

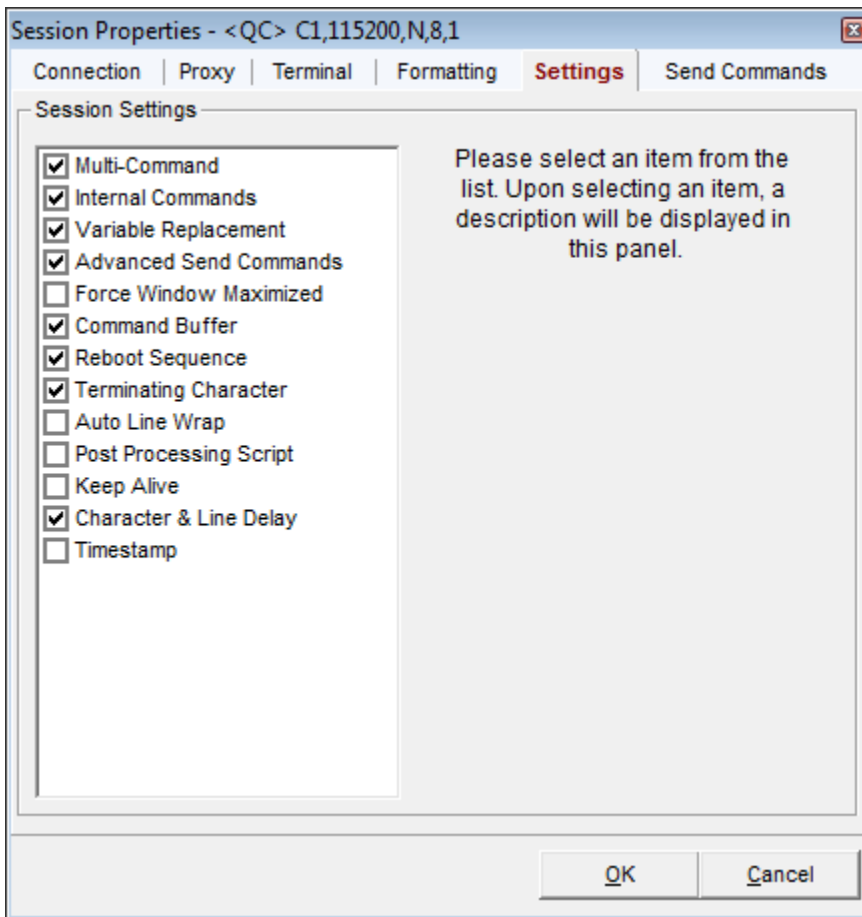
Advanced Session Settings

Overview

The *Settings* in the [Session Properties](#) editor tab is used to enable, disable, and configure some of the more advanced tools and options for a terminal session.

The individual features/tools are listed on the left side of the screen and include check boxes to designate if the feature is enabled or disabled.

To enable or disable a feature, you must place or remove the check from the check box in the feature listing.



If you select a feature in the listing, a feature description and additional feature configuration options will appear on the right side panel.

Please select from the following feature sections for more information on each feature:

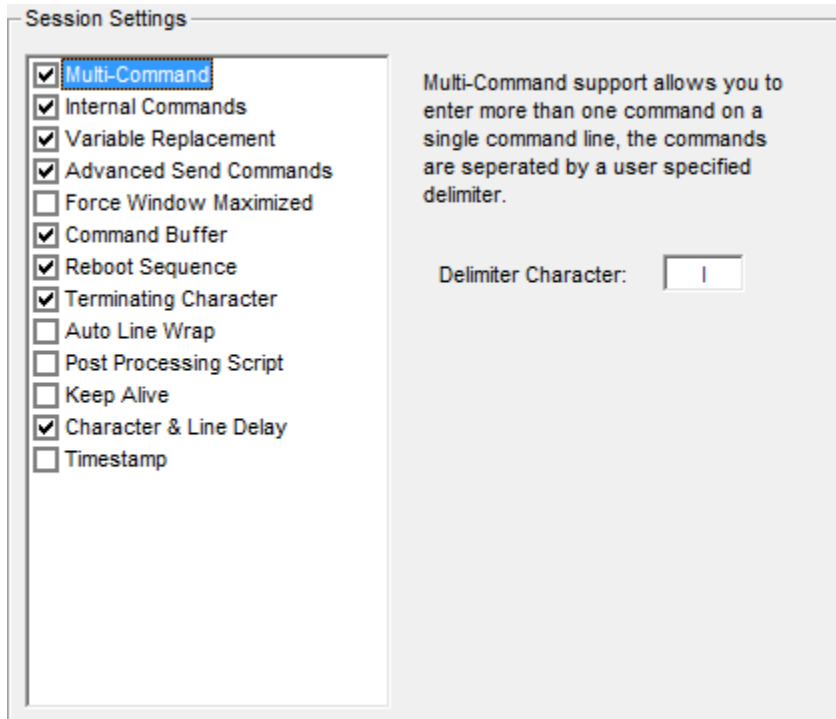
- [Multi-Command](#)
- [Internal Commands](#)
- [Variable Replacement](#)
- [Advanced Send Commands](#)
- [Force Window Maximized](#)
- [Command Buffer](#)
- [Reboot Sequence](#)
- [Terminating Characters/Bytes](#)
- [Auto Line Wrap](#)
- [Post Processing Script](#)
- [Keep Alive](#)

- [Character & Line Delay](#)
- [Timestamp](#)

Multi-Command

If this option is enabled, the terminal session will support [Multi-Commands](#).

You can also change the multi-command delimiter character on this configuration panel.



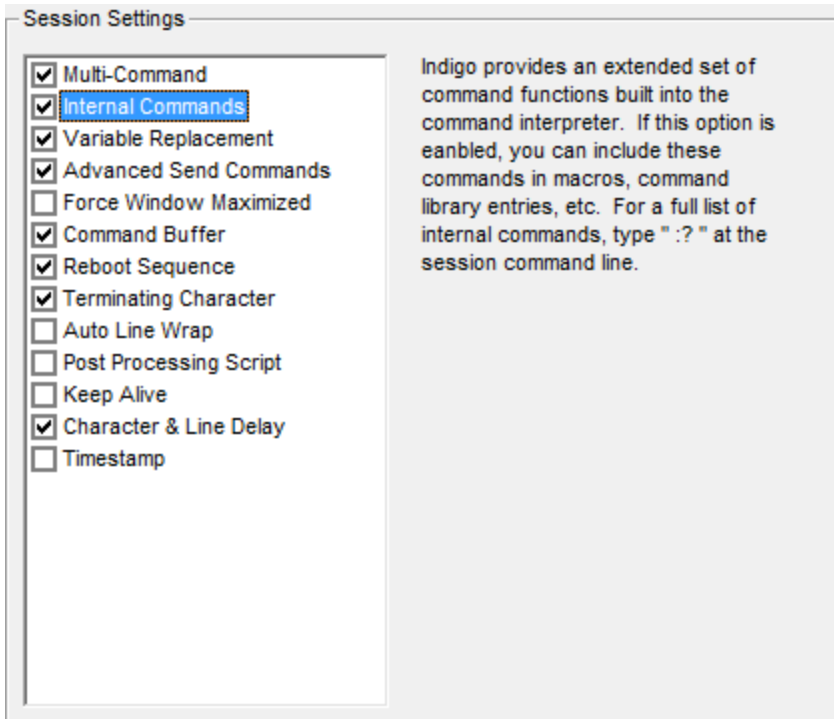
✓ Tip

Some devices/hosts may need the use of the pipe "|" character thus you can choose to disable the multi-command or change the multi-command delimiter to another character.

Internal Commands

The [Internal Commands](#) feature can be enabled or disabled on this configuration panel.

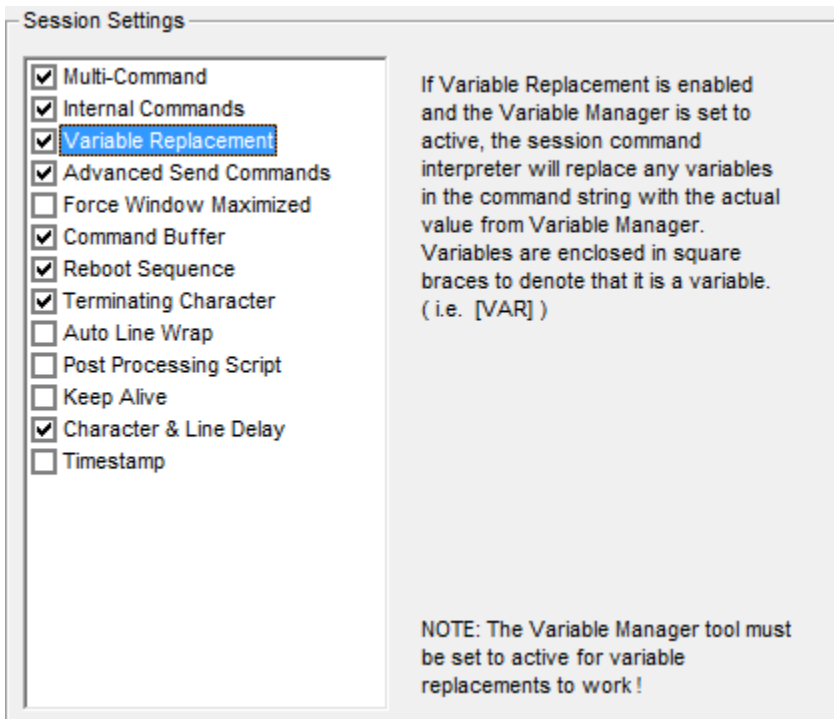
If the internal command processor is interfering with your terminal session, then you can disable it for the terminal session.



Variable Replacement

Indigo support a [variable replacement](#) feature allowing defined variables to be included in instruction commands and the Indigo interpreter would replaced the variables with the actual values.

This feature can be disabled for the terminal session.

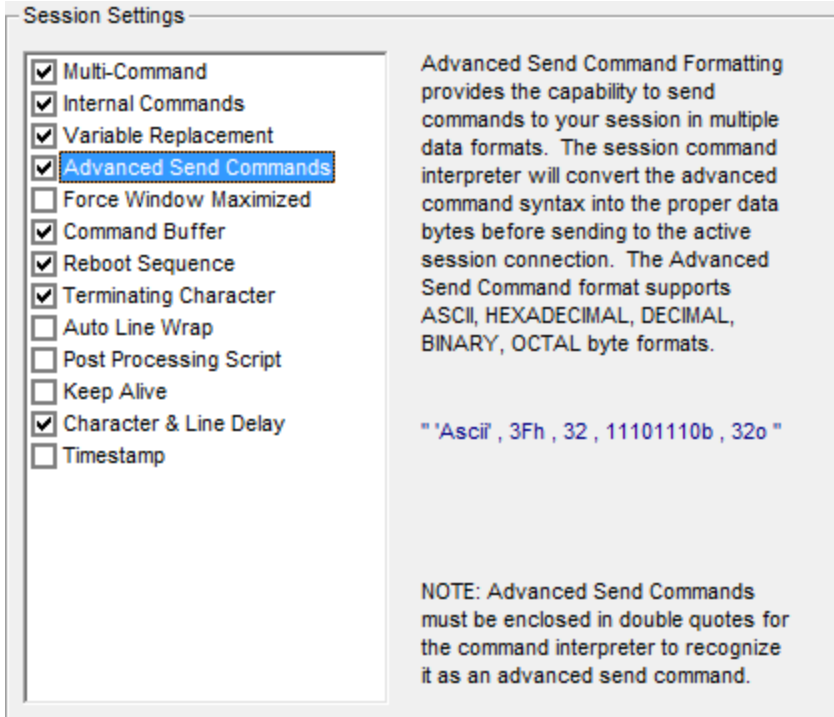


Advanced Send Commands

Indigo supports sending instruction commands in multiple data byte representations using the [Advanced Send](#)

[Commands](#) feature.

If the advanced command syntax interferes with your terminal communication, you can disabled the feature on this configuration panel.



The screenshot shows a 'Session Settings' window. On the left, a list of options is shown with checkboxes. The 'Advanced Send Commands' option is checked and highlighted with a blue box. To the right, there is a descriptive text block and a code example.

Multi-Command
 Internal Commands
 Variable Replacement
 Advanced Send Commands
 Force Window Maximized
 Command Buffer
 Reboot Sequence
 Terminating Character
 Auto Line Wrap
 Post Processing Script
 Keep Alive
 Character & Line Delay
 Timestamp

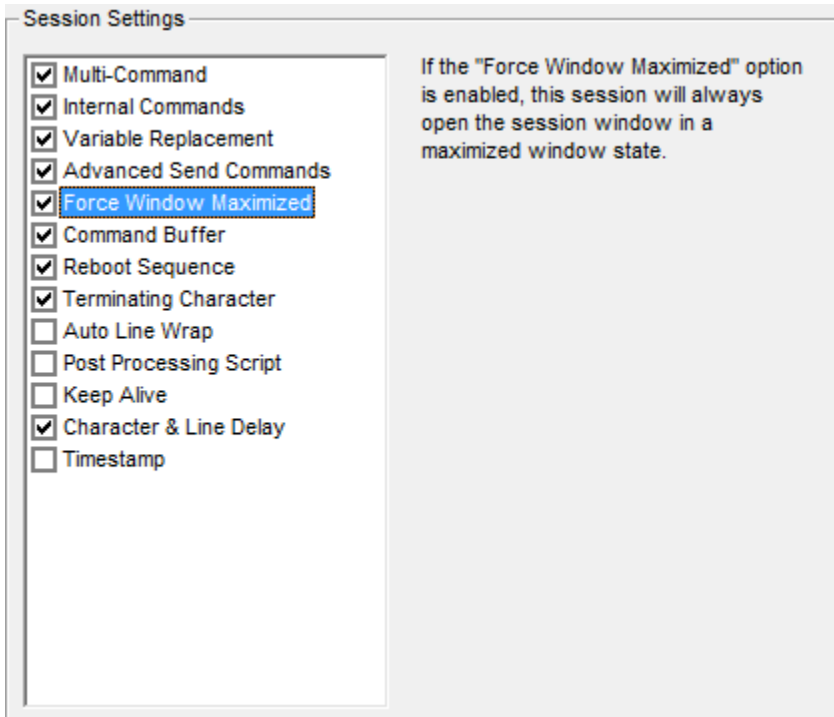
Advanced Send Command Formatting provides the capability to send commands to your session in multiple data formats. The session command interpreter will convert the advanced command syntax into the proper data bytes before sending to the active session connection. The Advanced Send Command format supports ASCII, HEXADECIMAL, DECIMAL, BINARY, OCTAL byte formats.

`" 'Ascii' , 3Fh , 32 , 11101110b , 32o "`

NOTE: Advanced Send Commands must be enclosed in double quotes for the command interpreter to recognize it as an advanced send command.

Force Window Maximized

If this option is enabled, this terminal session window will always be opened in the maximized states.



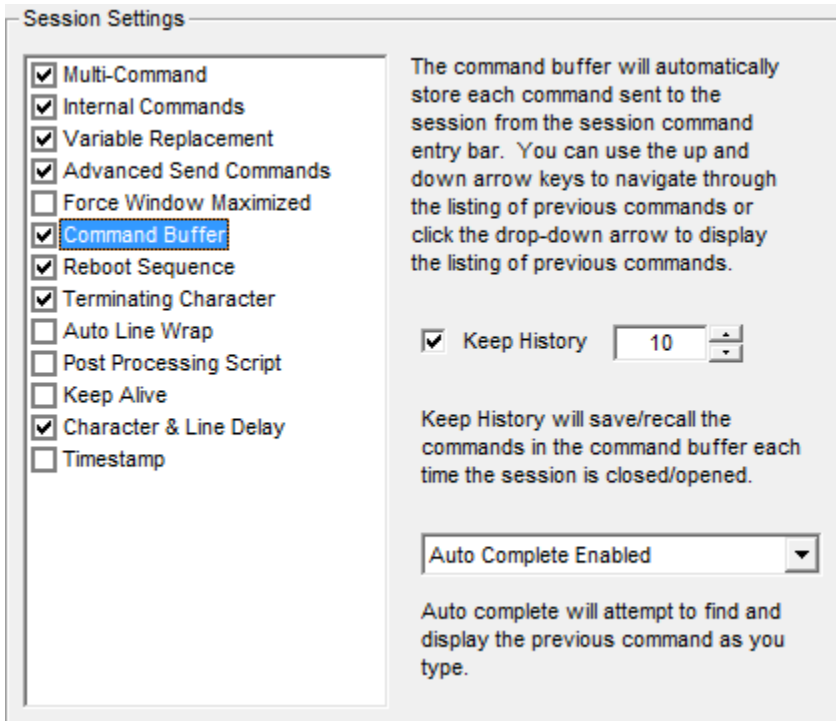
The screenshot shows a 'Session Settings' window. On the left, a list of options is shown with checkboxes. The 'Force Window Maximized' option is checked and highlighted with a blue box. To the right, there is a descriptive text block.

Multi-Command
 Internal Commands
 Variable Replacement
 Advanced Send Commands
 Force Window Maximized
 Command Buffer
 Reboot Sequence
 Terminating Character
 Auto Line Wrap
 Post Processing Script
 Keep Alive
 Character & Line Delay
 Timestamp

If the "Force Window Maximized" option is enabled, this session will always open the session window in a maximized window state.

Command Buffer

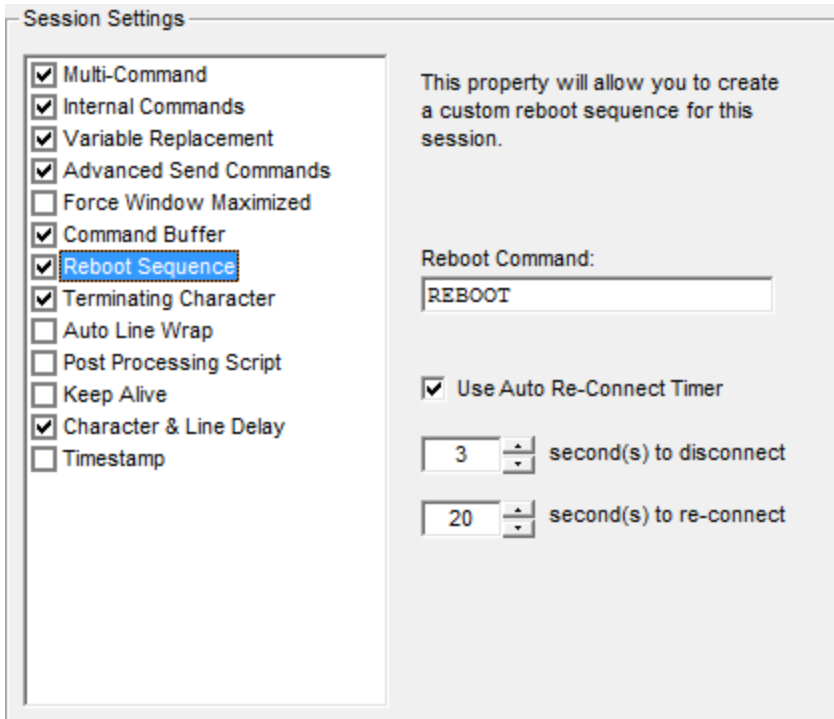
If this option is enabled, Indigo will buffer each instruction command submitted via the [command bar](#). Additional configuration options are available to control buffered command persistence and the auto-complete behavior of the command bar. Please see this [page](#) for more information on the command bar and command buffer.



Auto Complete Options	Description
Auto Complete Disabled	No auto-complete is attempted on the command bar.
Auto Complete Enabled	Command auto-complete is enabled on the command bar.
Case Sensitive Auto Complete Enabled	A case sensitive command auto-complete is enabled on the command bar.

Reboot Sequence

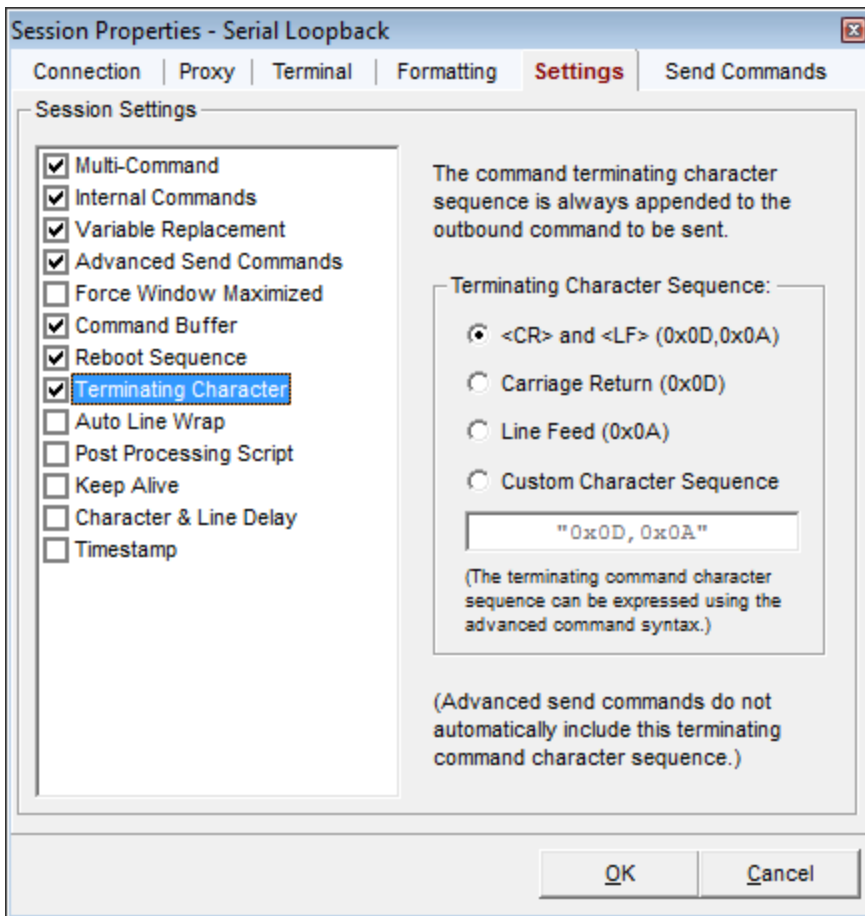
Indigo supports a [custom reboot sequence](#) feature that can issue a reboot instruction to the connected terminal session and then perform timed disconnect and re-connect of the session. This configuration panel provide the reboot sequence options for the terminal session. The [custom reboot sequence](#) can also be disabled if you have no need for it.



Terminating Characters/Bytes

Commands sent to a terminal connection automatically include a set of line [terminating characters/bytes](#). By default, Indigo includes both a carriage return (0x0D) and line feed (0x0A); however, if your device/host requires a different set of command terminating characters/bytes, you can modify them on this configuration panel.

The character sequence can be expressed using the [advanced send command](#) syntax.



Command Terminating Character Options	Description
<CR> and <LF> (0x0D,-0x0A)	This option will always send a carriage return and line feed appended to each command instruction submitted to the connected device/host in a terminal session. <i>(This is the default option in Indigo.)</i>
Carriage Return (0x0D)	This option will always send a carriage return appended to each command instruction submitted to the connected device/host in a terminal session.
Line Feed (0x0A)	This option will always send a line feed appended to each command instruction submitted to the connected device/host in a terminal session.
Custom Character Sequence	This option will allow the user to specify a custom character sequence that will be appended to each command instruction submitted to the connected device/host in a terminal session. <i>(The custom character sequence can be expressed using the advanced send command syntax.)</i>

Character	Byte in Decimal	Byte in Hex
Carriage Return <CR>	13	0x0D

Line Feed <LF>	10	0x0A
----------------	----	------

Note

The command terminating characters/bytes do not apply to [advanced send commands](#) or [keyboard redirected](#) input using the Terminal Emulation view mode.

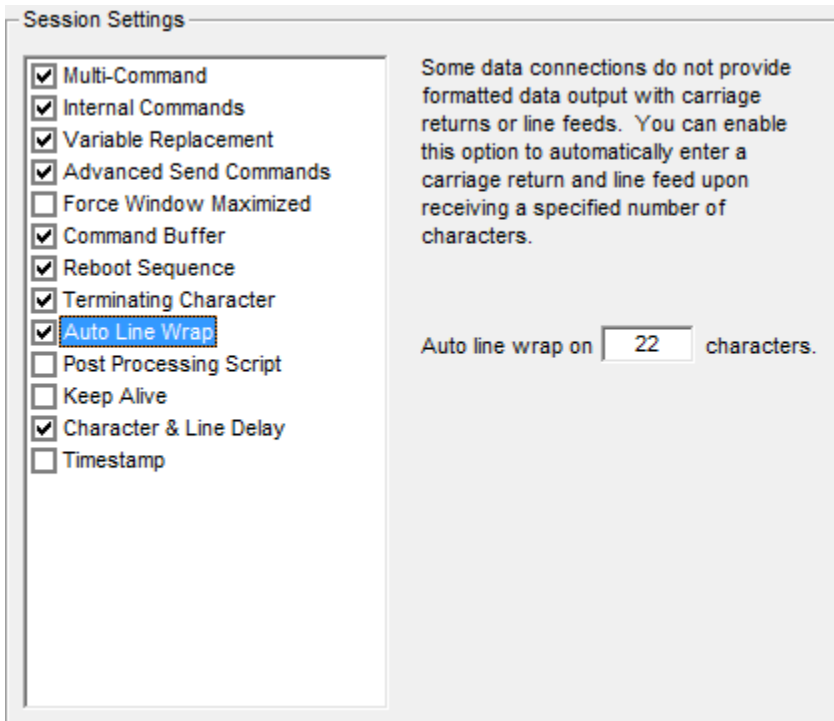
Auto Line Wrap

By default Indigo only performs line [wrapping](#) in the session data window when a line termination character is received.

Some devices/hosts do not send any carriage return (0x0D) or line feed (0x0A) characters and you may want the data to be displayed on multiple lines in the data window.

Enabling this option will force the terminal session to perform a line wrap after the specified number of characters have been rendered to a row on the screen.

This feature is only supported in the [Standard view mode](#).

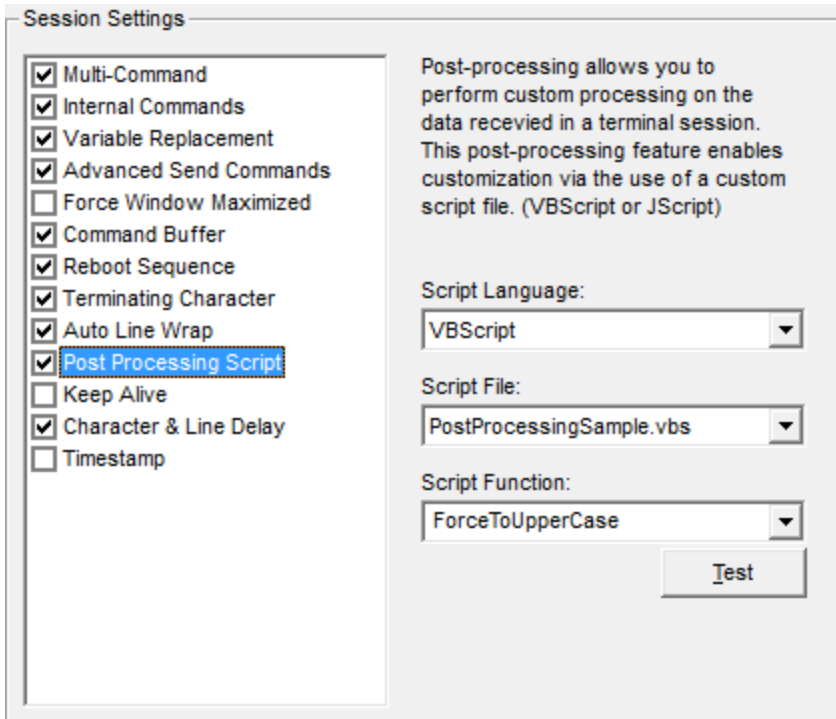


Post Processing Script

Indigo supports a [scripting](#) feature that allows a user created script to analyze and modify data after it has been received by the terminal session and before it is rendered to the screen.

This configuration panel allow you to select the custom script file and processing function to perform.

The scripting feature can be enabled or disabled for the terminal session using this configuration option.

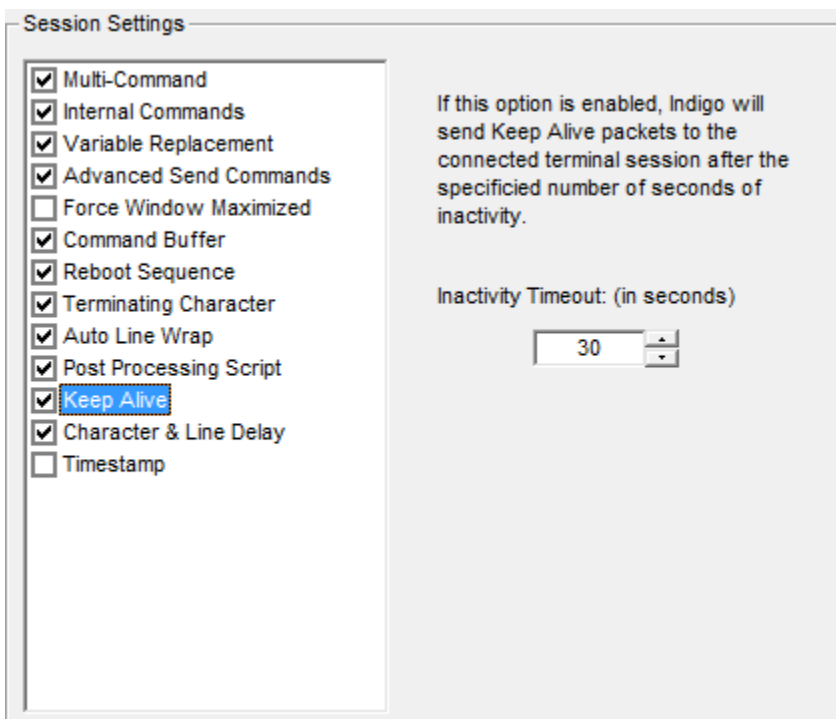


For more information on scripting, please see the [Scripting](#) help section.

Keep Alive

Some terminal connections may support [keep alive](#) packets and Indigo can be configured to send the keep alive instruction at regular intervals to prevent the device/host from terminating the connection.

This feature may be useful if you are monitoring a device/host over time and the device/host forcefully disconnects if no activity is detected.

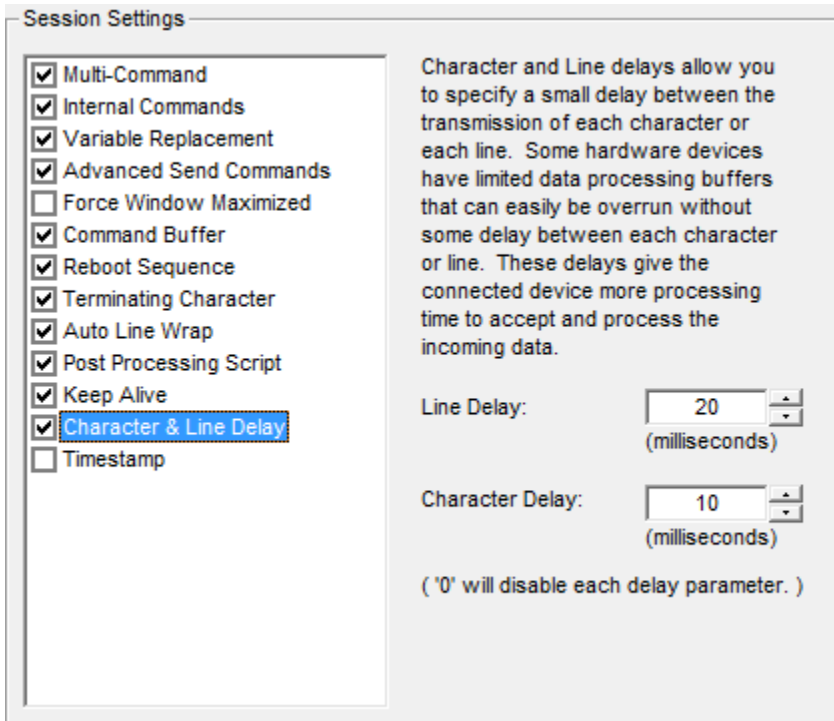


Character & Line Delay

Indigo typically transmits data commands as entire chunks of data in large packets. However, some device may not be able to process large data packets or fast data transmissions.

You can enable this feature to buffer the outgoing command data and [inject delays between character and line transmissions](#).

This feature may be necessary when working with limited resource embedded devices.



Timestamp

Indigo can automatically prepend data lines with [timestamps](#) to help record when data has been received by Indigo. Since the line formatting can vary between each type of terminal connection, you must configure the type of end line delimiter that Indigo should expect so that it can insert the timestamp at the appropriate locations.

Session Settings

- Multi-Command
- Internal Commands
- Variable Replacement
- Advanced Send Commands
- Force Window Maximized
- Command Buffer
- Reboot Sequence
- Terminating Character
- Auto Line Wrap
- Post Processing Script
- Keep Alive
- Character & Line Delay
- Timestamp**

This option displays a timestamp at the beginning of each line in the data receive window.

(Note: this timestamp does not apply to the Direct Terminal data window.)

End of Line Delimiter

This is used to determine where new lines begin and where to insert the timestamp data.

- Line Feed (\$0A)
- Carriage Return (\$0D)
- <CR> and <LF> (\$0D,\$0A)
- Ignore new line, add timestamp on each data packet.

Note

If using the option to include a timestamp on each data packet received, Indigo will not limit the timestamp to new lines, but inserts timestamp throughout the data stream.