User Manual

Get Console™

For Get Console Version: 1.80

Date: 5 April 2012

1.1 Table of Contents

2 Introduction to Get Console

2.1 What is the Get Console™ App?

Get Console is an Apple App Store distributed terminal application for Apple iPads and iPhones that allows IT engineers to connect and control RS232 serial ports (such as to the console port of Cisco devices) or run Telnet, RAW, or SSH(v2) terminal sessions over WIFI and 3G.

With the Get Console app, IT engineers can perform IT equipment maintenance, troubleshooting and disaster recovery operations directly on the serial device console ports, or over Wifi/3G using the instant on, portable iPad, iPhone or iPod Touch - all WITHOUT jailbreaking.

The Get Console app now has many features including:

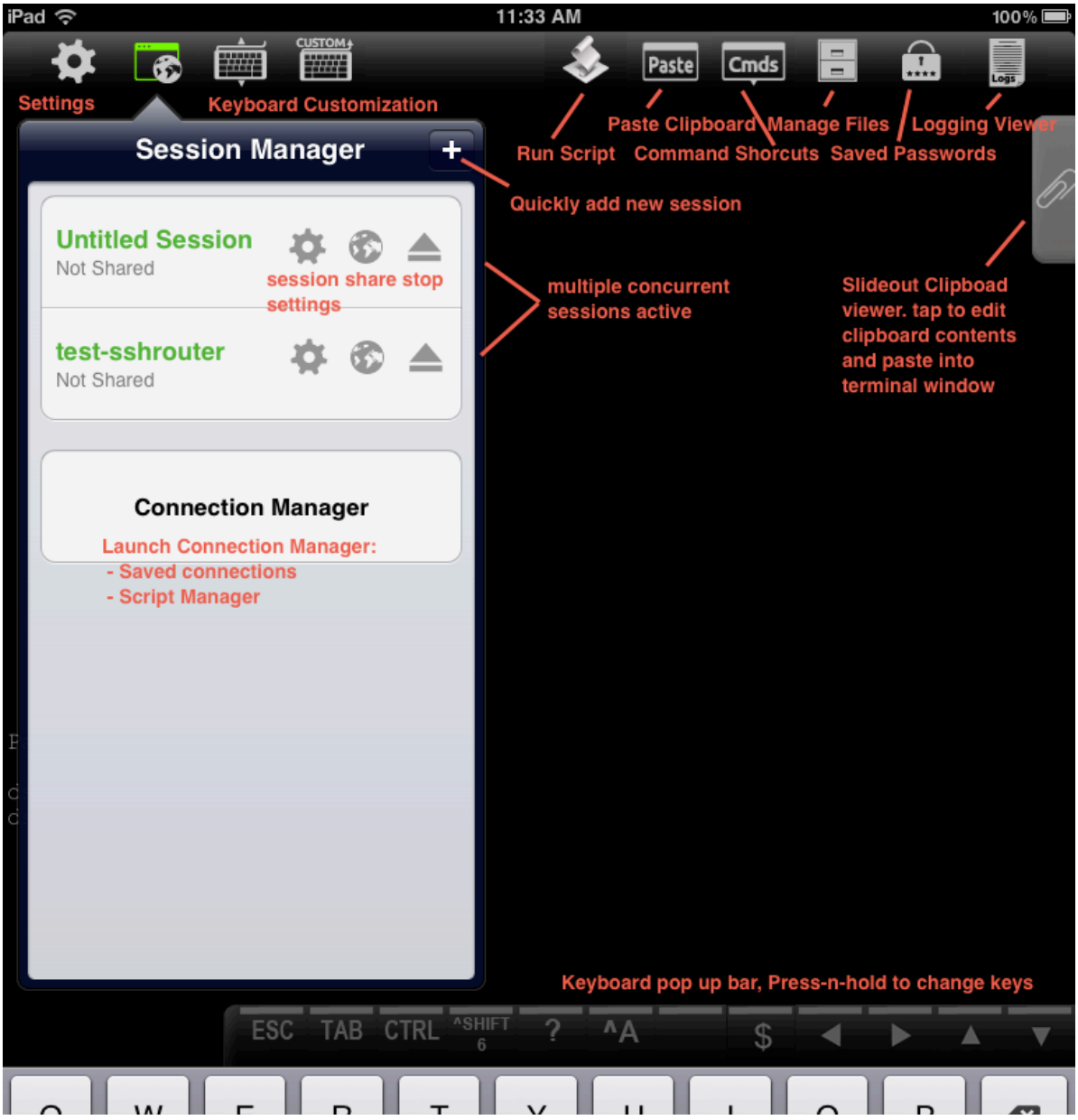
- Physical serial console access to network and other equipment via the Redpark C2-RJ45 cable (supports 300-57600 baud, optional parity, optional flowcontrol, variable stopbits and either 7 or 8 databits)
- SSHv2, Telnet, and RAW connectivity over WIFI/3G
- VT100, xterm and many other common Terminal emulations
- Encoding support for ASCII, UTF-8 and 17 other major encoding formats, including multibyte characters for non-latin text terminals
- Maintains multiple sessions concurrently, including when the App is in the background for upto 10 minutes.
- Session import from SecureCRT® and PuTTY via Windows tool and iTunes
- Full Session Scripting using simple “Expect X, then Send Y” type scripting.
- One Tap Secure Screen Sharing allows a remote web user to view and interact with the iPad/iPhones terminal window
- Cloud connectivity to both get-console.com and Dropbox.com for dynamically downloading configuration files and scripts and uploading session log files
- Tight integration with iPhone/iPad clipboard and cut/copy/paste directly from the terminal window
- Comprehensive logging support, Command Shortcuts, Stored Passwords, Bluetooth Keyboard and many other features.

2.2 Get Console App Requirement

- iPhone3 + (iOS version 4.2 or later is required)
- iPad1 onwards
- Get Console RJ-45 or serial cable or Wifi configured to connect to the network device
- Wifi/3G connection for sharing session

2.3 Terminal Overview

The below picture shows the main Get Console window and identifies the major UI features.



Clicking on the Main Settings button  will where most of the apps features are configured.

3 Get Console Main Settings

The Main Settings page is accessible via the  button. This sets the application Default settings for all sessions. Some of these defaults can be varied on individual saved connections (See Section 4 Connection Manager). The App settings cover 5 main areas:

- **Terminal Settings** – features specific to the terminal window – things like linewrap, local echo, scrollback size, but also access to the terminal scripting features.
- **Keyboard Settings** – features for how the keyboard behaves (with or without external Bluetooth keyboard running)
- **Serial Settings** – features specific to Serial Connections (ie baud rate)
- **SSH Settings** – features specific to SSH Connections (ie Certificates/Keys, Authentication methods)
  Sharing Settings – settings requiring configuration in order to share the terminal window with a remote user via the get-console.com website.

The following section details each of these setting areas.

3.1 Terminal Settings

3.1.1 Columns (Default: 80)

Supports any column width up to 132 characters wide

3.1.2 Rows (Default: 24)

Supports any number of rows on the screen

3.1.3 Scrollback (Default: 100)

Number of lines retained by the console for review. Setting this above 250 will impact performance on iPhone4 and below. iPad2, iPad3 and iPhone 4S supports 500 scrollback without impact due to improved hardware specifications.

3.1.4 Clear On Connect

Defines whether the terminal should clear the screen when starting a new connection

The default is ON

Options:
- **On**
- **Off**

3.1.5 LineWrap

Defines whether the terminal should automatically wrap the line to the next line when the output received from the remote device exceeds the column width and no carriage return has been received.

The default is ON

Options:
- **On**
- **Off**

3.1.6 Text Size

Sets font size in the terminal window

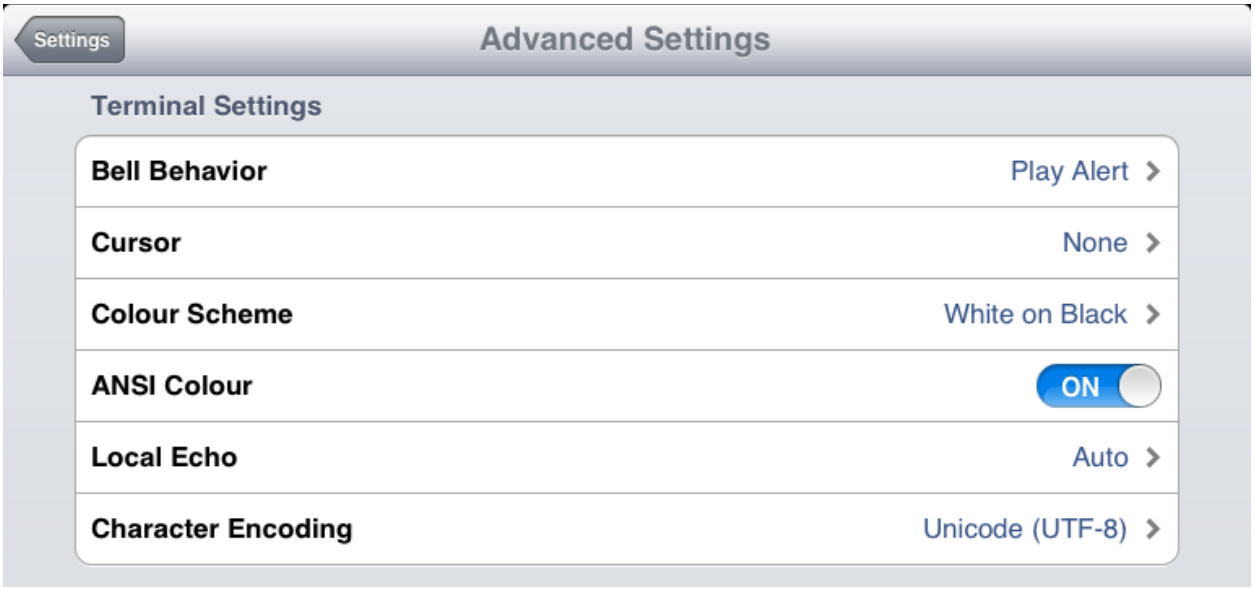
Smaller ---------- Larger

3.1.7 Console Logging (Default: Off)

Logs all printable output to file stored on iOS device. The log files can be uploaded to www.get-console.com website, Dropbox.com folder or extracted via iTunes. Log upload is performed via the Files section described below.

Options:
- **On**
- **Off**

3.1.8 Advanced Terminal Settings



3.1.8.1 Bell Behavior

Allows the iOS device to create an alert when there is an Terminal event or stay silent.

Options:
- **Play Alert (bell/vibrate phone)**
- **Do Nothing (silent)**

3.1.8.2 Cursor

Allow the choice of cursor style – the default is None

Options:
- **None**
- **Outline**
- **Block**
- **Vertical Line**
- **Underline**

3.1.8.3 Colour Scheme

Changes the colour scheme of the text and background of the console

Options:	
- **White on Black**	White text on Black background
- **Black on White**	Black text on White background
- **White on Black/Grey**	White text on Black/Grey background
- **Green on Black**	Green text on Black background
- **Rainbow**	Helps debugging by colouring different parts of the screen ○ Blue: Left and right border

	○ Red: Upper border of the scrollback limit ○ Black: Active area of the console within the scrollback limit

3.1.8.4 ANSI Colour

Allows up to 256 colours on terminal window following the ANSI spec for signaling colour. The default is ON.

Options:
- **On**
- **Off**

3.2 Script Manager

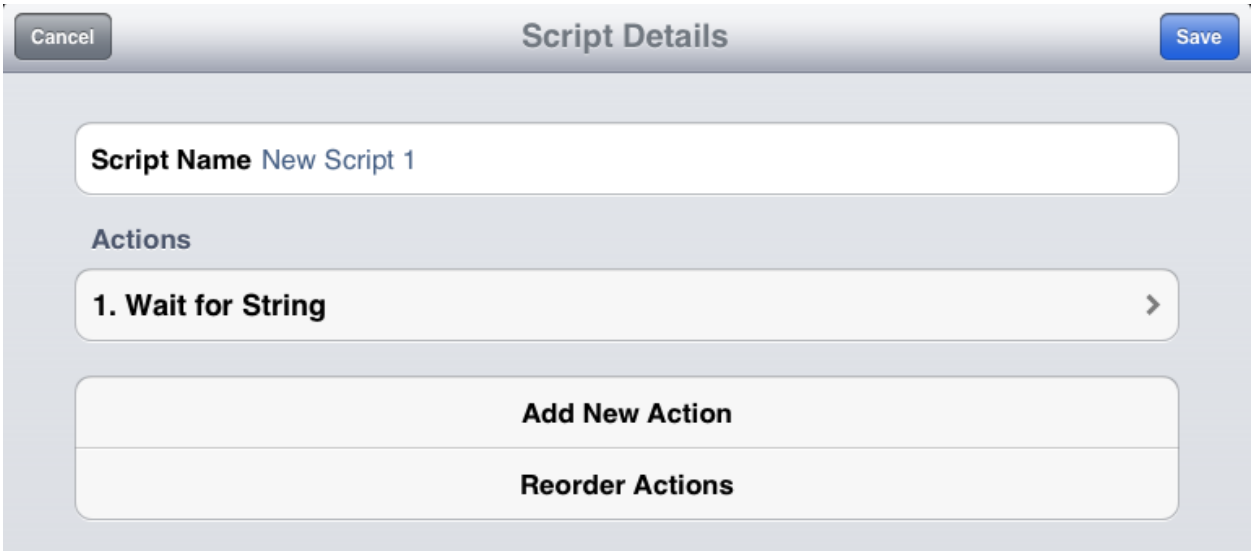
The script manager allows for the creation of terminal scripts. Once created, these scripts can be run immediately on connection by assigning them to saved connection in the Connection Manager, or alternatively by running once a terminal session is started via the Run-Script button on the top bar (iPad) or Tools menu (iPhone).

Scripts are built by adding various action items to the script in order that the terminal should execute them. Usually the script will begin by telling the terminal to wait to see something on the screen (ie the "Wait-for-String" action item) and then the next action item will be to do something (send command, start logging etc). These type of scripts are known as "Expect Scripts" in that while the script is running it will "Expect" to see "X" on the screen, and when it does it will respond with "Y". After creating the script and running it the items are completed sequentially from first item to last.

The below section describes the various action item options are and their configurable parameters. Note that poorly constructed scripts will possibly cause the terminal window to lock or perform unpredictably. A common cause for script problems not configuring a timeout for a Wait-for-String action item – if this is not configured, the script will never progress to the next action item as it has not seen the string it is looking for.

3.2.1 Create New Script

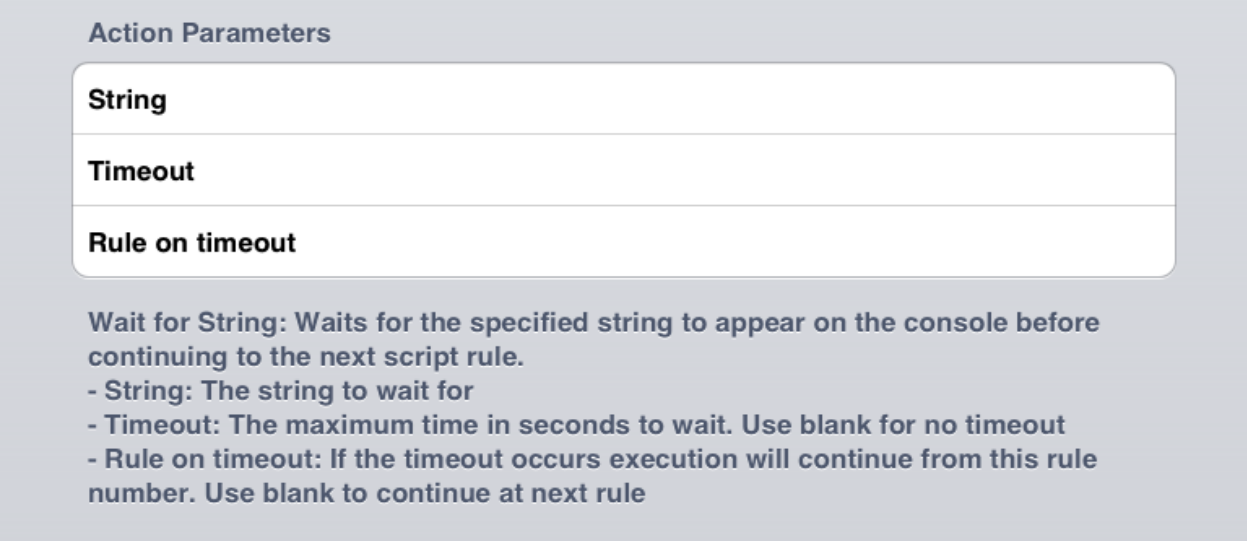
To create a new script, navigate to the Script Manager and tap the Create New Script button. Give the script a name, and then tap on Action 1.



By default the first action is Wait for String, however it can be any of the following action items. Change by tapping item 1 which then allows one of all of the possible action types to be selected.

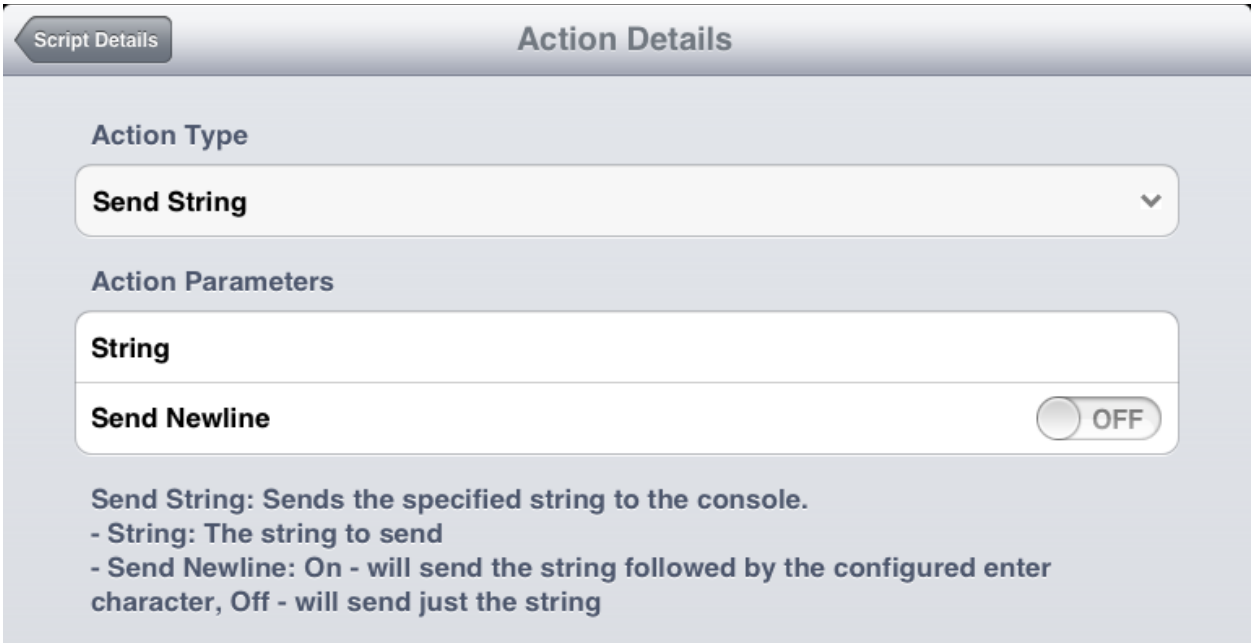
At the bottom of the screen is where the parameters for the action item is configured, along with some guidance text on which and the type values that can be entered.

3.2.2 Wait for String Action Item



Note that the Wait for String action is quite powerful in that it can be used to create loops via the "Rule on timeout" parameter. By default the script executes each item in the script sequentially, however using the Rule on timeout, the script could alternatively jump to earlier or later action items. For example if the script must keep pressing enter until it sees "End of File" then the Wait For String rule is configured to wait to see "End of File", and if it does not see it after 2 seconds then it jumps back to an earlier "Send String" action item that sends the enter key.

3.2.3 Send String Action Item

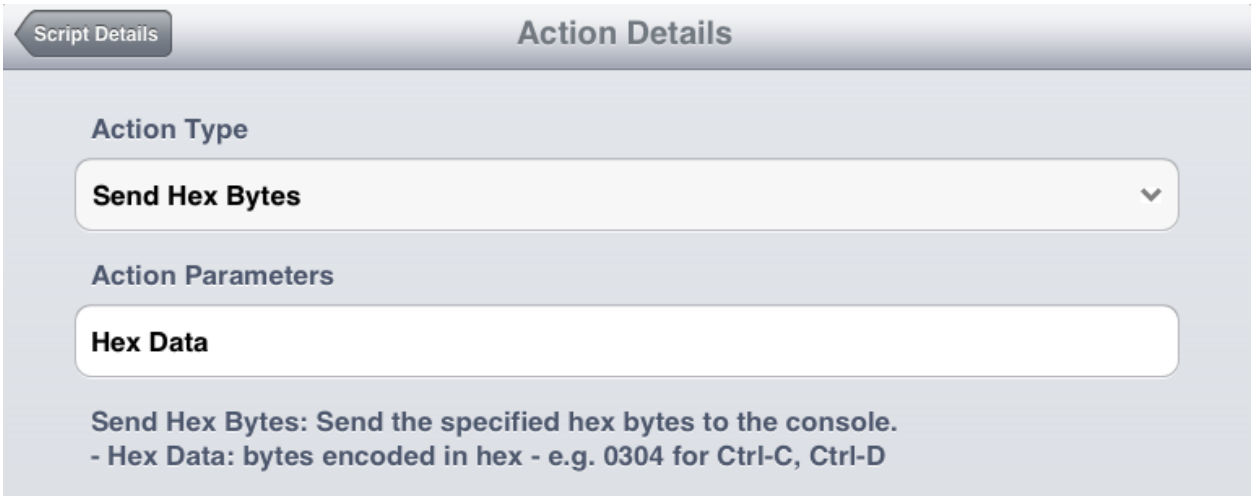


Use the send string to send normal ASCII text to the terminal window, with or without the enter key after it. If escape codes (ie CTRL-X) are needed to be sent, instead use the alternative “Send Hex Bytes” Action item.

3.2.4 Terminate Script Action Item

By default when the script completes its last action item it will terminate by itself, therefore this action is only used where jumps between rules are configured (see Wait-for-String above) and one leg of a script branch requires the script itself to terminate.

3.2.5 Send Hex Bytes Action Item

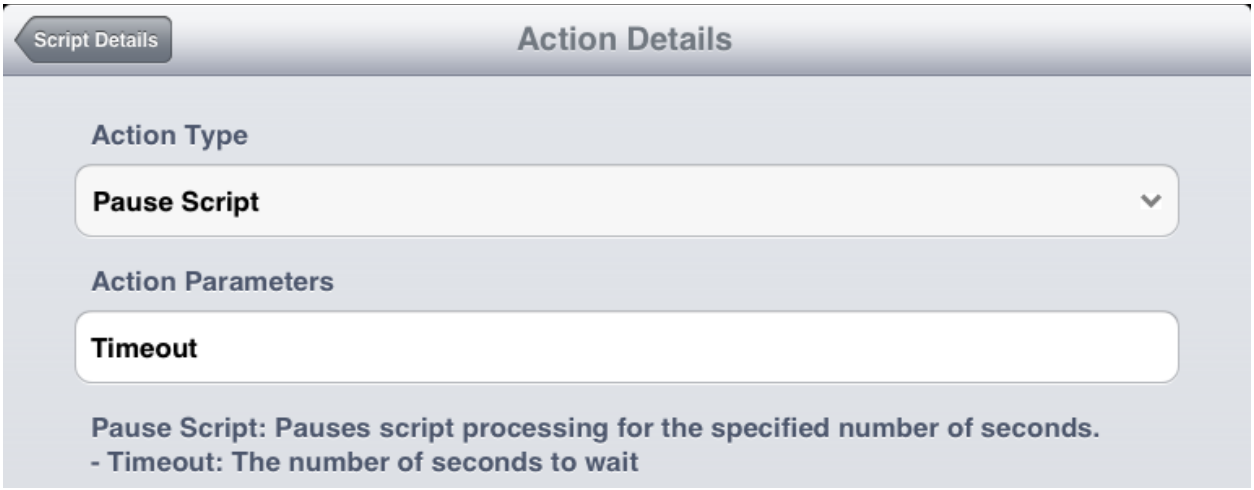


Other common hex byte codes are:

Hex Code	Sends
0D	Carriage Return
09	Tab
08	Backspace

Find more hex byte codes at www.asciitable.com

3.2.6 Pause Script Action Item



Use the Pause action to halt the script for a number of seconds – this is useful when the device connected to will not be ready to accept a “Send String” or “Send Hex Bytes” action immediately after a “Wait for String” has been matched. The pause action is also useful to give the remote device time to process the previous “Send String” before sending the next one.

3.2.7 Display Message Action Item

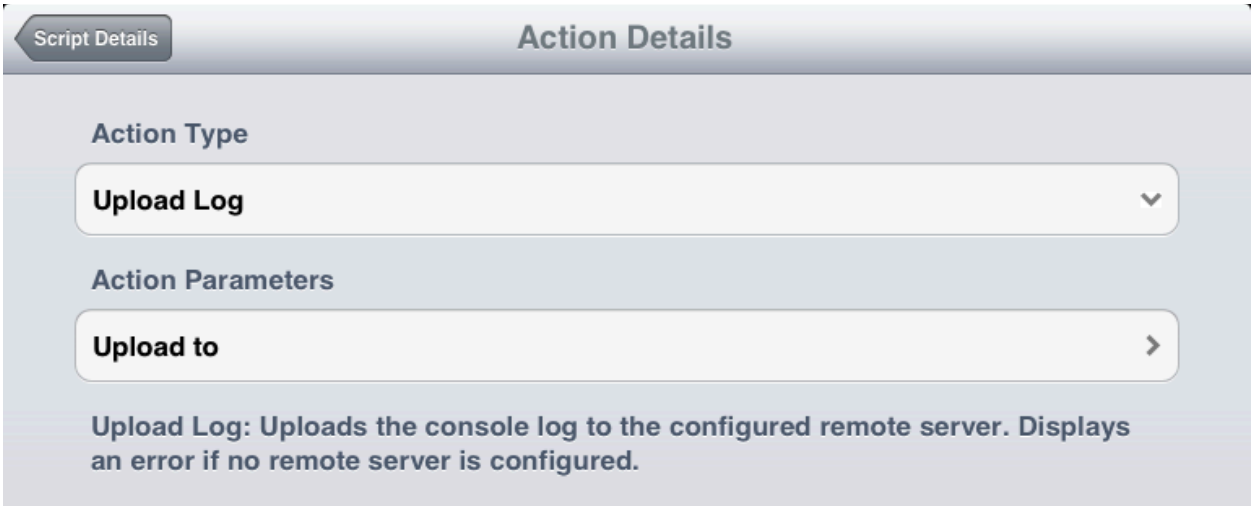
Display Message is useful to provide feedback to the iPad/iPhone user that the script has completed or is at a certain point. The pop-up message does not have to be acknowledged for the script to continue or complete.

3.2.8 Start and Stop Logging Action Item

Note that these actions override the default Terminal settings for Logging. If Logging is already enabled in the main App settings, there is no need to “Start Logging”, unless it had been previously stopped with the “Stop Logging” Action. These actions are useful if just a subset of terminal data is required to be

captured. The parameters allow the log file to be appended to the existing session log file rather than a new one be created.

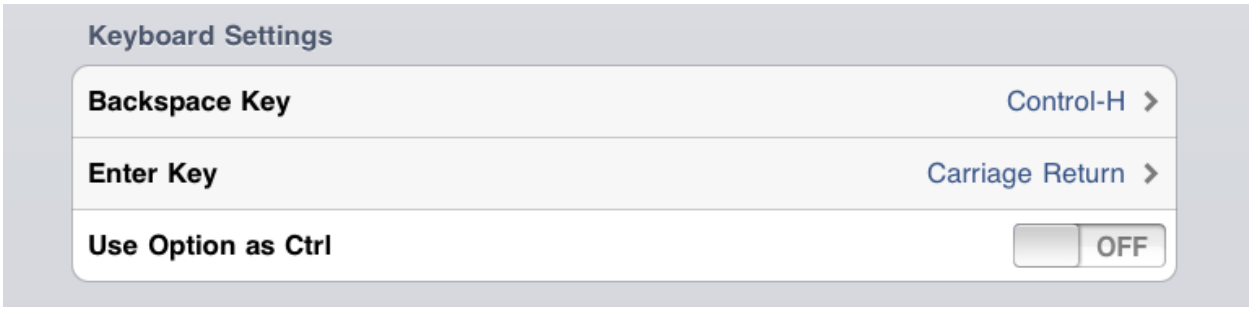
3.2.9 Upload Log Action Item



Upload log file created during the session (either because Logging is enabled by default, or due to Start Logging earlier action item). The log file can be either uploaded to the users portal space on the www.get-console.com website (Get Console option), or if the user has linked Get Console to their Dropbox.com account then uploaded to the My Apps/Get Console/ folder within their Dropbox.

3.3 Keyboard Settings

There are 3 configuration items under the Keyboard Settings section of the main App Settings.



Note that it is possible to use get-console app with a Bluetooth keyboard and the settings configured here will affect the Bluetooth keyboard. Note also that full Bluetooth Keyboard support is not yet implemented as at version 1.80 – ESC and Arrow keys will not work on Bluetooth keyboards.

3.3.1 Backspace Key

Certain terminals expect Backspace to be sent as CTRL-? (Delete – ASCII 0x7F) rather than the Backspace (CTRL-H – ASCII 0x08). The default setting is CTRL-H.

Options:
- **Control-H**
- **Control-? (127)**

3.3.2 Enter Key

Get Console has 3 options for what is sent when the enter key is pressed. By default it sends just Carriage Return (0x0D) which is the OSX standard. This can be changed to Line Feed (0x0A) or both Carriage Return followed by Line Feed (0x0D0A) each time the enter key is pressed.

Options:
- **Carriage Return**
- **Line Feed**
- **CR + LF**

3.3.3 Use Option as Ctrl

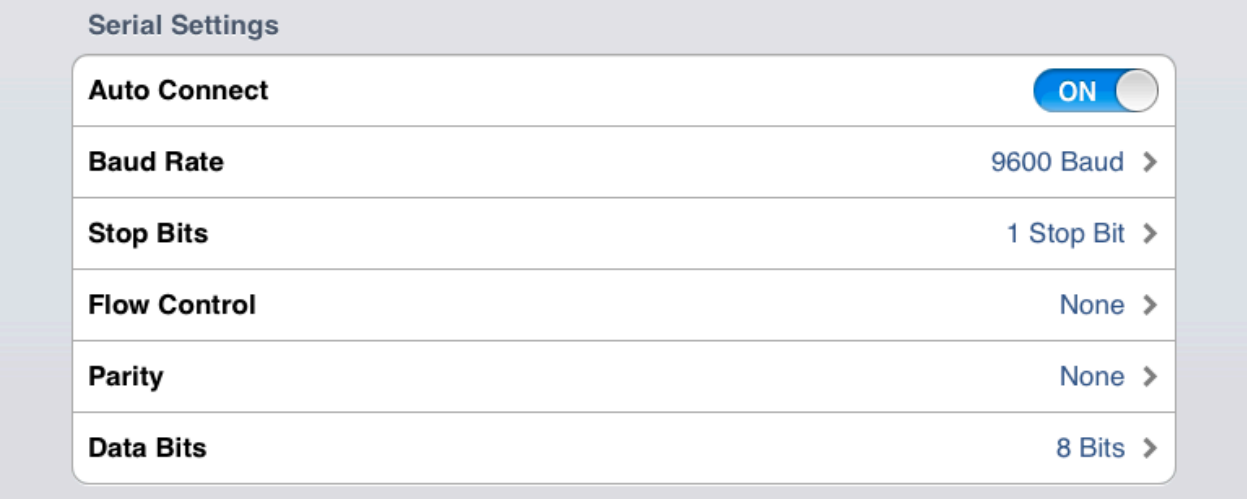
Get Console implements a workaround for the control key not working on Bluetooth keyboards. If using a Bluetooth keyboard with Get Console, you can select the “Use Option and Ctrl” to use the Alt/Option key on the keyboard to send CTRL-[key] sequences. The default setting is OFF.

Options:
- **On**
- **Off**

3.4 Serial Settings

Get Console’s Serial Settings allow for the full range of physical serial settings available on the Redpark C2-RJ45 adaptor to be set. Currently Get Console supports the Redpark C2-RJ45 and (from June 2012) the C2-RJ45V serial console cables. When the C2-RJ45 cable is adapted to either DB9 or DB25 presentation, only the pins available on RJ45 connector can be carried through – ie Pin 9 on DB9 connector (Ring Indicator) cannot be carried.

Below describes the parameters available for each Serial option:



3.4.1 Auto Connect

Get-console app automatically connects to the physical cable connection (RJ-45 or serial) when it's plugged into the iPhone/iPad/iPod port. This applies whether or not there is an existing Telnet or SSH connection (a new Serial connection will be added to your connection list).

Options:
- **On**
- **Off**

3.4.2 Baud Rate

Allows the choice of different baud rates to suit communication with the network device. Failure to select the correct Baud rate for your serial device will result in either no output or garbled screen output. Support for 115200 baud will be possible with the next Redpark C2-RJ45V cable.

Options:
- **1200 Baud**
- **2400 Baud**
- **4800 Baud**
- **9600 Baud**
- **19200 Baud**
- **38400 Baud**
- **57600 Baud**

3.4.3 Stop Bits (Default: 1 Stop Bit)

Options:
- **1 Stop Bit**
- **2 Stop Bits**

3.4.4 Flow Control (Default: None)

Use flow control where your serial device requires it. Examples of devices that require Hardware Flow control to be enabled include older Cisco 3500XL/2900XL series. If the Flow control setting is changed the Redpark C2-RJ45 cable must be removed and reinserted to reset its configuration.

Options:
- **None**
- **Hardware (RTS/CTS)**
- **Software (XON/XOFF)**

3.4.5 Parity (Default: Off)

Options:
- **Off**
- **Odd**
- **Even**

3.4.6 Databits (Default: 8 Bit)

Set Databits required by your serial device.

Options:
- **8 Bits**
- **7 Bits**

3.5 SSH Settings

Get Console supports SSH version 2 only. SSH version 1 support will be added in a future release.

The SSH Settings section is used to determine the default authentication method when password authentication is used, and also allows for the import of OpenSSH format certificates for use in SSH connections that use certificate based authentication.

3.5.1 Keyboard Auth (SSH)

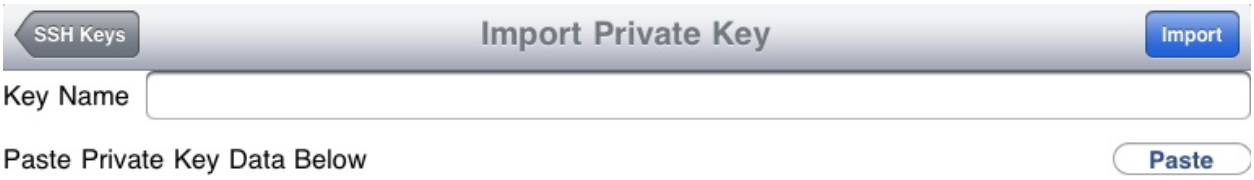
Allows for interactive keyboard password authentication in addition to the usual SSL security. Use Keyboard Interactive method to SSH into servers that require this method – for example to SSH to Mac OSX Lion Server requires “Keyboard Interactive” method. Try alternating the method if you can launch and to connect to an SSH server but no login or password: prompt appears. The default method is OFF.

Options:
- **On**
- **Off**

3.5.2 Private Keys

Allows for private keys to be imported and used in console sessions

Options:
- **Existing Key**
- **Import Key**



3.5.2.1 Import Key:

- Key Name:

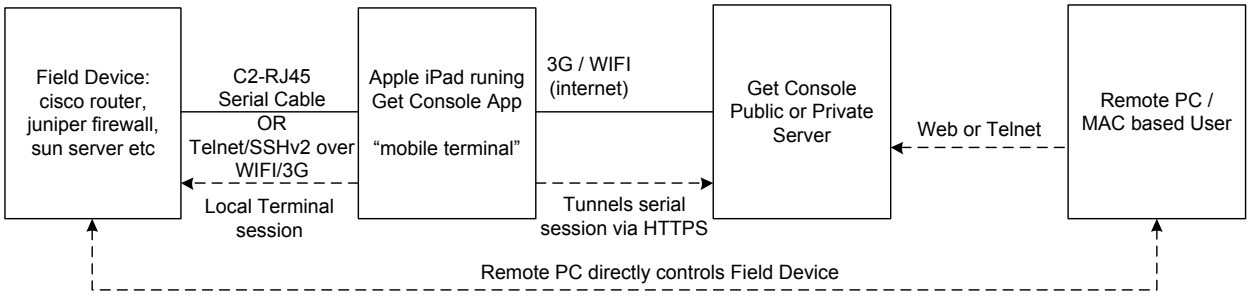
Define a key name for identification purpose when assigning to SSH sessions.

- Paste Private Key Data Below:

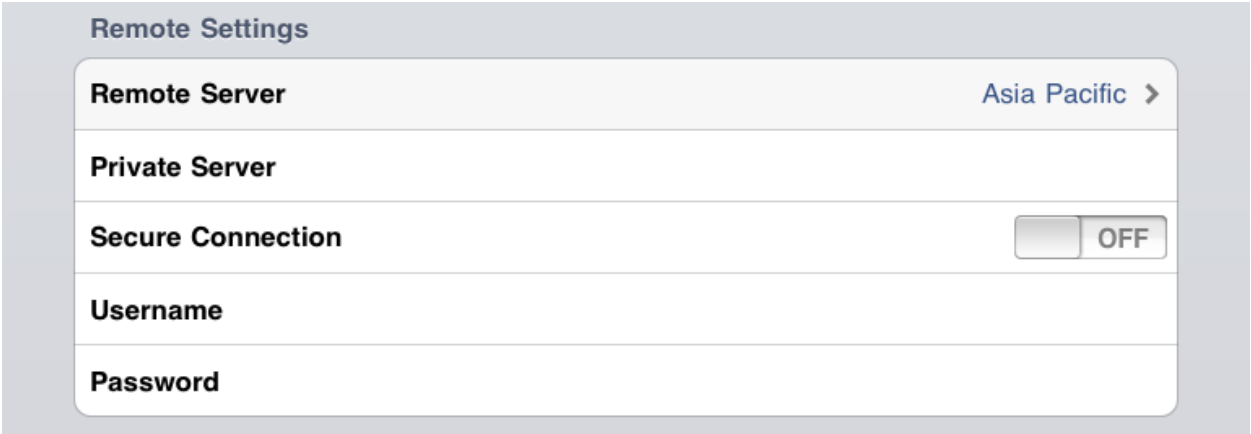
Paste the OpenSSH formatted key data into this window then press “Import” button. If successful a dialog box will appear confirming import and the key will be available to use / assign to saved SSH connections.

3.6 Sharing Settings

Get Console allows you to share your terminal window(s) with remote web users. Each concurrent terminal session on the iOS device can be shared independently with the get-console.com website, and made accessible via individual one-time token codes. The below drawing provides a simple overview of how the session sharing feature works:



In order to enable session sharing, the "Sharing Sessions" details must be populated in the App settings.



3.6.1 Remote Server

Select the closest Public server to the iPad/iPhone user for session sharing with a remote user, or if Private Server has been installed then select Private Server.

Public versions of the Server are hosted in US, UK and New Zealand. These can be used at no charge by any registered purchasers of the Get Console App to allow remote web access the terminal session via the public website www.get-console.com. The performance of the end to end terminal varies greatly on the current load of the server and the latency of both the Apple device and the Remote web users from the selected public server.

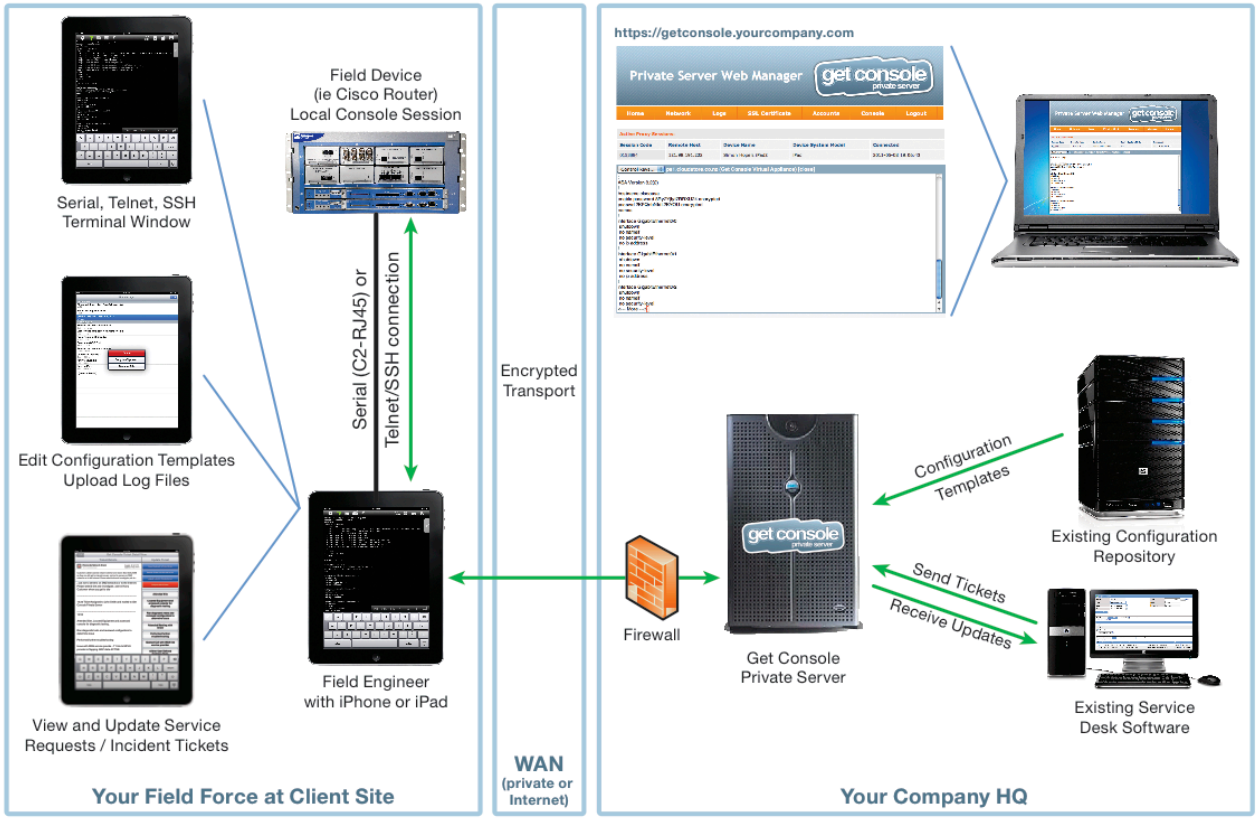
Options:	
- **Asia Pacific**	
- **Europe**	
- **North America**	
- **Private Server**	The Private Server is hosted on a customer's own network, and secured by their own network security policy
- **Disabled**	The option to share remote session will be disabled

3.6.2 Private Server

Private Server is an optional add on to Get Console that allows for end users to host their own Get Console server on their own network and infrastructure.

If Private Server is selected for Remote Server, set its hostname or IP address in this field.

Below is diagram representation of the working of the Private Server



For more information, please view the Get Console Private Server manual available from www.get-console.com/private-server

3.6.3 Secure Connection (Default: Off)

Use SSL for session sharing for more secure connection. SSL option should only be selected with Private Server if the Private Server has a valid and publicly trusted (ie trusted natively by iPhone / iPad) SSL certificate for the entered hostname.

Enabling Secure Connection has a small impact on the remote performance. For maximum responsiveness when encryption is not required leave Secure Connection OFF.

Options:

- **On**
- **Off**

3.6.4 Username

Registered email address on www.get-console.com – it's free to register. Or if Private Server has been chosen, the user name for use with the Private Server login

3.6.5 Password

Password for your registered account on www.get-console.com website. Or if Private Server has been chosen, the password for use with the Private Server login.

3.7 About Get Console

3.7.1 Version

Version number of the Get Console App. Certain features are only available in later versions. Updates to Get Console occur automatically via the App Update process in the iTunes App Store.

3.7.2 Copyright

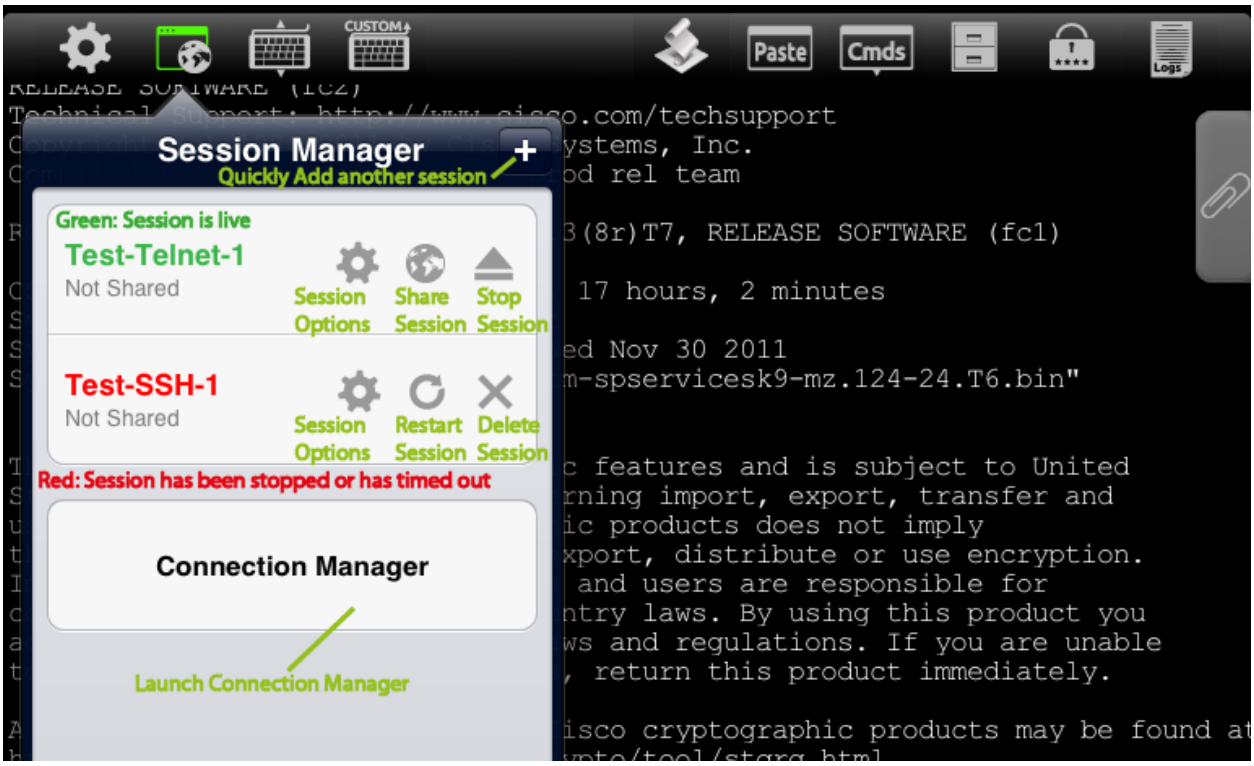
Year and holder of the copyright. Get Console is a registered trademark owned by Cloudstore Limited.

3.7.3 About Get Console

About Get Console app – Displays version, copyright information, developers, LibSSH2 license and any other licenses for third party software that is incorporated into Get Console.

4 Launching Connections (Session Manager)

Pressing the Connection button  will launch the new connection Session Manager popup (iPad) or full screen Session Manager (iPhone). Get Console provides 2 ways to launch connections – first via the plus button for a Quick connection, or via the Connection Manager to launch a saved or recent (unsaved) quick connection. The below drawing highlights the main features of the Session Manager.



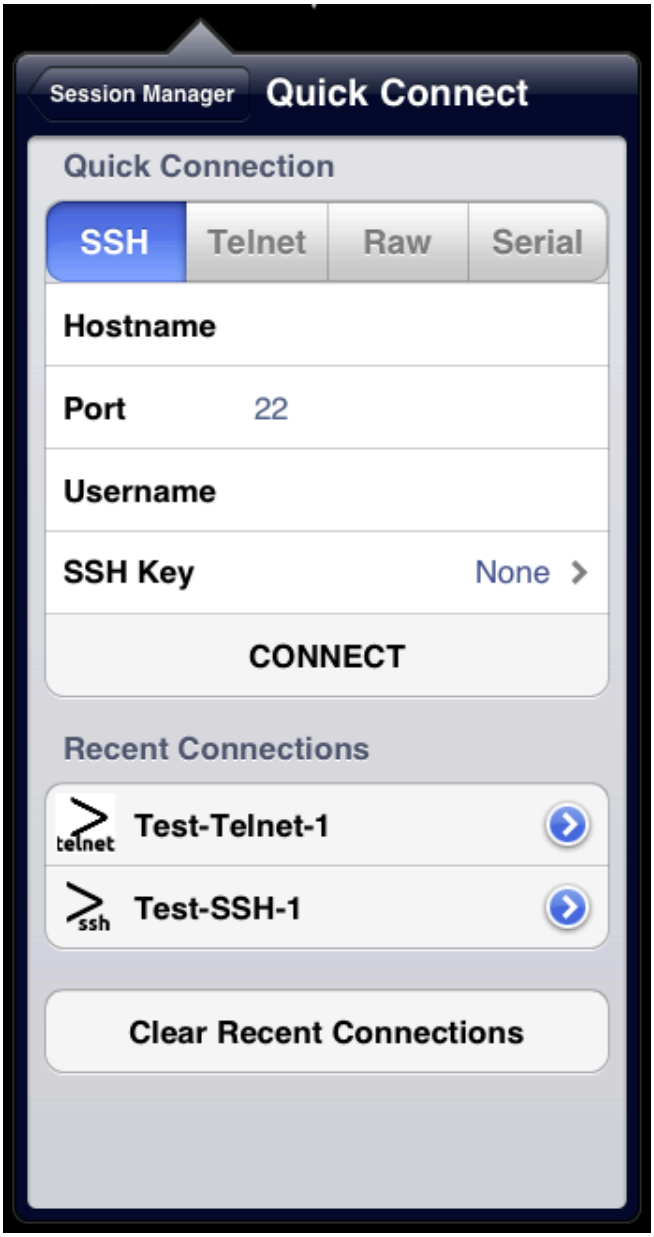
As can be seen, the Session Manager popup shows all active and also recently closed sessions that can be restarted. Within each session the options (ie like scrollback length, window size etc) can be varied from the Main Settings defaults via the session options button. Any live session can be shared a remote user via the “world” button and the session can be stopped via the “eject” button.

If a session is stopped or times out, then it still remains visible in the Session Manager until it is deleted via the X button or is restarted via the restart button.

There are hardware limits to the number of concurrent sessions that can be run in Get Console, depending on the Apple iOS device in use. The below table provides these limits:

Apple Hardware	**Number of Concurrent Live Sessions Allowed**
iPhone 3G	3
iPhone 3GS	5
iPhone 4	6
iPhone 4S	8
iPad 1	3
iPad 2	5
iPad 3	8
iPod Touch 3rd Gen	3
iPod Touch 4th Gen	5

4.1 Quick Connect



Pressing the "+" button on the Session Manager launches the Quick Connect popup. This dialog allows for the instant launching of a new Telnet, SSH, Raw or Serial connection in addition to launching an automatically saved recent connection (Either Quick connected or launched from the Connection Manager).

Depending on the type of connection, various session parameters (such as hostname/IP address) will need to be completed and then tap the Connect button.

Get Console can only maintain a single live Serial connection in the Session Manager as only one adaptor can be connected, however it can store multiple instances of serial connections each with a different session options (ie different baud rates) to make it easy to swap between serial connection templates.

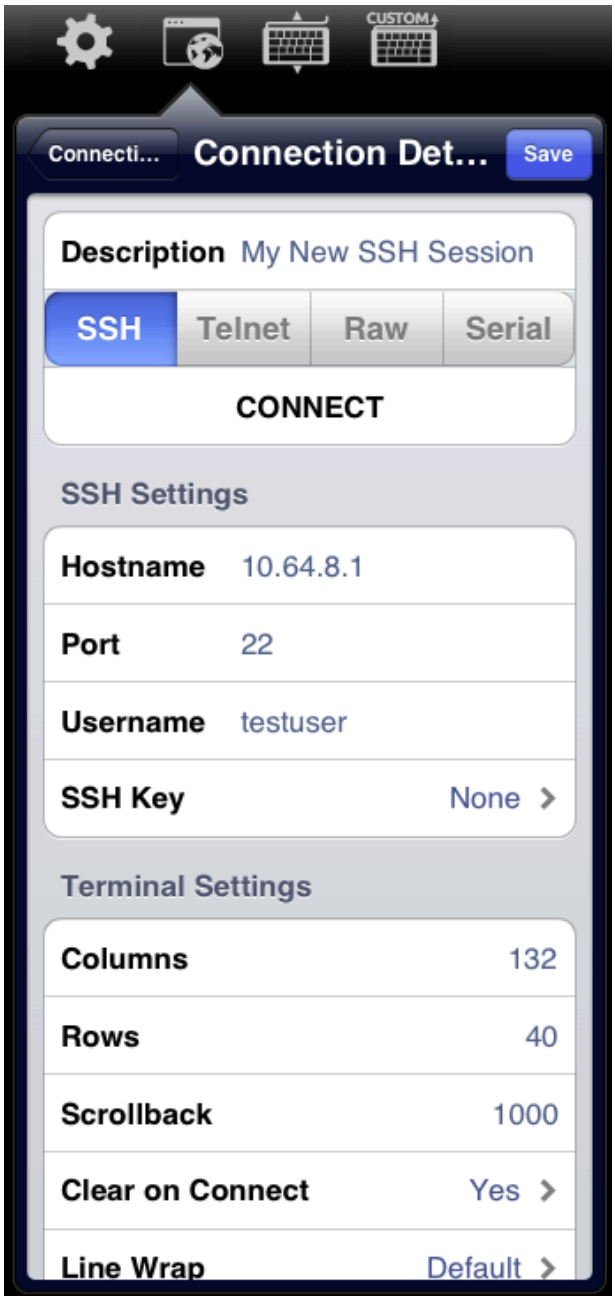
4.2 Connection Manager

The Connection Manager is used to create, modify and launch saved connections. Tapping Connection Manager in the Session Manager popup will launch Connection Manager where all saved connections can be administered.

Creating new connections can be done one of 3 ways

1) Via the Connection Manager “New Connection” dialog box (discussed in this section)
2) By converting a Recent Quick Connection to a saved connection via the blue arrow (Give the recent connection a Description and then tap Save)
3) By importing “.connection” files into Get Console via iTunes. Get Console provides a free tool to convert existing PuTTY and SecureCRT .ini files into Get Console “.connection” files. This tool can be downloaded from www.get-console.com/tools . The method to import the “.connection” files into Get Console via iTunes is documented at www.get-console.com/tools/importutility/

4.2.1 Creating New Connections via Connection Manager



4.2.1.1 SSH

For creating new SSH connections, complete the following items

Description: Define a name for this connection
Hostname: Enter hostname or IP address of the device to connect to via SSH
Port (Default: 22): Enter port number of the device to connect to via SSH
Username: Enter username to login to the device
SSH Key: Choose an SSH key to login to the device if using certificate based authentication. The default is None

Terminal Settings: This section allows for the override of the Main Settings default Terminal Settings to apply to this connection. Change any of the allowed settings.

Click Save when complete

4.2.1.2 Telnet

For creating new Telnet connections, complete the following items

Description: Define a name for this connection
Hostname: Enter hostname or IP address of the device to connect to via Telnet
Port (Default: 23): Enter port number of the device to connect to via Telnet. Note that if a non-standard port is chosen, the Telnet preamble will still be sent. If a connection over a non standard port is required

without the Telnet preamble (ie for when "Telnetting on port 80 to a webserver" to check that it responds

4.2.1.3 Raw

For creating new Telnet RAW connections, complete the following items

Description: Define a name for this connection
Hostname: Enter hostname or IP address of the device to connect to via Telnet
Port (Default: 23): Enter port number of the device to connect to via Telnet. Note even if port 23 is selected the Telnet preamble will NOT be sent. Common use of Telnet RAW is to test-open connections to web servers and mail servers to see if they are responding on their standard port 80 / port 25

4.2.1.4 Serial

For making serial connections via the Redpark C2-RJ45 cable. The following parameters can be configured on a per session basis, other serial settings are inherited from the Main Settings defaults:

Description: Define a name for this connection
Serial Settings: The Baud rate, Stopbits, Flow Control, Parity and Databits can be set on a per connection basis. All other settings are globally configured in the Main Settings page.

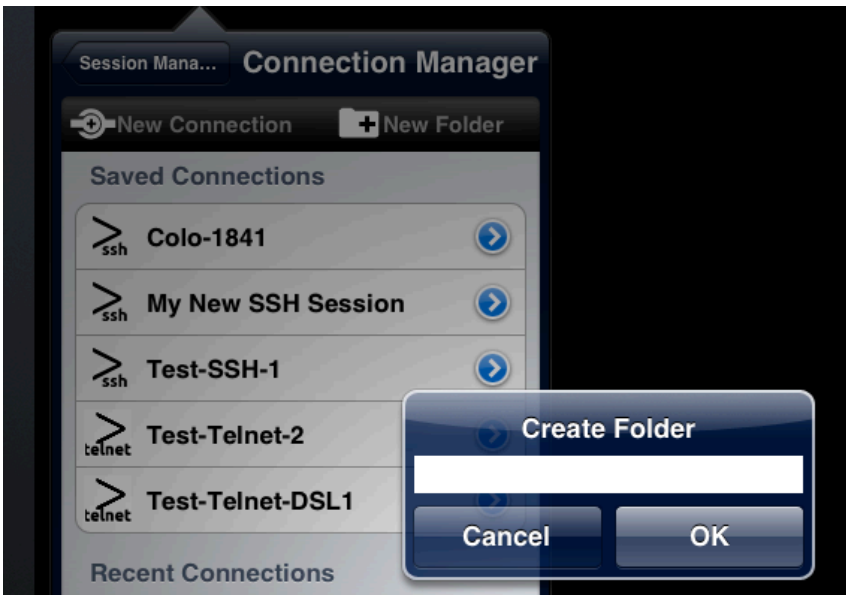
Terminal Settings: As per all other connections the terminal settings defines a non-default Terminal characteristics for just this connection.
Keyboard Settings: As per all other connections, the Keyboard settings can override the default Keyboard settings on a per connection basis.

4.2.2 Managing Connections in Folders

Connection Manager allows for the creation of connections, grouping of connections into subfolders and also for cloning connections and moving them between subfolders. These folders are stored in the Get Console Application's file system area within the iPhone/iPad. They can be exported via iTunes, however to import connections into a specific subfolder is not possible via iTunes. To perform this manipulation

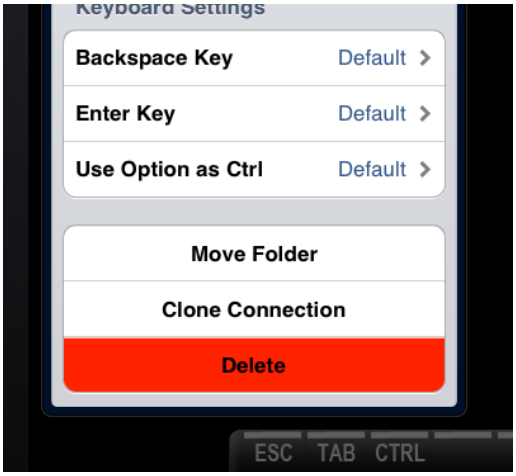
outside of Connection Manager requires a third party application. We have tested one such application called iExplorer which is discussed below.

4.2.2.1 Creating and Managing folders and sub-folders for Connections



Tap "New Folder" to create a new folder and enter name in dialog box. The folder appears at the top of the Saved Connections List. Folders can be nested, so to create a subfolder navigate to the new folder by tapping it then tap the New Folder button again.

Saved connections can be moved from one folder to another or to the root. To move a connection tap the blue arrow next to a saved connection, scroll to the bottom where there are options to Move (to a) Folder, as well as Clone the Connection or Delete.



A sub-folder can also be created under another folder with connections in it. Note that if you delete a folder which contains a connection, the connection will be moved to the root folder instead of being deleted together with the folder.

4.2.2.2 Recent Connections

Get Console Connection Manager presents the 10 most recent connections below the Saved Connections. The connections listed in Recent connections include Quick Connections and Saved Connections. This list can be cleared by tapping the "Clear Recent Connections" button at the very

bottom of the list. Alternatively, a recent connection that was made via “Quick Connect” method can be saved to the Connection Manager via tapping the blue arrow, providing a name (description) for the connection and then tapping save. The connection will then appear in the root Saved Connections folder of the Connection Manager.

4.2.3 iExplorer for Connection Manager

As mentioned in section 4.2 the Connection Manager content can also be populated or edited via your PC/Mac by a free program called iExplorer. Cloudstore makes no assurances about the stability and offers no support for iExplorer, however in our internal testing we have found the current version to be useful for bulk arranging connection files within folders and subfolders outside of the Get Console app.

iExplorer (formerly named “iPhone Explorer) can be downloaded from http://www.macroplant.com/iexplorer/ currently free of charge.

4.2.3.1 iExplorer Use

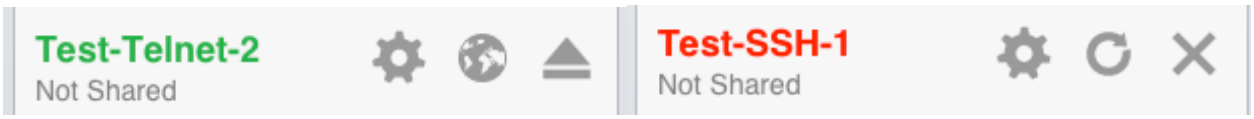
After the iExplorer is installed in your PC/Mac, follow these simple steps to create/edit/delete your own connections:

1. Ensure you have a suitably late version of iTunes installed
2. Connect your iPhone/iPod/iPad to the PC/Mac using its cable
3. Open iExplorer, navigate to the iDevice -> Apps -> com.amixcapital.CiscoConsoleCcompanion/Documents/Connections/[Folder]
    a. **To edit a connection:** Drag and drop any .connection file into your computer to edit its content using a text editor e.g. Notepad, Notepad++, you can then copy the file back to its original location to overwrite it with the new information
    b. **To add new connection(s):** Create a new .connection file in your computer then drag and drop it into iDevice -> Apps -> com.amixcapital.CiscoConsoleCompanion/Documents/Connections/[Folder]
    c. **To delete connection(s):** Delete a .connection file from the iDevice -> Apps -> com.amixcapital.CiscoConsoleCompanion/Documents/Connections/[Folder]

When adding or editing a connection, simply edit the .connection file accordingly using xml protocol.

4.3 In Session Options

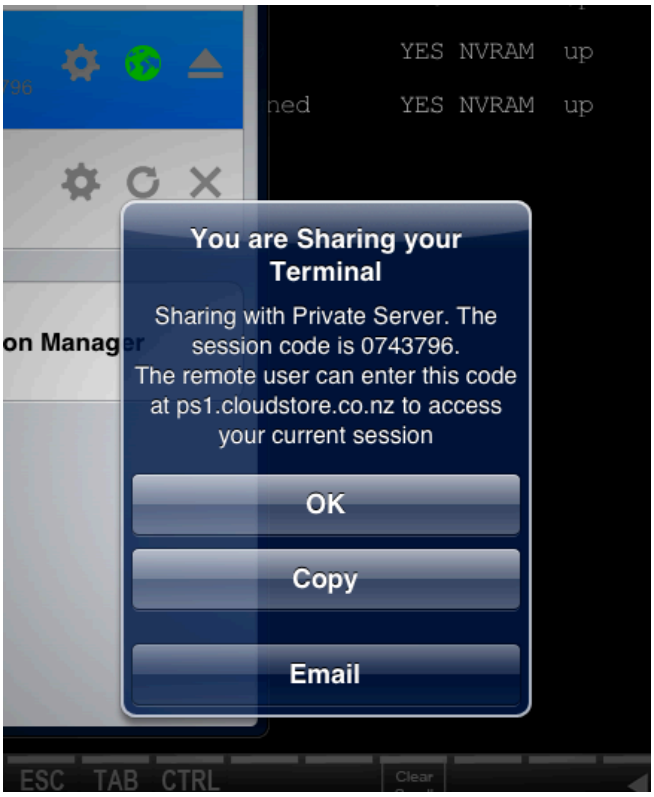
Once a connection has been established (or attempted to be established) it will appear in the Session Manager. From this popup the actions that can be performed depends on whether the connection is live (connected - green) or closed (disconnected - red)



The following table describes the actions available.

	Session Options: Allows for editing of changeable settings while the session is live, for example changing the terminal width, scrollback, colour scheme etc)
	Share Session: Will share the terminal screen with the configured remote server (Public or Private), and generate a one-time token code. See the session sharing Settings (section 3.6 above) and how to operation below.
	Disconnect Session: Disconnect from the remote server (Telnet/SSH/Raw) or disconnect the serial cable (Serial). The session will stay visible in session manager list of sessions with the 2 below options available.
	Restart Disconnected Session: Attempts to reconnect the session to the remote server (Telnet/SSH/Raw) or cable (Serial).
	Remove Disconnected Session from Session Manager

4.4 Session Sharing



Tapping the  button will attempt to share the selected terminal session with the Get Console website or configured Private Server so that a remote web user can see and interact with the terminal session at the same time. In order to use this feature the settings for session sharing must be configured in the main settings page (see section 3.6 above).

Each shared terminal session is dynamically given a one-time token code at the time session sharing is initiated. This code is used to secure access to the users iPad/iPhone. The remote web user must know this 7-digit token code in order to see the iPad/iPhones terminal window.

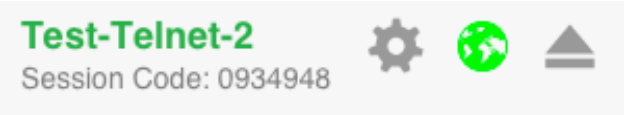
Only sessions that are shared are visible to remote users, however multiple concurrent sessions can be shared – each with their own unique one-time token code.

4.4.1 Start Session Sharing

Once the notification “You are Sharing your Terminal” appears, the remote user can access the iPad via the token code displayed in the notification. To make it easy for the remote user to learn the code it can be emailed or copied to clipboard via this notification window.

	Options	Function
-	**OK**	Return to the active terminal window
-	**Copy**	Copy the remote access token code to the clipboard so that it can be used in txt message or other iOS device message
-	**Email**	Email the remote access token code to a recipient with instructions how to connect to the shared session

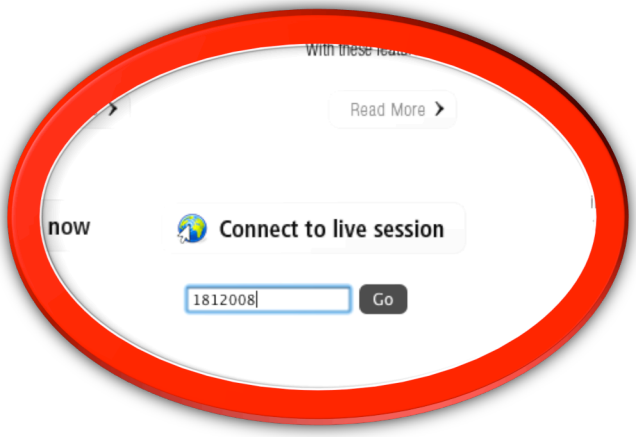
Once shared, the token code will also appear in the Session Manager, and the globe icon will show green



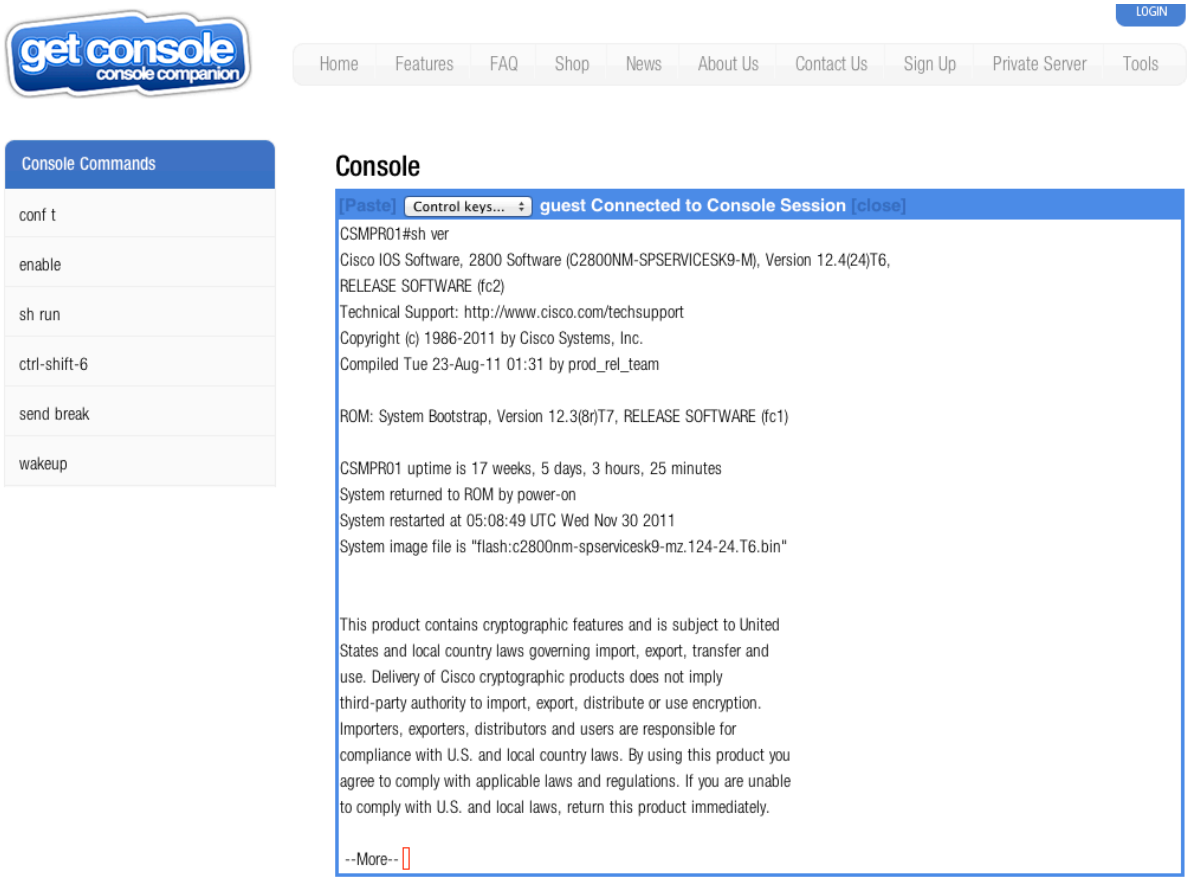
4.4.2 Remote User Access

Remote access to shared terminal sessions is dependent on whether they have been shared with the Get-Console.com public servers or via the users own privately hosted server (Private Server). For the purpose of this manual it is assumed shared with the Get-Console.com public servers. To learn more about Get Console Private Server visit www.get-console.com/private-server

When the remote user has obtained the one-time token code, they enter it on the home page at www.get-console.com, or if the token code was shared with the remote user via email there is a single link to click in the email to directly connect.



After entering the token code, the code is checked against currently active and available codes, and if correct the Get Console web terminal will launch. The remote user is presented with a plugin free web terminal that mirrors what is on the screen of the Get Console app running on the iPad/iPhone.



Both iPad/iPhone user and remote user can interact with the terminal session at the same time, however only one remote user can access any given session. If a later remote user enters the same token code, their later session will take over from any existing remote user.

Note that the remote user can only interact with the terminal while the Get Console app is foreground on the iPad/iPhone. If Get Console is in the background, then it will still maintain its Serial/Telnet/SSH session for upto 10 minutes, however the remote user will not be able to interact with the session until Get Console is brought to the iPad/iPhone foreground again.

4.4.3 Stop Sharing Session

To stop sharing an iPad/iPhone terminal session with a remote user, tap the  button to activate the Session Manager popover, then tap the  button for the terminal tab that sharing should be disconnected. The world icon will turn grey and the subtitle will change from the token code to say “Not Shared”.

5 Terminal Features

This section discusses the following Get Console terminal features:

- Keyboard popup bars
- Command Shortcuts
- Password Shortcuts
- Clipboard Viewer

5.1 Terminal Features

5.1.1 Keyboard Control

Use the  button to show and hide the onscreen iPad keyboard. Used to increase screen real estate, especially useful for debugging in the iPhone/iPod with its smaller screen. Note that when a Bluetooth keyboard is connected the

5.1.2 Keyboard Popup Bar Selector

Pressing the 

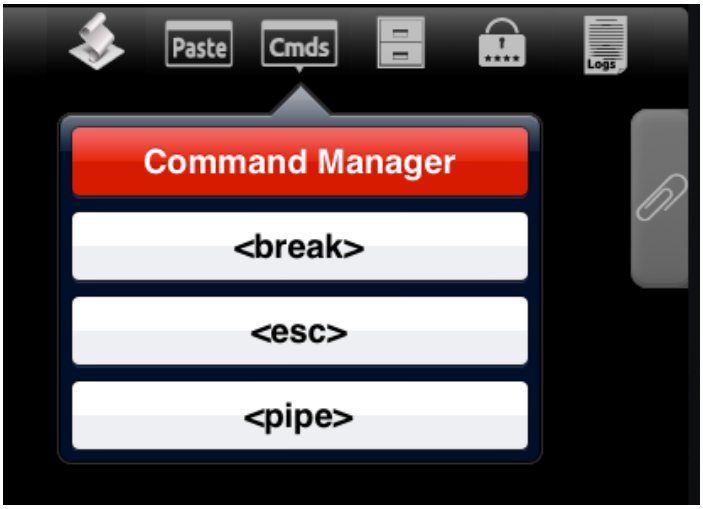
button changes or hides semi translucent popup bar of shortcuts that sits above the keyboard. Hiding the bar can be useful when trying to read the line at the bottom of the screen. Pressing this button cycles through some preconfigured popup bar options.

Options:
- **CISCO – keys useful when working on Cisco routers and switches**
- **SUN – keys useful when working on Sun Servers**
- **F1-F12 – Function Keys**
- **Custom – user selectable keys**
- **None – Popup bar disappears allowing full visibility of the bottom line of the terminal**

For the Default “Custom” option, the user can create their own combination of buttons on the popup bar by pressing and holding each individual button. A button selection popup will allow these keys to be changed to one of many options. Click on the top of an empty bar to assign a key to it, or click & hold on the top of the assigned bar to change the assigned key.

5.1.3 Command Manager

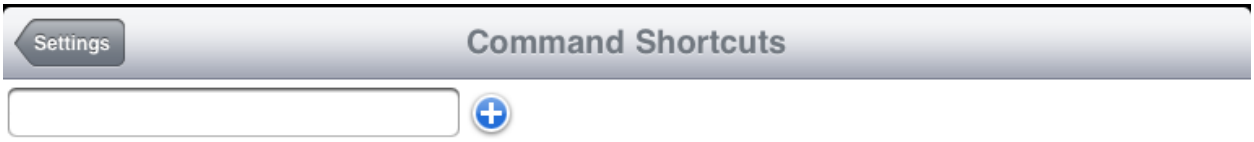
Command shortcuts are added in the Command Manager. Once entered they are available with one tap via the “Cmds” popup/dropdown from the top navigation bar. Each command is sent with a carriage return after it.



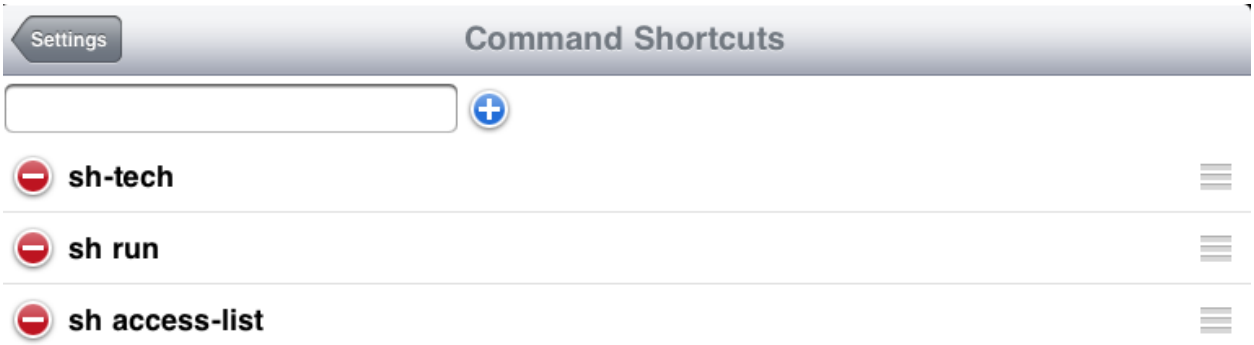
Tapping CMDS will show the list of current command shortcuts. A few shortcuts have already been defined by default and user can add their own custom command shortcuts for use in the terminal

Defaults:
- **Command Manager (Launches Command Manager page for entering more quick commands)**
- **<break> - sends terminal break sequence**
- **<esc> - sends ESC sequence for selected encoding method**
- **<pipe> - sends the | symbol WITHOUT a Carriage Return**

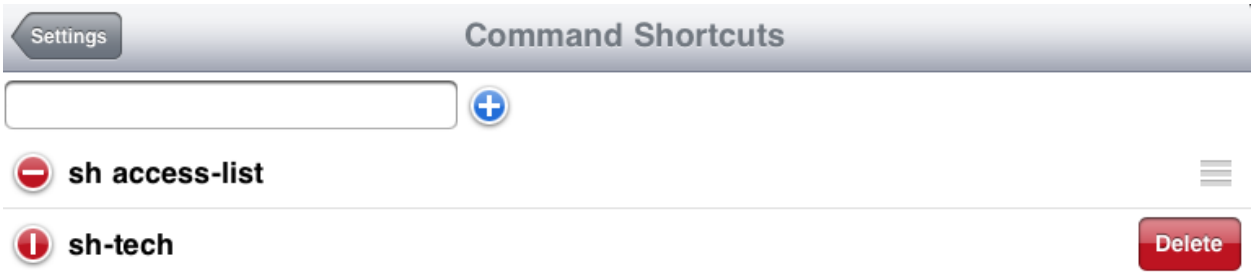
Adds command shortcuts here for use in terminal via dropdown icon



Rows on the right: Hold button to rearrange the order of the command shortcuts as they appear in the dropdown popover.



Stop sign on the left: Changes the rows on the right to delete button which then can delete the saved command shortcut



5.1.4 Password Manager

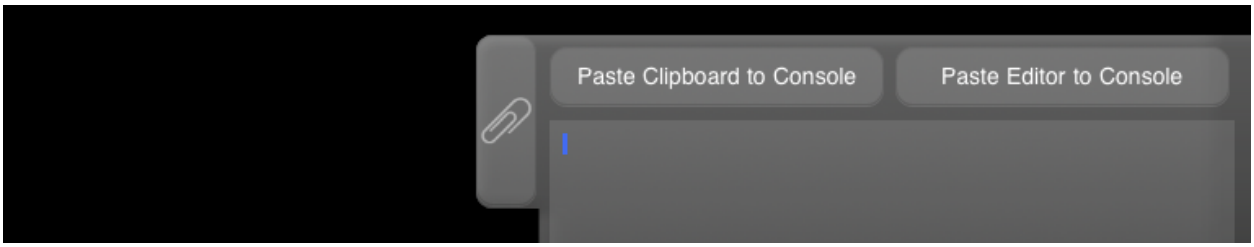
Like the Command Manager, Allows user to save passwords to be used as shortcut instead of typing the entire password all over again

Add passwords here for quick use in terminal via dropdown icon

Saved passwords are recalled in terminal sessions by tapping the  button. While the passwords as entered are visible in the Password Manager, from the terminal window the middle characters are obfuscated when recalled.

5.1.5 Clipboard Viewer

The iPad version of Get Console has a slide out clipboard window accessed via the paperclip icon. This allows the user to see and edit the contents of the iPad's clipboard and then paste



The iPhone version of Get Console has a whole page clipboard viewer with the same buttons for pasting to the terminal window.

5.1.5.1 Paste Clipboard to Console

Pastes the content of the clipboard to the console. If the user has made changes to the clipboard viewer text, but has not selected and copied the clipboard viewer to the internal clipboard, then this button will paste the clipboard viewer contents *prior* to any edits.

5.1.5.2 Paste Editor to Console

Pastes whatever is in the editor (the box containing the content of the clipboard) to the console. This allows the user to modify the content of the clipboard first before pasting to the console. Effectively what this button does is copy the edited view to the clipboard prior to then pasting to the terminal window.

Once the editor is closed, the content of the clipboard will revert to the original clipboard again unless the edited "Editor" is copied which then replaces the clipboard content.

6 Get Console File System

6.1 Get Console File Types

Get Console stores files within a portion of the iPad/iPhones file system that is dedicated to the Get Console App. There are 5 types of files that are stored:

- User created text based files that are typically used in a terminal session via cutting/pasting into the terminal window via the Clipboard Viewer.
- Log files, which are Get Console generated logs for each terminal session that has logging enabled.
- RSA Keys for use in certificate based authentication in SSH. These files are imported via the Main Settings -> Private Key dialog box.
- ".connection" files which are saved connections visible in the Connection Manager. These files are created either in the Get Console Connection Manager, or imported via iTunes.
- ".script" files which are saved scripts that can be executed on login or during a terminal session via the Script Manager. These files are created either in the Get Console Script Manager or imported via iTunes

6.2 User Created Text Files

While there are 5 types of files, the File Manager  is concerned only with the first type – User Created text based files. Other types of files are created / viewed / deleted / uploaded via their respective managers as discussed below.

User Created text files are typically used in terminal sessions – they could be snippets of configurations or other templates that are edited in the clipboard viewer before being pasted into the Terminal window.

To be used by Get Console, the files must be local to the iPad/iPhone device. Get Console offers 2 ways to get User Created text files into the local file system:

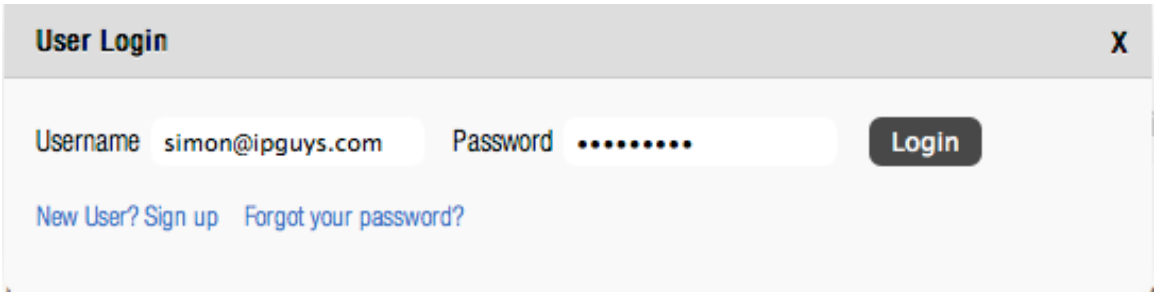
- via import from the users portal on the www.get-console.com website

- via import from the users /My Apps/Get Console/ folder in their Dropbox.com cloud storage

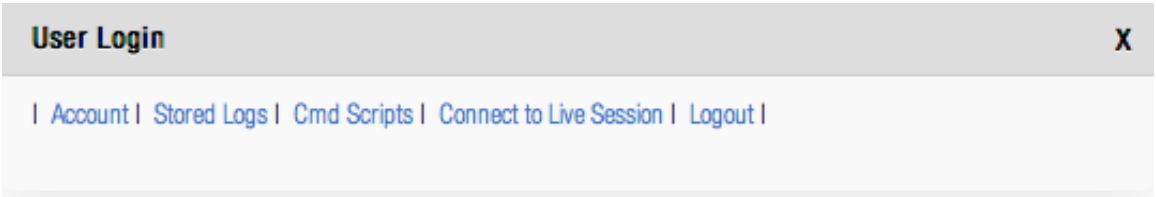
6.2.1 Importing files from Get-Console.com

To import text files from the Get-Console.com portal the user must first have signed up to a free account at www.get-console.com/signup . After sign up, the user logs in via the Login panel at the top of any webpage.

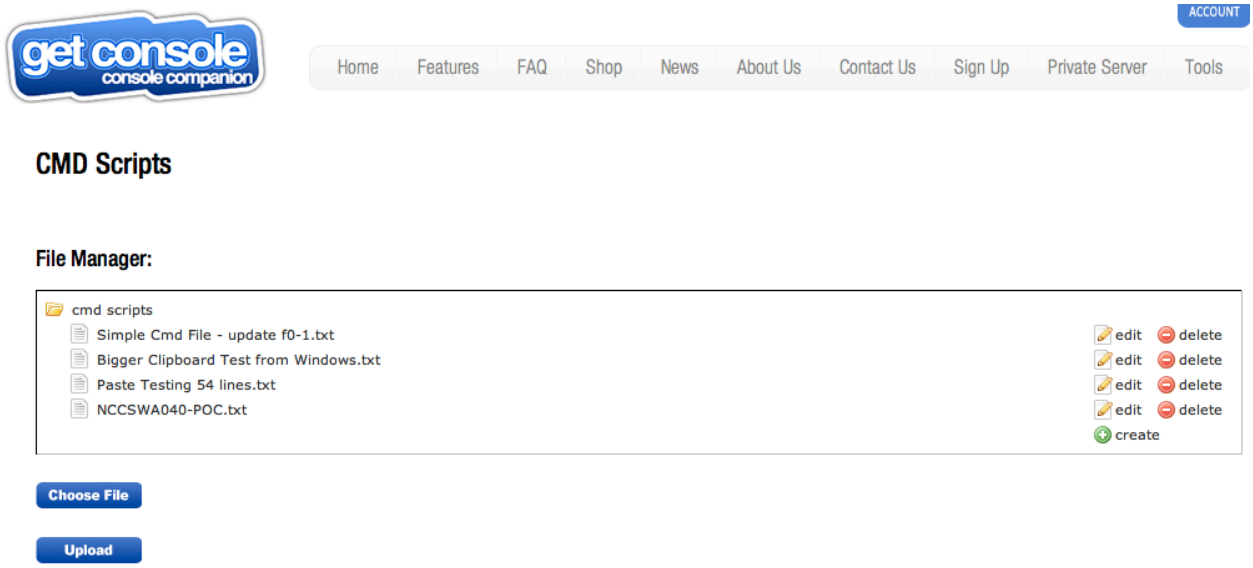
6.2.1.1 Creating / Uploading files on Get Console Website



Once signed in the website “Login” button changes to an “Account” button. Clicking on this reveals the users portal options:



User Created Text files are accessed via the “Cmd Scripts” button. This button will be renamed “User Created Text Files” in a later release of the website.

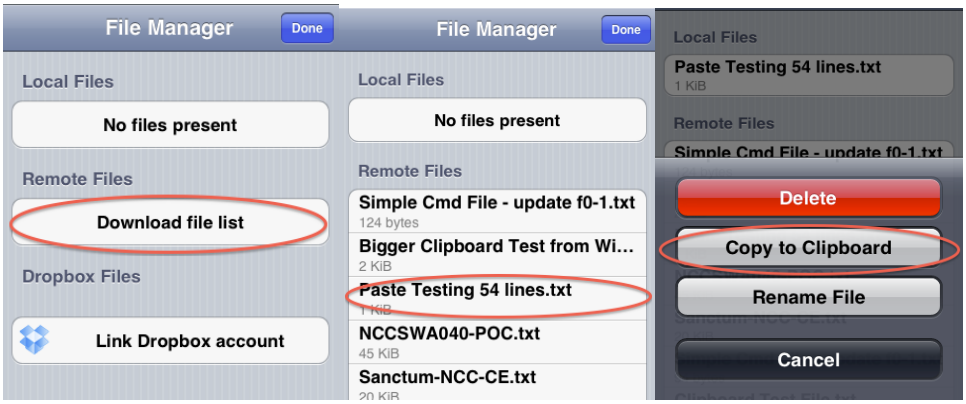


From the CMD Scripts user portal page, the user can upload, edit or create text files that will be visible to the iPad/iPhone Get Console application under the

6.2.1.2 Downloading Files from Get Console Website

Once files are uploaded to the users Get-Console.com portal page, they are visible via the File Manager by tapping the "Download File List" button. The users get-console.com credentials (email / password) must be entered into the Main Settings -> Session Sharing field as described in section 3.6 above in order for the remote files to be visible.

The list of files is presented to the user. Tapping on one of the remote files provides option to download to the local store. Once in the local store tapping the file name again allows for the file to be copied to the clipboard viewer for editing/pasting, renaming or deleting.



6.2.2 Importing files from Dropbox account

To import text files from a dedicated Get Console folder within the users Dropbox.com account, the user must first link their Dropbox account with the Get Console app. After successfully linking, the File Manager will be able to see User Created text files stored in the users Dropbox/Apps/GetConsole folder, and download them to the iPad/iPhones local storage.

Once in local storage the same operations are permitted as described for Get Console portal files are possible (Copy to Clipboard, Rename or Delete).

6.2.2.1 Linking Dropbox Account to Get Console

To link to Dropbox, from File Manager, tap the Link to Dropbox account button. This will start the mobile safari browser where the user will enter their dropbox.com credentials to login, or alternatively, if they already have the Dropbox iOS application installed on their iPad/iPhone it will present a simple allow/disallow dialog box. Tap Allow, after which control will return to Get Console. From Get Console File Manager, it will now be possible to download Dropbox folder contents to the local storage area.

Note that once linked, the user can unlink Get Console from their Dropbox account either within the File Manager or via the Dropbox website. Only a single folder is linked to the users Dropbox repository (not including any subfolders) therefore any files the user wants visible to the File Manager must be in this folder to be accessible.

6.3 Log Files

Get Console offers comprehensive logging of individual terminal sessions. Once logging is enabled in the main settings, all printable screen output is captured to a logfile.

One log file is created for each session. Unless a script is being used to append a log file, a new log file is created each time a session is stopped and restarted.

On iPad Logs are accessed via the  button, whereas on iPhone Logs are accessed via  and then “Logging”

6.3.1 Log File Naming

For connections that are started with the “Quick Connect” method the log file name will be:

Log_YYYY-MM-DD_HHMMSS.txt

For connections that are started from a saved connection in the Connection Manager, the file name will instead be appended with the connection name followed by date: ie

Server1_YYYY-MM-DD_HHMMSS.txt

Invoking log operations from a script can change the name of the log files. See the Script Manager section above for details on log operations that are possible via scripting.

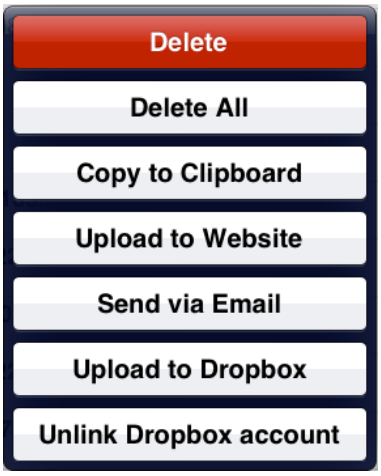
6.3.2 Uploading Log Files

Log files can be extracted from the iOS device via 2 possible methods:

- Via upload to the users portal page on the get-console.com website
- Via upload to the users Dropbox folder dedicated to Get Console (default is Apps/Get Console)

- Via email of individual log files from the iPad/iPhone mail client

Navigate to the Log Files page, then tap a log file name. Select the one of the 3 upload options. Note to upload to Dropbox, the users Dropbox account must be linked to the Get Console app as described in section 6.2.2.1 above.



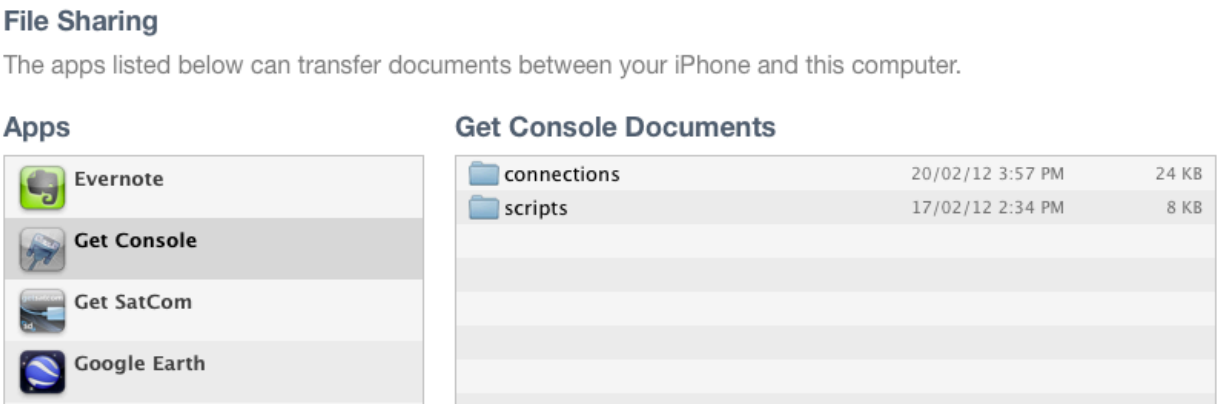
Upload to Get Console website user portal is invoked via the "Upload to Website" button. As per the File Manager, the user must have a (free) Get-Console.com account and the credentials entered as described in section 3.6 above.

Access to uploaded logs on the Get-Console.com website is via the Stored Logs button.

6.4 Other Get Console Files

Get Console stores .connection and .script files for saved connections and saved terminal scripts respectively. These are created and managed in the Connection Manager (see section 4.2) or Script Manager (see section 3.2). These are simple XML files and can be manipulated externally from the Get Console application and then re uploaded.

Existing .connection and .script files can be extracted via iTunes or iPhone Explorer (see section 4.2.3). To extract via iTunes, connect iPhone/iPad to PC/Mac, run iTunes, navigate to the iPad/iPhone device and then select Apps.



The .connection and .script files can be extracted via the "save to…" button at the bottom of the page.

Individual .connection and .script files that have been created or edited externally can be uploaded via the "Add…" button. When Get Console starts it checks the root folder for .script and .connection files and moves these into the correct internal folder.

7 General Troubleshooting

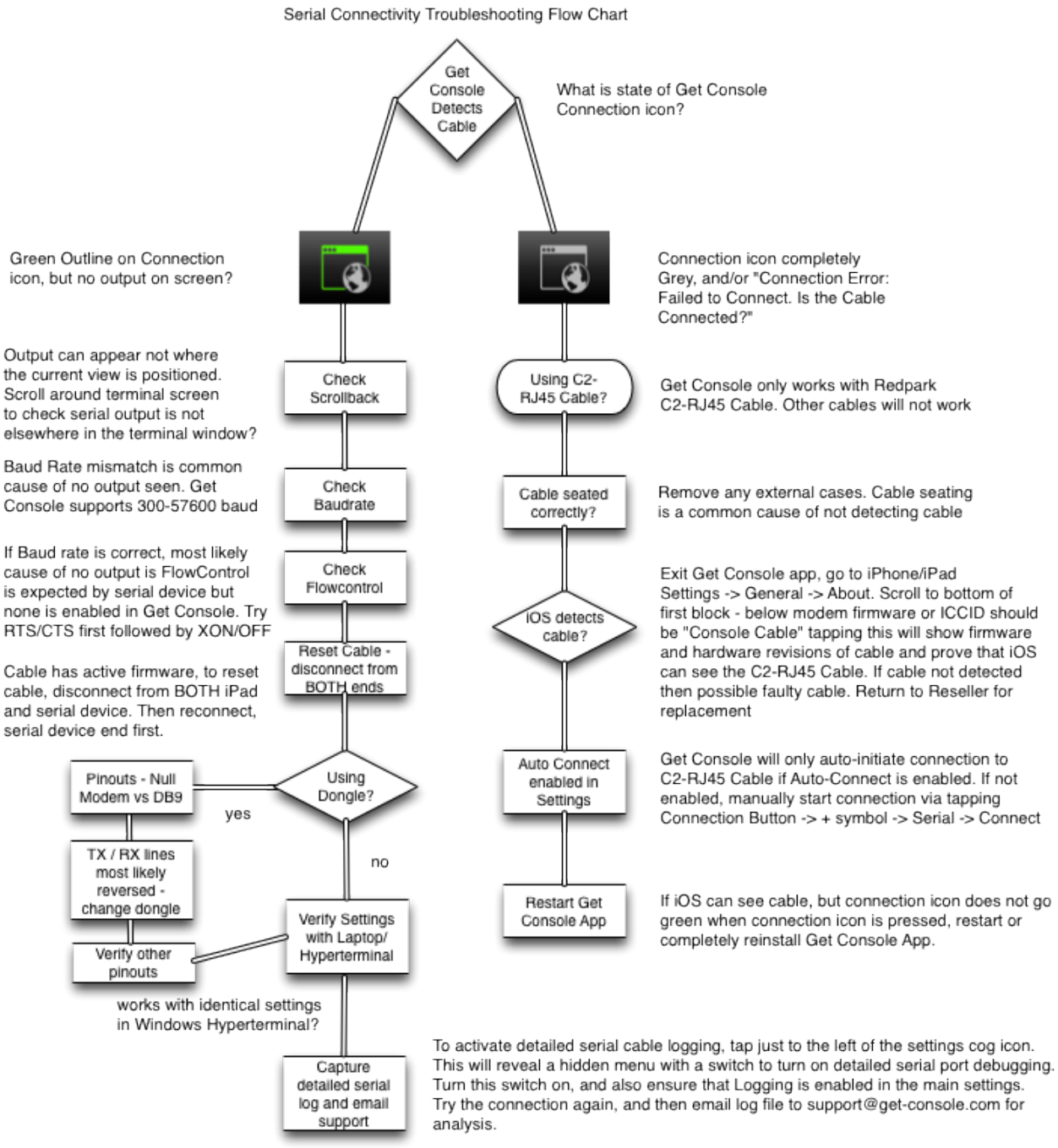
This section covers common problems reported by users, and typical fixes / work arounds

7.1 Serial Connectivity Issues

Serial connectivity issues can be divided into 2 types:

- Where Get Console cannot recognize / communicate with Redpark C2-RJ45 cable
- Where Get Console can recognize Redpark C2-RJ45 cable, but no console output is seen on the screen

The following flowchart helps answer the most common reasons why no output can be seen on screen when serial cable is connected to a device. Note that why Get Console is generally compatible with many serial devices, Cloudstore will only support / troubleshoot connections to devices that we have officially tested with. The most recent list of Cloudtore supported serial devices for use with Get Console is contained in the support forum at http://www.get-console.com/forum/



7.1.1 Console Cable Not Detected

If this error is seen and there is a cable plugged into the iOS device, then check that the iOS device is detecting the cable by going to the iOS devices general settings -> about -> console cable. This will show the cable hardware and firmware detected. Get Console only works with the Redpark C2-RJ45 cable. If you have a C2-RJ45 cable connected but it is not detected by the apple iOS operating system, then check it is seated correctly, reinstall Get Console and if still having issues contact the cable vendor for a replacement.

7.1.2 Console Cable Detected, No Output on Screen

If the cable is detected (Green outline on Connection icon for Serial session), but no output appears on the screen the most common reasons are:

- Baud Rate mismatch – note that Get Console does not support 115200 baud on current Redpark C2-RJ45 cables

- Flow control is required but not set. Note disconnect cable from both iPad/iPhone and serial device after changing Flow Control to reset cable.
- Pinouts of device are reversed from what is expected by C2-RJ45 cable. This is very common. The pinouts of the C2-RJ45 cable are as per Cisco console port. Using a RJ45-to-DB9 adaptor either makes the DB9 interface DTE or DCE depending on whether it reverses TX/RX and flow control/singalling pins. Use adaptors available from the get-console.com/shop which have been specifically wired to convert the C2-RJ45 to DB9 (std) and DB9 (null modem) including all the control/signaling.
- Cable needs to be reset – the cable has active electronics in it. Occasionally it needs to be reset by removing BOTH ends of the cable from iPad and Serial device, and then reconnecting it first to the Serial device end

7.2 Session Sharing Issues

Session Sharing to Get Console Public Servers fails. If user receives “Failed to connect / read from remote control server”:

- Check your internet connection (can you browse in mobile safari to www.get-console.com)
- Check Session Sharing settings configurations defined in Main Settings -> Session Sharing are correct (see section 3.6).
- If everything is correctly configured, try again – depending on latency the first attempt to connect can fail due to underlying encryption key exchange taking too long. The next attempt works as the encryption keys are cached by the Get Console app.
- If still having issues with a public Get Console server, try to use an alternative one (ie Asia Pacific rather than North America) .
- If having problems with a Private Server contact the Server Administrator, or if support has been purchased, contact Cloudstore.