About Connectivity Secure Shell

Connectivity Secure Shell is a client implementation of the Secure Shell protocol (SSH-2). It was developed to provide additional security for the existing suite of Hummingbird Connectivity products.

In addition to Secure Shell 2 support, Connectivity Secure Shell offers Connectivity Kerberos, Hummingbird's implementation of the Kerberos 4 and 5 protocols. It also offers Connectivity SSL which you can use to SSL-enable HostExplorer TN3270, TN5250, and VT sessions, as well as Hummingbird FTP and Classic FTP sessions.

Note:

- Connectivity Kerberos and Connectivity SSL are installed with Connectivity Secure Shell, and are also available independently as free downloads from the Hummingbird web site:
  `http://connectivity.hummingbird.com/getssl`
  `http://connectivity.hummingbird.com/getkerberos`
- This document describes only Secure Shell features. For information on Connectivity Kerberos, see Connectivity Kerberos Help.

Connectivity Secure Shell is comprised of the following components:

- Connectivity Secure Shell Management Console—This console is the hub of Connectivity Secure Shell functionality. Use it to create, configure, and launch Secure Shell sessions, and to monitor sessions that are running. It also provides access to Hummingbird Certificate and Key Manager.
- Connectivity Secure Shell engine—This is the SSH-2 client engine. It starts and maintains tunnel connections as configured in the management console. It also listens for port forwarding connections.
- Hummingbird Certificate and Key Manager—This management and data store component provides the Key Generation Wizard for generating Secure Shell key pairs for public-key authentication purposes. It also lets you generate self-signed certificates and certificate requests with the Certificate Creation Wizard. The Connectivity Secure Shell engine uses Certificate and Key Manager to access keys and certificates in the key and certificate stores.

Features and Functionality

You can use Connectivity Secure Shell to secure sessions initiated with other Hummingbird Connectivity products:

- Secure X Window sessions initiated with HostExplorer or Xstart.
- Secure VT terminal sessions initiated with HostExplorer.
- Secure FTP file transfers initiated with Hummingbird FTP and Classic FTP.

You can also secure connections initiated with third-party applications that run on TCP-based protocols such as TELNET, IMAP, and POP, to name a few.

Secure Shell Setting Configuration

Connectivity Secure Shell lets you configure a number of Secure Shell protocol settings. You can save the settings as tunnel profiles, which you can use to launch future sessions either from Connectivity Secure Shell itself, or from within Secure Shell-enabled Hummingbird Connectivity products.

The following is a sample of the settings you can configure:

- Choose from four supported authentication methods: Password, Keyboard interactive, Public/Private key, and Kerberos.
- Choose from several strong encryption algorithms.
- Choose from several Message Authentication Code (MAC) algorithms to ensure data integrity.
- Configure agent forwarding for remote authentication.
- Set tunnel parameters including window size, connection timeout, compression values, trace level, and so on.

Real-Time Monitoring

Connectivity Secure Shell console lets you monitor the Secure Shell sessions by displaying information about tunnels and channels connected to the client. This information is useful for debugging and troubleshooting purposes.

Certificate and Key Manager

The Certificate and Key Manager centralizes the management of public/private key pairs, user certificates, server certificates, and trusted root certificates. You can use it to import and export keys in different formats. Access Certificate and Key Manager from the Connectivity Secure Shell console.

Related Topics

About Secure Shell Sessions

Connectivity Secure Shell Management Console

About Certificate and Key Manager

About the Secure Shell Protocol

Secure Shell (SSH-2) is a TCP-based client/server protocol that provides the staples of secure networking:

- Authentication—Confirms the identity of both the server and the client user before initiating a Secure Shell session.
- Encryption—Encrypts information transferred over the network. Only the intended recipient can decrypt the information to view it.
- Data integrity—Ensures that the data that is sent is the same when it arrives at its destination.

A Secure Shell session, or tunnel, is an authenticated and encrypted SSH-2 connection initiated from a Secure Shell client to a host on the network that is running a Secure Shell server.

During the initial negotiation of this connection, both the server and client are authenticated. Server authentication is performed initially using public key exchange, and a number of authentication methods are supported for client authentication.

Note: In spite of this authentication, there remains the risk of a man-in-the-middle attack during the initial connection.

Once the tunnel is established, the information channelled through it is encrypted using any of a number of supported encryption ciphers including Blowfish, 3DES, CAST128, and the U.S. Advanced Encryption Standard (AES). To protect the integrity of the data, SSH-2 supports Hash Message Authentication Code (HMAC) algorithms.

Related Topics

About Secure Shell Sessions

About Certificate and Key Manager

Direct Secure Shell Session Scenario

This scenario requires that the applications whose sessions you want to secure are Secure Shell aware. Applications such as HostExplorer and Hummingbird FTP, for example, let users configure their terminal or FTP sessions to use any tunnels you have configured.

Although both client-side applications may use the same Secure Shell tunnel, each one would send and receive data on a separate Secure Shell channel within the tunnel. (Each tunnel may have only one X11-enabled channel.)

In the case of FTP communication, the Connectivity Secure Shell acts as an FTP server for the local FTP client, and as a SFTP client toward the remote host Secure Shell server.

Related Topics

About Secure Shell Terminals

About SFTP File Transfers

General Port Forwarding Scenario

In this scenario, local applications that communicate with a remote server need not be Secure Shell enabled. Instead, you can configure them to send their data to a local port. The Secure Shell session you configure and initiate with Connectivity Secure Shell listens on the port used by the applications and forwards the data to the remote Secure Shell server through the tunnel.

As in the previous scenario, each application data stream is sent between the Connectivity Secure Shell engine and the Secure Shell server through a separate channel. Data that is received by the Secure Shell server, is forwarded to the remote server applications.

Related Topics

About Secure Port Forwarding

X Window Port Forwarding Scenario

If you enable X11 port forwarding in a tunnel profile, when you start the tunnel, the Connectivity Secure Shell engine requests X forwarding when connecting to the Secure Shell server. If the server supports X forwarding, it runs an X11 proxy on the remote host. The $DISPLAY variable is configured so that initiated X applications connect to the X11 proxy, which sends the connection through the tunnel. What is displayed on your desktop is a terminal window from which you can run commands.

Note: The workstation is not limited to using the TELNET protocol to launch the X Window client on the server machine. It can use any other startup method, including REXEC, RSH, and RLOGIN.

Related Topics

About Secure X Window Sessions

SFTP Scenario

Built into Connectivity Secure Shell is an SFTP proxy server which you can configure to listen on multiple listening interfaces (IP address and port combinations). It is through these listening interfaces that the SFTP proxy server accepts inbound FTP connections (both control and data connections) as would an FTP server. It then interprets the incoming FTP commands and executes them as SSH2 SFTP commands through a tunnel connection to an SFTP server launched by the Secure Shell server on the target host.

Related Topics

About SFTP File Transfers

About Secure Shell Sessions

A Secure Shell session, or tunnel, is an encrypted network connection from a workstation to a host on the network that is running a Secure Shell server. In subsequent sections, you will find information on how to use tunnels to secure remote terminal logins and file transfers. You will also find information on how to direct otherwise insecure TCP-based protocols through an active tunnel to ensure that the data transmitted is secure.

Tunnels and Channels

Application data is carried over channels within tunnels. Establishing a new tunnel for each new application connection results in an unnecessary time overhead due to the authentication process required to establish each new tunnel. Therefore, Hummingbird applications that support Secure Shell attempt to create new channels within existing tunnels whenever possible. A tunnel is reused when its host address, port, and user ID values match those specified by the new connection.

In the case of SFTP connections, you may specify <> as the user name in the FTP client to suppress user name matching. Then, only the host address and port values must match for the new connection to reuse an established tunnel. This is useful if the established tunnel was configured with a blank user ID value.

Secure Shell Tunnel Profiles

Using the Connectivity Secure Shell Management Console, you can create and configure tunnel profiles. Each profile you create has the information required to launch a tunnel connection to a specific port on a specific host using specific authentication credentials. These profiles also record other tunnel configuration settings you may need to specify to suit different tasks, network configurations, and preferences.

You can access and launch tunnels either from Connectivity Secure Shell, or directly through Hummingbird applications that support Connectivity Secure Shell.

Related Topics

About SFTP File Transfers

About Secure Shell Terminals

About Secure X Window Sessions

Creating Tunnel Profiles

Before you can start a tunnel in Connectivity Secure Shell, you must create a tunnel profile. You can create as many profiles as is necessary to start Secure Shell sessions with different hosts and different settings.

Initially, profiles receive the default Secure Shell settings. However, you can select a profile and configure these settings to suit your needs.

To create a new tunnel profile with default settings:

1. In the Connectivity Secure Shell Management Console, open the Tunnels folder and click Connection. The three-paned Connection window is displayed.
2. Click the Open Tunnel button in the top pane. The Open Tunnel dialog box opens.
3. Click the Add New Tunnel button.
4. In the Add New Tunnel dialog box, provide the following information:
   - Tunnel Name—Provide a name that identifies the tunnel profile.
   - Host Name—Specify the IP address or fully qualified domain name of the Secure Shell server host to which you want the tunnel to connect.
   - User Name—Specify your login name for the host. You can also use the following variables to supply this value:

Variable	User Name Value Supplied
`$USERNAME$`	Workstation user’s login name.
`$username$`	Workstation user’s login name in lower-case characters only.
`$COMPUTERNAME$`	Name of the workstation computer. This is useful if the workstation account is set up as a general account.
`$computername$`	Name of the workstation computer in lower-case characters only. This is useful if the workstation account is set up as a general account.

   - TCP Port—Specify the host port to which the tunnel will connect.
5. Click OK.

You can now select the new tunnel profile and click Open to start the tunnel with the default settings. You can customize the tunnel settings at any time.

Related Topics

Configuring Tunnel Profiles

Copying Tunnel profiles

Instead of creating a new tunnel profile with default settings, you can create a new profile using a previously generated profile as the basis.

To create a tunnel profile based on another profile:

1. In the Connectivity Secure Shell Management Console, open the Tunnels folder and click Connection. The three-paned Connection window is displayed.
2. Click the Open Tunnel button in the top pane. The Open Tunnel dialog box opens.
3. Select the tunnel profile on which you want to base the new profile.
4. Click the Copy button.
5. Click the Paste button. A copy of the selected tunnel profile appears on the profile list with same name as the original. The name is followed by a number in brackets to distinguish it from the original. The numbers are incremented for each successive copy made to distinguish one copy from another.

You can now start the tunnel with the same settings as the original, or customize the tunnel settings first.

Related Topics

Configuring Tunnel Profiles

Configuring Tunnel Profiles

The Tunnel Profile dialog box lets you edit tunnel configuration settings. The settings are grouped into three categories:

Connection These settings let you specify information about the host to which you want to connect. You can also specify display and communication settings, environment variables, remote tasks, exit settings and trace settings.

Security These settings let you select encryption algorithms and Message Authentication Code (MAC) algorithms. You can also specify other parameters such as authentication type, password or user key.

Port Forwarding These settings let you specify hosts and configure listening and destination ports, and connection types for incoming and outgoing port forwarding. You can also specify settings for X11 port forwarding.

To modify tunnel profile settings:

1. In the Connectivity Secure Shell Management Console, open the Tunnels folder and click Connections. The three-paned Connection window is displayed.
2. Click the Open Tunnel button in the top pane. The Open Tunnel dialog box opens.
3. Select the tunnel profile you want to configure and click the Edit Tunnel Info button. The Tunnel Profile dialog box opens displaying the setting categories in the left pane.

   Tip: Remember that some tunnel profiles may be referenced by other applications such as HostExplorer or Exceed. If that is the case, consider creating a new tunnel profile.

4. Select the setting group you want to modify and make the necessary changes. Repeat for each group as required.

   For additional information on a particular setting, refer to the following topics:

   - For Connection settings

    - For Security settings
    - For Port Forwarding settings

5. Click OK.

Related Topics

Starting a Tunnel

Deleting Tunnel Profiles

You can selectively delete tunnel profiles listed in the Open Tunnel dialog box, or delete all of the profiles at once. Keep in mind that some tunnel profiles may be referenced by other applications such as HostExplorer or Exceed. Deleting them can affect sessions launched in these applications.

To delete tunnel profiles:

1. In the Connectivity Secure Shell Management Console, open the Tunnels folder and click Connections. The three-paned Connection window is displayed.
2. Click the Open Tunnel button in the top pane. The Open Tunnel dialog box opens.
3. Do one of the following:
   - To delete all listed profiles, click the Delete All button.
   - To delete a specific tunnel profile, select it and click the Delete button.
4. Confirm the deletion when prompted.

Starting a Tunnel

To start a tunnel from the Connectivity Secure Shell Management Console, you must first create a tunnel profile.

If you have created one or more tunnel profiles, you can start a Secure Shell tunnel by selecting a profile to open the tunnel according to the configuration settings saved in that profile. The Open Tunnel dialog box lists all of the tunnel profiles you have created.

Tip: Applications that support Connectivity Secure Shell also let you configure and start tunnels. For more information, see the application-specific section in this Help.

After starting a tunnel, you can use it to carry any number of different communication protocols, provided they each use a single socket.

Note: Most communication types that you initiate within the tunnel occur through their own channels. FTP is an example of a communication protocol that can use multiple connections.

If you start a tunnel to secure one connection, and then, using the same profile, attempt to start another tunnel to secure another connection, one of the following occurs:

- The active tunnel is re-used. A new channel is created within this tunnel for each connection that uses same tunnel profile.
- If you are using Connectivity Secure Shell or Exceed to start a tunnel, or if you enable X11 Port Forwarding when using another application such as HostExplorer, a new tunnel is created each time you create a connection using that profile.

To start a tunnel in Connectivity Secure Shell:

1. In the Connectivity Secure Shell Management Console, open the Tunnels folder and click Connections. The three-paned Connection window is displayed.
2. Click the Open Tunnel button in the top pane. The Open Tunnel dialog box opens.
3. Select the tunnel profile you want to use to start the tunnel and click Open. The tunnel status is displayed in the top pane of the Secure

Shell Management Console.

Related Topics

Creating Tunnel Profiles

Viewing Information about Tunnels

About Secure X Window Sessions

Connection Settings

General

Host Name—Specify a new IP address or fully qualified domain name of a Secure Shell server host to which you want the tunnel to connect. Every time you specify a new host, the previous host is added to the drop-down list so that you can select it again at a later time. The User Name and TCP Port values are not saved with the selected host. That is, if these values differ depending on the host, you must change them accordingly after you select a new host from the list. Consider creating a separate tunnel profile for each host.

User Name—Specify your login name for the host specified above. This name is used for authentication.

TCP Port—Specify the host port to which the tunnel will connect. The default port is 22.

Advanced

Default Window Size—Specify the amount of data that can be transferred before acknowledgement is required. The Connectivity Secure Shell engine must acknowledge receipt once the specified amount of data is transferred. The smaller the window size, the more acknowledgement is required.

Maximum Packet Size (KB)—Specify the maximum size of transferred packets. Generally speaking, smaller maximum packet sizes can increase system responsiveness. Larger maximum packet sizes are more suited for large file transfers, such as FTP.

Initial Buffer Allocation (KB)—Specify the initial size of the internal buffer allocations. Change this setting if recommended by Hummingbird Technical Support.

Connection Timeout (Seconds)—Specify the amount of time that Connectivity Secure Shell will wait for a host response before terminating a connection attempt. Increase this value if a host is slow to respond, or if the network is slow.

Protocol Keepalive Interval (Seconds)—If the server uses a heartbeat function to test sockets for inactivity, socket connections can be disconnected. This setting lets you specify the amount of time after connecting that Connectivity Secure Shell waits before sending an SSH2_MSG_IGNORE packet with a random amount of data to keep sockets active. You may also use this option to confound attempts at traffic analysis, a method used to deduce information that is helpful in connection attacks.

Level of Outbound Compression—Specify the degree of compression for outbound data. A value of `0` indicates no compression, and `9` indicates the highest level of compression.

Enable Socket Keepalive—Enable this option to allow the server to disconnect in the event that the connection fails. The applicability of this option is dependent on whether or not a keepalive mechanism is enabled on the server.

Enable Nagle Algorithm—This option is selected by default for telnet and most other applications. When using X-Windows through Secure Shell, however, you may experience better performance with the Nagle Algorithm disabled.

Note: If this option is cleared, Connectivity Secure Shell enables the TCP_NODELAY option in the TCP/IP protocol stack.

Prompt for Banner Message—Clear this option to disable the automatic display of the host’s banner message when you connect. If you enable this option, banner messages will be displayed if sent by the host.

Warning! In cases where one machine serves as the SFTP proxy server for a number of other machines on the local area network, banners from various hosts accumulate on the server machine. (One banner is posted per new connection.) To avoid this, FTP clients should use tunnel profiles in which the Prompt for Banner Message option is turned off.

Server Key Action

Host authentication with Secure Shell is performed by distributing and storing keys when a host is first accessed. For each subsequent connection attempt, the keys are compared to ensure that they match.

You can configure the action to be taken when a host sends an unidentifiable public key to Connectivity Secure Shell for authentication:

- If the host address/port number entry is not found in the server key, then Connectivity Secure Shell performs the action that you specify in the New Server Key Action drop-down list.
- If the host address/port number entry is found, but the public key no longer matches (has changed), then Connectivity Secure Shell performs the action that you specify in the Changed Server Key Action drop-down list.

  Tip: Beware of changed keys as they may signal a potential security breach.

Select one of the following options in each drop-down list:

- Prompt—The user is prompted with the following choices. This option is the default.
- Add (Replace) Key and Connect—The key is added or replaced and the connection continues. If this is selected, the user is not notified.
- Do Not Add (Replace) Key and Connect—No change is made to the key database and the connection continues. If this is selected, the user is not notified.
- Deny Connection—The connection fails. The user is not notified.

  Note: In all of the above cases, if logging is turned on, all actions are logged.

Other

Generate trace files if recommended by Hummingbird Technical Support. Tracing generates large files and adds significant overhead to the communication process, which can increase the memory requirements. In some cases, traces can cause connections to work incorrectly with default timeout values. You can choose the following trace levels:

Tip: For information on timeout values, see Advanced.

- Basic—Select to log only major events such as connection, disconnection, errors. This option has the least impact on performance.
- Detail—Select to log all important actions and events with a high level of detail. The data traffic itself is not logged.
- Verbose—Select to perform a trace that includes all the elements of a Detail trace, plus all of the packets read and written from the network. This trace lets you view encrypted as well as plaintext packet contents so that you can match up trace events with protocol analyzer output.

Warning! The output of a Verbose trace can easily reach hundreds of megabytes. This trace can have a significant impact on performance.

Security Settings

Encryption Algorithm

Encryption is negotiated to be the same in both directions, although the Secure Shell protocol allows incoming and outgoing streams to have different encryption types. The algorithm used to select a cipher is the first algorithm that is common to both the client and server, as determined by the order of the client’s list. If you have a preference, you can modify the list.

By default, all algorithms are enabled. You can enable or disable them by selecting Specify List on the drop-down list. Use the arrow buttons to alter the order in which the algorithms are negotiated. Click the Reset All button to return to the default settings.

MAC Algorithm

Message Authentication Code (MAC) algorithms are integrity checksums used to ensure that the contents of a packet have not been altered during transmission. The Secure Shell protocol allows incoming and outgoing streams to have different MAC types. The algorithm used to select a MAC is the first algorithm that is common to both the client and server, as determined by the order of the client. If you have a preference, you can modify the list.

By default, all MAC algorithms are enabled. You can enable or disable them by selecting Specify List on the drop-down list. Use the arrow buttons to alter the order in which the algorithms are negotiated. Click the Reset All button to return to the default settings.

Authentication

Select the type of authentication you want to use to identify yourself to the remote host. Use the arrow buttons to alter the order in which the authentication methods are attempted. Click the Reset All button to return to the default settings.

The authentication methods you enabled are tried once in the order you specify. The tunnel is established with the first method that succeeds. If all authentication methods fail, the tunnel disconnects.

Use Entered Password—This method uses your password to log into the remote host. When you select this method, the Password box appears in which you can specify the password.

Use Selected User Key—This method requires you to generate a public/private key pair with Hummingbird Certificate and Key Manager. When you select this method, the Browse button appears. Click it to access the Select User Key dialog box which lets you create a key or select a previously generated key.

> **Allow Agent Forwarding**—This option is available only if the Use Selected User Key authentication type is selected and you have selected a key. It lets remote hosts send authentication requests to Connectivity Secure Shell. For example, if you connect to a host, and then create a Secure Shell connection to a second host, the second host can send an authentication request to Connectivity Secure Shell through the original host. When the client receives a remote authentication request from the Secure Shell server, it returns all of the public keys in the user key store. The server selects a key and returns an authentication request to the client. When public key authentication is used, the client attempts to open all keys in the user key store. If each has a different passphrase, you will receive multiple prompts for keys. If you use the same passphrase for all keys, you will be prompted only once.

> **Keyboard Interactive**—This option provides a generic keyboard method for use with supported authentication tools such as Smart

Cards.

Kerberos—This method requires you to configure Kerberos settings. These settings are displayed when you select the method. To alter the settings, click the Kerberos Settings icon to open the Kerberos Settings dialog box. Select one of the following options:

GSSAPI Service Name—The name of the service principal defined on the target host. This is the default principal as specified in the standards document. Ask your system administrator if your host configuration specifies a different service principal name.

Note: `Host` is commonly used for shells, while `ftp` is commonly used for file transfer. If you select ftp, and there is no service principal for ftp on that machine, you will not connect to the host.

Kerberos Client—You can select Hummingbird Connectivity Kerberos or MIT Kerberos if installed.

Use HMS2MIT—Imports Kerberos tickets (for Kerberos client authentication) from the Microsoft ticket store to the Kerberos ticket store.

Delegate Credentials—Exports the ticket-granting ticket (acquired at login, or via kinit or your Kerberos client) to the host. Once you are logged onto the host, you can check for the `krbtgt@Realm@Realm,` which is the ticket-granting ticket, in your Kerberos cache. (On a Unix host, issue the klist command.)

Port Forwarding Settings

Outgoing

This describes outgoing connections from the local computer (that is, TCP connections that the local computer forwards from a specified local port to the specified port on the remote host computer). The display pane lists all outgoing port definitions you create. Use the buttons in this pane to add, edit, delete, copy, paste, and change the order of outgoing ports.

Incoming

This describes incoming connections to the local computer (that is, TCP connections that the remote host forwards from a specified remote port to the specified port on the local computer). The display pane lists all of the incoming port definitions you create. Use the buttons in this pane to add, edit, delete and change the order of incoming ports.

X11

These settings let you enable and configure X11 port forwarding. If you enable X11 port forwarding in a tunnel profile, when you start the tunnel, the Connectivity Secure Shell engine requests X forwarding upon connection to the Secure Shell server. If the server supports X forwarding, the Connectivity Secure Shell engine listens on the specified port.

You can also enable and disable X11 Authentication. When enabled, X11 authentication ensures that no other user account is permitted to open an X window on the Connectivity Secure Shell workstation. Advanced users may need to disable this option if, for example, they want to open an X window after they `su` (substitute user) to a different account.

Warning!

- If you `su` without a hyphen (-), the $DISPLAY variable is retained. (This variable is required to run an X11 command back to the PC.)
- If you `su` with a hyphen (-), then the $DISPLAY variable is not set. You will receive a "`Can’t open display`" message if you attempt to initiate a connection. You must set the $DISPLAY variable.

In either case, if X11 authentication is enabled, the X window connection will fail.

Related Topics

Setting Up an Outgoing Port Forwarding

Setting Up an Incoming Port Forwarding

About Secure X Window Sessions

About Secure Shell Terminals

Typically, terminal connections to remote UNIX hosts use the TELNET protocol. TELNET, however, presents several vulnerabilities that an attacker can exploit to gain access to sensitive data in transit over the network. Not only can attackers view the data to acquire secret login information, for example, but they can also alter and reroute the data.

When you install Connectivity Secure Shell alongside HostExplorer, you gain the option of creating Secure Shell-enabled HostExplorer VT session profiles. You can then use these session profiles to launch secure VT sessions, which are tunnelled through the Secure Shell protocol instead of using unsecured TELNET connections.

About Creating Secure Shell-Enabled VT Profiles

Securing a HostExplorer VT session requires that you create a VT session profile in HostExplorer that specifies Secure Shell, rather than TELENET, as the connection protocol. You must also provide the parameters necessary to establish a Secure Shell tunnel to the target host.

New Profile dialog box □

When creating a Secure Shell-enabled VT session profile, the New Profile dialog box lets you specify all the parameters required to start the session quickly with default tunnel settings. From this dialog, you can also access options that let you customize the settings to varying degrees. You may, for example, want to change the authentication method from its default, which is password authentication, to user key authentication. To do so, use the Properties button to access the session profile properties. This dialog box lets you set custom port, user name, and authentication parameters. You can also enable X11 port forwarding.

Session Profile dialog box □

For even more control over tunnel settings, you also have the option of specifying a custom Connectivity Secure Shell tunnel profile from which the tunnel settings will then be read into the HostExplorer profile.

Related Topics

Creating Secure Shell-Enabled VT Session Profiles

Using Tunnel Profiles to Secure VT Sessions

Overriding Tunnel Profile Parameters

Creating Secure Shell-Enabled VT Session Profiles

The New Profile dialog box lets you specify all the parameters required to start a Secure Shell-enabled VT session with default tunnel settings. If you want to customize these settings, you can also access the session profile properties.

To create a Secure Shell-enabled VT session profile:

1. In the HostExplorer Open Session dialog box, use the Folder drop-down list to specify a location for the new session profile, and then click the Create New Profile button. The New Profile dialog box opens.
2. In the Profile Name box, type a name for the new profile. This name cannot contain any of the following characters: `\ / : * ? “ < > |`
3. In the Profile Type drop-down list, select VT Display.
4. If you have created a theme, you can select it from the Theme drop-down list. For more information on creating themes, see HostExplorer help.
5. In the Connect By drop-down list, select Secure Shell.
6. In the Host Name box specify the name of the host to which you want to connect.
7. Enable the Connect option if you want the session to start automatically after you create the profile.
8. At this point, you have provided the minimum requirements to establish a Secure Shell tunnel to the remote host. This configuration uses default settings. To view or customize the default settings, continue with this procedure. Otherwise, click OK.
9. Click Properties to access the Secure Shell options in the Session Profile dialog box.

   Note: The Properties button is divided in two. Click the right side of the button and click Connection to go directly to the Secure Shell options. Alternatively, you can click the left side of the button, expand the Connection folder, and click Secure Shell.

.0. The Secure Shell page is displayed in the right pane. Specify the following connection parameters:

Tip: If you specified a tunnel profile, these boxes are populated with the parameters specified in the tunnel profile. Any changes you make to the parameters will be saved in the HostExplorer profile as tunnel parameter overrides.

- Host Name—Specify a new IP address or fully qualified domain name of a Secure Shell server host to which you want the tunnel to connect.
- Port—Specify the host port to which the tunnel will connect.
- User Name—Specify your login name for the host specified above. This name is used for authentication.

Variable	User Name Value Supplied
`$USERNAME$`	Workstation user's login name.
`$username$`	Workstation user's login name in lower-case characters only. For UNIX logins.
`$COMPUTERNAME$`	Name of the workstation computer. This is useful if the workstation account is set up as a general account.
`$computername$`	Name of the workstation computer in lower-case characters only. This is useful if the workstation account is set up as a general account. For UNIX logins.

- Authentication—Click the Authentication Settings button to select the type of authentication you want to use to identify yourself to the remote host. For more information on configuring authentication, see Authentication.
- Enable X11 Port Forwarding—Select this option if you want to use the Secure Shell tunnel to secure X11 sessions. If you do not select this option, you cannot start an X11 session in HostExplorer unless you have configured the DISPLAY properly, and have launched Exceed on your workstation.

.1. Click OK. The new HostExplorer session profile appears in the Open Session dialog box.

Related Topics

Using Tunnel Profiles to Secure VT Sessions

Overriding Tunnel Profile Parameters

Using Tunnel Profiles to Secure VT Sessions

When you create a Secure Shell-enabled VT session profile in HostExplorer, you have to provide the parameters necessary to establish a Secure Shell tunnel to the target host. To do so, you have the option of calling the parameters from a Connectivity Secure Shell tunnel profile instead of specifying them directly as described in the previous procedure.

Connectivity Secure Shell tunnel profiles contain all the parameters necessary to start and configure a Secure Shell tunnel. When you create your HostExplorer profile, you can configure it to call the necessary connection parameters from the tunnel profile of your choice. You can create tunnel profiles with Connectivity Secure Shell before hand, and then select one to be used for your VT session, or you can create them on the fly while you configure your HostExplorer VT session profile.

To create a Secure Shell-enabled VT session profile:

1. In the HostExplorer Open Session dialog box, use the Folder drop-down list to specify a location for the new session profile, and then click the Create New Profile button. The New Profile dialog box opens.
2. In the Profile Name box, type a name for the new profile. This name cannot contain any of the following characters: `\ / : * ? “ < > |`
3. In the Profile Type drop-down list, select VT Display.
4. If you have created a theme, you can select it from the Theme drop-down list. For more information on creating themes, see HostExplorer help.
5. In the Connect By drop-down list, select Secure Shell.
6. To specify the tunnel profile you want to use to establish the Secure Shell tunnel for the terminal session, do one of the following:

   Tip: After you specify the tunnel, you can view the Secure Shell Connection settings by clicking Properties. The Host Name, Port, User Name, Authentication and X11 settings displayed are those called from the tunnel profile you specified. If you modify the parameters, they are saved in the HostExplorer profile as tunnel parameter overrides.

- Type the name of the tunnel profile (`*profilename*.csp`) into the Tunnel Profile box.
- Click the Browse Tunnel Profiles button to select from a list of your tunnel profiles, and then click Open.
- If you have not yet created a tunnel profile, you can click the Browse Tunnel Profiles button, and then click the Add New Tunnel button in the Open Tunnel dialog box. For more information on creating a Tunnel profile, see Creating Tunnel Profiles.

7. Click OK. The new HostExplorer session profile appears in the Open Session dialog box.

Related Topics

Overriding Tunnel Profile Parameters

Overriding Tunnel Profile Parameters

When creating a Secure Shell-enabled VT session profile you have three options for specifying the connection parameters:

- The first option is to specify each parameter directly in the HostExplorer profile, as described in About Creating Secure Shell-Enabled VT Profiles. In this case, the parameters are stored in the HostExplorer profile itself
- Otherwise, you can specify the connection parameters indirectly by associating a tunnel profile with your HostExplorer profile, as described in Using Tunnel Profiles to Secure VT Sessions.

  In this case, the tunnel profile name is referenced in the HostExplorer profile, and connection parameters are retrieved from the tunnel profile when the connection is established.
- The third option is a combination of the previous two. That is, you can specify a tunnel profile, and then modify one or more of the parameters called from it.

In this case, any modifications you make to connection parameters are saved in the *HostExplorer* profile as overrides to the tunnel profile. The tunnel profile itself is not altered. When you start a session, the tunnel profile values are read and applied first, and then the HostExplorer values override these values.

To override tunnel profile parameters:

You can use the Session Profile dialog box to override the tunnel profile parameters either while creating a Secure Shell-enabled HostExplorer profile, or while editing an existing profile.

The Host Name, Port, User Name, Authentication, and Enable X11 Port Forwarding settings are called from the specified tunnel profile. Specify new parameters to override the tunnel values. To revert to the original tunnel values, click the Reload button.

Editing Secure Shell Settings for VT Session Profiles

After you create a session profile, you can edit the settings at any time. If a session is running, you must restart it for your modifications to be applied.

To edit a Secure Shell-enabled VT session profile:

1. In the HostExplorer Open Session dialog box, right-click the profile you want to edit and click Properties in the pop-up menu.
2. In the Session Profile dialog box, expand the Connection folder and click Secure Shell. The Secure Shell page appears in the right pane.
3. The Tunnel Profile box identifies the tunnel profile, if any, currently associated with the VT session profile you are editing.
   - To add or to specify a different tunnel profile, click the Browse Tunnel Profiles button.
   - To remove the tunnel profile, delete it from the Tunnel Profile box.
   - To edit the tunnel profile settings, click the Edit This Tunnel Profile button and refer to Connection Settings for information on specific settings.

     Tip: You can also access and edit tunnel profiles from the Secure Shell Management Console. For more information, see .
   - To create a new tunnel profile, click the Create A New Tunnel Profile button and refer to To create a new tunnel profile with default settings:.
   - To reload the Host Name, Port, User Name, Authentication, and X11 settings specified in the tunnel profile, click the Reload button.
4. Modify individual parameters as necessary, bearing in mind that if you specified a tunnel profile in the previous step, any changes you make to the Host Name, Port, User Name, Authentication, and X11 parameters will be saved in your HostExplorer profile as overrides of the tunnel profile parameters.
   - Host Name—Specify a new IP address or fully qualified domain

name of a Secure Shell server host to which you want the tunnel to connect.

- Port—Specify the host port to which the tunnel will connect.
- User Name—Specify your login name for the host specified above. This name is used for authentication.

Variable	User Name Value Supplied
`$USERNAME$`	Workstation user's login name.
`$username$`	Workstation user's login name in lower-case characters only.
`$COMPUTERNAME$`	Name of the workstation computer. This is useful if the workstation account is set up as a general account.
`$computername$`	Name of the workstation computer in lower-case characters only. This is useful if the workstation account is set up as a general account.

- Authentication—Click the Authentication Settings button to select the type of authentication you want to use to identify yourself to the remote host. For more information on configuring authentication, see Authentication.
- Enable X11 Port Forwarding—Select this option if you want to use the Secure Shell tunnel to secure X11 sessions.

5. Click OK.

About SFTP File Transfers

The Secure Shell protocol alone was not designed to perform secure file transfers. Strictly speaking, it is not a file transfer protocol. Furthermore, traditional file transfer protocols, such as FTP, are not amenable to being channelled easily over Secure Shell as they require more than one socket connection per session.

To make secure file transfer possible over Secure Shell, it was necessary to implement a tool that could execute FTP commands and file transfers over a single socket connection. That tool is the SFTP protocol, which is layered on top of the Secure Shell protocol to move files securely between FTP client applications and their target hosts.

The Connectivity Secure Shell implementation of the SSH2 protocol includes an SFTP tool, making it possible for you to secure FTP connections configured and initiated with Hummingbird FTP, Hummingbird's FTP client application, as well as with third-party FTP clients.

Overview of the SFTP File Transfer Process

Built into Connectivity Secure Shell is an SFTP proxy server which you can configure to listen on multiple listening interfaces (IP address and port combinations). It is through these listening interfaces that the SFTP proxy server accepts inbound FTP connections (both control and data connections) as would an FTP server. It then interprets the incoming FTP commands and executes them as SSH2 SFTP commands through a tunnel connection to an SFTP server launched by the Secure Shell server on the target host.

Once the connections are established, the files you send and retrieve move securely between the SFTP proxy server and the host’s SFTP server.

Securing Hummingbird FTP Sessions

Securing an FTP session requires that you create an FTP session profile in Hummingbird FTP that specifies Secure Shell as the security method. For Connectivity Secure Shell to establish a Secure Shell tunnel to the target host of the FTP session, it also requires that you specify certain connection parameters including:

- host address
- user name
- authentication method
- authentication method parameters such as a password or key

When you configure an FTP profile to use Secure Shell, all parameters used to establish the Secure Shell tunnel are stored in the FTP profile.

Using Tunnel Profiles to Secure FTP Sessions

Instead of specifying the connection parameters explicitly when you create a Secure Shell-enabled FTP session profile, you also have the option of specifying these parameters by associating a tunnel profile with your FTP profile. Tunnel profiles contain all the connection parameters needed to establish the Secure Shell tunnel to the remote host.

You can also do a combination of both. That is, in the FTP profile, you can specify a tunnel profile from which to load the necessary parameters. Then, in the same FTP profile, you can specify and save parameters that override some of those loaded from the specified tunnel profile.

When you launch a Secure Shell-enabled FTP session, Hummingbird FTP uses a combination of the original parameters in the tunnel profile and the Secure Shell parameter overrides stored in the FTP session profile.

Related Topics

Creating Secure Shell-Enabled FTP Profiles

About Securing Third-Party FTP Clients

Creating Secure Shell-Enabled FTP Profiles

Hummingbird FTP profiles are saved session configurations. Typically, they contain the settings to connect to an FTP server on a remote machine to and from which you want to transfer files. In the case of Secure Shell-enabled FTP profiles, however, you must provide the parameters to establish a Secure Shell tunnel to the target host.

To create a Secure Shell-enabled FTP profile:

1. In the Hummingbird Neighborhood window, click New FTP Profile. The FTP Site Properties dialog box opens displaying the connection parameters on the General page.
2. Type the host address, user name and password.
3. Select Secure Shell from the Security drop-down list.

   At this point, you have provided the minimum requirements to establish a Secure Shell tunnel to the remote host. This configuration supports password authentication only. If you want to configure different authentication methods, or want to use a tunnel profile, proceed with the next step. Otherwise, click OK.
4. Click Configure. The SSH properties dialog box opens.
5. To specify a tunnel profile, do one of the following. Otherwise, proceed to the next step.
   - Type the name of the tunnel profile (`*profilename*.csp`) into the CSS Tunnel Profile box.
   - Click Browse to access the Connectivity Secure Shell Open Tunnel dialog box which lets you select from a list of your tunnel profiles. Then click Open.
   - Click Create to access the Connectivity Secure Shell Add New Tunnel dialog box, which lets you create a new tunnel profile with default settings. For more information, see To create a new tunnel profile with default settings:. After you create the new tunnel, you can click Edit in the Secure Shell Configuration dialog box to access the Connectivity Secure Shell Tunnel

Profile dialog box, which lets you edit the parameters in the tunnel profile you just created. For more information, see Configuring Tunnel Profiles.

Note: Every time you select a new tunnel profile, change the name of a tunnel profile, or click Reload, all override values you have specified in the FTP profile are reset.

6. To configure the authentication method, click Config and refer to Authentication. Otherwise proceed to the next step.

   Note: If you specified a tunnel profile, any changes you make to the Authentication Methods here will be saved in the FTP profile as overrides of the authentication parameters specified in the tunnel profile. The tunnel profile will not be altered.

7. In the Secure Shell Configuration dialog box, click OK. The General page now displays the parameters as you configured them in the previous two steps.

   Note: Some parameters may have no specified value depending on your previous configurations.

8. You can modify the following parameters on the General page. Otherwise, click OK to save the FTP profile as it is.

   Note: If you specified a tunnel profile, any changes you make to the following parameters are saved in the FTP profile as overrides of the parameters specified in the tunnel profile. The tunnel profile will not be altered.

   - Host Address—Specify the IP address or fully qualified domain name of the host to which you want to connect.
   - TCP/IP Port—Specify the host port to which the Secure Shell tunnel will connect.
   - User Name—Specify your login name for the host. Use “<>” to suppress user name matching for tunnel reuse. For more information on tunnel reuse, see Tunnels and Channels.
   - Password—If you are using password authentication, specify your password for the host.

9. Click OK. The FTP profile is saved. You can now use it to launch a secure FTP session.

About Securing Third-Party FTP Clients

In addition to securing Hummingbird FTP client sessions, Connectivity Secure Shell lets you secure FTP sessions established with third-party FTP clients, provided they let you send arbitrary FTP commands.

Securing an FTP connection requires you to perform the following steps:

- Start the Connectivity Secure Shell proxy server and configure it to listen on a specific address and port. For more information, see Setting up the SFTP Proxy Server for Third-Party Clients.
- Configure the FTP client to connect to the SFTP proxy server. For more information see Configuring the FTP Client Connection.
- Connect the client to the SFTP proxy server, and then follow your FTP client's instructions to establish an FTP connection.
- Instruct the SFTP proxy server to establish an SFTP connection to the remote server through a Secure Shell tunnel. For more information, see Establishing an SFTP Connection.

You can use command line arguments to start the SFTP proxy server, and to instruct it to listen for FTP connections on the same interface to which you configured the FTP client to send its connection requests.

Instead of connecting directly to the target host, your FTP client must be configured to connect to Connectivity Secure Shell's SFTP proxy server using the specified listening interface (the IP address and port on which the SFTP proxy server is listening). You must also configure it to send the necessary instructions to the SFTP proxy server to start a Secure Shell connection to the target host.

Configuring the FTP Client Connection

To launch a secure FTP session, you must configure your FTP client to connect to a listening interface that you have, or will, set up on the SFTP server. A listening interface is comprised of two parts: the IP address of the machine on which the SFTP server runs (that is, the machine on which Connectivity Secure Shell is installed), and an unused port.

Some FTP client applications, let you type the entire listening interface, as *`ipaddress`*:*`port,`* into the box where you would normally specify the target host's address. Most FTP applications have a dedicated port parameter, in which case, you can specify them separately. When attempting to make these configuration changes, bear in mind that some FTP client applications may not offer this functionality.

Related Topics

Setting up the SFTP Proxy Server for Third-Party Clients

Establishing an SFTP Connection

An FTP client connected to the SFTP proxy server can instruct the server to establish an SFTP connection by sending it a `SITE` connect request. The syntax for the `SITE` connect request is as follows:

```
SITE CONNECT [parametername=value,]...
```

where *`parametername`* is a parameter recognized by the SFTP proxy server, and *`value`* is the value you want to send for the specified parameter. For a list of available parameters, see the following table.

Parameter Name	Value
PROFILE	Specify the tunnel profile `(.CSP)` you want to use to establish the tunnel connection. You can type the fully qualified path to the profile, or, if the profile is located in the default location, you can type: `$PROFILEDIR$\`*`profilename`*`.csp`
USER	Specify the user name for the host to which the tunnel will connect.
PASSWORD	Specify the password to be used if password authentication is requested. See the AUTHMETHOD parameter.
HOST	Specify the IP address or name of the target host. It is to this host that Connectivity Secure Shell will establish a tunnel connection.
PORT	Specify the host port to be used for the Secure Shell tunnel connection.
USERKEY	Specify the key fingerprint to be used if user-key authentication is requested.
AUTHMETHODS	Specify which authentication method(s) to use. The string will consist of space-separated numbers, each of which represents an authentication method. `0`—Password protection `1`—User Key `3`—Keyboard Interactive `4`—Kerberos authentication
USEHMS2MIT	Specify a value of `1` to populate the Kerberos ticket store with tickets from the Microsoft ticket store. Used for Kerberos authentication only.
KERBCLIENT	Specify a value of `0` to instruct Connectivity Secure Shell to use the

	Hummingbird Kerberos client. Specify a value of 1 to use the MIT Kerberos client. Used for Kerberos authentication only.
KERBSERVICE	Specify a string containing the Kerberos service name to be used for Kerberos authentication. Used for Kerberos authentication only.
AGENTFORWARD	Specify a value of 1 to enable agent forwarding.
KERBDELEGATE	Specify a value of 1 to enable the delegation of Kerberos credentials when Kerberos authentication is used. Used for Kerberos authentication only.
REMOTECLIENT	Specify a value of 1 to indicate that the FTP client does not reside on the same machine as the SFTP proxy server. If this is the case, see Instructing the Remote SFTP Proxy Server to Establish an SFTP Connection.
SERVERKEYACTION	Specify the action you wish the SFTP proxy server to take when a new or changed server key is returned during tunnel authentication. Set this value as follows to specify the action: 0—Prompt (The SFTP Proxy server will prompt the user for the necessary action.) **Note:** If this parameter is used in conjunction with REMOTECLIENT, then the prompt is delivered to the client on an FTP 300 level reply. The action should be reported back to the SFTP Proxy server using the SITE AUTHREPLY extension. 1—Add Server Key to keystore and connect 2—Do not add Server Key to keystore and connect 3—Do not add Server Key to keystore and cancel connection

Setting up the SFTP Proxy Server for Third-Party Clients

For third-party FTP clients to connect to the SFTP proxy server, the server must be started and configured to listen on a convenient and accessible listening interface. A listening interface is comprised of two parts: the IP address of the machine on which the SFTP server runs, and an unused port.

The SFTP proxy server resides in Hummingbird's Connectivity Secure Shell server (`humshsrv.exe`). You can use the following command line arguments for HUMSHSRV to start the SFTP proxy server and to configure it to listen for FTP client connection requests.

Command	Definition
-startsftp	Instructs HUMSHSRV to load into memory and to start the SFTP proxy server. An icon appears in the system tray if the server is started successfully. By default, the proxy is configured to listen on `localhost:0`, in which case, the interface used will be determined as follows: ▪ If the server is already listening on a local port, that interface is used. ▪ If there are no active local interfaces, and you do not specify another interface, the server uses the next unused port on the local machine. You can use the `-i` and `-p` command line arguments to specify a different IP address and/or port. Separate all commands following the -startsftp command with spaces. **Note:** If you require more than one interface, you can call HUMSHSRV with `-startsftp` repeatedly, once for each unique IP address/port combination. Only one instance of an interface can be active at one time.
-i <address>	Instructs the SFTP proxy server to listen for FTP connections on the specified address instead of the default address, `localhost` (`127.0.0.1`).
-p <port>	Instructs the SFTP proxy server to listen for FTP connections on the specified port instead of the default port, `0.` A port value of `0` instructs the server to start an arbitrary unused port, unless the server is already listening on an active interface with the same address as the one being requested. In which case, that interface is used.
-showinterface	Instructs HUMSHSRV to show a message box that lists the active listening interfaces on which the SFTP server is listening. The interfaces are listed as *`ipaddress`*`:`*port*. You can configure FTP clients to connect to any listed interface.

Creating Listening Interfaces with the Console

In addition to using command line arguments, you can use the Connectivity Secure Shell Console to configure listening interfaces on the SFTP proxy server. You can also use the console to edit and perform other interface management tasks.

To create a listening interface with the Connectivity Secure Shell Console:

1. In the Connectivity Secure Shell Management Console, click Listening Interfaces under SFTP in the left pane. The SFTP pane appears on the right.
2. Click the Open Listening Interface button.
3. In the Open Listening Interface dialog box, click the Add New Listening Interface button.
4. In the Add New Listening Interface dialog box, provide the following information:
   - In the Interface box, specify the address which the SFTP proxy will use to listen for inbound FTP connections. If the server will serve only those FTP clients that are running on the same machine, enter `localhost` or `127.0.0.1`. If the server will serve FTP clients running on remote machines, then enter the server machine’s network name or IP address.
   - In the Port box, specify the port on which you want the SFTP proxy server to listen. If you specify a port value of 0, an arbitrarily selected port will be used.
   - Enable Auto Start if you want the listening interface to be open (active) whenever the SFTP proxy server is running. If you do not enable this option, you must select and open the interface before attempting to connect to the server through it.
5. Click OK. The new interface is listed in the Open Listening Interface dialog box.
6. If you have created multiple listening interfaces, you can use the Move Up and Move Down buttons to change the order in which the

interfaces are listed. This is useful if you do not configure your FTP client to use a specific listening interface. In such a case, the FTP client uses the first available interface on the list.

7. You can open the listening interface by clicking Open. If you do not require the listening interface to be active at this time, click Cancel.

Related Topics

Managing Listening Interfaces

Setting up the SFTP Proxy Server for Third-Party Clients

Managing Listening Interfaces

You can perform the following tasks to manage existing listening interfaces. For more information, see the procedures following the list.

- Open and close interfaces to control their availability for FTP client connections.
- Edit listening interface settings.
- Delete listening interfaces.
- Configure the order in which interfaces are listed. This is useful if you do not configure your FTP client to use a specific listening interface. In such a case, the FTP client uses the first available interface on the list.

To open a previously configured listening interface:

1. In the Connectivity Secure Shell Management Console, click Listening Interfaces under SFTP in the left pane. The SFTP pane appears on the right, and lists all currently open listening interfaces.
2. At the bottom of the pane, click the Open Listening Interface button. The Open Listening Interface dialog box appears listing all previously configured interfaces.
3. Select the interface you want to open, and click Open.

The selected interface appears in the SFTP interface pane, and is now available to accept connections from FTP clients.

To close an open listening interface:

In the Connectivity Secure Shell Management Console, click Listening Interfaces under SFTP in the left pane. The SFTP pane appears on the right, and lists all currently open listening interfaces. Select the interface you want to close, and click the Unload button. To close all open interfaces, click the Unload All button.

To edit, delete, or arrange the order of listening interfaces

1. In the Connectivity Secure Shell Management Console, click Listening

Interfaces under SFTP in the left pane. The SFTP pane appears on the right.

2. Click the Open Listening Interface button.
3. In the Open Listening Interface dialog box you can do the following:
    - To add a new listening interface, click the Add New Listening Interface button, and provide the required information. For more detailed information, see Creating Listening Interfaces with the Console.
    - To edit a listening interface, click the Edit Listening Interface Information button and make the necessary changes to the Interface, Port, or Auto Start settings.
    - To delete listening interfaces, select the interface you want to delete and click the Delete button, or click the Delete All button.
4. Click Open.

Using a Remote SFTP Proxy Server

Because the FTP client and SFTP proxy communicate through TCP/IP, they can reside on different machines. For example, you can set up a single machine that is running the SFTP proxy server to be used as a gateway for all machines on the Local Area Network.

To configure Hummingbird FTP for a remote SFTP proxy:

1. Follow all but the last step of the procedure for configuring an Secure Shell-enabled FTP profile. See Creating Secure Shell-Enabled FTP Profiles.
2. On the Firewall/Proxy page of the FTP Site Property dialog box do the following:
   a. From the Firewall Type drop-down list, select Hummingbird CSS SFTP Proxy.
   b. In the Firewall Address box, type the IP/address of an SFTP proxy server's listening interface.
   c. In the Port box, specify the port of the SFTP proxy server's listening interface.

   Note:

   - If you specified a tunnel profile when creating the FTP profile, it must reside on the SFTP proxy server machine.
   - The SFTP proxy server must be configured to listen on an accessible listening interface comprised of a proper IP address or machine name (not localhost or the loopback address).

To configure third-party FTP clients for a remote SFTP proxy server:

See the procedure Configuring the FTP Client Connection.

Note: The SFTP proxy server must be configured to listen on an accessible listening interface comprised of a proper IP address or machine name (not localhost or the loopback address).

Instructing the Remote SFTP Proxy Server to Establish an SFTP Connection

In the process of establishing a Secure Shell tunnel, Connectivity Secure Shell may require user interaction. When the FTP client resides on the same machine as the SFTP proxy server, the server posts the necessary dialogs to prompt for information. However, when the FTP client is on a different machine, the SFTP proxy server initiates these interactions by means of authentication requests. In such a case, you can supply the necessary information using the SITE AUTHREPLY extensions.

To set up an SFTP connection with a remote SFTP proxy server:

Follow the instructions provided in Establishing an SFTP Connection, ensuring that you set the REMOTECLIENT parameter to a value of 1. To provide the information requested by the server, you can use the SITE AUTHRPLY extensions.

Using scp2

Hummingbird scp2 (Secure Copy) is a command line program used to secure file transfers over a network using the SSH2 protocol. Two methods of authentication are supported: password and public key.

The following information is also available from the scp2 usage information (scp2 –h):

scp2 usage:

```
scp2 [-aBCdhIpQqruV] [--auto] [--bin] [-b buffer-size] [-c cipher]
     [-g fingerprint] [-m mask] [--overwrite[=no]]
     [-P ssh2-port] [--SKA action] [--tracefile file] [-T level]
     [[user[#password]@]host[#port]:]file ...
     [[user[#password]@]host[#port]:]file_or_dir
```

Option	Action
`-a`	Transfer files in ascii mode.
`--auto`	Automatically select transfer type.
`--bin`	Transfer files in binary mode (default).
`-B`	Set batch-mode on.
`-b buffer-size`	Define maximum buffer size for one request. Default is 32768 bytes.
`-C`	Set compression on. Default is off.
`-c cipher`	Select encryption algorithm. Multiple -c options are allowed, and a single -c flag can have only one cipher.
`-d`	Force target to be a directory.
`-g fingerprint`	Select user key to use for authentication.
`-h`	Display help.
`-I`	Display interactive prompt before overwriting.
`-m mask`	Set the default file permission mask.
`--overwrite`	Overwrite files (default).
`-p ssh2-port`	Indicate the remote ports on which sshd2 listens.
`-p`	Preserve file attributes and timestamps.
`-Q`	Suppress progress indicator.
`-q`	Make scp quiet (only fatal errors are displayed).

`-r`	Recurse subdirectories.
`--SKA action`	Indicate server key action: `0`—Prompt (default) `1`—Add key and connect `2`—Do not add key and connect `3`—Deny connection
`-T level`	Indicate trace level: `1`—Basic `2`—detailed `3`—Verbose (default)
`--tracefile file`	Indicate the name of the trace file.
`-u`	Remove source files after copying.
`-v`	Display version.

Notes on selected options:

-B Batch mode is used for scripting file transfers. With this option, scp2 cannot stop execution for any reason. Therefore, all authentication information must be present on the command line or the transfer will fail.

-g fingerprint Scp2 uses the keys you create in Connectivity Secure Shell. You can view the value of the fingerprint in the Key Information dialog box accessible from the User Keys pane in the Connectivity Secure Shell Console.

-m mask This is the 3-number file permission mask. The first number corresponds to the owner of the file (you), the second corresponds to groups you might have created, and the third is for the general public. Each of the three numbers ranges from 0 to 7 and correspond to a bit-style flag-addition for [Read, Write, Execute], Read being in the most significant binary position, its value is 4, Write=2, and Execute=1. Thus, adding up the values gives full permission =7. Turning them all off means no permissions are given =0.

-r When the source is a directory, use this option to copy its subtrees.

--SKA Scp2 is considered the client to an SSH2 server. When connecting to this server, the server sends its certificate to scp2. This allows scp2 to determine if it’s safe to connect to this server. The server key action is an issue only when a server’s key is not already added to Connectivity Secure Shell’s Server Key store. What to do with this new server key is up to the user. The values are:

`0`—for prompting the user with the server certificate fingerprint

`1`—for connect and adding the key without prompting

`2`—for connecting without adding the key

`3`—for deny connection if the key is not already found in the store

If you choose to connect without being prompted for a server key, you will not have the opportunity to authenticate the server you are connecting to. Your safest option is to be prompted so that you can verify the server certificate's fingerprint.

--tracefile file Creates 2 files. One is for the FTP protocol trace events, and one for the SSH2 protocol events. Both protocols are used in Hummingbird Scp2.

Support for Other scp2 Tools

Hummingbird scp2 supports the command line formats of a number of other scp2 tools. Specifically, Hummingbird scp2 offers command line argument compatibility with the versions of scp2 offered by SSH.COM, F-Secure, PuTTY, and OpenSSH.

To use Hummingbird scp2 in one of the compatibility modes, run scp2 and use the `–emulate` argument, passing one of the following values as a parameter:

1—for native Hummingbird operation

2—for SSH.com

3—for F-Secure

4—for PuTTY

5—for OpenSSH

Hummingbird scp2 will remember the compatibility type you select, and will run in that compatibility mode until you run the command again to indicate another mode.

In each of the compatibility modes, running scp2 without any arguments will give you usage information appropriate for that compatibility mode.

About Secure Port Forwarding

The Transmission Control Protocol (TCP) offers strong protection against general data transmission failures caused by failed links, or network noise and congestion. It does not, however, provide protection against intentional attempts to sabotage these transmissions or intercept data. As a result, many protocols that run over TCP such as TELNET, NNTP, SMPT, IMAP, and POP are open to attacks.

With Connectivity Secure Shell, you can secure these otherwise unsecured TCP-based protocols by forwarding their connections through a Secure Shell session or tunnel. This process is generally referred to as port forwarding. In the following example, the TCP connection between POP client and server applications is forwarded through a Secure Shell tunnel.

To set up port forwarding, you create a tunnel profile with Connectivity Secure Shell and configure it for the type of forwarding you require. It may also be necessary to make some minor configuration changes at the TCP application level. After you launch the tunnel, the system works transparently, channelling the TCP/IP connection through a Secure Shell tunnel.

Port Forwarding Scenarios

There are three general scenarios for port forwarding:

- Outgoing Port Forwarding
- Incoming Port Forwarding
- Off-Host Port Forwarding (incoming/outgoing)

Outgoing Port Forwarding

In outgoing (or local) port forwarding scenarios, the TCP channel is initiated by Connectivity Secure Shell and sent through the tunnel to the remote Secure Shell server. For example, if a TCP client application such as an e-mail client is running locally on the same machine as Connectivity Secure Shell, and you want to create a secure connection to the application server (POP3), which resides on a remote host along with the Secure Shell server, you must set up an outgoing port forwarding.

Incoming Port Forwarding

In incoming (or remote) port forwarding scenarios, the Secure Shell server sends the initial TCP connection through the tunnel. In other words, the direction of the tunnel and the direction of the initial TCP connection are not the same.

For example, if the TCP client and server from the previous example are reversed, so that the e-mail client resides on the remote Secure Shell server machine, while the POP3 server application resides locally along with Connectivity Secure Shell, you can set up an incoming port that remote POP3 e-mail clients can use to access the local POP3 server. The server must be configured to listen for incoming connections to forward into the tunnel.

Off-Host Port Forwarding

Off-host port forwarding can be outgoing or incoming in nature. Its distinguishing feature is it requires that there be more than one host, which is to say that the hosts involved are not limited to those on either end of the Secure Shell connection. Another distinguishing feature of off-host port forwarding, however, is it does not offer the same level of security as simple outgoing or incoming port forwarding.

In this scenario, either the e-mail client or POP3 server application, or both can reside on machines other than the Secure Shell server machine or the machine on which Connectivity Secure Shell is installed. The following diagram depicts an incoming off-host setup.

You can set up an incoming or outgoing port forwarding, and configure it to allow remote connections. Then, on the remote TCP client or server machine, as the case may be, you can telnet to the forwarded port and connect through the Secure Shell tunnel. This configuration is useful, for example, in lieu of a VPN to traverse the Internet securely.

About Setting Up Port Forwarding

When you use Connectivity Secure Shell to set up any type of port forwarding, what you are doing is creating a Secure Shell tunnel, thereby providing a safer route for network traffic, and then directing the traffic so that it automatically assumes this detour.

Before endeavouring to set up an outgoing or incoming port forwarding, it will help to take a closer conceptual look at the mechanics of directing network traffic in this manner. As its name implies, port forwarding depends largely on port configuration for this task.

Normally, a POP server on remote host listens for connections on the default TCP port (110).

In an example of an outgoing port forwarding, the connection occurs as follows:

1. The e-mail client on the workstation sends a connection request to a local port instead of port 110 on the server. For the purpose of this example, we can specify port 3800.
2. The Connectivity Secure Shell tunnel you created must be configured to listen on local port 3800. When it receives the data, the connectivity Secure Shell engine encrypts it and sends it through the Secure Shell tunnel to the Secure Shell server on Host B.
3. After it is decrypted by the Secure Shell server, the data arrives at the destination port, 110, just as it would have were it not secured in transit. After the TCP connection is established through the Secure Shell tunnel, data returned by the POP server follows the same path in reverse.

Setting Up an Outgoing Port Forwarding

The following procedure guides you through the process of setting up an outgoing port forwarding in Connectivity Secure Shell. It also applies if you are setting up off-host port forwarding that is outgoing in nature.

Once you have set up the outgoing port forwarding in connectivity Secure Shell, you must configure any involved third-party applications accordingly to complete the set up process.

To set up an outgoing port forwarding:

1. In the Connectivity Secure Shell Management Console, open the Tunnels folder and click Connection. The three-paned Connection window is displayed.
2. Click the Open Tunnel button in the top pane. The Open Tunnel dialog box opens.
3. If you want to use an existing tunnel that is configured to connect to the appropriate host, select it. Otherwise, follow these steps to create a new tunnel:
   a. Click the Add New Tunnel button.
   b. In the Add New Tunnel dialog box, provide the required information.
   c. Click OK.

   The tunnel profile appears in the Open Tunnel dialog box with the name you specified. It will also be accessible through Hummingbird applications that support Connectivity Secure Shell.
4. In the Open Tunnel dialog box, click the Edit Tunnel Info button.
5. In the Tunnel Profile dialog box, click Outgoing in the Port Forwarding folder.
6. In the right pane, click the Add New Outgoing Port Forwarding button.
7. In the Add New Outgoing Port Forwarding dialog box, specify the following information:

- Display Name—Specify a name for the outgoing port forwarding profile. You may find it helpful to choose display names that indicate the defining features of the profiles you create, as this will simplify your future outgoing port selection.
- Destination Host—Specify either the IP address or the fully qualified domain name of the remote host to which you want to connect. This host is the Secure Shell server machine in the case of a simple outgoing port forwarding. If you are setting up an off-host forwarding, you will specify the host to which the Secure Shell server sends the unencrypted data.

  Note: If you are setting up off-host outgoing port forwarding:

  - Ensure the Allow Local Connections Only option is not selected.
  - Server configuration may not support or allow port forwarding.

- Listen Port—Specify the local port on which the Connectivity Secure Shell engine listens for connection attempts. You can choose any port between 1024 and 65535. Make a note of the port you select as this is the port to which you must configure third-party applications such as e-mail clients to send their connections.
- Destination Port—Specify the port (on the destination host) to which you want the Secure Shell server to send the unencrypted data. In most cases, you will specify a default protocol port. For example, specify port 110 for POP3 servers, port 23 for TELNET, and so on.

  Tip: For information on standardized protocol ports, visit http://www.iana.org/assignments/port-numbers.

- Allow local connections only—Select this option to restrict connections to the listening port to those initiated locally. If you are setting up an off host port forwarding and intend to connect to the listening port from a remote machine, do not select this option.
- Enable—Select this option to enable this port forwarding profile within the tunnel profile you are editing. Enabled forwardings

have a checkmark next to their display names in the Tunnel Profile dialog box.

8. Click OK. The new profile appears in the Outgoing Ports list.
9. In the Tunnel Profile dialog box, click OK.
0. In the Open Tunnel dialog box, you can click Open to start the tunnel. However, it may be necessary to configure third-party applications before you can successfully use the Secure Shell session to secure their connections.

Setting Up an Incoming Port Forwarding

The following procedure guides you through the process of setting up an incoming port forwarding in Connectivity Secure Shell. It also applies if you are setting up off-host port forwarding that is incoming in nature.

Once you have set up the incoming port forwarding in Connectivity Secure Shell, you must configure any involved third-party applications accordingly to complete the set up process.

To set up an incoming port forwarding:

1. In the Connectivity Secure Shell Management Console, open the Tunnels folder and click Connection. The three-paned Connection window is displayed.
2. Click the Open Tunnel button in the top pane. The Open Tunnel dialog box opens.
3. If you want to use an existing tunnel that is configured to connect to the appropriate host, select it. Otherwise, follow these steps to create a new tunnel:
    a. Click the Add New Tunnel button.
    b. In the Add New Tunnel dialog box, provide the required information.
    c. Click OK.

   The tunnel profile appears in the Open Tunnel dialog box with the name you specified. It will also be accessible through Hummingbird applications that support Connectivity Secure Shell.
4. In the Open Tunnel dialog box, click the Edit Tunnel Info button.
5. In the Tunnel Profile dialog box, click Incoming in the Port Forwarding folder.
6. In the right pane, click the Add New Incoming Port Forwarding button.
7. In the Add New Incoming Port Forwarding dialog box, specify the following information:

- Display Name—Specify a name for the incoming port forwarding profile. You may find it helpful to choose a display name that indicate the defining features of the profiles you create, as this will simplify your future outgoing port selection.
- Destination Host—If you are setting up a simple incoming port forwarding, you can type `localhost` to specify the local machine as the destination host. If you are setting up an off-host forwarding, specify either the IP address or the fully qualified domain name of the host to which you want to forward incoming connections.

  Note: If you are setting up off-host incoming port forwarding:

  - Do not select the Allow Local Connections Only option in this dialog box.
  - Server configuration may not support or allow port forwarding.

- Listen Port—Specify the remote port on which the Secure Shell server listens for connection attempts. You can choose any port between 1024 and 65535. Make a note of the port you select as this is the port to which you must configure third-party applications such as e-mail clients to send their connections.
- Destination Port—Specify the port on the local host to which you want the Connectivity Secure Shell engine to send the unencrypted data. In most cases, you will specify a default protocol port. For example, specify port 110 for POP3 servers, port 23 for TELNET, and so on.

  Tip: For information on standardized protocol ports, visit http://www.iana.org/assignments/port-numbers.

- Enable—Select this option to enable this port forwarding profile within the tunnel profile you are editing. Enabled forwardings have a checkmark next to their display names in the Tunnel Profile dialog box.

8. Click OK. The new profile appears in the Incoming ports list.
9. In the Tunnel Profile dialog box, click OK.
0. In the Open Tunnel dialog box, you can click Open to start the

tunnel. However, it may be necessary to configure third-party applications before you can successfully use the Secure Shell session to secure their connections.

About Secure X Window Sessions

The mechanics of securing the X protocol with Secure Shell are similar to the mechanics of securing other third-party protocols by forwarding their connections through Secure Shell tunnels. Though the mechanics are similar, the process of rerouting the X protocol (as opposed to the IMAP protocol, for example) is made simpler by Secure Shell’s built-in ability to handle X communications.

If you enable X11 port forwarding in a tunnel profile, when you start the tunnel, the Connectivity Secure Shell engine requests X forwarding upon connection to the Secure Shell server. If the server supports X forwarding, the process continues as follows:

1. The initial terminal session is channelled through the tunnel. You are authenticated by the server and logged in.
2. Because X11 forwarding is enabled in the tunnel profile, the Secure Shell server takes on the roll of a Proxy X server, and forwards the DISPLAY in the remote shell to point to itself (the X proxy display).
3. When you launch an X client application, the X client connects to the X proxy server (the Secure Shell server), which then instructs the Connectivity Secure Shell engine to act as proxy X client. The X protocol is sent through the tunnel between the Connectivity Secure Shell engine and the Secure Shell server over a second channel.

Even though these socket connections are not encrypted, they are secure because they do not traverse the external network. The X client on the remote server uses a socket connection on localhost to the proxy X server established by the Secure Shell server. The X protocol data are then transferred through the external network via the encrypted Secure Shell tunnel. The Connectivity Secure Shell engine sends this data to the proxy X client, which uses a socket connection on localhost to the actual X server (Exceed) running on the user's workstation.

Securing Hummingbird X Window Sessions

Securing an X Window session requires that you create an Xstart profile that specifies Secure Shell as the startup method. The Connection parameters you specify in Xstart are used to establish a Secure Shell tunnel to the target host. You can specify the following connection parameters:

- Host
- User ID
- Authentication Method

If you want to custom configure other Secure Shell tunnel parameters, you must associate a customized tunnel profile with your Xstart profile. For more information, see Using Tunnel Profiles to Secure X Window Sessions.

To create a Secure Shell-enabled Xstart profile:

1. Launch Xstart.
2. In the Xstart window, select Secure Shell as the startup method.
3. In the Connection area, provide the following information:
    - Host—A host name or IP address. By default, the most recent host to which you connected appears at the top of the drop-down list. Click Browse to open the Browse for Hosts dialog box.
    - User ID—Type a valid name for a user account on the host.
    - Password—A valid password for the user account. An asterisk displays for each character typed. The password is encrypted before being stored in the Xstart file.

    Note: You can type login macros in the User ID and Password boxes.

    - Command—Type a command or command macro to send to the host. The command depends on the Program Type, the host system, and how the host is set up. Click Browse to open the Browse for Applications dialog box.
4. To configure Authentication Methods, click Settings. The Xstart

Settings dialog box opens.

a. On the Security page, click Properties. The Secure Shell Configuration dialog box opens.

b. Click Config. The the Authentication Settings dialog box opens. See Authentication for further instruction.

5. When you have provided all required information, do the following in the Xstart dialog box:

   - Click the Save button to save the Xstart profile in a location and and using a name of your specification.
   - Click the Run button to launch the session.

Using Tunnel Profiles to Secure X Window Sessions

This procedure is useful if you want to use a customized Secure Shell tunnel to connect to the host, instead of using the default tunnel as described in Securing Hummingbird X Window Sessions. This is necessary if you want to modify Secure Shell tunnel settings that are not accessible in Xstart.

Connectivity Secure Shell tunnel profiles contain all the parameters necessary to start and configure a Secure Shell tunnel. When you create your Xstart profile, you can configure it to call the necessary connection parameters from a custom tunnel profile of your choice. You can create tunnel profiles with Connectivity Secure Shell before hand, and then select one to be used for your X Window session, or you can create them on the fly while you configure your Xstart profile.

To create a customized, Secure Shell-enabled Xstart profile:

1. Launch Xstart.
2. In the Xstart window, select Secure Shell as the startup method.
3. Click Settings. The Xstart Settings dialog box opens.
4. Click the Security tab.
5. With the Use For Authentication Only option, specify which data you want to secure. If you do not enable this option, Xstart secures your authentication to the host as well as all X application data that is passed thereafter. If you enable the option, only the authentication process is secured.
6. Click Properties. The Secure Shell Configuration dialog box opens.
7. To specify the tunnel profile you want to use to establish the Secure Shell tunnel for the session, do one of the following and then click OK:
    - Type the name of the tunnel profile (`*profilename*.csp`) into the CSS Tunnel Profile box.
    - Click Browse to select from a list of your tunnel profiles. Click Open.

- If you have not yet created a tunnel profile or if you want to create a new one, click Create. The Add New Tunnel dialog opens. For more information on creating a Tunnel profile, see Creating Tunnel Profiles.

8. In the Xstart Settings dialog box, specify a command, and then click OK.
9. The connection parameters configured in the tunnel profile now appear in the Connection area of the Xstart window.
   - Click the Save button to save the Xstart profile.
   - Click the Run button to launch the session.

Related Topics

Overriding Tunnel Profile Connection Parameters

Overriding Tunnel Profile Connection Parameters

If you created an Xstart profile that references connection parameters from a tunnel profile as described in Using Tunnel Profiles to Secure X Window Sessions, you can modify the imported connection parameters without altering the tunnel profile itself. Xstart lets you save the new parameters as overrides.

You can use the Connection area of the Xstart Window to specify tunnel parameter overrides either while you are creating an Xstart profile, or while editing an existing profile.

Specify new parameters to override the tunnel value. To revert to the original tunnel values, click the Reload button.

Overriding Authentication Method Settings

To override the authentication method settings specified by the tunnel profile, click Settings in the Xstart window. In the Xstart Settings dialog box, click Properties on the Secure Shell page. The Secure Shell Configuration dialog box opens. Click Config to access the authentication settings. See Authentication for further instruction.

Editing Secure Shell Settings for X Window Sessions

When editing a Secure Shell-enabled Xstart profile, keep in mind that the changes you make to the configuration may be saved differently depending on whether you have associated a tunnel profile with the Xstart profile, and depending on how you make the changes.

To edit Secure Shell-enabled Xstart profiles:

1. In the Xstart window, click the Open button and select the profile you want to edit.
2. To edit the parameters saved in the Xstart profile, specify the new parameters in the Connection area. Bear in mind that, if you associated a tunnel profile with the Xstart profile you are editing, any changes you make to the Host, User ID and Password parameters are saved in your Xstart profile as overrides of the tunnel profile parameters.
3. To configure Authentication Methods, click Settings. The Xstart Settings dialog box opens.
   a. On the Security page, click Properties. The Secure Shell Configuration dialog box opens.
   b. Click Config. The the Authentication Settings dialog box opens. See Authentication for further instruction.
4. When you have made all of the required modifications, click the Save button in the main Xstart window.

Editing an Associated Tunnel Profile

There are two ways you can edit a tunnel profile that you have associated with your Xstart profile. You can select and edit the tunnel profile using Connectivity Secure Shell. For more information, see Configuring Tunnel Profiles.

Tip: You can override some tunnel profile parameters instead of editing the tunnel profile itself. For more information, see Overriding Tunnel Profile Connection Parameters.

You can also perform tunnel profile-related modifications from within Xstart, whether you want to associate a different tunnel with your Xstart profile or edit the tunnel profile that is currently associated with your Xstart profile.

To edit tunnel profiles with Xstart:

1. In the Xstart window, click the Open button and select the profile you want to edit.
2. In the Startup area, click Settings.
3. In the Xstart Settings dialog box, click the Security tab.
4. In the Secure Shell area, click Properties. The Secure Shell Configuration dialog box opens.
5. To specify the tunnel profile you want to use to establish the Secure Shell tunnel for the session, do one of the following:
   - Type the name of the tunnel profile (*`profilename`*`.csp`) into the CSS Tunnel Profile box.
   - Click Browse to select from a list of your tunnel profiles. Click Open.
   - If you have not yet created the tunnel profile, click Create. The Add New Tunnel dialog opens. For more information on creating a Tunnel profile, see Creating Tunnel Profiles.
6. Click OK.

Connectivity Secure Shell Management Console

After you configure and establish one or more Secure Shell tunnel connections, you can use the Connectivity Secure Shell Management Console to view important information about the tunnels, and about the channels established within them. This information along with general settings, which are also accessible from the console, will assist you in managing, monitoring, and troubleshooting your connections.

To view information about the tunnels that are currently active, in the left pane of the Connectivity Secure Shell Console, click Connection in the Tunnels folder. The following three viewing areas appear from top to bottom in the Connection window on the right side of the console:

- Tunnel pane—For more information, see Viewing Information about Tunnels.
- Channel pane—For more information, see Viewing Channel Information.
- Output window—For more information, see Viewing Channel Output.

Related Topics

Generating Log files

Viewing Information about Tunnels

The Tunnel pane displays a list of active tunnels and provides information about each tunnel’s configuration and performance. To display the Tunnel pane, in the left pane of the Connectivity Secure Shell Console, click Connection in the Tunnels folder.

Information	Description
Tunnel Name	Indicates the name of the tunnel profile used to establish the tunnel.
Host Name	Indicates the host to which the tunnel is connected.
Port	Indicates the host port to which the tunnel is connected.
User Name	Indicates the login name used to connect to the host.
X11	Indicates if the tunnel is configured to carry X11 sessions.
Channel	Indicates the number of active channels in the tunnel.
Bytes In	Indicates the amount of data received.
Bytes Out	Indicates the amount of data sent.
Started Time	Indicates when the tunnel was established.

The buttons in this viewing area let you access and configure existing tunnels, unload tunnels that are currently started, and manage the information displayed.

Related Topics

Configuring Tunnel Profiles

Viewing Channel Information

The middle pane in the Connection window is the Channel pane. It displays a list of active channels currently established in the tunnel selected in the Tunnel Profile pane above it. To display the Channel pane, in the left pane of the Connectivity Secure Shell Console, click Connection in the Tunnels folder. You can use the Toggle Tunnel Tracing button in the top pane to enable and disable this view.

Information	Description
Channel	Indicates the application that is using the channel.
Channel Type	Indicates how the channel is being used.
Bytes In	Indicate the amount of data received.
Bytes Out	Indicates the amount of data sent.
Started Time	Indicates when the tunnel was established.

Related Topics

Creating Tunnel Profiles

Configuring Tunnel Profiles

Viewing Channel Output

The bottom pane in the Connection window is the Output window displays the output for the channel selected in the Channel pane. To display this window, in the left pane of the Connectivity Secure Shell Console, click Connection in the Tunnels folder. Use the Toggle Channel Tracing button in the top pane to enable and disable this view.

To configure the format of the information displayed in the Output window, click the Output Window Settings button in the Tunnel pane to access the Output Window Settings dialog box, which lets you set the following options:

- Data Format—Specify a format for the displayed data. You can view the data as ASCII, EBCDIC, or ASCII and EBCDIC. Otherwise, you can specify no conversion.
- Data Update Time—Specify the amount of time in seconds between output window data updates.
- Number Of Lines For The Output Window—Specify the number of lines you want to be displayed in the output window.

To clear all output, click the Clear Output Window button.

Related Topics

Creating Tunnel Profiles

Configuring Tunnel Profiles

Generating Log files

To view information about the behavior of Connectivity Secure Shell, you can generate a log file. When you enable logging, a log file is generated and saved as `CSSLog.txt` in the `My Documents` system folder by default.

To generate a log file for Connectivity Secure Shell:

1. Under Settings, in the left pane of the console, click General. The general settings appear.
2. Select Enable Logging.
3. If necessary, click the Browse button to alter the default log file name or location.
4. Click Apply.

About Certificate and Key Manager

Certificate and Key Manager is a centralized repository and management tool for keys and certificates that reside in the Hummingbird certificate and key store.

Certificate and Key Manager lets you manage keys and certificates for use with Secure Shell and SSL connections initiated with a number of Hummingbird Connectivity products including Exceed, HostExplorer, Hummingbird FTP, and classic FTP. You can perform the following tasks:

- view information about your keys
- view information about your certificates
- generate user keys
- generate certificate requests
- generate self-signed certificates
- import and export keys
- import and export certificates

Viewing Key Information

Keys are grouped by type in the left pane of the Secure Shell Management Console:

- User Keys—Click to view a list of the user keys contained in the `my.hks` file. These are keys that you either created or imported.
- Server Keys—Click to view a list of the server keys imported into the `server.hks` file.

Note: Changing `.hks` files manually is not recommended.

To view the details of a user or server key:

Click the key type in the left pane. In the list pane that appears on the right, double-click the key to open the Key Information window.

Viewing Certificate Information

Certificates are grouped by type in the left pane of the Secure Shell Management Console:

- User Certificates—Click to view the user certificates stored in the `my.hcs` file. These certificates were either created or imported by you, and signed by Certificate Authorities (CA) or system administrators.
- Server or Intermediate Certificate Authorities—Click to view the certificates stored in the `ca.hcs` file. These certificates belong to the commercial or private CAs that are below the root CAs in the certificate chain.
- Trusted Root Certificate Authorities—Click to view the certificates stored in the `root.hcs` file. These certificates belong to the top-level (or parent) CAs who sign their own certificates. Generally, they establish trust by publishing their public keys widely.

Note: Changing `.hcs` files manually is not recommended.

To view the details of a certificate:

Click the certificate type in the left pane. In the list pane that appears on the right, double-click the certificate to open the Certificate Information window.

Generating Private/Public Keypairs

Part of the process of generating a certificate is to create a private/public keypair. You can also use these keys on their own for authentication and encryption. Keypairs are stored on your machine and protected by a passphrase that will be necessary to open and use them.

Tip: You can use one key for multiple certificates.

You can create keys before you generate the certificate, and then select the keys during the certificate generation process. Otherwise, you can create the keys as part of the certificate generation process itself. In either case, you will use the Key Generation Wizard to create the keys.

To create a private/public keypair:

1. Open the Key Generation Wizard by clicking User Keys in the left pane of the console, and then clicking the Create New Key button at the bottom of the key pane.
2. Click Next on the welcome screen. The Key Parameters screen opens.
3. Provide the necessary information and click Next. The wizard generates the key.

   Note: Depending on the size of the key to be generated and the speed of your CPU, the key generation process may take some time.

4. Click Finish. The key is displayed in the key pane and saved in the `my.hcs` file located in the `certs` directory where the user files are stored on your machine.
5. Click Close.

Related Topics

Using the Passphrase Cache

Generating Certificate Requests

If you do not want to use a self-signed certificate, you can request a certificate from a certificate authority (CA). A certificate request contains personal information describing the individual requesting the certificate. This certificate request is sent to the appropriate certificate authority or security administrator who, in turn, sends you a signed certificate.

To generate a certificate request:

1. Open the Certificate Creation Wizard by clicking User Certificates in the left pane of the console, and then clicking the Create New Certificate button at the bottom of the certificate pane.
2. On the Wizard Type drop-down list, select Create A Certificate Request.
3. Click Next on the welcome screen. The Certificate Information screen opens.
4. Provide the required information and click Next.
5. Do one of the following:
   - If you have already created a keypair, select it from the User Keys list, and type the passphrase for the selected key. You can use the buttons below the User Keys list to view key details and change the passphrase.
   - If you have not created a keypair, click the Create New Key button and use the Key Generation Wizard to create a keypair. For more information, see Generating Private/Public Keypairs.
6. Click Next. The wizard generates the certificate request. The public key part of the private/public key combination you specified is stored into the certificate.
7. Click Finish.
8. The certificate is listed in the certificate pane, and the request is now saved in the `my.hcs` file located in the `certs` directory where the user files are stored on your machine. Use the Export button to export the certificate to `.pem` format.

9. Manually forward the `.pem` exported request to the appropriate CA (for example, a commercial CA such as a Verisign or your organization's private CA) or to your security administrator. You can send the certificate request through e-mail.

When you receive the signed certificate (for example, the user certificate or a new root certificate), you must update the certificate request with the certificate you received. To do so, double-clicking your original certificate request in the right pane of the console. In the Certificate Information dialog box, click Update. In the Open dialog box, select the file where you saved the response from the CA and click Open.

Generating Self-Signed Certificates

Self-signed certificates are certificates in which the subject and the issuer are the same. There is no independent means of verifying the trustworthiness of the certificate, as is the case with Root certificates. They are ideal if you need a certificate to test whether or not the connection works. Also, an organization might want to store user information in a certificate, but it may not want (or need) this certificate information to be validated. Self-signed certificates can also be used for authentication and authorization on a private network.

For example, when connecting to a host, the host must provide a certificate. This certificate can be self-signed (and then exported) by the server administrator. If there is no doubt as to the validity of this information and your application is configured to accept self-signed certificates, then the self-signed certificate are sufficient.

To create a self-signed certificate:

1. Open the Certificate Creation Wizard by clicking User Certificates in the left pane of the console, and then clicking the Create New Certificate button at the bottom of the certificate pane.
2. On the Wizard Type drop-down list, select Create A Self-Signed Certificate.
3. Click Next on the welcome screen. The Certificate Information screen opens.
4. Provide the required information and click Next.
5. Do one of the following:
   - If you have already created a keypair, select it from the User Keys list, and type the passphrase for the selected key. You can use the buttons below the User Keys list to view key details and change the passphrase.
   - If you have not created a keypair, click the Create New Key button and use the Key Generation Wizard to create a keypair. For more information, see Generating Private/Public Keypairs.

6. Click Next. The wizard generates the certificate. The public key part of the private/public key combination you specified is stored into the certificate.
7. Click Finish.
8. The certificate is encoded and listed in the certificate pane. It is saved in the `my.hcs` file located in the `certs` directory where the user files are stored on your machine.

Modifying Keys and Certificates

Generally speaking, keys and certificates are not editable. However, different types of keys and certificates do allow you to alter certain parameters.

Changing the Passphrase for a User Key

Select User Keys in the left pane. In the list pane, select the key and click the Change Passphrase button to open the Change Passphrase dialog box.

Changing the Host Identification of a Server Key

Select Server Keys in the left pane. In the list pane, select the key and click the Change Host Identification button to open the Change Host Identification dialog box. Provide the required information. If you do not know the IP address of the specified host, click Get IP.

Enabling Cryptographic Service Providers

A cryptographic service provider (CSP) is an independent software module that performs cryptography algorithms for authentication, encoding, and encryption. The Connectivity Secure Shell Management Console retrieves the list of installed CSP modules from your registry and uses only the ones you have selected from the Select Cryptographic Service Providers dialog box. Normally, the defaults should not be changed; the providers selected by default are the only ones you will likely use. It is possible, however, to try an unsupported provider, or to temporarily disable a certain provider in order to stop using certificates from that provider.

To access the cryptographic service provider settings, select User Certificates in the left pane and click the Cryptographic Service Providers button. The Select Cryptographic Service Providers dialog box opens. You may use the dialog box to filter the list of user certificates by selecting or disabling providers.

Importing and Exporting Keys

Hummingbird key store files (`.hks`) are repositories for the keys you create or import. User keys are comprised of a public/private key pair and are stored in the `my.hks` file. Server keys are public keys only and are stored in the `server.hks` file. These files are located in the `Certs` directory where the user files are stored on your machine. When you import or export either type of key, you do so to and from these files.

Note: Modifying key files manually is not recommended.

To export a key from the user or server key store

To import a key to the user or server key store:

1. Under Keys, in the left pane of the console, click the key store (User Keys or Server Keys) into which you want to import a key.
2. Click the Import button.
3. In the Open dialog box, locate and select the keypair you want to import. By default, Connectivity Secure Shell auto-detects all key formats, but you can specify the following file filters:
    - Hummingbird HKS Files (*.hks)—Lists Hummingbird key stores.
    - PEM Files (*.pem)—Lists OpenSSL-derived keys.
    - SSH2 Public Key Files (*.pub)—Lists OpenSSH and SSH2 public key formats.
4. Click Open.
5. If you are prompted, type the passphrase for the selected key.

The key is added to the key store and is displayed in the list pane of the console. If you imported an `*.hks` file, the keys it contains are extracted and added to the `my.hks` or `server.hks` store.

Note: If the imported key does not appear in the list pane, right click in the pane and click Refresh in the pop-up menu.

To export a key from the user or server key store:

1. Under Keys, in the left pane of the console, click the key store (User

Keys or Server Keys) from which you want to export a key.

2. From the list that appears, select the key you want to export, and click the Export button. The Save As dialog box opens.
3. Use the Save In drop-down list to specify the target location for the key.
4. In the File Name box, specify a key name.
5. In the Save as Type box, specify the format to which you want to export the key:
    - Hummingbird HKS Files (*.hks)—Exports the public and, in the case of user keys, the private portion of the selected key to a Hummingbird key store file.
    - PEM Files (*.pem)—Exports the public and, in the case of user keys, the private portion of the selected key to a `.pem` file.
    - SSH2 Public Key Files (*.pub)—Exports the public portion of the specified key as an SSH2 public key.
    - OpenSSH Public Key Files (*.pub)—Exports the public portion of the specified key as an OpenSSH public key.
6. Click Save.

Uploading Public Keys

Hummingbird Key Upload Wizard lets you upload public keys to a host for public key authentication. You can access the wizard from the Connectivity Secure Shell Management Console, or from the Certificate Creation Wizard while creating a certificate.

To upload a public key to a host:

1. Under Keys, in the left pane of the console, click User Keys.
2. In the right pane, right-click the key you want to upload, and click Upload on the pop-up menu. Hummingbird Key Upload Wizard opens.
3. Click Next on the welcome screen, and provide the required information on each subsequent screen. For more information on a particular screen, click Help.

Importing and Exporting Certificates

The Hummingbird certificate store is comprised of three files. Each stores a specific type of key:

- `my.hcs` stores user certificates.
- `ca.hcs` stores server certificates are certificates from intermediate certification authorities.
- `root.hcs` stores certificates from trusted root certification authorities.

These files are located in the `Certs` directory where the user files are stored on your machine. When you import or export a certificate, you do so to and from these certificate stores.

For procedural information, select the task you want to perform:

- Importing Certificates
- Exporting Certificates

Importing Certificates

To send a certificate to the server during connection negotiation, you need to install the certificate by importing it to your Hummingbird certificate store. The certificate you import may be any of the following:

- a new or updated root certificate that is sent to you from a system administrator
- a root certificate installed with your Windows operating system, that you deleted and need to re-import
- a signed user certificate that is sent to you by your system administrator or certificate authority (CA) as a result of your submitted certificate request

To import a certificate:

1. Under Certificates, in the left pane of the console, click the certificate store into which you want to import a certificate.
2. Click the Import button.
3. In the Open dialog box, locate the certificate you want to import. You can select one of the following file filters in the Files Of Type drop-down list:
   - Hummingbird HCS Files (*.hcs)—Lists Hummingbird certificate stores.
   - X.509 Certificate (*.cer, *.crt, *.der, *.pem)—Lists different variations of X.509 certificates.
   - PKCS#12 (*.p12, *.pfx)—Lists different variations of PKCS#12 files.
4. Click Open.

   Note: If the imported certificate does not appear in the list pane, right click in the pane and click Refresh in the pop-up menu.

Exporting Certificates

You can export certificates currently saved in the Hummingbird certificate store in order to distribute them to other machines. This is necessary if, for example, you generate a self-signed certificate for a user or for a host machine on the network, or if you want to connect to a host from a machine other than the one on which the certificate resides.

To export a certificate:

1. Under Certificates, in the left pane of the console, click the certificate store from which you want to export a certificate.
2. In the list pane, select the certificate you want to export.
3. Click the Export button.
4. In the Save As dialog box, use the Save In drop-down list to specify the target location for the certificate.
5. In the File Name box, specify the certificate name.
6. In the Save as Type box, specify the format to which you want to export the certificate:
   - Hummingbird HCS Files (*.hks)—Exports the certificate to the Hummingbird certificate store file (`my.hcs`, `ca.hcs`, or `root.hcs` depending on the type of certificate).
   - Base-64 Encoded X.509 (*.cer)—Exports a base-64-encoded, X.509-format certificate file.
   - DER Encoded Binary X.509 (*.cer)—Exports a DER-encoded, binary X.509-format certificate file.
7. Click Save As.

Using the Passphrase Cache

The passphrase cache lets you store user key passphrases. You can use this cache to minimize the number of times you have to enter passphrases for public key authentication requests.

Typically, you are prompted for a passphrase each time a public key authentication request is made. However, if you enable passphrase caching and set the cache expiration time, you enter the passphrase for the first request. Thereafter, up to the specified expiration time, the passphrase is retained in memory and used against subsequent public key requests. If all of your user keys use same passphrase, you are not prompted again. If all passphrases in the cache fail against a particular key, then you are prompted for the passphrase for that particular key.

Passphrase Caching and S-Agent Requests

The passphrase cache is always available for s-agent requests, even if it is disabled. When you attempt to connect a public-key-authenticated session for which s-agent authentication is enabled, Connectivity Secure Shell opens all of the user keys in the keystore and prompts you if passphrases for these keys are not already in the cache. The passphrases are cached in advance in the event that you cannot respond when subsequent s-agent authentication requests are received.

To cache passphrases:

1. Under Settings, in the left pane of the console, click General. The general settings appear.
2. Select Use Passphrase Cache.
3. In the Passphrase Timeout box, specify the amount of time in minutes that the passphrases in the cache are to remain active.
4. Click Apply.

Related Topics

Generating Private/Public Keypairs

General Accessibility

Hummingbird products are accessible to all users. Wherever possible, our software adheres to Microsoft Windows interface standards and contains a comprehensive set of accessibility features.

Access Keys All menus have associated access keys (mnemonics) that let you use the keyboard, rather than a mouse, to navigate the user interface (UI). These access keys appear as underlined letters in the names of most UI items. (If this is not the case, press Alt to reveal them.) To open any menu, press Alt and then press the key that corresponds with the underlined letter in the menu name. For example, to access the File menu in any Hummingbird application, press Alt+F.

Once you have opened a menu, you can access an item on the menu by pressing the underlined letter in the menu item name, or you can use the arrow keys to navigate the menu list.

Keyboard Shortcuts Some often-used menu options also have shortcut (accelerator) keys. The shortcut key for an item appears beside it on the menu.

Directional Arrows Use the directional arrows on the keyboard to navigate through menu items or to scroll vertically and horizontally. You can also use the directional arrows to navigate through multiple options. For example, if you have a series of radio buttons, you can use the arrow keys to navigate the possible selections.

Tab Key Sequence To navigate through a dialog box, press the Tab key. Selected items appear with a dotted border. You can also press Shift+Tab to go back to a previous selection within the dialog box.

Spacebar Press the Spacebar to select or clear check boxes, or to select buttons in a dialog box.

Esc Press the Esc key to close a dialog box without implementing any new settings.

Enter Press the Enter key to select the highlighted item or to close a dialog box and apply the new settings. You can also press the Enter key to close all About boxes.

ToolTips ToolTips appear for all functional icons. This feature lets users use Screen Reviewers to make interface information available through synthesized speech or through a refreshable Braille display.

Microsoft Accessibility Options

Microsoft Windows environments contain accessibility options that let you change how you interact with the software. These options can add sound, increase the magnification, and create sticky keys.

To enable/disable Accessibility options:

1. In Control Panel, double-click Accessibility Options.
2. In the Accessibility Options dialog box, select or clear the option check boxes on the various tabs as required, and click Apply.
3. Click OK.

If you installed the Microsoft Accessibility components for your Windows system, you can find additional accessibility tools under Accessibility on the Start menu.

Technical Support

You can contact the Hummingbird Technical Support department Monday to Friday between 8:00 a.m. and 8:00 p.m. Eastern Time.

Hummingbird Ltd. **1 Sparks Avenue, Toronto, Ontario, Canada M2H 2W1**		
	Canada and the USA	International
Technical Support:	1-800-486-0095	+1-416-496-2200
General Enquiry:	1-877-FLY-HUMM	+1-416-496-2200
Main:	+1-416-496-2200	
Fax:	+1-416-496-2207	
E-mail:	support@hummingbird.com	
FTP:	ftp.hummingbird.com	
Web Support:	support.hummingbird.com/customer	
Web Site:	www.hummingbird.com	