

Telnet and SSH

ON THIS PAGE

- [Enabling Telnet](#)
 - [Disabling Telnet](#)
- [Enabling SSH](#)
 - [Disabling SSH](#)
- [Accessing the Telnet/SSH Connection](#)
- [Using Telnet/SSH with Serial](#)
- [Additional Notes](#)

This page outlines how to enable Telnet and/or SSH on a networked BrightSign player. You can use Telnet/SSH to connect to the BrightSign application console, which hosts the [BrightScript debug console](#) and receives `print` messages from BrightScript. This is useful when the serial port (the default I/O for the BrightSign application console) is inaccessible. Telnet and SSH have been tested and proven to work with PuTTY, but any client with Telnet/SSH support should work.

Telnet offers no security features, while SSH uses encrypted sessions and requires credentials to access the shell. However, for security reasons, we do not recommend enabling either Telnet or SSH in a production environment; they should only be used for debugging presentations in a test environment.

Enabling Telnet

Important

Telnet support requires firmware version 4.8.88 or later.

To enable Telnet on a player, run an *autorun.brs* script containing the following code:

```
reg = CreateObject("roRegistrySection", "networking")
reg.write("telnet", "23")
reg.flush()
```

This will allow a Telnet connection on port 23 (the default port for Telnet). You can change the second parameter on the `write()` method to open Telnet on a different port if desired.

Telnet will not be enabled until the script exits and the player reboots. You can use the `RebootSystem()` function to terminate the *autorun.brs* script and reboot the player.

Disabling Telnet

To disable Telnet, run an *autorun.brs* script containing the following code:

```
reg = CreateObject("roRegistrySection", "networking")
reg.delete("telnet")
reg.flush()
```

The player will need to be rebooted for this setting to take effect.

Enabling SSH

Important

SSH support requires firmware version 5.0.22 or later.

To enable SSH on a player, run an *autorun.brs* containing the following code:

```
reg = CreateObject("roRegistrySection", "networking")
reg.write("ssh", "22")

n=CreateObject("roNetworkConfiguration", 0)

n.SetLoginPassword("password")
n.Apply()
reg.flush()
```

This will allow an SSH connection on port 22 (the default port for SSH). You can change the second parameter on the `Write()` method to open SSH on a different port. Note that the interface number used to create the *roNetworkConfiguration* instance (0 for Ethernet or 1 for WiFi) is not significant.

The code also sets the SSH login password to "password", which can be changed by editing the `SetLoginPassword()` parameter. The username will always be "brightsign".

SSH will not be enabled until the script exits and the player reboots. You can use the `RebootSystem()` function to terminate the *autorun.brs* script and reboot the player. Alternatively, you can use the `END` function to exit the script, then reboot the player manually.

Disabling SSH

To disable SSH, run an *autorun.brs* script containing the following code:

```
reg = CreateObject("roRegistrySection", "networking")
reg.delete("ssh")
reg.flush()
```

The player will need to be rebooted for this setting to take effect.

Accessing the Telnet/SSH Connection

To connect to the player via Telnet/SSH, enter the IP address of the player with the configured port into your client application. You can obtain the IP address of the player by booting up the player without an SD card.

Since DHCP-configured devices often change IP addresses when rebooted, this process will be easier if you assign the player a static IP address during setup. If this is not feasible, you may be able to connect to the player using the following (this requires Linux/macOS or Windows with Bonjour installed):

```
telnet brightsign-<serial-number>.local
```

Using Telnet/SSH with Serial

By default, enabling debugging output via Telnet or SSH will disable such output over serial. The serial port can still be used for interaction with serial devices via the *roSerialPort* object.

It is possible to enable debugging output over both Telnet/SSH and serial; however, interaction with serial devices via the *roSerialPort* object will become highly unreliable. To enable debugging output over serial as well, run a script containing the following code after Telnet/SSH has been enabled:

```
reg = CreateObject("roRegistrySection", "networking")

print "Trying to enable serial with Telnet/SSH"
print reg.write("serial_with_telnet", "1")
reg.flush()
```

Note

Though the registry key is labeled "serial_with_telnet", it applies to both Telnet and SSH configurations.

Additional Notes

- Multiple simultaneous connections are permitted via Telnet or SSH. The output is mirrored as a single, shared session.
- A running script can be interrupted by pressing Ctrl-C (if script debugging is disabled, you may need to press Ctrl-C twice).
- The SSH password remains set even if SSH access is disabled. There is currently no way to clear the password.
- There is currently no support for authenticating against an SSH public key.