer. The larger the window, the greater the compression. However, increasing the window uses more virtual memory.	Valid values are 9–15. The default is 13.
retry.codes	The codes to recognize as a file allocation retry attempt. File allocation retry enables a Process with a file allocation or open error on either the local or remote node to run the Process again, beginning at the copy step where the error occurred. This feature supports the ability to retry a Process that failed when a file is already in use. When a file allocation or open error occurs on either the local or remote node, the PNODE searches for the error or message ID in the retry.codes and retry.msgids parameters. If the error code or message ID is found, the Process is retried. Since error codes can vary from one operating system to another and the same error code can have different meanings, use message IDs to identify retry conditions when communicating between two different platforms. You can perform retry attempts based on codes only, IDs only, or a combination of the two. When a retry condition is detected, the session is terminated cleanly and the Process is placed in the Timer queue.	Any valid error code

Record	Description	Value
retry.msgids	Identifies the message IDs to use to support a file allocation retry attempt. Since error codes can vary from one operating system to another and the same error code can have different meanings, use message IDs to identify retry conditions when communicating between two different platforms. When a file allocation or open error occurs on either the local or remote node, the PNODE searches for the message ID in the retry.msgids parameters. If the message ID is found, the Process is retried. You can perform retry attempts based on codes only, message IDs only, or a combination of the two. When a retry condition is detected, the session is terminated cleanly and the Process is placed in the Timer queue.	Any of the valid file allocation retry messages.
tcp.crc	Globally turn on or off the CRC function for TCP/IP Processes.	y | n y—Turns on the CRC function globally. n—Turns off the CRC function globally. The default is n.
tcp.crc.override	Determines whether netmap remote 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 y—Allows netmap remote node and Process statement overrides for CRC checking. n—Prevents netmap remote node and Process statement overrides for CRC checking. The default is n.
strip.blanks	Determines whether trailing blank characters are stripped from the end of a record. If strip.blanks is not defined in the initialization parameter, the default value of i is used.	y | n | i y—Strips blanks from the end of a record n—Does not strip blanks from the end of a record i—Setting for strip.blanks is determined by the default value of the remote node type as follows: • z/OS, VM, VSE, and i5OS—y • All other platforms—n
insert.newline	Arbitrarily appends an LF character at the end of each record when receiving a datatype=text file. By default, an LF character is not appended if one already exists at the end of a record.	y | n y—Arbitrarily appends an LF character n—Appends an LF character if needed

Global Run Task record

The Global Run Task record called runtask.parms is used if the pnode and snode cannot resynchronize during a restart. If a Process is interrupted when a run task on an SNODE step is executing, Sterling Connect:Direct attempts to synchronize the previous run task step on the SNODE with the current run task step. If synchronization fails, Sterling Connect:Direct reads the **restart** parameter to determine whether to perform the run task step again. The following parameter is available for this record:

Parameter	Description	Value
restart	If processing is interrupted when a run task on an SNODE step is executing and if synchronization fails after a restart, Sterling Connect:Direct reads the **restart** parameter to determine whether to perform the run task step again. Set this parameter in the initialization parameters file of the SNODE. **Note:** When a load balancing cluster is used and the snode.work.path is specified, the **restart** parameter takes effect only when resynchronization fails.	y | n y—The run task program runs again. The default is **y**. n—The Process skips the run task step.

Statistics file information record

The statistics file information record called **stats** defines the statistics facility. The following parameters are available for this record:

Parameter	Description	Value
file.size	The maximum size in bytes of an individual statistics data file. The statistics file name is written in the format of *Syyyymmdd*.ext, where *yyyy* indicates year, *mm* indicates month, and *dd* indicates day. The extension (ext) begins as 001. When a statistics file reaches the defined size within a 24-hour period, a new file is created with the same file name. The extension value is incremented by one.	nnnnnnnn, nnnnnnnnK, nnnnnnnM, or nnnnG—Establishes a default output file size limit for the statistics files. K denotes 1024 bytes. M denotes 1048576 bytes. G denotes 1073741824 bytes. The maximum value you can specify is 1 TB.
log.commands	Determines whether commands are written to the statistics file. If you want to log all commands except the select statistics and select process commands, set this parameter to y and the **log.select** parameter to n.	y | n y—Commands are written to the statistics file. n—Commands are not written to the statistics file. The default is n.

Parameter	Description	Value
log.select	Specifies whether Sterling Connect:Direct creates a statistics record when a select process or select statistics command is executed.	y | n y—A statistics record is created. n—A statistics record is not created. The default is n.
max.age	Specifies how old a statistics file must be before it is archived. Once a day, a shell script is executed that identifies the statistics files that are as old as the **max.age**, runs the tar command and the compress command to create a compressed archive, and then deletes the statistics files that have been archived.	A 3-digit decimal number. The default is 8 days. 0—no archiving.

Running a Process generates multiple statistics records. To accommodate the large number of statistics records generated, Sterling Connect:Direct closes the current statistics file and creates a new statistics file at midnight every day. It can also close the current file before midnight if the file size exceeds the value set for the **file.size** initialization parameter. The default file size is 1 megabyte.

Statistics files are stored in the *d_dir*/work/*cd_node* directory. Names of the statistics files are in the format *Syyyymmdd*.ext, where *yyyy* indicates year, *mm* indicates month, and *dd* indicates day. The extension (ext) begins as 001. The extension is incremented by one each time a new statistics file is created in a single day.

Sterling Connect:Direct for UNIX provides a utility to archive and purge statistics files. You identify when to archive a statistics file by setting the parameter, **max.age**. The **max.age** parameter defines how old a statistics file must be before you want to archive the file. Once a day, the script called statarch.sh is started. This script identifies the statistics files that are greater than or equal to the **max.age**. It then runs the tar command and the compress command to create a compressed archived file of all the statistics records that match the **max.age** parameter. Once the statistics files are archived, these files are purged.

The archived files are stored in the directory where the statistics files and TCQ are stored. The shell script, statarch.sh, is located in the ndm/bin directory. If necessary, modify the script to customize it for your environment.

If you want to restore statistics files that have been archived, run the statrestore.sh script. It uses the tar command to restore all the statistics files in the archive. Once files are restored, the statistics records can be viewed using the select statistics command.

Server authentication record

The server authentication record called authentication is used during the authentication procedure. The following parameters are available for this record:

Parameter	Description	Value
server.program	The name and location of the server program used during the authentication procedure.	The default is ndmauths.

Parameter	Description	Value
server.keyfile	The name and location of the key file used during the authentication procedure.	The default is keys.server.

User exit record

The user exit record called **user.exits** provides interfaces to specified programs. The available user exits include Statistics Exit, File Open Exit, and Security Exit. The following parameters are available for this record:

Parameter	Description	Value
stats.exit.program	The gateway control program used during the user exit procedure. This exit is given control for each statistics record that is written.	Name of the gateway control program.
file.open.exit.program	The file open exit program used during the user exit procedure. It enables you to control file names on both the sending and receiving node. The exit is located so that it takes control on the receiving (remote) node before the file is opened. This exit applies only to the copy statement and provides access to all file control parameters (including the data set name file name, sysopt parameters, and disposition).	Name of the file open exit program.
security.exit.program	The security exit program used during the user exit procedure. This exit generates and verifies passtickets, and it also supports other password support programs, such as PASSTICKET, part of the RACF security system available on MVS hosts and also supported by IBM on UNIX AIX and OS/2 computers using the NETSP product.	Name of the security exit program.
security.exit.flag	Modifies the default behavior of **security.exit.program**. This is an optional parameter.	snode_sec_exit_only | sec_exit_only snode_sec_exit_only— Causes Sterling Connect:Direct to use the security exit, when it is acting in the role of the SNODE. After Sterling Connect:Direct receives a valid message, it evaluates the proxy and the secure point-of-entry to establish the local user. The security exit is not used when Sterling Connect:Direct is the PNODE. sec_exit_only—Causes Sterling Connect:Direct to always use the security exit. After Sterling Connect:Direct receives a valid message, it evaluates the proxy and the secure point-of-entry to establish the local user.

Firewall navigation record

The firewall navigation record, called **firewall.parms**, enables you to assign a specific TCP/IP and UDT source port number or a range of port numbers with a

particular TCP/IP and UDT address for outbound Sterling Connect:Direct sessions. These ports also need to be open on the firewall of the trading partner to allow the inbound Sterling Connect:Direct sessions. This feature enables controlled access to an Sterling Connect:Direct server if it is behind a packet-filtering firewall without compromising security policies.

Before you configure firewalls for use with UDT, review all information regarding firewall navigation and rules beginning with Firewall Navigation.

The following parameters are available for this record:

Parameter	Description	Value
tcp.src.ports	For TCP/IP connections, remote IP addresses and the ports permitted for the addresses when using a packet-filtering firewall. This parameter is required only if the local node acts as a PNODE. Place all values for an address inside parentheses and separate each value for an address with a comma.	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.*, 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) A wildcard character (*) is supported to define an IP address pattern. If the wildcard character is used, the optional mask is not valid. For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports.
tcp.src.ports.list. iterations	The number of times that Sterling Connect:Direct scans the list of available ports to attempt a connection before going into a retry state.	Any numeric value from 1–255. The default value is 2.
udp.src.ports	For UDT connections, remote IP addresses and the ports permitted for the addresses when using a packet-filtering firewall. This parameter is recommended if a firewall is used whether the local node acts as a PNODE or an SNODE. Place all values for an address inside parentheses and separate each value for an address with a comma.	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.*, 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) A wildcard character (*) is supported to define an IP address pattern. If the wildcard character is used, the optional mask is not valid. For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports.
udp.src.ports.list. iterations	The number of times that Sterling Connect:Direct scans the list of available ports to attempt a connection before going into a retry state.	Any numeric value from 1–255. The default value is 2.

Chapter 3. Maintaining the client configuration file

Client configuration file

The client configuration file consists of parameter records that interface with End User Applications (EUA). The client file includes the following parameters:

- Sterling Connect:Direct API configuration parameters
- Sterling Connect:Direct CLI configuration parameters
- Client authentication parameters

You can modify Sterling Connect:Direct configuration files using any text editor. If you want to create a new configuration file, use the cdcust command.

Contents of the client configuration file

The client configuration file is created during the customization procedure and resides in *d_dir*/ndm/cfg/cliapi/ndmapi.cfg, where *d_dir* is the directory where Sterling Connect:Direct is installed.

Sample client configuration file

The following example shows a sample client configuration file:

```
# Connect:Direct for UNIX Client configuration file
  cli.parms:\
   :script.dir=/home/qatest/jsmith/cdunix/hp/ndm/bin/:\
  :prompt.string="Test CD on Medea":

api.parms:\
   :tcp.hostname=alicia:\
   :tcp.port=1393:\
   :wait.time=50:

  # Authenticator
 authentication:\
   :client.program=/home/qatest/jsmith/cdunix/hp/ndm/bin/ndmauthc:\
   :client.keyfile=/home/qatest/jsmith/cdunix/hp/ndm/sc/keys.client:
```

API configuration record

The Sterling Connect:Direct API Configuration record, **api.parms**, is used by the API to communicate. The parameters for the API configuration record are described in the following table:

Parameter	Description	Value
tcp.hostname	The host name or IP address to which the API usually connects.	Host name or IP address. For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports.
tcp.port	The TCP/IP port number to which the API usually connects.	Port number. The default is 1363.

Parameter	Description	Value
wait.time	The number of seconds to wait for responses from the server. If this limit is exceeded, the message ID **XCMG000I** is displayed.	Seconds to wait. The default is 50 seconds.

CLI configuration record

The CLI configuration record, cli.parms, identifies the location of the script files to format the output of the select statistics and select process commands and allows you to customize the CLI prompt. If you customize the script to format the output of the **select statistics** and **select process** command, update the **script.dir** parameter to identify the location of the scripts. If you want to display a customized prompt at the CLI command line, in place of the default "Direct" prompt, identify the prompt to use in the **prompt.string** parameter. The cli.parms parameters are described in the following table:

Parameter	Description	Value
script.dir	The directory where customized script files are stored. Specify this parameter if you have created a custom script to format the output of the **select statistics** and **select process** commands. The file names must be ndmstat and ndmproc.	Directory name. The default directory is ndm/bin/.
prompt.string	Identifies the CLI prompt to display on the command line when the client is started. If the prompt string includes spaces or special characters, enclose it in single or double quotation marks. You can set the customized prompt in this parameter and at the command line (using the -P parameter). If the prompt string is specified in both places, the -P parameter at the command line takes precedence. When the default prompt is overridden, the new prompt string is displayed in the Welcome banner and at the command prompt.	Prompt string up to 32 characters. The default is "Direct".

Client authentication record

The client authentication record, **authentication**, is used during the authentication procedure. The client authentication parameters are described in the following table:

Parameter	Description	Value
client.program	The client program to use during authentication.	Client program name. The default is ndmauthc.
client.keyfile	The key file to use during authentication.	Client key file. The default is keys.client.

Chapter 4. Maintaining the network map file

Network map file overview

The network map file is created when you install Sterling Connect:Direct. If necessary, use a text editor to add or modify remote node records in the network map file. You can modify the network map file dynamically while the server is running.

You can also use the Sterling Connect:Direct Browser User Interface to perform some of the procedures related to the initialization parameters file. To learn more about the Sterling Connect:Direct Browser User Interface, see the documentation related to that product in the IBM Documentation Library.

Contents of the network map file

The network map contains connectivity information that describes the local node and the remote nodes in the network. One remote node information record is created for each node with which the local node communicates.

The network map file resides in *d_dir*/ndm/cfg/*cd_node*/netmap.cfg where *d_dir* is the location where Sterling Connect:Direct is installed and *cd_node* is the node name.

If you are using TCP/IP, the local node can communicate with a remote node without a remote node information record. Specify the required connection information in the submit command or the Process statement.

Sample remote node records in a network map

The following sample shows network map remote node entries for a TCP/IP connection and a Sun LU6.2 connection to remote nodes. To insert comments about fields in the network map, be sure to place a # in the first column. If the # is not in the first column, the comment is not ignored and the field is read.

```
# Sample Network Map remote node entry for a TCP/IP connection
 remote.customer.node:\
   :conn.retry.stwait=00.00.30:\
   :conn.retry.stattempts=3:\
   :conn.retry.ltwait=00.10.00:\
   :conn.retry.ltattempts=6:\
  :tcp.max.time.to.wait=180;\
  :runstep.max.time.to.wait=0:\
   :contact.name=:\
   :contact.phone=:\
   :descrip=:\
   :sess.total=255:\
   :sess.pnode.max=255:\
   :sess.snode.max=255:\
   :sess.default=1:\
   :comm.info=10.20.246.49;9974:\
   :comm.transport=tcp:\
   :comm.bufsize=65536:\
   :pacing.send.delay=0:\
   :pacing.send.count=0:
 # Sample Network Map remote node entry for a Sun LU6.2 connection
 # hostl1 is the profile name
 MVS.SAM1.NODE:\
   :conn.retry.stwait=00.00.30:\
   :conn.retry.stattempts=3:\
   :conn.retry.ltwait=00.10.00:\
   :conn.retry.ltattempts=6:\
   :contact.name=:\
   :contact.phone=:\
   :descrip=:\
   :sess.total=255:\
   :sess.pnode.max=128:\
   :sess.snode.max=127:\
   :sess.default=1:\
   :comm.info=hostl1:\
   :comm.transport=blklu62:\
   :comm.bufsize=65536:
```

Local node connection record

The **local.node** record serves two separate purposes:

- Configures settings for the local node
- Provides default configuration values that can be overridden in the remote node entries.

Two sets of connection retry parameters are created:

- Short-term parameters define retry attempts in the event of a short-term connection failure.
- Long-term parameters are used after exhausting short-term attempts. Long-term attempts are set for less frequent retries, because long-term attempts assume that the connection problem cannot be fixed quickly.

Following are the **local.node** parameters. The parameters in bold are required.

Parameter	Description	Value
api.max.connects	The maximum number of concurrent API connections permitted for the local node. The value of **api.max.connects** and **sess.total** cannot exceed the number of file descriptors available. This value is system dependent. A Command Manager (CMGR) is created for every API connection that is successfully established. The number of Command Managers that a PMGR can create is system-dependent and limited by the number of file descriptors available for each UNIX Process. The number of file descriptors set up by the UNIX operating system may affect Sterling Connect:Direct operation.	1–256 The default is 16.
comm.bufsize	The buffer size for transmitting data to and from a remote node for TCP/IP connections.	The value for TCP/IP has no limit (up to 2,147,483,623). For LU6.2, the maximum is 32000. The default is 65536 bytes.
conn.retry.stwait	The time to wait between retries immediately after a connection failure occurs. The format is *hh.mm.ss*, where *hh* specifies hours, mm specifies minutes, and *ss* specifies seconds.	The maximum value is limited to the highest value in the clock format, 23.59.59. The default is 00.00.30, which is 30 seconds.
conn.retry.stattempts	The number of times to attempt connection after a connection failure occurs.	0–9999 The default is 3.
conn.retry.ltwait	The time to wait between long-term retry attempts after all short-term retry attempts are used. The format is *hh.mm.ss*, where *hh* specifies hours, *mm* specifies minutes, and *ss* specifies seconds.	00.00.00–23.59.59 The default is 00.10.00, or 10 minutes.
conn.retry.ltattempts	The number of times to attempt a connection after all short-term retry attempts are used.	0–9999 The default is 6.
conn.retry.exhaust.action	Action to take after the specified short and long-term retries have been used.	hold | delete hold—Places Processes in the hold queue in "Held in Error" status, after all retry attempts are used. This is the default value. delete—Causes the Processes to be deleted from the TCQ.
contact.name	The name of the Sterling Connect:Direct administrator or operator.	Name
contact.phone	The phone number of the Sterling Connect:Direct administrator or operator.	Phone number
descrip	Comments to include as part of the record.	An unlimited text string
lu62.writex.wait	If you are using SNA on an IBM AIX operating system, use this parameter to identify how long to wait before retrying the connection.	0:00:00–59:59:59 The value is in *hh:mm:ss* format. The default value is 1.

Parameter	Description	Value
lu62.writex.retry. attempts	If you are using SNA on an IBM AIX operating system, use this parameter to identify how many times to attempt a connection.	0–2,147,483,647 The default value is `0`.
netmap.check	Enhanced security testing performed on the SNODE. For TCP/IP connections, the remote IP address of the incoming socket connection is compared to the **comm.info** record of the netmap.cfg file. These values must match for an Sterling Connect:Direct session to be established. The **comm.info** record can be the official network name, an alias name listed in the appropriate file (for example, `/etc/hosts`, if the system is not running NIS or DNS), or the IP address. For all connections, the remote node name must be in the netmap.cfg.	`y` | `n` `y`—Specifies that the security checks are made to verify that the remote node name is in the netmap.cfg file. `n`—Specifies that none of these security checks are made. The default value is `n`.
outgoing.address	If running in a high availability environment, this parameter enables you to specify the virtual IP address for the remote node to use for network map checking and prevents the Process from failing when initiated from within a high availability environment. Specify the IP address for this value and network map checking verifies the address instead of the value set in **comm.info** in the SNODE network map record.	*nnn.nnn.nnn.nnn* (IPv4) or *nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn* (IPv6) For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports.
pacing.send.delay	The time to wait between send operations to the remote node. The decimal number is the number of milliseconds between the end of one packet and the beginning of the next packet. Time-based pacing does not contribute to network traffic. The value for this parameter has no effect on LU6.2 connections.	The format is *nnn*. No limit exists for the size of this value. The default is `0`, which indicates no pacing of this type.
pacing.send.count	The number of send operations to perform before automatically waiting for a pacing response from the remote node. The value for this parameter has no effect on LU6.2 connections.	No limit exists for the size of this value. The default is `0`, which indicates no pacing of this type.
proxy.attempt	Enables the **ID** subparameter of **snodeid** to contain a proxy, or dummy user ID to be used for translation to a local user ID on the remote system. Using a dummy user ID improves security because neither the local system nor the remote system requires a valid user ID from the other side.	`y` | `n` `y`—Specifies that the remote users can specify a dummy user ID in **snodeid** parameter. `n`—Specifies that the remote users cannot specify dummy user ID in **snodeid** parameter. The default is `n`.
	The following code illustrates the logic used to perform a security check for the user ID:	

Parameter	Description	Value

	`if (snodeid is coded with ID and PSWD)`
	`attempt OS validation`
	`if (OS validation succeeds)`
	`security OK`
	`else if (proxy.attempt=yes)`
	`if (ID@PNODE proxy found)`
	`security OK`
	`else`
	`security check fails`
	`else`
	`security check fails`
	`else if (snodeid is coded with ID only)`
	`if (proxy.attempt=yes)`
	`if (ID@PNODE proxy found)`
	`security OK`
	`else`
	`security check fails`
	`else`
	`security check fails`
	`else if (snodeid is not coded)`
	`if (submitter&;PNODE proxy found)`
	`security OK`
	`else`
	`security check fails`	

runstep.max.time.to.wait	The maximum time to wait for remote run steps to complete. Remote run steps include remote run task, run job, or submit statements. This wait time is different from the wait time specified by the **tcp.max.time.to.wait parameter**. Using **runstep.max.time.to.wait** prevents a Process from failing when a remote step takes longer to complete than specified in **tcp.max.time.to.wait**.	`0–10000` The value is in seconds. The default value is `0`.
sess.total	The maximum number of concurrent connections between all nodes and the local node. The sum of **api.max.connects** and **sess.total** cannot exceed the number of file descriptors available. This value is system dependent. You must define enough file descriptors to handle the number of concurrent Sterling Connect:Direct sessions allowed, which can be as many as `999`.	`0–999` A 1–3 digit number. The default is `255`.
sess.pnode.max	The maximum concurrent connections, where the local node is the initiator of the session. Number of PNODE sessions cannot exceed the total number of sessions. If **sess.pnode.max** is larger than **sess.total**, the value of **sess.pnode.max** is silently rounded down to the value of **sess.total**.	`0–999` The default is `255`.

Parameter	Description	Value
sess.snode.max	The maximum concurrent connections, where the local node is the secondary node in a session. Number of SNODE sessions cannot exceed the total number of sessions. If **sess.snode.max** is larger than **sess.total**, then it is silently changed to the value of **sess.total**.	0–999 The default is 255.
sess.default	The default session class for starting session managers. A Process executes on the specified class or any higher session class.	1–50 The default is 1.
tcp.api.bufsize	The buffer size for transmitting data to and from an Sterling Connect:Direct CLI/API.	This value has no limit. The default is 32768 bytes.
tcp.api	The information needed to monitor connection requests from the CLI or API using TCP/IP. The *host* is the host name or IP address where Sterling Connect:Direct is running. The *port* identifies the communications port for Sterling Connect:Direct. Multiple *host name/IP addresses* and *port* combinations can be specified when they are separated by a comma. This parameter is required.	*host name/IP address;nnnn* *host nam*e—is the name of the Sterling Connect:Direct host computer. *IP address*—is the IP address of a machine running Sterling Connect:Direct: *nnn.nnn.nnn.nnn* (IPv4) or *nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn* (IPv6) *port*—identifies the communications port for Sterling Connect:Direct. The format is *nnnn*, where *nnnn* is a decimal number. A semi-colon separates the *host name/IP address* from the *port*: msdallas-dt;1363 You can specify multiple *address/host name* and *port* combinations (separated with a comma): 10.23.107.5;1363, fe00:0:0:2014::7;1363, msdallas-dt;1363 For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports.
tcp.api.inactivity. timeout	This is the maximum time a CMGR waits before exiting when it has not received a command from a client program.	0–86399 (23 hours, 59 minutes, and 59 seconds) The value is in seconds. The default is 0, which indicates no timeout occurs.
tcp.max.time.to.wait	The maximum time the local node waits for a message from the remote node when using TCP/IP. When the time expires, the Process is moved to the Timer queue and Sterling Connect:Direct attempts to re-establish a session with the remote node. When set to 0, wait time is unlimited unless limited by the operating system.	0–10000 The value is in seconds. The default value is 180.

TCP/IP Default Record

The **tcp.ip.default** record defines default information to use when the remote node is specified by IP address. The **tcp.ip.default** record parameters are described in the following table:

Parameter	Description	Value
conn.retry.stwait	The time to wait between retries immediately after a connection failure occurs. The format is **hh.mm.ss**, where **hh** specifies hours, mm specifies minutes, and ss specifies seconds.	The maximum value is limited to the highest value in the clock format, 23.59.59. The default is **00.00.30**, which is 30 seconds.
conn.retry.stattempts	The number of times to attempt connection after a connection failure occurs.	0–9999 The default is **3**.
conn.retry.ltwait	The time to wait between long-term retry attempts after all short-term retry attempts are used. The format is **hh.mm.ss**, where **hh** specifies hours, **mm** specifies minutes, and **ss** specifies seconds.	0–23.59.59 The default is **00.10.00**, or 10 minutes.
conn.retry.ltattempts	The number of times to attempt a connection after all short-term retry attempts are used.	0–9999 The default is **6**.
comm.bufsize	The buffer size for transmitting data to and from a remote node.	The value for TCP/IP has no limit (up to 2,147,483,623). For LU6.2, the maximum is 32000. The default is **65536** bytes.
conn.retry.exhaust. action	Action to take after the specified short and long-term retries have been used.	hold | delete hold—Places Processes in the Hold queue in Held in Error status, after all retry attempts are used. This is the default value. delete—Causes the Processes to be deleted from the TCQ.
tcp.max.time.to.wait	The maximum time the local node waits for a message from the remote node when using TCP/IP. When the timer expires, the Process is moved to the Timer queue and Sterling Connect:Direct attempts to re-establish a session with the remote node.	0–10,000 The value in seconds. The default value is **180**. When set to 0, wait time is unlimited unless limited by the operating system.
runstep.max.time.to. wait	The maximum time to wait for remote run steps to complete. Remote run steps include remote run task, run job, or submit statements. This wait time is different from the wait time specified by the tcp.max.time.to.wait parameter. Using runstep.max.time.to.wait prevents a Process from failing when a remote step takes longer to complete than specified in tcp.max.time.to.wait.	0–10000 The value in seconds. The default value is **0**.

Parameter	Description	Value
contact.name	The name of the administrator or operator.	Name
contact.phone	The phone number of the Sterling Connect:Direct administrator or operator.	Phone number
descrip	Comments to include as part of the record.	An unlimited string
sess.total	The maximum number of concurrent connections between all nodes and the local node. The sum of api.max.connects and sess.total cannot exceed the number of file descriptors available. This value is system dependent.	0–999 A 1–3 digit number. The default is **255**.
sess.pnode.max	The maximum concurrent connections, where the local node is the initiator of the session.	0–999 The default is **255**.
sess.snode.max	The maximum concurrent connections, where the local node is the secondary node in a session.	0–999 The default is **255**.
sess.default	The default session class for starting session managers. A Process executes on the specified class or any higher session class. The value for this parameter overrides the equivalent value in the local.node record.	1–50 The default is **1**.
pacing.send.delay	How long to wait between send operations to the remote node. The decimal number is the number of milliseconds between the end of one packet and the beginning of the next packet. Time-based pacing does not contribute to network traffic. The value for this parameter has no effect on LU6.2 connections.	nnn The size of this number has no limit. The default is **0**, which indicates no pacing of this type.
pacing.send.count	The number of send operations to perform before automatically waiting for a pacing response from the remote node. The value for this parameter has no effect on LU6.2 connections.	No limit exists for the size of this value. The default is **0**, which indicates no pacing of this type.

Remote Node Connection Record

The remote node connection record contains information you can use to define default values for a generic remote node connection or customize to for a particular new remote node. Following are the remote node connection parameters.

Parameter	Description	Value
alternate.comminfo	Provides support for establishing netmap-checked sessions with systems with multiple IP addresses, such as Sterling Connect:Direct/Plex z/OS. Use this parameter to list all IP addresses or host names that are part of the multiple IP address environment. For Sterling Connect:Direct/Plex, this list should include the address of each Sterling Connect:Direct/Server with a different IP address from the Sterling Connect:Direct/Plex Manager.If the IP address of the initiating node does not match the IP address specified on the **comm.info** parameter, the **alternate.comminfo** parameter is checked for other valid IP addresses. For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports.	host name/IP address or * host name—Host name associated with the IP address, for example: :alternate.comminfo=hops (where hops is a machine on the local domain) :alternate.comminfo=hops.csg.stercomm.com (fully-qualified host name) IP address——the IP address of a machine running Sterling Connect:Direct in IPv4 or IPv6 format: nnn.nnn.nnn.nnn (IPv4) or nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn (IPv6) For example: :alternate.comminfo=10.23.107.5 :alternate.comminfo=fe00:0:0:2014::7 Specify multiple addresses/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 *—Accepts any IP address. This turns off IP address validation. **Note:** Partial pattern matches are not supported, such as: *.mydomain.com, myplex??.mydomain.com.
alt.comm.outbound	Alternate communication address (communication path) used for outbound Processes. This parameter provides the alternate addresses for a remote node that has multiple NIC cards. When the local node is the PNODE, the alternate addresses are tried if an initial attempt to the primary address (specified in the comm.info parameter) 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. When the local node is the SNODE, the alternate addresses are used in the Netmap check.	Fully-qualified host name/IP address;nnnn The host name/IP address and port are separated by a semi-colon (;). A comma separates the list of alternate communication paths, and the list is processed from the top down. For example: salmon;9400, 10.20.40.65;9500 For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports.

Parameter	Description	Value
comm.bufsize	The buffer size for transmitting data to and from a remote node on TCP/IP connections.	The value for TCP/IP has no limit (up to 2,147,483,623). For LU6.2, the maximum is 32000. The default is **65536** bytes.
comm.info	The information needed to initiate connection requests to remote nodes using TCP/IP or LU6.2. This information refers to the network card that the local Sterling Connect:Direct node uses to initiate outbound requests. This value is required. • For TCP/IP connections, specify the host name or the IP address and port number. If specifying IP address and port, separate parameters with a semicolon (;). • For LU6.2 connections, identify the profile name to identify the name of an SNA configuration profile for the remote connection. Sterling Connect:Direct generates the default name, hostl1, during the customization procedure. For AIX SNA, hostl1 refers to the side information profile name. For HP SNA, SunLink SNA, and Brixton SNA, hostl1 refers to the SNA profile file located in the same directory as the configuration files.	For TCP/IP connections, specify the host name or the IP address and port number. The default port is **1364**. For LU6.2 connections, specify a profile name, up to 8 characters. For more information on specifying IP addresses and host names, see IP Addresses, Host Names, and Ports.
comm.transport	The transport protocol for the remote node.	tcp | lu62 |blklu62 | udt33 tcp—TCP/IP connections lu62—AIX SNA LU6.2 connections blklu62—Other LU6.2 connections udt33—UDT connections
conn.retry.stwait	Time to wait between retries immediately after a connection failure occurs. The format is **hh.mm.ss**, where **hh** specifies hours, **mm** specifies minutes, and ss specifies seconds.	The maximum value is limited to the highest value in the clock format, 23.59.59. The default is **00.00.30**, which is 30 seconds.
conn.retry.stattempts	Number of times to attempt connection after a connection failure occurs.	0–9999 The default is **3**.
conn.retry.ltwait	Time to wait between long-term retry attempts after all short-term retry attempts are used. The format is **hh.mm.ss**, where **hh** specifies hours, **mm** specifies minutes, and **ss** specifies seconds.	0–23.59.59 The default is **00.10.00**, or 10 minutes.
conn.retry.ltattempts	Number of times to attempt a connection after all short-term retry attempts are used.	0–9999 The default is **6**.

Parameter	Description	Value
conn.retry.exhaust.action	Action to take after the specified short and long-term retries have been used.	hold | delete hold—Places Processes in the Hold queue in Held in Error status, after all retry attempts are used. This is the default value. delete—Causes the Processes to be deleted from the TCQ.
contact.name	The name of the Sterling Connect:Direct administrator or operator.	Name
contact.phone	The phone number of the Sterling Connect:Direct administrator or operator.	Phone number
descrip	Comments to include as part of the record.	An unlimited string
pacing.send.count	The number of send operations to perform before automatically waiting for a pacing response from the remote node. The value for this parameter has no effect on LU6.2 connections.	No limit exists for the size of this value. The default is **0**, which indicates no pacing of this type.
pacing.send.delay	The time to wait between send operations to the remote node. The decimal number is the number of milliseconds between the end of one packet and the beginning of the next packet. Time-based pacing does not contribute to network traffic. The value for this parameter has no effect on LU6.2 connections.	nnn The size of this number has no limit. The default is **0**, which indicates no pacing of this type.
runstep.max.time.to.wait	The maximum time to wait for remote run steps to complete. Remote run steps include remote run task, run job, or submit statements. This wait time is different from the wait time specified by the tcp.max.time.to.wait parameter. Using runstep.max.time.to.wait prevents a Process from failing when a remote step takes longer to complete than specified in tcp.max.time.to.wait. The value is in seconds.	0–10000 The default value is **0**.
sess.total	The maximum number of concurrent connections between all nodes and the local node. The sum of api.max.connects and sess.total cannot exceed the number of file descriptors available. This value is system dependent.	0–999 A 1–3 digit number. The default is **255**.
sess.pnode.max	The maximum concurrent connections, where the local node is the initiator of the session. Number of PNODE sessions cannot exceed the total number of sessions. If sess.pnode.max is larger than sess.total, the value of sess.pnode.max is silently rounded down to the value of sess.total.	0–999 The default is **255**.

Parameter	Description	Value
sess.snode.max	The maximum concurrent connections, where the local node is the secondary node in a session. Number of SNODE sessions cannot exceed the total number of sessions. If sess.snode.max is larger than sess.total, then it is silently changed to the value of sess.total.	0–999 The default is **255**.
tcp.crc	Turn on or off the CRC function for TCP/IP Processes on the remote node.	y | n The default is **n**.

Chapter 5. Maintaining access information files

Access Information Files

You can control access to Sterling Connect:Direct through the following components:

- User authorization information file which contains local and remote user information records
- Strong access control file
- Program directory to limit access
- Sterling Connect:Direct's ability to detect shadow passwords
- Security exit

User Authorization Information File

In order for users to have access to Sterling Connect:Direct and use Sterling Connect:Direct commands and statements, you need to define a record for each user ID in the user authorization information file, called userfile.cfg. The user ID is the key to the local user information record. It must be a valid user ID on the local system and must be unique. To disable access to the software for a local user, delete or comment out the local user information record.

You can create a generic user ID by specifying an asterisk (*) as the user ID. If a user does not have a specific local user information record, the user authorizations will default to those specified in this generic record. If no generic local user information record is defined and no specific local user information record is defined for the user, the user cannot use Sterling Connect:Direct.

Sterling Connect:Direct may optionally use remote user information records to translate remote user IDs to valid local user IDs where Sterling Connect:Direct is installed. If an snodeid parameter is not coded on the incoming Process, Sterling Connect:Direct uses this proxy relationship to determine the rights of remote users to issue Sterling Connect:Direct commands and statements.

Sterling Connect:Direct for UNIX uses the asterisk (*) character to establish generic mappings that facilitate mapping remote user IDs to local user IDs. The asterisk matches the node name or the host name. For example, you can specify *@node name to map the remote user ID to all user IDs at one node name, specify id@* to map to a specific user ID at all node names, or specify *@* to match all users at all node names.

Sample Mapping of Remote User IDs to Local User IDs

The following table displays sample remote user ID mappings to local user IDs using the special characters:

Remote User ID	at	Remote Node Name	is mapped to	Local User ID	Result of Mapping
user	@	*	=	test02	Remote user ID “user” on all remote nodes is mapped to local user ID test02.
*	@	mvs.node3	=	labs3	All remote user IDs on remote node mvs.node3 are mapped to local user ID labs3.
*	@	*	=	vip01	All remote user IDs on all remote nodes are mapped to local user ID vip01.

You can generate all the records through the script-based customization procedure or generate only one or two records and use a text editor to generate additional records. After customization, you may want to modify some of the parameters. Use cdcust to create a new user file or a text editor to modify the file as necessary.

Sample User Authorization File

The following sample displays a user authorization file. In the sample, SAM1 is the remote user ID, MVS.SAM1.NODE is the remote node name, and sam is the local UNIX user ID.

```
SAM1@MVS.SAM1.NODE:\
   :local.id=sam:\
   :pstmt.upload=y:\
   :pstmt.upload_dir=/home/qatest/username/ndm/uploaddir:\
   :pstmt.download=y:\
   :pstmt.download_dir=/home/qatest/username/ndm/downloaddir:\
   :pstmt.run_dir=/home/qatest/username/ndm/rundir:\
   :pstmt.submit_dir=/home/qatest/username/ndm/submitdir:\
   :descrip=:
 sam:\
   :admin.auth=y:\
   :pstmt.copy.ulimit=y:\
   :pstmt.upload=y:\
   :pstmt.upload_dir=/home/qatest/username/ndm/uploaddir:\
   :pstmt.download=y:\
   :pstmt.download_dir=/home/qatest/username/ndm/downloaddir:\
   :pstmt.run_dir=/home/qatest/username/ndm/rundir:\
   :pstmt.submit_dir=/home/qatest/username/ndm/submitdir:\
   :name=:\
   :phone=:\
   :descrip=:
 :cmd.s+conf=n:
```

Local User Information Record Format

The local user record, userid, defines the default values for each user ID. Most of the parameters in the local user information record can take the following values:

- y—Indicates that a user can perform the function. In the case of process and select statistics commands, the user can affect Processes and view statistics owned by this user ID
- n—Indicates that a user cannot perform the function.

- a—Indicates that a user can issue commands for Processes owned by all users and generate statistics records for all users.

If the same parameter is specified in the remote user information record and the local user information record, the parameter in remote user information record takes precedence unless it is a null value. When a null value is specified in the remote record, the local user record takes precedence.

The following table defines the local user information parameters. The default values are underlined.

Parameter	Description	Value
admin.auth	Determines if the user has administrative authority. If set to y, the user can perform all of the commands by default, but the specific command parameters override the default. If set to n, the specific command parameters must be granted individually.	y | n y—User has administrative authority. n—User does not have administrative authority. The default is **n**.
cmd.chgproc	Determines if the user can issue the change process command. A "y" value enables a user to issue the command to targets owned by that user. Whereas, "a" allows a user to issue the command to targets owned by all users.	y | n | a y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**. a—Allows the user to issue the command against targets owned by all users.
cmd.delproc	Determines if the user can issue the delete process command. A "y" value enables a user to issue the command to targets owned by that user. Whereas, "a" allows a user to issue the command to targets owned by all users.	y | n | a y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**. a—Allows the user to issue the command against targets owned by all users.
cmd.flsproc	Determines if the user can issue the flush process command. A "y" value enables a user to issue the command to targets owned by that user. Whereas, "a" allows a user to issue the command to targets owned by all users.	y | n | a y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**. a—Allows the user to issue the command against targets owned by all users.
cmd.selproc	Determines if the user can issue the select process command. A "y" value enables a user to issue the command to targets owned by that user. Whereas, "a" allows a user to issue the command to targets owned by all users.	y | n | a y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**. a—Allows the user to issue the command against targets owned by all users.

Parameter	Description	Value
cmd.viewproc	Determines if the user can issue the view process command. A "y" value enables a user to issue the command to targets owned by that user. Whereas, "a" allows a user to issue the command to targets owned by all users.	y | n | a y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**. a—Allows the user to issue the command against targets owned by all users.
cmd.selstats	Determines if the user can issue the select statistics command. A "y" value enables a user to issue the command to targets owned by that user. Whereas, "a" allows a user to issue the command to targets owned by all users.	y | n | a y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**. a—Allows the user to issue the command against targets owned by all users.
cmd.stopndm	Determines if the user can issue the stop command.	y | n y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**.
cmd.s+conf	Determines if the user can issue commands from network clients, such as IBM Sterling Control Center or Java API, to configure Sterling Connect:Direct Secure Plus. **Note:** This parameter has no effect on local tools, such as spadmin.sh and spcli.sh.	y | n y—Allows the user to issue commands. The default is **y**. n—Prevents the user from issuing commands.
cmd.submit	Determines if the user can issue the submit process command.	y | n y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**.
cmd.trace	Determines if the user can issue the trace command.	y | n y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**.
pstmt.crc	Enables the user to override the initial settings to use the keyword CRC in a Process statement.	y | n y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**.
descrip	Permits the administrator to add descriptive notes to the record.	Unlimited text string
name	The name of the user.	User name
phone	The phone number of the user.	user phone number

Parameter	Description	Value
pstmt.copy	Determines if the user can issue the **copy** statement.	y | n y—Allows the user to issue the command. n—Prevents the user from issuing the command. The default is **n**.
pstmt.copy.ulimit	The action taken when the limit on a user output file size is exceeded during a copy operation. The value for this parameter overrides the equivalent value for the ulimit parameter in the initialization parameters file.	y | n | nnnnnnnn | nnnnnnnnK | nnnnnnnM | nnnnG y—Honors the user file size limit. If this limit is exceeded during a copy operation, the operation fails. n—Ignores the limit. The default is **n**. nnnnnnnn, nnnnnnnnK, nnnnnnnM, or nnnnG—Establishes a default output file size limit for all copy operations. K denotes 1024 bytes. M denotes 1048576 bytes. G denotes 1073741824 bytes. The maximum value you can specify is 1 TB.
pstmt.upload	Determines if the user can send files from this local node. If a file open exit is in use, this parameter is passed to the exit, but it is not enforced.	y | n y—Allows the user to send files. The default is **y**. n—Prevents the user from sending files.
pstmt.upload_dir	The directory from which the user can send files. If a value is set for this parameter, then files can only be sent from this directory or subdirectories. If a file open exit is in use, this parameter is passed to the exit, but it is not enforced. If this parameter is defined, file names in Copy statements must be relative to this directory. Absolute path names can be used, but the path must coincide with this specification.	Directory path name
pstmt.download	Determines if the user can receive files to this local node. If a file open exit is in use, this parameter is passed to the exit, but it is not enforced.	y | n y—Allows the user to receive files. The default is **y**. n—Prevents the user from receiving files.
pstmt.download_dir	The directory to which the user can receive files. If a value is set for this parameter, then files can only be received to this directory or subdirectories. Otherwise, they can be received to any directory. If a file open exit is in use, this parameter is passed to the exit, but it is not enforced.	Directory path name

Parameter	Description	Value
pstmt.run_dir	The directory where Sterling Connect:Direct is installed that contains the programs and scripts the user executes with run job and run task statements. Any attempt to execute a program or script outside the specified directory fails. The UNIX Restricted Shell provides enhanced security by restricting the user to the commands contained in the pstmt.run_dir. If the user does not specify pstmt.run_dir, the commands are started with the Bourne shell. To restrict the use of special characters in the run directory, be sure to configure Y for the **restrict:cmd** initialization parameter. For more information on specifying the restrict:cmd initialization parameter, see Restrict Record.	Directory path name
pstmt.runjob	Specifies whether the user can issue the **run job** statement.	y | n y—Allows the user to issue the statement. n—Prevents the user from issuing the statement. The default is **n**.
pstmt.runtask	Specifies whether the user can issue the **run task** statement.	y | n y—Allows the user to issue the statement. n—Prevents the user from issuing the statement. The default is **n**.
pstmt.submit	Specifies whether the user can issue the **submit** statement.	y | n y—Allows the user to issue the statement. n—Prevents the user from issuing the statement. The default is **n**.
pstmt.submit_dir	The directory from which the user can submit Processes. This is for **submit**s within a Process.	Directory path name
snode.ovrd	Specifies whether the user can code the **snodeid** parameter on the **submit** command and **process** and **submit statements.**	y | n y—Allows the user to code the **snodeid** parameter n—Prevents the user from coding the **snodeid** parameter. The default is **n**.
pstmt.crc	Gives the user the authority to specify the use of CRC checking in a Process statement. Setting this parameter to y enables the user to override the initial settings in the initialization parameters or network map settings files.	y | n y—Allows a user to specify CRC checking on a Process statement. n—Prevents a user from specifying CRCchecking on a Process statement. The default is **n**.

Remote User Information Record

The remote user information record contains a remote user ID and a remote node name that become the key to the record. The local.id parameter identifies a local user information record for this user. You must create a local user information record for the remote user.

Note: To prevent the remote user from using Sterling Connect:Direct, delete or comment out the remote user information, unless the remote user specifies an SNODEID parameter in the Process.

The remote user information record is remote userid@remote node name. It specifies the user and remote node name pair defined as a remote user. This value becomes the key to the record and must be unique. Create a remote user information record for each user on a remote node that will communicate with this local node.

Following are the parameters for the remote user information record:

Parameter	Description	Value
local.id	The local user ID to use for security checking on behalf of the remote user. The **local.id** parameter must identify a local user information record.	Local user ID
pstmt.copy	Determines if the user can issue the **copy** statement.	y | n y—Allows a user to issue the statement. n—Prevents a user from issuing the statement. The default is **n**.
pstmt.upload	Determines if the user can send files from this local node. If a file open exit is in use, this parameter is passed to the exit, but it is not enforced.	y | n y—Allows a user to send files. The default is **y**. n—Prevents a user from sending files.
pstmt.upload_dir	The directory from which the user can send files. If a value is set for this parameter, then files can only be sent from this directory or subdirectories. If a file open exit is in use, this parameter is passed to the exit, but it is not enforced. If this parameter is defined, file names in Copy statements must be relative to this directory. Absolute path names can be used, but the path must coincide with this specification.	Directory path name

Parameter	Description	Value
pstmt.download	Determines if the user can receive files to this local node. If a file open exit is in use, this parameter is passed to the exit, but it is not enforced.	y | n y—Allows a user to receive files. The default is **y**. n—Prevents a user from receiving files.
pstmt.download_dir	The directory to which the user can receive files. If a value is set for this parameter, then files can only be received to that directory or subdirectories. Otherwise, they can be received to any directory. If a file open exit is in use, this parameter is passed to the exit, but it is not enforced.	Directory path name
pstmt.run_dir	The directory that contains the programs and scripts the user can execute with **run job** and **run task** statements. Any attempt to execute a program or script outside the specified directory fails. To restrict the use of special characters in the run directory, be sure to configure Y for the **restrict:cmd** initialization parameter. For more information on specifying the restrict:cmd initialization parameter, see Restrict Record.	Directory path name
pstmt.submit_dir	The directory from which the user can submit Processes. This is for **submit**s within a Process.	Directory path name
pstmt.runjob	Specifies whether the user can issue the **run job** statement.	y | n y—Allows a user to issue the statement. n—Prevents a user from issuing the statement. The default is **n**.
pstmt.runtask	Specifies whether the user can issue the **run task** statement.	y | n y—Allows a user to issue the statement. n—Prevents a user from issuing the statement. The default is **n**.
pstmt.submit	Specifies whether the user can issue the **submit** statement.	y | n y—Allows a user to issue the statement. n—Prevents a user from issuing the statement. The default is **n**.
descrip	Permits you to add descriptive notes to the record.	Text string

Strong Access Control File

To provide a method of preventing an ordinary user from gaining root access through Sterling Connect:Direct, a strong access control file called sysacl.cfg is created at installation in the d_dir/ndm/SACL/ directory. By default, an ordinary user cannot access the root through Sterling Connect:Direct for UNIX. If you want to give an ordinary root user access through Sterling Connect:Direct for UNIX, you must access and update the **sysacl.cfg** file.

Note: Even if you do not want to limit root access through Sterling Connect:Direct for UNIX, the sysacl.cfg file must exist. If the file is deleted or corrupted, all users are denied access to Sterling Connect:Direct for UNIX.

The file layout of the sysacl.cfg file is identical to the user portion of the userfile.cfg file. Setting a value in the sysacl.cfg file for a user overrides the value for that user in the userfile.cfg file.

The root:deny.access parameter, which is specified in the sysacl.cfg file, allows, denies, or limits root access to Sterling Connect:Direct. This parameter is required. The following values can be specified for the **root:deny.access** parameter:

Parameter	Description	Value
deny.access	Allows, denies, or limits root access to IBM Sterling Connect:Direct	y | n | d y—No Processes can acquire root authority n—PNODE Processes can acquire root authority, but SNODE Processes can not. This is the default value. d—Any Process can acquire root authority

If a user is denied access because the **root:deny.access** parameter is defined in the sysacl.cfg file for that user, a message is logged, and the session is terminated. If a user is running a limited ID, an informational message is logged.

Automatic Detection of Shadow Passwords

Because shadow password files are available on some versions of the UNIX operating system, Sterling Connect:Direct for UNIX detects the use of shadow passwords automatically, if available.

Limiting Access to the Program Directory

The program directory provides enhanced security for the run task and run job process statements by limiting access to specified scripts and commands. Any attempt to execute a program or script outside the specified directory fails. The program directory is identified with the **pstmt.run_dir** parameter. If the program directory is specified, the UNIX restricted shell is invoked, providing enhanced security. If the program directory is not specified, the regular (Bourne) shell is invoked for executing commands with no restrictions.

The restricted shell is very similar to the regular (Bourne) shell, but it restricts the user from performing the following functions:

- Changing the directory (cd)
- Changing PATH or SHELL environment variables
- Using command names containing a slash (/) character
- Redirecting output (> and >>)

Additional information about the restricted shell can be found in the appropriate UNIX manual pages or UNIX security text books.

The restricted shell is started using only the environment variables HOME, IFS, PATH, and LOGNAME, which are defined as follows:

```
HOME=run_dir
IFS=whitespace characters (tab, space, and newline)
PATH=/usr/rbin and run_dir
LOGNAME=user's UNIX ID
```

Because environment variables are not inherited from the parent Process, no data can be passed to the script or command through shell environment variables. The restricted shell restricts access to specified scripts and commands, but it does not restrict what the scripts and commands can do. For example, a shell script being executed within the run_dir directory can change the value of PATH and execute command names containing a slash (/) character. For this reason, it is important that the system administrator controls which scripts and commands the user has access to and does not give the user write privileges to the run_dir directory or any of the files in the run_dir directory.

Security Exit

The Security Exit in the initialization parameters file, initparm.cfg, provides an interface to password support programs.

This exit generates and verifies passtickets and it also supports other password support programs. An example of other programs is PASSTICKET, part of the RACF security system available on MVS hosts and also supported by IBM on UNIX AIX and OS/2 computers using the NETSP product.

For more information on the Security Exit, see User Exit Record.

Chapter 6. Maintaining client and server authentication key files

Client and Server Authentication Key Files

Sterling Connect:Direct client/server security depends on a key, similar to a password, in a Sterling Connect:Direct server and an identical key in each API that communicates with that server. The keys are defined and coordinated by the system administrator. You can edit both key files with any text editor installed on your system.

The client key file is called keys.client on the node on which the API resides. The server key file is keys.server on the node on which the server resides. The key files are located in the directory d_dir/security.

Key File Format

A record in a key file can contain up to four keys that match entries in another API or server key file. The key file can contain as many key file records as necessary. The format of a key file entry is illustrated in the following sample:

```
hostname MRLN SIMP key [key [key [key] ] ]
```

Key File Parameters

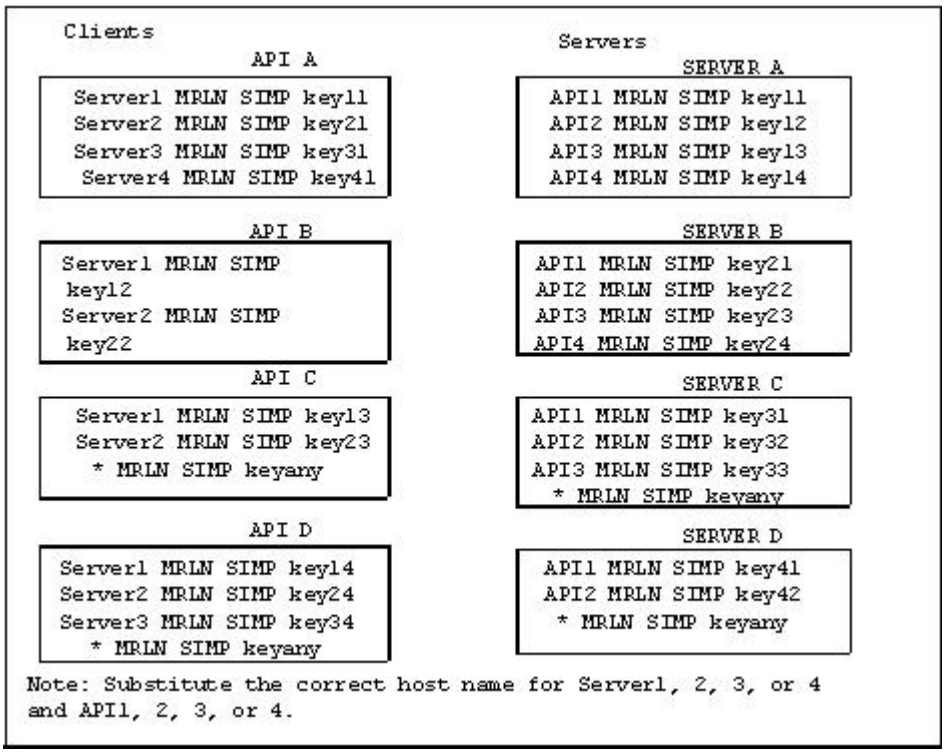
The following table describes the available key file parameters:

Parameter	Description	Value
hostname	The host name of the server with which you want to communicate or the host name of the API you will allow to communicate with your server. The hostname is followed by one or more space characters. If you replace the host name with an asterisk (*) character in the server configuration file, the server accepts a connection from any API with a matching key. You can use only one asterisk per file. Always place the entry with the asterisk after entries with specific host names.	1—16 characters and must be unique within its key file.
MRLN SIMP	A required character string, separated from the other fields by one or more spaces.	Character string
key	The security key. Separate the key from SIMP by one or more spaces.	Up to 22 characters long including A to Z, a to z, 0 to 9, period (.), and ⌂slash (/). **Note:** Only the last 22 characters entered are used. Any characters added prior to the last 22 are ignored.

Sample Client Authentication Key File

The following figure illustrates API key lists in the Clients column and server key lists in the Servers column.

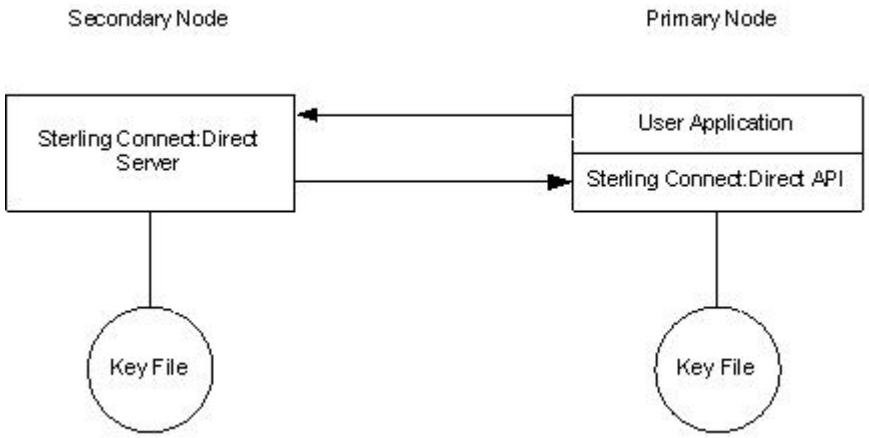
- API A contains key11, key21, key31, and key41. Key11 enables API A to communicate with Server A because Server A also contains the key11 entry. You must ensure that API1 is the host name on which API A resides and that Server1 is the host name on which Server A resides.
- API D contains key14, key24, and key34. Key14 enables API D to communicate with Server A because Server A also contains the key14 entry. You must ensure that API4 is the host name on which API D resides and that Server1 is the host name on which Server A resides.
- API C can communicate with Server A and Server B through matching keys. API C also can communicate with Server C and Server D only through the * MRLN SIMP keyany line.



Authentication Process

The Sterling Connect:Direct authentication process determines if the user is authorized to access the system.

The goal of Sterling Connect:Direct security is to reliably determine the identity of each user without requiring logon repetition. In addition, the security design ensures that all requests originate from the Sterling Connect:Direct API, to ensure that the authentication process is not bypassed by an unauthorized user. The following figure displays the components that perform authentication:



Server Authentication Parameters

The server authentication parameters are specified in initparm.cfg. You must have ownership and permissions to modify these files. Ownership is established during the installation procedure.

Additionally, the directory containing the keys.server file must have UNIX permission 0700, and keys.server must have UNIX permission 0600. These files cannot be owned by root.

The following server authentication parameters are used by the CMGR during the authentication procedure:

Parameter	Description
server.program	The server program to use during the authentication procedure.
server.keyfile	The key file to use during the authentication procedure.

Client Authentication Parameters

The client authentication parameters are specified in ndmapi.cfg. You must have ownership and permissions to modify these files. Ownership is established during the installation procedure.

Additionally, the directory containing the keys.client file must have UNIX permission 0700, and keys.client must have UNIX permission 0600.

The following client authentication parameters are used by the CLI/API during the authentication procedure:

Parameter	Description
client.program	The client program to use during the authentication procedure.
client.keyfile	The key file to use during the authentication procedure.

Firewall Navigation

Firewall navigation enables controlled access to an Sterling Connect:Direct system running behind a packet-filtering firewall without compromising your security policies or those of your trading partners. You control this access by assigning a specific TCP or UDT source port number or a range of source port numbers with a specific destination address (or addresses) for Sterling Connect:Direct sessions.

Before you configure source ports in the Sterling Connect:Direct initialization parameters, you need to review all information regarding firewall navigation and rules, especially if you are implementing firewalls for UDT.

Implement Firewall Navigation

To implement firewall navigation in Sterling Connect:Direct:

Procedure

1. Coordinate IP address and associated source port assignment with your local firewall administrator before updating the firewall navigation record in the initialization parameters file.
2. Add the following parameters to the Sterling Connect:Direct initialization parameters file as needed, based on whether you are using TCP or UDT:
   - tcp.src.ports
   - tcp.src.ports.list.iterations
   - udp.src.ports
   - udp.src.ports.list.iterations
3. Coordinate the specified port numbers with the firewall administrator at the remote site.

Firewall Rules

Firewall rules need to be created on the local firewall to allow the local Sterling Connect:Direct node to communicate with the remote Sterling Connect:Direct node. A typical packet-filtering firewall rule specifies that the local firewall is open in one direction (inbound or outbound) to packets from a particular protocol with particular local addresses, local ports, remote addresses, and remote ports. Firewall Navigation differs between TCP and UDT; as a result, firewall rules for TCP and UDT should be configured differently.

TCP Firewall Navigation Rules

In the following table, the TCP rules are presented in two sections: the first section applies to rules that are required when the local node is acting as a PNODE; the second section applies to rules that are required when the local node is acting as an SNODE. A typical node acts as a PNODE on some occasions and an SNODE on other occasions; therefore, its firewall will require both sets of rules.

TCP PNODE Rules			
Rule Name	**Rule Direction**	**Local Ports**	**Remote Ports**
PNODE session	Outbound	Local C:D's source ports	Remote C:D's listening port
TCP SNODE Rules			
Rule Name	Rule Direction	Local Ports	Remote Ports

TCP PNODE Rules			
Rule Name	**Rule Direction**	**Local Ports**	**Remote Ports**
SNODE session	Inbound	Local C:D's listening port	Remote C:D's source ports

UDT Firewall Navigation Rules

UDT firewall rules are applied to the UDP protocol. The recommended default firewall rule for UDP packets is to block packets inbound to the local system and outbound from the local system to prevent the confusion that could occur due to the callback feature of UDT session establishment.

In the following table, the UDT rules are presented in two sections: the first section applies to rules that are required when the local node is acting as a PNODE; the second section applies to rules that are required when the local node is acting as an SNODE. A typical node acts as a PNODE on some occasions and an SNODE on other occasions; therefore, its firewall will require both sets of rules.

UDT PNODE Rules			
Rule Name	**Rule Direction**	**Local Ports**	**Remote Ports**
PNODE Session Request	Outbound	Local C:D's source ports	Remote C:D's listening port
PNODE Session	Outbound	Local C:D's source ports	Remote C:D's source ports
UDT SNODE Rules			
Rule Name	Rule Direction	Local Ports	Remote Ports
SNODE listen	Inbound	Local C:D's listening port	Remote C:D's source ports
SNODE session	Inbound	Local C:D's source ports	Remote C:D's source ports

Firewall Configuration Examples

In the firewall configuration examples for TCP and UDT, the following IP addresses and source ports will be used:

Note: The IP addresses in the examples have been chosen to be distinctive and are not intended to be valid IP addresses.

- The **local node** has IP address 222.222.222.222 and listening port 2264. Its source ports for communicating with the remote node are 2000–2200.
- The **remote node** has IP address 333.333.333.333 and listening port 3364. Its source ports for communicating with the local node are 3000–3300.

See Session Establishment for a discussion of the differences between UDT and TCP session establishment.

TCP Firewall Configuration Example

The Sterling Connect:Direct administrator configures the **local node** to listen on port 2264, and the following initialization parameter settings are used to configure the local node's source ports:

- tcp.src.ports = (333.333.333.333, 2000–2200)
- tcp.src.ports.list.iterations = 1

This configuration specifies to use a source port in the range 2000–2200 when communicating with the remote node's address 333.333.333.333 and to search the port range one time for an available port. The local node will act as both a PNODE and an SNODE when communicating with the remote node.

Based on this scenario, the firewall rules for the local node are the following:

Rule Name	Rule Direction	Local Ports	Remote Ports
PNODE session request	Outbound	2000–2200	3364
SNODE session	Inbound	2264	3000–3300

UDT Firewall Configuration Example

The Sterling Connect:Direct administrator configures the **local node** to listen on port 2264, and the following initialization parameter settings are used to configure the local node's source ports:

- udp.src.ports = (333.333.333.333, 2000-2200)
- udp.src.ports.list.iterations = 1

This configuration specifies to use a source port in the range 2000–2200 when communicating with the remote node's address 333.333.333.333 and to search the port range one time for an available port. The local node will act as both a PNODE and an SNODE when communicating with the remote node.

Based on this scenario, the firewall rules for the local node are the following:

Rule Name	Rule Direction	Local Ports	Remote Ports
PNODE session request	Outbound	2000–2200	3364
PNODE session	Outbound	2000–2200	3000–3300
SNODE listen	Inbound	2264	3000–3300
SNODE session	Inbound	2000–2200	3000–3300

Blocking Outbound Packets

The recommended default rule for outbound UDP packets from the local system is to block the packets. If you do not follow this recommendation, port usage may, at first sight, appear to violate the firewall's inbound rules.

An example will help illustrate this situation. Suppose that in the example in the previous section:

- The local node is the SNODE.
- The default outbound rule allows all outbound UDP packets from the local system.
- The “SNODE session” rule is accidently omitted.

Because of the callback feature of UDT session establishment, SNODE sessions are still likely to succeed on ports 2000–2200. This may cause confusion because ports 2000–2200 are blocked to inbound UDP packets.

If you use the recommended default outbound rule and apply the PNODE and SNODE rules described in the previous section, there will be no confusion about which port to use, and the UDT callback feature will function as designed, thus supporting reliability.

Session Establishment

Session establishment differs between TCP and UDT; these differences affect how you set up firewall rules and configure the firewall navigation initialization parameters in Sterling Connect:Direct.

TCP Session Establishment

An Sterling Connect:Direct TCP client contacts an Sterling Connect:Direct TCP server on its listening port. The Sterling Connect:Direct client scans the list of ports (specified using the **tcp.src.ports** initialization parameter) and looks for a port to bind to. The number of times Sterling Connect:Direct scans the list is specified using the **tcp.src.ports.list.iterations** initialization parameter. If Sterling Connect:Direct finds an available port, communication with the remote node proceeds.

UDT Session Establishment

When an Sterling Connect:Direct UDT client contacts an Sterling Connect:Direct UDT server on its listening port to request a session, the UDT server responds with a different server port to use for the session. The client attempts to contact the server on the session port. The Sterling Connect:Direct client scans the list of ports (specified in the **udp.src.ports** initialization parameter) and looks for an available port to bind to. The number of times Sterling Connect:Direct scans the list is specified using the **udp.src.ports.list.iterations** initialization parameter. If the Sterling Connect:Direct client finds an available port, communication with the remote Sterling Connect:Direct server proceeds. If a session cannot be established after a certain time interval, the server attempts to contact the client.

Chapter 7. Specifying connection information

IP Addresses, Host Names, and Ports

Sterling Connect:Direct accepts both Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) versions of the Internet Protocol as well as host names. You can enter IP addresses/host names and ports in several ways, depending on the field you are specifying:

- Address or host name only
- Port number only
- Address/host name with a port number
- Multiple address/host name and port combinations

When specifying IP addresses/host names and ports for Sterling Connect:Direct, use the following guidelines.

IP Addresses

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

IPv4 Addresses

IPv4 supports 2^{32} addresses written as 4 groups of dot-separated 3 decimal numbers (0 through 9), for example, 10.23.107.5.

IPv6 Addresses

IPv6 supports 2^{128} addresses written as 8 groups of colon-separated 4 hexadecimal digits, for example, 1001:0dc8:0:0:0:ff10:143e:57ab. The following guidelines apply to IPv6 addresses:

- If a four-digit group contains zeros (0000), the zeros may be omitted and replaced with two colons (::), for example:

```
2001:0db8:85a3:0000:1319:8a2e:0370:1337
can be shortened as
2001:0db8:85a3::1319:8a2e:0370:1337
```

- Any number of successive 0000 groups may be replaced with two colons (::), but only one set of double colons (::) can be used in an address, for example:

```
001:0db8:0000:0000:0000:0000:1319:58ab
Can be shortened as:
2001:0db8:0000:0000::1319:58ab
```

- Leading zeros in a four-zero group can be left out (0000 can be shortened to 0), for example:

```
2001:0db8:0000:0000:0000:0000:1319:58ab
Can be shortened as:
2001:0db8:0:0:0:0:1319:58ab
```

- You can write a sequence of 4 bytes that occur at the end of an IPv6 address in decimal format using dots as separators, for example:

```
::ffff:102:304
Can be written as:
::ffff:1.2.3.4
```

  This notation is useful for compatibility addresses.

Host Names

When you specify a host name, rather than an IP address, Sterling Connect:Direct gets the IP address from the operating system. The first IP address returned by the operating system is used regardless of whether it is in IPv4 or IPv6 format.

A host name (net, host, gateway, or domain name) is a text string of up to 24 characters comprised of the alphabet (A–Z), digits (0–9), minus sign (-), and period (.), for example, msdallas-dt.

The following guidelines also apply:

- No blank or space characters are permitted as part of the name.
- Periods are allowed only when they are used to delimit components of domain-style names.
- Host names are not case sensitive.
- The first and last character must be a letter or digit.
- Single-character names or nicknames are not allowed.

Port Numbers

Port numbers can be appended to the end of IP/host addresses when they are preceded by a semi-colon (;), for example, 10.23.107.5;1364. This convention is specific to Sterling Connect:Direct and is not an industry standard.

A port number must be in the range of 0 through 65535. Port numbers lower than 1024 are designated as reserved and should not be used. The following examples show port numbers appended to IP/host addresses using these conventions:

```
10.23.107.5;1364
fe00:0:0:2014::7;1364
msdallas-dt;1364
```

Multiple Addresses, Host Names, and Ports

You can specify multiple IPv4 and IPv6 addresses and host names by separating them with a comma (,). A space can be added after the comma for readability, for example:

```
10.23.107.5, fe00:0:0:2014::7, msdallas-dt
```

You can also specify a port number for each address or host name. The port is separated from its corresponding address/host name with a semi-colon (;), and each address/host name and port combination is separated by a comma (,). A space may be added after the comma for readability. The following example shows multiple address/host name and port combinations:

```
10.23.107.5;1364, fe00:0:0:2014::7;1364, msdallas-dt;1364
```

Multiple address/host names (and combinations with port numbers) are limited to 1024 characters.

About Using Masks for IP Address Ranges

When you specify a value for the **tcp.src.ports** parameter in the initialization parameters file, you can use masks to specify the upper boundary of a range of IP addresses that will use a specific port, multiple ports, or a range of ports. Sterling Connect:Direct 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 will use 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 will use a port in the range of 2000 through 3000.

`(199.2.4.0/255.255.255.0, 4000-5000)`—Any IPv4 address that falls in the range from 199.2.4.0 through 199.2.255.255 will use a port in the range of 4000 through 5000.

`(fd00:0:0:2015::0/48, 6000, 7000)`—Any IPv6 address that falls in the range from fd00:0:0:2015:0:0:0:0 through fd00:0:0:ffff:ffff:ffff:ffff:ffff will use port 6000 or port 7000.

As shown in the sample entry above, the wildcard character (*) is supported to define an IP address pattern. You can specify up to 255 unique IP address patterns or up to 1024 characters in length, each with its own list of valid source ports. If the wildcard character is used, the optional mask is not valid.

Chapter 8. Using Sterling Connect:Direct in a test mode

Test Mode Overview

You can enable a test mode for production instances of Sterling Connect:Direct to perform the following functions:

- Test new applications and customer connections
- Prevent future production work from executing until testing is complete after you have terminated all active production work using the Flush Process command
- Resume regular production work after testing
- Control individual file transfers by application
- Enable and disable individual nodes and applications

While testing is being conducted, only Processes, particularly file transfers, involved with the testing activity are executed. No production data is transferred to applications being tested while at the same time no test data is transferred to production applications.

Processing Flow of the Test Mode

You enable the testing mode using the **quiesce.resume** initialization parameter and specify which Sterling 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 /ndm/src directory. After you have updated the file for your testing environment, place it in the installation ndm/cfg/<nodename> directory. If you enable the quiesce.resume parameter, you must have an NDMPXTBL table to operate Sterling Connect:Direct in a test mode.

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 Sterling Connect:Direct submitter ID and submitter node combination

In addition to telling Sterling 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.

When the testing mode is enabled, Sterling Connect:Direct performs a syntax check on the parameter table and fails initialization if the table is invalid. If the table is valid, Sterling Connect:Direct 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.

For Processes initiated on remote nodes, the testing mode functions in the same manner as it does for Processes submitted on the local Sterling Connect:Direct node except that the remote node is the PNODE (Process owner) for that Process, and the local node is the SNODE (secondary node). The NDMPXTBL Parameter Table is searched for a matching entry, and the remotely-initiated Process is either permitted to execute or is excluded from execution. Because the local node is the SNODE for this type of transfer, it cannot enforce the Process disposition setting in the NDMPXTBL parameter table. The remote PNODE determines how the Process is handled. Typically, the remote node places the Process in the Hold queue with a status of “HE” (Held in Error).

Preparing the NDMPXTBL Parameter Table

You can use any text editor to modify the sample NDMPXTBL parameter table supplied with Sterling Connect:Direct. When you update the parameter table, name it NDMPXTBL and save it to the Server directory of the installation. The parameter table file can be created or updated while the server is active, and any changes made to the file take effect for sessions that begin after the changes are made. Similarly, the **quiesce.resume** initialization parameter can be modified while the server is active. For more information on the **quiesce.resume** initialization parameter, see Quiesce/Resume Record.

Note: If you enable the quiesce.resume initialization parameter, you must have an NDMPXTBL parameter table.

NDMPXTBL Parameter Table

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.

Command Code	Description	Subparameters/Examples
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.”
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 Sterling 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.

Command Code	Description	Subparameters/Examples
I	Includes Processes for execution that match the patterns in the table which follow this command code. Either "I" or "X" must be the second non-comment entry in the table. Processes which do not match a pattern in the table are not executed. **Note:** To choose which command code to use to select Processes, determine which group is smaller and use the corresponding command Code. For example, if the number of Processes to be executed is smaller than the number of Processes to exclude from execution, specify "I" as the command code and add patterns to match that group of Processes.	ER 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 SDALLASOPS@*⌂ Excludes Processes for execution submitted by the ID DALLASOPS from any node
L	Last entry in table.	

Sample Test Scenarios

The following examples show different applications of the test mode using the NDMPXTBL parameter table to define which Sterling Connect:Direct Processes to run and not run.

Specifying Which Processes Run

In this example, Sterling Connect:Direct executes all Processes that start with ACH or are named DITEST01 or DITEST02. All other Processes are placed in the Hold queue.

```
* Enable processing. Only permit processes matching one of the patterns
* to execute. Hold processes that don't execute.
EH
I
PACH*
PDITEST01
PDITEST02
L
```

Specifying Which Processes to Exclude

In this example, Sterling Connect:Direct does not execute any Process that starts with ACH or is named DITEST01 or DITEST02. All other Processes are executed.

```
* Exclude matching processes.  Permit all others to execute.
EH
X
PACH*
PDITEST01
PDITEST02
L
```

Permitting Process Execution by Secondary Node and Submitter User ID/Node

In this example, Sterling 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 Sterling Connect:Direct submitter ID from node DI0049
- The specific Sterling Connect:Direct submitter ID ACHAPP from any node

All Processes not matching one of the above criteria are flushed from the queue.

```
* Only permit matching processes to execute.  Flush those that do not.
EF
I
NDI.NODE1
NDI0017*
S*@DI0049
SACHAPP@*
L
```

Stopping the Test Mode

In this example, no Processes will 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
```

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-IBMproducts. 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.

Trademarks

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

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

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®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce®, 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.

Index

R

S

T

U

W

X

Product Number:

Printed in USA