



## Bridging and Connecting with SSH for LES Series Console Servers

**Works with LES1200, LES1300, LES1400,  
and LES1500 Series console servers.**

### About this Document

This document describes two procedures related to SSH and your Console Servers:

- Serial bridging with SSH tunnels
- SSH connecting using an SSH client

### Serial Bridging with SSH Tunnels

You can configure a pair of LES1200/LES1300/LES1400/LES1500 console servers to support serial bridging—for interconnecting serial devices over a network. The serial data is encapsulated into network packets for transport over the IP network between the two console servers. For secure transport, this can be directed through a secure SSH tunnel:

Serial Bridge Settings	
Serial Bridging Mode	<input checked="" type="radio"/> Create a network connection to a remote serial port via RFC-2217.
Server Address	<input type="text" value="250.258.2.16"/> The network address of an RFC-2217 server to connect to.
Server TCP Port	<input type="text" value="5002"/> The TCP port the RFC-2217 server is serving on.
RFC 2217	<input checked="" type="checkbox"/> Enable RFC 2217 access.
SSH Tunnel	<input checked="" type="checkbox"/> Redirect the serial bridge over an SSH tunnel to the server

## Bridging and Connecting with SSH for LES Series Console Servers

Select SSH Tunnel when configuring the Serial Bridging Setting

Next, you will need to set up SSH keys for each end of the tunnel and upload these keys to the Server and Client gateways.

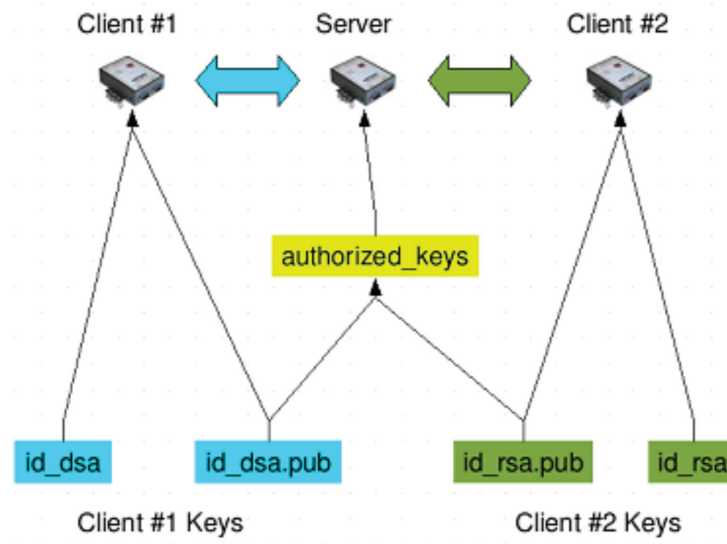
### Client Keys

The first step in setting up ssh tunnels is to generate keys. Ideally, you will use a separate, secure machine to generate and store all keys to be used on the LES1200/LES1300/LES1400/LES1500 console servers. If this is not ideal to your situation, keys may be generated on the console servers themselves.

You can generate only one set of keys, and reuse them for every SSH session. While this is not recommended, each organization will need to balance the security of separate keys against the additional administration they bring.

Generated keys may be one of two types—RSA or DSA (and it is beyond the scope of this document to recommend one over the other). RSA keys will go into the files `id_rsa` and `id_rsa.pub`. DSA keys will be stored in the files `id_dsa` and `id_dsa.pub`.

For simplicity going forward, the term private key will be used to refer to either `id_rsa` or `id_dsa` and public key to refer to either `id_rsa.pub` or `id_dsa.pub`.



To generate the keys using OpenBSD's OpenSSH suite, we use the `ssh-keygen` program:

```
$ ssh-keygen -t [rsa|dsa]
```

Generating public/private [rsa|dsa] key pair.

Enter file in which to save the key (/home/user/.ssh/id\_[rsa|dsa]):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/user/.ssh/id\_[rsa|dsa].

Your public key has been saved in /home/user/.ssh/id\_[rsa|dsa].pub.

The key fingerprint is:

```
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
```

```
$
```

You can create a new directory to store your generated keys. You can name the files after the device they will be used for. For example:

```
$ mkdir keys
```

```
$ ssh-keygen -t rsa
```

Generating public/private rsa key pair.

Enter file in which to save the key (/home/user/.ssh/id\_rsa):

```
/home/user/keys/control_room
```

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/user/keys/control\_room

Your public key has been saved in /home/user/keys/control\_room.pub.

The key fingerprint is:

```
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
```

```
$
```

*NOTE: Make sure there is no password associated with the keys. If there is a password, then the LES1200/1300/1400/1500 servers will have no way to supply it at runtime.*

### Authorized Keys

If the LES1200/1300/1400/1500 selected to be the server will only have one client device, then the authorized\_keys file is simply a copy of the public key for that device. If one or more devices will be clients of the server, then the authorized\_keys file will contain a copy of all of the public keys. RSA and DSA keys may be freely mixed in the authorized\_keys file. For example, assume we already have one server, called bridge\_server, and two sets of keys, for the control\_room and the plant\_entrance:

```
$ ls /home/user/keys control_room control_room.pub plant_entrance plant_entrance.pub $ cat /home/user/keys/control_room.pub/home/user/keys/plant_entrance.pub > /home/user/keys/authorized_keys_bridge_server
```

### Uploading Keys

The keys for the server can be uploaded through the web interface, on the System: Administration page:

SSH RSA Public Key	<input type="text"/>	<input type="button" value="Browse..."/>
	Upload a replacement RSA public key file.	
SSH RSA Private Key	<input type="text"/>	<input type="button" value="Browse..."/>
	Upload a replacement RSA private key file.	
SSH DSA Public Key	<input type="text"/>	<input type="button" value="Browse..."/>
	Upload a replacement DSA public key file.	
SSH DSA Private Key	<input type="text"/>	<input type="button" value="Browse..."/>
	Upload a replacement DSA private key file.	
SSH Authorized Keys	<input type="text"/>	<input type="button" value="Browse..."/>
	Upload a replacement authorized keys file.	

## Bridging and Connecting with SSH for LES Series Console Servers

If only one client will be connecting, then simply upload the appropriate public key as the authorized keys file. Otherwise, upload the authorized keys file constructed in the previous step.

Each client will then need its own set of keys uploaded through the same page. Take care to ensure that the correct type of keys (DSA or RSA) go in the correct spots, and that the public and private keys are in the correct spot.

### SSH connecting using an SSH client

Use a secure protocol such as SSH, HTTPS or a VPN when connecting to (and through) LES1200/1300/1400/1500 console servers—particularly when connecting over the Internet or other public network.

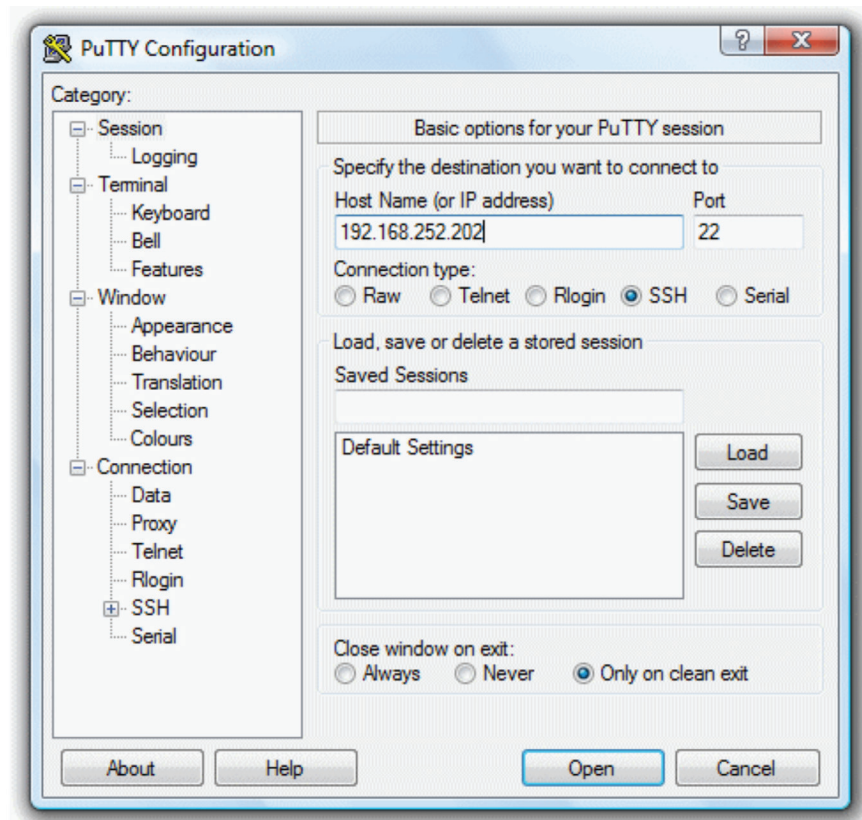
SSH will provide authenticated encrypted communications between the remote client program and the console server. Supported SSH client programs include:

- PuTTY is a complete (though not user-friendly) freeware implementation of SSH for Windows and UNIX platforms.
- SSHTerm is a useful open source SSH communications package.
- SSH Tectia is leading end-to-end commercial communications security solution for the enterprise.
- Reflection for Secure IT (formerly F-Secure SSH) is another good commercial SSH-based security solution.
- SDTConnector is a free open source SSH Java client that simplifies secure connectivity with attached network and serial devices.

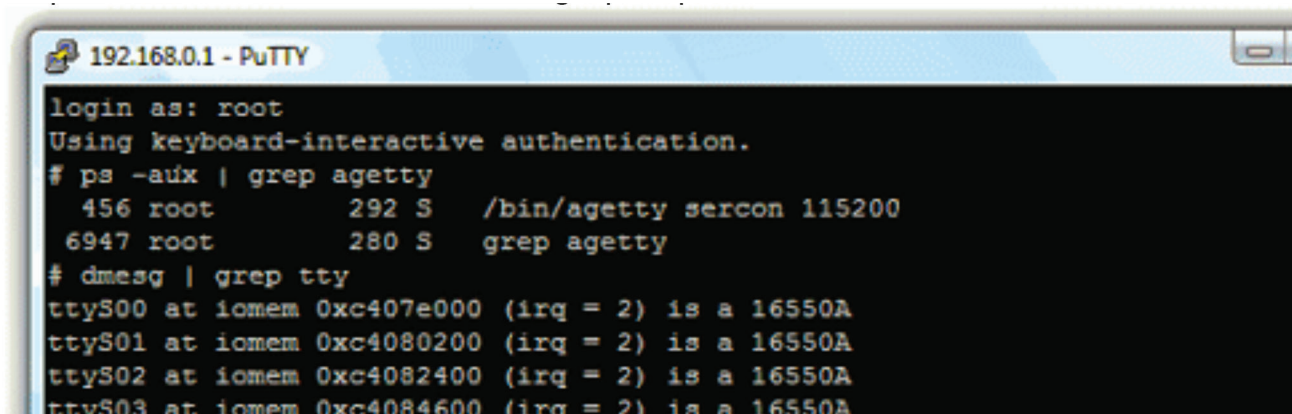
### SSH connection using PuTTY

PuTTY runs as an executable application that can be freely downloaded from the PuTTY Download Page.

- Enter the console server's IP address as "Host Name (or IP address)."



- To connect to the console server itself, select "SSH" and enter 22 as the Port. Then click the "Open" button and you will be presented with the command line login prompt to access the console server's Linux kernel:



```
login as: root
Using keyboard-interactive authentication.
# ps -aux | grepagetty
  456 root      292 S    /bin/agetty sercon 115200
 6947 root      280 S    grepagetty
# dmesg | grep tty
ttyS00 at iomem 0xc407e000 (irq = 2) is a 16550A
ttyS01 at iomem 0xc4080200 (irq = 2) is a 16550A
ttyS02 at iomem 0xc4082400 (irq = 2) is a 16550A
ttyS03 at iomem 0xc4084600 (irq = 2) is a 16550A
```

You can now check to see if agetty is running ( `# ps -aux | grepagetty` ), or check which serial ports were detected during boot ( `# dmesg | grep tty` ), or execute other commands.

- You can also SSH connect directly to serially connected Managed Devices. The SSH port address for direct access to a serially connected device is IP Address - Port (3000 + serial Port #). So to connect to a Managed Device on Port 1, set the "TCP port" to 3001. Again, click "Open" and you will be presented with the login prompt from the remote Managed Device.
- Alternately, SSH connections to attached devices can be configured using the standard SSH port 22. The serial port being accessed is then identified by appending a descriptor to the username. This syntax supports any of:

< username> :< portXX>

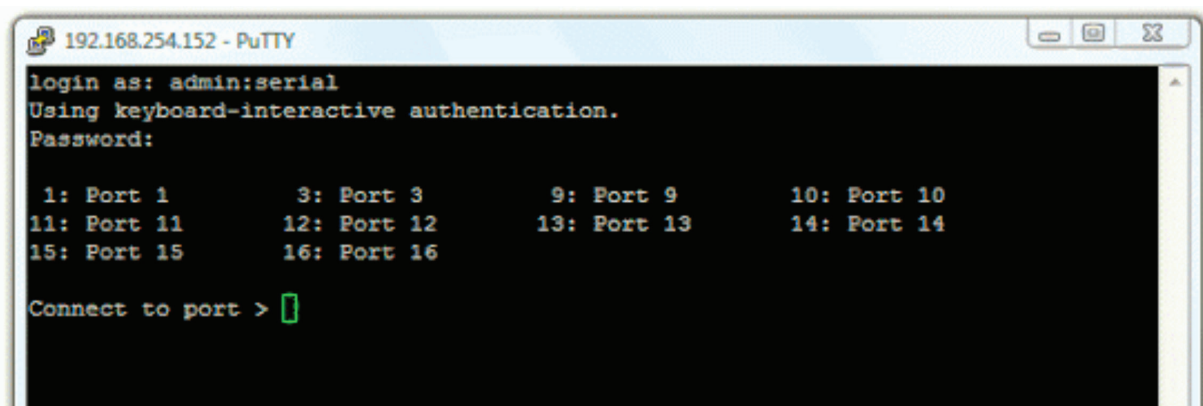
< username>:< port label>

< username>:< ttySX>

< username>:< serial>

So for a User named "fred" to access serial port 2, when connecting with the SSH client (e.g. SSHTerm or PuTTY SSH) instead of responding with username = "fred" and ssh port = "3002" the alternate is to type username = "fred:port02" (or username = "fred:ttyS1") and ssh port = 22.

Another option is to type username="fred:serial" and ssh port = 22 ... and the user will be presented with a port selection menu option:



```
login as: admin:serial
Using keyboard-interactive authentication.
Password:

  1: Port 1      3: Port 3      9: Port 9     10: Port 10
11: Port 11     12: Port 12    13: Port 13    14: Port 14
15: Port 15     16: Port 16

Connect to port > 
```

The above syntax enables users to set up SSH tunnels to all serial ports on a console server with only a single IP port 22 having to be opened in the console server/firewall.

When you have finished, you can logout using the escape keys. The default escape key is "~" and the key to close is the "." key. So to close the session, first press the enter key to be on a new line and then press the ~. keys to terminate the ssh connection.





**Black Box Tech Support: FREE! Live. 24/7.**

Tech support the  
way it should be.



Great tech support is just 60 seconds away at 877-877-2269 or [blackbox.com](http://blackbox.com).



**Disclaimer:**

Black Box Network Services shall not be liable for damages of any kind, including, but not limited to, punitive, consequential or cost of cover damages, resulting from any errors in the product information or specifications set forth in this document and Black Box Network Services may revise this document at any time without notice.

**About Black Box**

Black Box provides an extensive range of networking and infrastructure products. You'll find everything from cabinets and racks and power and surge protection products to media converters and Ethernet switches all supported by free, live 24/7 Tech support available in 60 seconds or less.

© Copyright 2016. Black Box Corporation. All rights reserved. Black Box® and the Double Diamond logo are registered trademarks of BB Technologies, Inc. Linux is a registered trademark of Linus Torvalds. Any third-party trademarks appearing in this manual are acknowledged to be the property of their respective owners.