



# AVcontrol Programming Guide

application version 4.3.x ©2012-2024 [avcontrolapp.com](http://avcontrolapp.com)

## AVcontrol for iOS • Programming guide

Introduction .....	1
About premium.....	1
Connection method .....	1
About custom resources .....	2
Protocol command structure.....	3
About percent (URL) encoding .....	3
About unicode .....	3
iOS Today Extension .....	4
Apple Watch Extension .....	4
Menu and (sub) page commands (control system ⇒ AVcontrol) .....	5
User interface commands (control system ⇒ AVcontrol).....	10
Table (list) view commands (control system ⇒ AVcontrol).....	19
Tcp socket connection commands (control system ⇒ AVcontrol) .....	23
Miscellaneous commands (control system ⇒ AVcontrol) .....	24
Command feedback (AVcontrol ⇒ control system).....	29
Watch API (control system ⇒ AVcontrol) .....	34
Watch API Command feedback (AVcontrol ⇒ control system) .....	37
Appendix A • Supported resource file formats.....	38
Appendix B • background image resolutions .....	39
Appendix C • color value table.....	40
Appendix D • symbol font keyboard map .....	41
Appendix E • icon number reference .....	42

## Introduction

AVcontrol is a universal iOS application for control of your Crestron, AMX or any other ethernet enabled control system which allows for custom programming.

The application connects via a Wi-Fi, wired or cellular network using TCP/IP sockets. The programming of your control system needs to be updated to enable the use of AVcontrol.

Unlike traditional AV touch panels the user interface of AVcontrol is fully served via custom programming. Every UI element is positioned and customized using serial commands from the control system to AVcontrol.

The control system program should open a TCP/IP server on a port of your own choice (default port is 2048) and should respond to AVcontrol commands as described in the chapter [Connection method](#).

The free version of AVcontrol allows for a basic set of commands. To use all the options AVcontrol offers consider buying the Premium version via in-app purchase.

## About premium

The premium version of AVcontrol can be purchased via in-app purchase right from within the app. Without premium the app will still be functional and connect to control systems without time limitations but with a limited set of options.

A subset of commands will require the premium version. Every command in this guide that requires premium will be prefixed with **[P]**.

## Connection method

Set the control system's IP address or DNS name in iOS Settings under tab 'AVcontrol'.

Program an IP server in the control system's program listening on the port specified in the settings. The default port is 2048. Make sure to correctly close and re-open the port programmatically in case the device gets disconnected from the control system.

You have the option to use different IP/DNS addresses and ports for connection via Wi-Fi/wired and connection via cellular network. When Wi-Fi/wired is not available the app will automatically connect to the control system using the IP/DNS and port specified in cellular connection settings. Note that cellular connection information is empty when first installing AVcontrol.

AVcontrol will send a '[#Ping!](#)' request every 3 seconds, the control system must be programmed to reply with '[@Pong#](#)' in order to maintain the connection.

If AVcontrol does not receive a '[@Pong#](#)' reply after 3 '[#Ping!](#)' requests it will automatically disconnect from the control system and retry the connection.

## About custom resources

Prepare all resources (refer to [appendix A](#) for a list of supported file formats) and upload them to the control system's http or https web server. It is recommended that all resources reside in a folder named 'AVcontrol' (case sensitive). Alternatively all resources can be uploaded in the root folder as well (be sure to change the folder setting to 'Root' in AVcontrol Settings under 'Advanced').

Create and upload a text file called 'Resources.txt' with a comma delimited list of all resource file names. For example;

BlueButtonLarge\_Off.png,  
BlueButtonLarge\_On.png,  
StartAV.wav

'Resources.txt' must also be uploaded to the 'AVcontrol' or 'Root' web server directory.

Provide 2 image files called 'bg\_portrait.png' and 'bg\_landscape.png' if a custom (initial) background is required.

- For retina equipped iPhone/iPod Touch devices append <@2x> or <@3x> to the filename.
- For iPad devices append <~ipad> to the filename.
- For retina equipped iPad devices append <@2x~ipad> to the filename.
- For iPad Pro 10.2" devices append -810h@2x to the filename.
- For iPad Pro 10.5" devices append -834-1h@2x to the filename.
- For iPad Pro 11" devices append -834-2h@2x to the filename.
- For iPad Pro 12.9" devices append -1024h@2x to the filename.
- For iPad mini 8.3" devices append -744h@2x to the filename.
- For iPhone 5/SE (1st gen) and iPod Touch (5th gen) devices append -568h@2x to the filename.
- For iPhone 6/7/8/SE (2nd gen) devices append -667h@2x to the filename.
- For iPhone 6/7/8 Plus and iPad 12.9 inch devices append -736h@3x to the filename.
- For iPhone X/XS/11 Pro devices append -812h@3x to the filename.
- For iPhone XR devices append -896h@2x to the filename.
- For iPhone XS Max/11 Pro Max devices append -896h@3x to the filename.
- For iPhone 12/13 (Pro)/14 devices append -844h@3x to the filename.
- For iPhone 12/13 (Mini) devices append -780h@3x to the filename.
- For iPhone 12/13 Pro Max/14 Plus devices append -926h@3x to the filename.
- For iPhone 14 Pro devices append -852h@3x to the filename.
- For iPhone 14 Pro Max devices append -932h@3x to the filename.

To support custom backgrounds for all currently supported devices refer to [appendix B](#)

The @2x and @3x modifiers are also applicable for all other image resources. For @2x resources double the pixel size of the original item and for @3x resources triple the pixel size of the original item. If no device specific resource is loaded the application will use the resource with the original name.

- Tip: use graphic resources with double or triple the intended size so you can use 1 resource for multiple devices. AVcontrol will scale the image to the correct resolution for the device it is currently running on.

Special care must be taken in the naming of animation sequences. Provide a list in the following format: <Name>\_xx.extension where xx refers to the image number in the image sequence. For example;

LogoAnimation\_01.png,  
LogoAnimation\_02.png,  
etc...

AVcontrol does not verify the actual existence of the file(s) during (re)loading so make sure that all file names correspond with the file names specified in 'Resources.txt'.

In addition to playing custom sounds by sending a command from the control system you can add a default click sound when a button is pressed. Name this sound 'click.wav' and add it to the resource list and items. If no default click sound is needed then don't include this file.

For resources you wish to use on the Watch App prefix the names with 'wk\_' (wk stands for WatchKit). For example;

wk\_MainButton.png,  
wk\_Logo.png,  
etc...

## Protocol command structure

Commands from the control system to AVcontrol are ASCII based and must start with '@' and end with '#'.

Commands from AVcontrol to the control system are ASCII based strings and will start with '#' and end with '!'.

AVcontrol will send a '#Ping!' request every 3 seconds, the control system must be programmed to reply with '@Pong#' in order to maintain a healthy connection.

When no ping reply has been received for approximately 10 seconds AVcontrol will assume network connection has been lost and will try to reconnect.

Parameters in a command must be separated by '|' (pipe character).

<tag> parameters are used to uniquely identify each control and visual element. <tag> must be a number in the range from 1 up to 10000.

## About percent (URL) encoding

Commands from the control system to AVcontrol can contain special characters using percent (URL) encoding. AVcontrol will decode the percent codes to their common character counterparts. Refer to [this page](#) for more information on percent (URL) encoding.

Tip: when button text needs to be multi line you can add the %0d encoding to force a carriage return.

## About unicode

Buttons and text labels support Unicode text codes. Prefix Unicode hex codes with '\u' and AVcontrol will automatically display the correct Unicode character. Refer to [this page](#) for a comprehensive overview of available Unicode codes. Refer to [this page](#) for a Unicode converter utility.

## iOS Today Extension

AVcontrol support the use of an iOS Today Extension *when using Premium*. With an iOS Today Extension (often called widget) you can quickly access functionality that you use often such as adjusting volume or controlling lights. Your control system needs to be programmed to support the use of a widget.

A separate IP server needs to be programmed for communicating with the widget. The IP port is the normal AVcontrol IP port +1. For instance; if you have defined port 2048 as the IP port in AVcontrol Settings then the widget will communicate on port 2049.

After enabling the widget you can use a subset of the commands described in this document. Commands that are supported for use with widgets are prefixed with **[TE]**.

The available screen size for a widget is 320 \* 240 (iPhone or iPad portrait) or 480 \* 240 points (all other configurations).

## Apple Watch Extension

AVcontrol supports the Apple Watch for quick access to functionality *when using Premium*, much like the Today Extension.

Similar to the Today Extension you need to program a separate IP server for communication with the Watch App. The IP port is the normal AVcontrol IP port +2. For instance; if you have defined port 2048 as the IP port in AVcontrol Settings then the Watch App will communicate on port 2050.

*\*Please note that the Watch needs to be in (Bluetooth) range of your iPhone. This is due to the fact that TCP/IP socket connections are not directly supported on watchOS. You cannot connect from a cellular Watch without a paired iPhone in range.*

The Watch App can present control options in the form of a list. You can add or remove items from this list as required.

The Watch features unique interface elements that you can add, remove and control using the [API](#). This API is a subset of the regular API but has been completely redesigned specifically for the Apple Watch.

## Menu and (sub) page commands (control system ⇒ AVcontrol)

<a href="#">@initMenu</a>	Initialize a new menu
<a href="#">@initUpdatedMenu</a>	Initialize a new updated menu
<a href="#">@configureMenuItemProperties</a>	Configure global settings for all menu items
<a href="#">@setArrowColor</a>	Set the scroll arrow color
<a href="#">@setMenuScrollIndicatorStyle</a>	Set the scroll indicator style
<a href="#">@setMenuBackgroundColor</a>	Set the menu background color
<a href="#">@clearMenu</a>	Clear the current menu
<a href="#">@clearUpdatedMenu</a>	Clear the current updated menu
<a href="#">@scrollMenu</a>	Scroll the menu programmatically
<a href="#">@pageMenu</a>	Scroll the menu programmatically
<a href="#">@bounceMenu</a>	Set menu scroll bounce behavior
<a href="#">@initPage</a>	Initialize a new subpage
<a href="#">@showPage</a>	<b>[TE]</b> Show the initialized subpage (or widget page)
<a href="#">@showPageModal</a>	<b>[P]</b> Show the initialized subpage as a modal page
<a href="#">@centerPage</a>	Center the subpage
<a href="#">@removePage</a>	Remove the subpage
<a href="#">@clearPage</a>	Clear the subpage
<a href="#">@aPageControl</a>	<b>[P]</b> Add page control on the bottom of the page
<a href="#">@appendPage</a>	<b>[P]</b> Append a page to the current subpage
<a href="#">@commitPage</a>	<b>[P]</b> Finalize the appended page
<a href="#">@scrollPage</a>	<b>[P]</b> Scroll to an appended subpage
<a href="#">@initFullPage</a>	<b>[P]</b> Initialize a new full screen page
<a href="#">@showFullPage</a>	<b>[P]</b> Show the initialized full screen page
<a href="#">@removeFullPage</a>	<b>[P]</b> Remove a full screen page
<a href="#">@appendFullPage</a>	<b>[P]</b> Append a page to the current full page
<a href="#">@commitFullPage</a>	<b>[P]</b> Finalize the appended full page
<a href="#">@scrollFullPage</a>	<b>[P]</b> Scroll to an appended full page

**'@initMenu|<tag>-<\*image name>|...'**: Initialize the menu, where x is the tag number and y is the image name for the menu item(s). Repeat tag and image name for the desired number of items. This menu style will arrange the items in a swipe-able style display.

- \*All menu items are 160\*160 points in size, to prevent scaling artifacts provide images with the same size. If smaller images are needed use .png files with transparent areas around the actual image. Use 320px\*320px images for retina support.
- \*Substitute <image name> with a [color value](#) for a solid colored button.
- Alternatively you can specify an [icon number](#) to select a built in icon as a graphic representation of the menu item, for example: @initMenu|1-icon.100|2-icon.101 etc.

**'@initUpdatedMenu|<tag>-<\*image name>|...'**: Initialize the menu, where x is the tag number and y is the image name for the menu item(s). Repeat tag and image name for the desired number of items. This menu will arrange the items in a grid style display.

- \*All menu items are 160\*160 points in size, to prevent scaling artifacts provide images with the same size. If smaller images are needed use .png files with transparent areas around the actual image. Use 320px\*320px images for retina support.
- \*Substitute <image name> with a [color value](#) for a solid colored button.
- As an alternative you can specify an [icon number](#) to select a built in icon as a graphic representation of the menu item, for example: @initUpdatedMenu|1-icon.100|2-icon.101 etc.

**'@configureMenuItemProperties|<cornerRadius>|<fill color>|<border width>|<border color>|<button text color>|<font name>|<font size>|<text alignment>|<vertical text alignment>|<icon color>|<icon vertical justification offset>'**: Configure global properties for all menu items. Font name can be left empty if the default font is desired. Icon vertical justification offset sets the number of pixels that icons are vertically offset (negative values can be used).

**'@setArrowColor|<color value>|<alpha value>'**: Set the menu scroll arrow color. Default color is black with an alpha value of 0.8. Refer to [appendix C](#) for a list of possible color values. Valid alpha values are float values ranging from 0.0 to 1.0.

**'@setMenuScrollIndicatorStyle|<\*style>'**: Set the menu scroll indicator style.

- \*'0' = Black with a white border (default).
- \*'1' = Black.
- \*'2' = White.
- \*'3' = Hidden.

**'@setMenuBackgroundColor|<color value>'**: Set the menu background color. Default color is transparent (value 35). Refer to [appendix C](#) for a list of possible color values.

**'@clearMenu'**: Remove all menu items.

**'@clearUpdatedMenu'**: Remove all updated menu items.

**'@scrollMenu|<itemNumber>'**: Scrolls the specified item number in the menu into visible range.



**`@pageMenu|<\*mode>`**: Enable paging (menu scroll always stops with items centered) for menu. This feature is not available for iPhone and iPod Touch devices.

- **\*'0'** = Disabled (default).
- **\*'1'** = Enabled.

**`@bounceMenu|<\*mode>`**: Set menu scroll bounce behavior (when bounce is enabled the menu bar will appear to bounce back when the user swipes the menu to the beginning or the end).

- **\*'0'** = Disabled.
- **\*'1'** = Enabled (default).

**`@initPage|<\*optional:background image name>`**: Start a new subpage (put all UI control commands in between [`initPage`](#) and [`showPage`](#)).

- Subpages are 640\*640 points in size (portrait mode as well as in landscape mode). iPhone and iPod touch devices will automatically scale this to a suitable value for the specific device.
- Specifying a color value according to [appendix C](#) will generate a solid background color for the subpage.
- Subpages will be tagged with a value of 33001. You can use this to perform a [corner radius](#) command for example.

**[TE] `@showPage|<\*optional: subpage animation transition>`**: Shows a previously initialized subpage.

- **\*'0'** = No animation.
- **\*'1'** = Flip from left.
- **\*'2'** = Flip from right.
- **\*'3'** = Curl up.
- **\*'4'** = Curl down.
- **\*'5'** = Dissolve.
- **\*'6'** = Flip from top.
- **\*'7'** = Flip from bottom.
- **\*'8'** = Reveal from last pressed button.
- **\*'9'** = Non scaled iPhone subpage.
- **\*'10'** = Push.
- **\*'11'** = Move in.
- **\*'12'** = Reveal.
- **\*'13'** = Ripple. (deprecated in iOS13, defaults to fade)
- **\*'14'** = Cube.

**[P] `@showPageModal|<\*optional:subpage animation transition>`**: Shows a previously initialized subpage as a modal page.

- **\*'0'** = Cover vertical.
- **\*'1'** = Cover horizontal.
- **\*'2'** = Dissolve.

**`@centerPage|<\*state>|<\*\*\*animated>`**: Controls the subpage position.

- **\*'0'** = Not centered (default).
- **\*'1'** = Centered.
- **\*\*\*'0'** = Without animation.
- **\*\*\*'1'** = With animation.

**`@removePage|<\*optional:subpage animation transition>`**: Remove the current subpage.

- **\*'0'** = No animation.
- **\*'1'** = Flip from left.
- **\*'2'** = Flip from right.
- **\*'3'** = Curl up.
- **\*'4'** = Reserved (will default to "Curl up").
- **\*'5'** = Dissolve.
- **\*'6'** = Flip from top.
- **\*'7'** = Flip from bottom.
- **\*'8'** = "Un" reveal from last pressed menu item.

**`@clearPage`**: Remove all elements from the current subpage.

**[P] `@aPageControl|<dot color value>|<page indicator dot color value>|<alpha value>|<optional: fade in duration>`**: Add a page control indicator on the bottom of the page (sub- or fullpage). This indicator will be shown only when there are multiple pages added via the [@appendPage](#) and [@commitPage](#) or [@appendFullPage](#) and [@commitFullPage](#) commands.

**[P] `@appendPage|<optional:background image name>`**: Append a page to the current subpage. use [@commitPage](#) to finalize the page.

**[P] `@commitPage`**: Finalize the appended page.

**[P] `@scrollPage|<pageNumber>|<\*animated>`**: Scroll to an appended subpage.

- **\*'0'** = Not animated.
- **\*'1'** = Animated.

**[P] `@initFullPage|<\*desired orientation>|<optional:background image name>`**: Start a new full screen page (put all UI control commands in between [`initFullPage`](#) and [`showFullPage`](#)). *This command will lock the device to the requested orientation for as long as the page is showing. For a comprehensive list of resolutions refer to [appendix B](#)*

- **\*'1'** = Portrait (home button below).
- **\*'2'** = Portrait (home button above).
- **\*'3'** = Landscape left (home button right).
- **\*'4'** = Landscape right (home button left).
- **\*'5'** = Portrait (page will rotate upon device rotation).
- **\*'6'** = Landscape (page will rotate upon device rotation).
- **\*'7'** = Universal – page will be sized full screen regardless of the current device or orientation.
- **\*\*** Devices *will not scale these pages* (as opposed to subpages) so provide separate designs programmatically if multiple devices need to be supported.
- Specifying a color value according to [appendix C](#) will generate a solid background color for the page.

**[P] '@showFullPage|<\*optional: page animation transition>':** Shows a previously initialized full screen page.

- \*'0' = No animation.
- \*'1' = Slide from right.
- \*'2' = Slide from left.
- \*'3' = Slide from top.
- \*'4' = Slide from bottom.
- \*'5' = Dissolve.
- \*'6' = Reveal from last pressed button.
- \*'7' = Reserved.
- \*'8' = Reserved.
- \*'9' = Reserved.
- \*'10' = Push.
- \*'11' = Move in.
- \*'12' = Reveal.
- \*'13' = Ripple. (deprecated in iOS13, defaults to fade)
- \*'14' = Cube.

**[P] '@removeFullPage|<\*optional: page animation transition>':** Removes a previously initialized full screen page.

- \*'0' = No animation.
- \*'1' = Slide to right.
- \*'2' = Slide to left.
- \*'3' = Slide to top.
- \*'4' = Slide to bottom.
- \*'5' = Dissolve.

**[P] '@appendFullPage|<optional:background image name>':** Append a page to the current full page. use '[commitFullPage](#)' to finalize the page.

**[P] '@commitFullPage':** Finalize the appended page.

**[P] '@scrollFullPage|<pageNumber>|<\*animated>':** Scroll to an appended full page.

- \*'0' = Not animated.
- \*'1' = Animated.

## User interface commands (control system ⇒ AVcontrol)

<a href="#">@tLabel</a>	Add a text label
<a href="#">@iLabel</a>	Add an input text label
<a href="#">@iLabelPrivate</a>	<b>[P]</b> Add a private input text label
<a href="#">@iLabel 0</a>	Remove keyboard programmatically (no text feedback)
<a href="#">@mLabel</a>	<b>[P]</b> Add a marquee (scrolling) label
<a href="#">@sText</a>	<b>[TE]</b> Set text
<a href="#">@sTextAlignment</a>	Set horizontal text alignment
<a href="#">@sVerticalTextAlignment</a>	Set vertical text alignment
<a href="#">@tFontSize</a>	<b>[TE]</b> Set individual tag font size
<a href="#">@tFont</a>	<b>[P] [TE]</b> Set individual tag font
<a href="#">@tFontSizeGlobal</a>	<b>[TE]</b> Set global font size
<a href="#">@tFontGlobal</a>	<b>[P] [TE]</b> Set global font
<a href="#">@cButton</a>	<b>[TE]</b> Add a control system feedback button
<a href="#">@tFeedback</a>	<b>[TE]</b> Set feedback
<a href="#">@mButton</a>	<b>[TE]</b> Add a momentary feedback button
<a href="#">@bButton</a>	Add a blink feedback button
<a href="#">@tButton</a>	Add a toggle feedback button
<a href="#">@sImage</a>	Add a static image
<a href="#">@sIconTintColor</a>	Set icon tint color
<a href="#">@dImage</a>	<b>[P] [TE]</b> Add a dynamic image
<a href="#">@aImage</a>	<b>[P]</b> Add an animated image
<a href="#">@aIndicator</a>	<b>[TE]</b> Add a indicator (spinner)
<a href="#">@pBargraph</a>	<b>[TE]</b> Add a passive bargraph
<a href="#">@aBargraph</a>	Add an active bargraph
<a href="#">@cBargraphColors</a>	<b>[P] [TE]</b> Custom bargraph colors
<a href="#">@sLevel</a>	Set bargraph level
<a href="#">@aLevelControl</a>	<b>[P] [TE]</b> Add relative level control to button
<a href="#">@aView</a>	Add an alert view
<a href="#">@aView 0</a>	Remove active alert view
<a href="#">@aViewTextColor</a>	<b>[P]</b> Customize alert view text color
<a href="#">@actionSheet</a>	<b>[P]</b> Add an action sheet view
<a href="#">@actionSheet 0</a>	<b>[P]</b> Remove active alert sheet
<a href="#">@aSwitch</a>	<b>[P] [TE]</b> Add a on/off switch
<a href="#">@sSwitch</a>	<b>[P] [TE]</b> Set switch state
<a href="#">@segControl</a>	<b>[P] [TE]</b> Add a segmented control
<a href="#">@segControlColor</a>	<b>[P] [TE]</b> Set segmented control tint color
<a href="#">@sSegControl</a>	<b>[P] [TE]</b> Select a segment programmatically
<a href="#">@aPickerView</a>	<b>[P]</b> Add a picker view control
<a href="#">@sPickerViewRow</a>	<b>[P]</b> Select a picker view element programmatically
<a href="#">@sPickerViewColor</a>	<b>[P]</b> Set the picker view color
<a href="#">@aStepper</a>	<b>[P] [TE]</b> Add a stepper view control

<a href="#">@sStepper</a>	<b>[P] [TE]</b> Set stepper view control value
<a href="#">@aWebView</a>	<b>[P]</b> Add a web view
<a href="#">@loadWebView</a>	<b>[P]</b> Load a web view
<a href="#">@aVideo</a>	<b>[P]</b> Add a video
<a href="#">@analogClock</a>	<b>[TE]</b> Add an analog style clock
<a href="#">@sAlpha</a>	<b>[TE]</b> Set control alpha
<a href="#">@sLabelTextColor</a>	Set the text color of a label
<a href="#">@sButtonTextColor</a>	Set the text color of a button
<a href="#">@tSwipeGesture</a>	<b>[P]</b> Add a swipe gesture recognizer
<a href="#">@tTapGesture</a>	<b>[P]</b> Add a tap gesture recognizer
<a href="#">@tPanGesture</a>	<b>[P]</b> Add a pan gesture recognizer
<a href="#">@sColorPicker</a>	<b>[P]</b> Add the iOS built in color picker
<a href="#">@authenticateUser</a>	<b>[P]</b> Request authentication via touchID
<a href="#">@sCornerRadius</a>	<b>[TE]</b> Add a rounded corner to an image or button
<a href="#">@sBorder</a>	<b>[TE]</b> Add a border to an image or button
<a href="#">@showsTouchWhenHighlighted</a>	<b>[TE]</b> Control touch on highlight behavior
<a href="#">@rControl</a>	<b>[TE]</b> Remove control

**`@tLabel|<tag>|<x>|<y>|<width>|<height>|<text>|<\*optional:color value>`:** add a dynamic or static text label to the subpage.

- \*Default color value = black, refer to [appendix C](#) for a list of possible values

**`@iLabel|<tag>|<x>|<y>|<width>|<height>|<placeholder text>|<\*border style>|<\*\*keyboard style>|<optional:color value>`:** add a user input enabled text label to the subpage.

- \*'0' = No border.
- \*'1' = Line border.
- \*'2' = Bezel border.
- \*'3' = Rounded rectangle border.
- \*\*'0' = Default keyboard.
- \*\*'1' = ASCII keyboard.
- \*\*'2' = Numbers and punctuation keyboard.
- \*\*'3' = URL keyboard.
- \*\*'10' = Default keyboard, sentences start with a capital letter.
- \*\* '11', '12', '13', '20' = same as above except keyboard appearance is dark.

**[P] `@iLabelPrivate|<tag>|<x>|<y>|<width>|<height>|<placeholder text>|<\*border style>|<\*\*keyboard style>|<optional:color value>`:** same functionality as iLabel with hidden characters (suitable for passwords).

**`@iLabel|0`:** Remove the keyboard programmatically (will not send text feedback to the control system).

**[P] '@mLabel|<tag>|<x>|<y>|<width>|<height>|<text or url>|<\*type>|<\*\*rate>|<\*\*\*fade length>|<optional:color value>'**: add a marquee (scrolling) label. This label will scroll automatically if the text is wider than the frame. If <text> is prefixed with http:// then it will be treated as a RSS feed URL and parsed accordingly.

- **\*'0'** = RSS time+subject+description.
- **\*'1'** = RSS time+subject.
- **\*'2'** = RSS subject only.
- **\*\*rate** refers to the speed of the label. It is defined as number of pixels per second.
- **\*\*\*fade length** refers to the number of pixels on either side where the view fades to transparent.

**[TE] '@sText|<tag>|<text>|<optional:counter direction\*>'** : Set the text of the button, label, switch, or menu item identified by tag number.

- **\*'-'** = text value decreases by one every second. When using ':' in the text value AVcontrol assumes timecode behavior.
- **\*'+'** = text value increased by one every second. When using ':' in the text value AVcontrol assumes timecode behavior.

**'@sTextAlignment|<tag>|<\*alignment or left alignment value>|<optional:right alignment value>'** : Set the horizontal text alignment of the label or button identified by tag number.

- **\*'0'** = Align text left.
- **\*'1'** = Align text centered (default).
- **\*'2'** = Align text right.
- when the optional right alignment value is used you can specify left and right alignment in points for example: '@sTextAlignment|1|20|10 '

**'@sVerticalTextAlignment|<tag>|<\*alignment or top alignment value>|<optional:bottom alignment value>'** : Set the vertical text alignment of the label or button identified by tag number.

- **\*'0'** = Align text center (default).
- **\*'1'** = Align text top.
- **\*'2'** = Align text bottom.
- when the optional bottom alignment value is used you can specify top and bottom alignment in points for example: '@sVerticalTextAlignment|1|20|10 '

**[TE] '@tFontSize|<tag>|<size>'** : Set the font size of the control identified by tag number.

**[P] [TE] '@tFont|<tag>|<font name>|<optional:size>'** : Set the font of the control identified by tag number. Font name must contain the exact [name of the font](#). A special application enclosed symbol font is available with the name "**Guifx V2 Transports**". refer to [appendix D](#) for a detailed keyboard map.

**[TE] '@tFontSizeGlobal|<size>'** : Set the global font size of all (following) items.

**[P] [TE] '@tFontGlobal|<font name>|<optional:size>'** : Set the font of all (following) controls. Font name must contain the exact [name of the font](#). A special application enclosed symbol font is available with the name '**Guifx V2 Transports**'. refer to [appendix D](#) for a detailed keyboard map.

**[TE] '@cButton|<tag>|<x>|<y>|<width>|<height>|<text>|<\*image name off state>|<\*image name on state>|<\*\*optional:text colorValue off state>|<\*\*optional:text colorValue on state>'**: Add a channel (feedback from control system) button.

- \*Substitute <image name off state> and <image name on state> with a [color value](#) for a solid colored button.
- \*\*Default color value = white, refer to [appendix C](#) for a list of possible values.

**[TE] '@tFeedback|<tag>|<\*state>|<\*\*optional:animation transition>'**: Set the state for control identified by tag number.

- \*'0' = set off state.
- \*'1' = set on state.
- \*\*'0' = No animation (same as not including the parameter).
- \*\*'1' = Flip from left.
- \*\*'2' = Flip from right.
- \*\*'3' = Curl up.
- \*\*'4' = Curl down.
- \*\*'5' = Dissolve.
- \*\*'6' = Flip from top.
- \*\*'7' = Flip from bottom.
- \*\*'8' = Reserved.
- \*\*'9' = Reserved.
- \*\*'10' = Push.
- \*\*'11' = Move in.
- \*\*'12' = Reveal.
- \*\*'13' = Ripple.
- \*\*'14' = Cube.

**[TE] '@mButton|<tag>|<x>|<y>|<width>|<height>|<text>|<\*image name off state>|<\*image name on state>|<\*\*optional:text colorValue off state>|<\*\*optional:text colorValue on state>'**: Add a momentary (no feedback from control system) button.

- \*Substitute <image name off state> and <image name on state> with a [color value](#) for a solid colored button.
- \*\*Default color value = white, refer to [appendix C](#) for a list of possible values.

**'@bButton|<tag>|<x>|<y>|<width>|<height>|<text>|<\*image name off state>|<\*image name on state>|<\*\*optional:text colorValue off state>|<\*\*optional:text colorValue on state>'**: Add a blink button to the subpage. Set the blink rate with the [sBlinkRate](#) command.

- \*Substitute <image name off state> and <image name on state> with a [color value](#) for a solid colored button.
- \*\*Default color value = white, refer to [appendix C](#) for a list of possible values.

**`@tButton|<tag>|<x>|<y>|<width>|<height>|<text>|<\*image name off state>|<\*image name on state>|<\*\*optional:text colorValue off state>|<\*\*optional:text colorValue on state>`:** Add a toggle button to the subpage. Whenever this button is tapped the state changes.

- \*Substitute <image name off state> and <image name on state> with a [color value](#) for a solid colored button.
- \*\*Default color value = white, refer to [appendix C](#) for a list of possible values.

**`@sImage|<tag>|<x>|<y>|<\*width>|<\*height>|<image name>`:** Add a static image (push/release enabled).

- \*Image gets scaled (if image size does not match the frame size) to width/height size of the frame.
- Alternatively you can specify an [icon number](#) to select a built in icon for the image, for example: *sImage|1|100|100|100|100|icon.100.*

**`@sIconTintColor|<tint color\*>`:** Set icon tint color. \*\*refer to [appendix C](#) for a list of possible values.

**[P] [TE] `@dImage|<tag>|<x>|<y>|<\*width>|<\*height>|<image URL>`:** Add a dynamic image (push/release enabled) to the subpage. AVcontrol will [notify](#) the control system when the image has finished loading.

- \*Image gets scaled (if image size does not match the frame size) to width/height size of the frame.

**[P] `@aImage|<tag>|<x>|<y>|<\*width>|<\*height>|<\*\*image name>|<number of frames>|<\*\*\*duration>`:** Add a animated image (push/release enabled).

- \*Image gets scaled to width/height size of the frame.
- \*\*Image name must refer to an array of images in the resource list, see chapter 'Using custom resources' for correct file naming convention.
- \*\*\*Number of seconds to complete a single animation cycle.

**[TE] `@aIndicator|<tag>|<center x>|<center y>|<\*style>|<\*\*optional:color value>`:** Add an activity indicator (spinner).

- \*'0' = style white large.
- \*'1' = style white.
- \*'2' = style gray.
- \*\*refer to [appendix C](#) for a list of possible values.

**[TE] `@pBargraph|<tag>|<x>|<y>|<width>|<height>|<\*orientation>|<min value>|<max value>|<initial value>|<\*\*off image name>|<\*\*on image name>`:** Add a passive bargraph (progress bar).

- \*'h' = horizontal, 'v' = vertical.
- \*\*Custom images for the appearance of the bargraph. These parameters can be left empty (*do* provide the required number of '|' characters!) to use the default iOS progress bar appearance. You can also provide solid color values, refer to [appendix C](#) for a list of possible values.



**'@aBargraph|<tag>|<x>|<y>|<width>|<height>|<\*orientation>|<min value>|<max value>|<initial value><\*\*off image name><\*\*on image name><\*\*thumb image name>':** Add an active bargraph (slider).

- **\*'h'** = horizontal, **'v'** = vertical.
- **\*\*Custom images** for the appearance of the bargraph. These parameters can be left empty (*do* provide the required number of **'|'** characters!) to use the default iOS touch slider appearance. You can also provide solid color values, refer to [appendix C](#) for a list of possible values.

**[P] [TE] '@cBargraphColors|<tag>|<off color>|<on color>|<\*optional:thumb color>':** Specify custom colors for the default iPad/iPhone/iPod bargraph appearance. Refer to [appendix C](#) for a list of possible values.

**'@sLevel|<tag>|<level>':** Set the level of an active or passive bargraph identified by tag number.

**[P] [TE] '@aLevelControl|<tag>|<level number>|<value>|<repeat time>':** Add relative level control to a button. **<value>** can be plus or minus and repeat time indicates the number of value changes per second.

**'@aView|<tag>|<title>|<message>|<button 1 text>|<button 2 text>|etc...':** Display an alert that presents options to the user. Use as many button text parameters as needed. **<tag>** is limited from 1 up to 100. Leave the button parameter empty to display an alert without buttons (this is not recommended since it leaves the user without an option to dismiss the alert).

**'@aView|0':** Will dismiss a previously activated alert programmatically.

**[P] '@aViewTextColor|<color value\*>':** Customize text colors for alerts.

- \*refer to [appendix C](#) for a list of possible values.

**[P] '@actionSheet|<tag>|<title>|<cancel button title>|<destructive (red) button title>|<button 1 text>|<button 2 text>|etc...':** Display an action sheet that presents options to the user. Use as many button text parameters as needed. **<tag>** must be an active view on the screen so the application can determine from where the sheet must originate. iPhone action sheets do not use originating tags. Leave the cancel and/or destructive button parameter(s) empty to remove those options.

**[P] '@actionSheet|0':** Will dismiss a previously activated action sheet programmatically.

**[P] [TE] '@aSwitch|<tag>|<center x>|<center y>|<\*initial state>|<\*\*optional:color value>':** Add a switch control.

- **\*'0'** = state off.
- **\*'1'** = state on.
- **\*\*refer to [appendix C](#)** for a list of possible values.

**[P] [TE] '@sSwitch|<tag>|<\*state>':** Set the state of a switch identified by tag number.

- \*'0' = state off.
- \*'1' = state on.

**[P] [TE] '@segControl|<tag>|<x>|<y>|<width>|<\*style>|<segment 1 title>|...':** Add a segmented control. The amount of segment titles determine the amount of segments in the control. The style parameter has been deprecated in iOS7; it will not change the appearance of the control. The height of the control is fixed.

- \*'0' = Plain style.
- \*'1' = Bordered style.
- \*'2' = Bar style.

**[P] [TE] '@segControlColor|<tag>|<color value>|<optional:text color>|<optional:selected segment fill color>':** Set tint, text and selected segment color for the segmented control identified by tag number. Refer to [appendix C](#) for a list of possible values. This command needs iOS13.x or higher.

**[P] [TE] '@sSegControl|<tag>|<index>':** Select a segment in a segmented control identified by tag number programmatically. <index> starts a 0 for the leftmost segment.

**[P] '@aPickerView|<tag>|<x>|<y>|<width>|<height>|<element 1 title>|...':** Add a picker view control. The amount of element titles determine the amount of elements in the control.

**[P] '@sPickerViewRow|<tag>|<index>':** Select an element in a picker control identified by tag number programmatically. <index> starts a 0 for the first element.

**[P] '@sPickerViewColor|<tag>|<text color value>|<optional:background color value>':** Set the picker view color.

**[P] [TE] '@aStepper|<tag>|<x>|<y>|<min>|<max>|<initial value>|<optional:color value>':** Add a stepper view control. This control is available on iOS6 or later.

**[P] [TE] '@sStepper|<tag>|<value>':** Set stepper view control value programmatically.

**[P] '@aWebView|<tag>|<x>|<y>|<width>|<height>|<url>|<enabled/disabled>|<initial vertical offset in pixels>|<optional: initial alpha>':** Add a web view. Use the <enabled> or <disabled> parameter to enable or disable user interaction for this view. If needed use button tags 10001 (back) and 10002 (forward) for navigation controls. Use prefix <local:> to display local files, including pdf and office files.

**[P] '@loadWebView|<tag>|<url>':** Load a wegview in the control identified by tag.

**[P] '@aVideo|<tag>|<x>|<y>|<width>|<height>|<url>|<enabled/disabled/paused>|<optional:initial alpha>':** Add a web video. Use the <enabled> or <disabled> parameter to enable or disable user interaction for this view. For volume control add an active bargraph control with a tag value of 10003. Use prefix <local:> to display local files.

AVcontrol will sent the following state notifications during the video lifecycle:

- #Video went full screen
- #Video did exit full screen
- #Video did finish

**[TE] '@analogClock|<tag>|<x>|<y>|<width>|<height>|<optional:second hand color>|<optional:clock color>':** Add a predesigned analog clock. Refer to [appendix C](#) for a list of possible color values for the second hand and the clock.

**[TE] '@sAlpha|<tag>|<value>|<optional:animation duration>':** Set the alpha value on a control identified by tag. Valid values are float values ranging from 0.0 to 1.0. Use the optional animation duration (seconds) parameter to animate the value change. Use tag 33001 to change the subpage color.

**'@sLabelTextColor|<tag>|<red>|<green>|<blue>|<optional:alpha>':** Set the color value on a label identified by tag. Valid values are float values ranging from 0.0 to 1.0. Alpha value is optional. When only the red parameter is given the text color will use a color value.

**'@sButtonTextColor|<tag>|<\*state>|<red>|<green>|<blue>|<optional:alpha>':** Set the color value on a button identified by tag. Valid values are float values ranging from 0.0 to 1.0. Alpha value is optional. When only the red parameter is given the text color will use a color value.

- \*'0' = off state.
- \*'1' = on state.

**[P] '@tSwipeGesture|<tag>|<\*direction>|<number of touches required>':** Add a swipe gesture recognizer to control identified by tag number (removes push/release handlers!).

- \*'l' = direction left.
- \*'r' = direction right.
- \*'u' = direction up.
- \*'d' = direction down.

**[P] '@tTapGesture|<tag>|<number of taps required>|<number of touches required>':** Add a tap gesture recognizer to control identified by tag number (removes push/release handlers!).

**[P] '@tPanGesture|<tag>|<number of touches required>':** Add a pan gesture recognizer to control identified by tag number (removes push/release handlers!).

**[P] '@sColorPicker|<title>|<\*supports alpha>|<\*\*continuous updates>|<optional:initial red value>|<optional:initial green value>|<optional:initial blue value>|<optional:initial alpha value>':** Show the iOS built in color picker. The optional initial color values are float values ranging from 0.0xxxx to 1.0xxxx. If set these values will set the initial selected color in the picker. Upon dismissal of the color picker the selected color values will be reported back to the control system (see [colorPickerColorSelected](#)).

- **\*'YES'** = color picker will enable alpha values
- **\*'NO'** = color picker will disable alpha values
- **\*\*'YES'** = color picker will report selected color values continuously as the user selects
- **\*\*'NO'** = color picker will report selected color value when the user closes the picker

**[P] '@authenticateUser|<user message>':** Request authentication via biometrics (touchID or faceID) with a user message string.

**[TE] '@sCornerRadius|<tag>|<radius>|<\*optional:corner>':** Add a rounded corner (mask) to an image or button.

- **\*'1'** = top left
- **\*'2'** = top right
- **\*'3'** = top left + top right
- **\*'4'** = bottom left
- **\*'5'** = bottom left + top left
- **\*'6'** = bottom left + top right
- **\*'7'** = bottom left + top left + top right
- **\*'8'** = bottom right
- **\*'9'** = bottom right + top left
- **\*'10'** = bottom right + top right
- **\*'11'** = bottom right + top left + top right
- **\*'12'** = bottom right + bottom left
- **\*'13'** = bottom right + bottom left + top left
- **\*'14'** = bottom right + bottom left + top right
- **\*'15'** = all corners

**'@sBorder|<tag>|<width>|<optional:colorValue>':** Add a border to an image or button, refer to [appendix C](#) for a list of possible color values. Use <tag>=0 to set this globally.

**[TE] '@showsTouchWhenHighlighted|<tag>|<\*value>|<\*\*optional:disable click sound on touch>':** Control touch on highlight behavior. Use <tag>=0 to set this globally.

- **\*'0'** = disabled.
- **\*'1'** = enabled (default).
- **\*\*'0'** = click sound enabled (default).
- **\*\*'1'** = click sound disabled.

**[TE] '@rControl|<tag>|<optional:animated value>':** Remove control identified by tag number.

## Table (list) view commands (control system ⇒ AVcontrol)

<a href="#"><u>@sTableView</u></a>	<b>[P]</b> Show the table view
<a href="#"><u>@rTableView</u></a>	<b>[P]</b> Remove/hide the table view
<a href="#"><u>@cTableView</u></a>	<b>[P]</b> Clear the table view contents
<a href="#"><u>@aTableViewCell</u></a>	<b>[P]</b> Add a table view item
<a href="#"><u>@iTableView</u></a>	<b>[P]</b> Index the table view
<a href="#"><u>@tTableViewHeader</u></a>	<b>[P]</b> Set title for table view header
<a href="#"><u>@hTableViewHeader</u></a>	<b>[P]</b> Hide table view header
<a href="#"><u>@tTableViewFooter</u></a>	<b>[P]</b> Set title for table view footer
<a href="#"><u>@hTableViewFooter</u></a>	<b>[P]</b> Hide table view footer
<a href="#"><u>@sTableViewRow</u></a>	<b>[P]</b> Select table view row programmatically
<a href="#"><u>@sTableViewRowWithName</u></a>	<b>[P]</b> Select table view row based on its content
<a href="#"><u>@dTableViewRow</u></a>	<b>[P]</b> Deselect table view row programmatically
<a href="#"><u>@sTableViewAnimation</u></a>	<b>[P]</b> Specify the animation type for row changes
<a href="#"><u>@TableViewAccessoryType</u></a>	<b>[P]</b> Set an accessory type for table view
<a href="#"><u>@sTableViewFontSize</u></a>	<b>[P]</b> Set table view font size (including header & footer)
<a href="#"><u>@sTableViewFont</u></a>	<b>[P]</b> Set table view font (including header & footer)
<a href="#"><u>@sTableViewColor</u></a>	<b>[P]</b> Set table view color
<a href="#"><u>@sTableViewTextColor</u></a>	<b>[P]</b> Set table view text color
<a href="#"><u>@sTableViewRowHeight</u></a>	<b>[P]</b> Set table view row height
<a href="#"><u>@TableViewPullToRefresh</u></a>	<b>[P]</b> Add "pull to refresh" functionality
<a href="#"><u>@startTableViewRefresh</u></a>	<b>[P]</b> Start the table view refresh mode
<a href="#"><u>@endTableViewRefresh</u></a>	<b>[P]</b> End the table view refresh mode
<a href="#"><u>@TableViewDeleteEnabled</u></a>	<b>[P]</b> Add "slide to delete" functionality
<a href="#"><u>@aTableViewLongPress</u></a>	<b>[P]</b> Add long press event handler
<a href="#"><u>@sTableViewImageProperties</u></a>	<b>[P]</b> Set table view image properties
<a href="#"><u>@sTableViewMultiline</u></a>	<b>[P]</b> Set table view image properties
<a href="#"><u>@sTableViewSeparatorColor</u></a>	<b>[P]</b> Set table view separator color
<a href="#"><u>@storeTableViewData</u></a>	<b>[P]</b> Store table view data to device
<a href="#"><u>@retrieveTableViewData</u></a>	<b>[P]</b> Retrieve table view data

**[P] '@sTableView|<x>|<y>|<width>|<height><optional:fade duration>|<optional|\*user interaction enabled>':** Show the table (list) view (1 view per (sub) page supported). If the optional fade duration is specified the view will fade in. Valid duration values are 0.0 to 5.0 (in seconds).

- **\*'enabled'** = user interaction enabled.
- **\*'disabled'** = user interaction disabled.

**[P] '@rTableView|<optional:fade duration>':** Remove/hide the table (list) view. If the optional fade duration is specified the view will fade out. Valid duration values are 0.0 to 5.0 (in seconds).

**[P] '@cTableView':** Clears the table (list) view (all items will be removed).

- **[P] '@aTableViewItem|<item name>|<optional:detail title>|<optional:resource name or URL>':** Add item to the table (list) view. If the table view is indexed adding items is disabled. [Clearing](#) the table view contents will re-enable item adding. You can specify an [icon number](#) to select a built in icon as a resource, for example: @aTableViewItem|Item name|detail name|icon.100.

**[P] '@iTableView|<\*optional:sort alphabetically>':** Index the table (list) view. This will analyze the contents of the table view and divide the content into sections based on the first letter of the items. A fast scroll helper bar on the right side of the view will be added. The application will notify the control system when it has finished indexing (see [tViewIndexingDoneReply](#)). Specify **\*YES** for sorting the table view alphabetically.

**[P] '@tTableViewHeader|<title>':** Set title for table (list) view header (will automatically show the header as well).

**[P] '@hTableViewHeader':** Remove/hide the header.

**[P] '@tTableViewFooter|<title>':** Set title for table (list) view footer (will automatically show the footer as well).

**[P] '@hTableViewFooter':** Remove/hide the footer.

**[P] '@sTableViewRow|<x>':** Select row programmatically (number based).

**[P] '@sTableViewRowWithName|<content>':** Select row programmatically (content based).

**[P] '@dTableViewRow':** Deselect the currently selected row programmatically.

**[P] '@sTableViewAnimation|<\*value>':** Specify the animation type for row changes.

- \*'0' = fade (default).
- \*'1' = slide from right.
- \*'2' = slide from left.
- \*'3' = slide from top.
- \*'4' = slide from bottom.
- \*'5' = no animation.
- \*'6' = slide to keep centered.
- \*'7' = automatic (iOS chooses the style).

**[P] '@TableViewAccessoryType|<\*value>':** Set an accessory type for table (list) view.

- \*'0' = none.
- \*'1' = disclosure indicator.
- \*'2' = detail disclosure button.
- \*'3' = checkmark.

**[P] '@sTableViewCellFontSize|<size>|<optional:detail label font size>':** Set the font size of the table view text (including header & footer).

**[P] '@sTableViewCellFont|<font name>|<optional:size>':** Set the font of the table view text (including header & footer). Font name must contain the exact [name of the font](#).

**[P] '@sTableViewCellColor|<background color>|<section index tracking color>|<section index text color>':** Set the color of the table view, refer to [appendix C](#) for a list of possible color values

**[P] '@sTableViewCellTextColor|<red>|<green>|<blue>|<optional:alpha>':** Set the color value on a button identified by tag. Valid values are float values ranging from 0.0 to 1.0. Alpha value is optional. When only 1 parameter is given the text color will use a color value. When 2 parameters are used the second parameter sets the color value for the detail text label.

**[P] '@sTableViewCellRowHeight|<height>':** Set the table view row height (points).

**[P] '@TableViewCellPullToRefresh|<\*status>':** Add or remove "pull to refresh functionality".

- \*'0' = off (default).
- \*'1' = on.

**[P] '@startTableViewCellRefresh':** Start the table view refresh mode programmatically.

**[P] '@endTableViewCellRefresh':** End the table view refresh mode programmatically.

**[P] '@TableViewCellDeleteEnabled|<\*status>':** Add or remove "slide to delete functionality".

- \*'0' = off (default).
- \*'1' = on.

**[P] '@aTableViewLongPress|<duration>':** Add a table view long press handler. Valid duration values are from 0.1 to 10.0 seconds. When duration is set to 0 the long press handler will be removed. Refer to [tViewLongPress](#) for the callback message.

**[P] '@sTableViewImageProperties|<border width>|<optional:border color>|<optional:background color>|<\*optional:image justification>|<optional:image corner radius>':** Set table view image properties.

- \*'1' = left.
- \*'2' = right( default).

**[P] '@sTableViewMultiline|<number of label multilines>|<number of detail multilines>':** Set table view text multiline options.

**[P] '@sTableViewSeparatorColor|<color value>':** Set table view separator color, refer to [appendix C](#) for a list of possible color values.

**[P] '@storeTableViewData|<store id>':** Store all currently loaded table view data to the device disk storage. <store id> is a number between 1 and 100.

**[P] '@retrieveTableViewData|<store id>':** Retrieve table view data with <store id> and fill the table view with the retrieved data.



## Tcp socket connection commands (control system ⇒ AVcontrol)

<a href="#">@openTcpSocketConnection</a>	[P] Open tcp socket connection
<a href="#">@closeTcpSocketConnection</a>	[P] Close tcp socket connection
<a href="#">@sendTcpSocketString</a>	[P] Send tcp socket string
<a href="#">@registerTcpSocketString</a>	[P] Register tcp socket string
<a href="#">@aButtonString</a>	[P] Add a string to a button push
<a href="#">@aButtonStringOnRelease</a>	[P] Add a string to a button release
<a href="#">@aLevelString</a>	[P] Add a string to a level

**[P] '@openTcpSocketConnection|<identifier>|<host>|<port>':** Open a tcp socket connection. <identifier> is a (self-chosen) name that will be used to identify further tcp socket commands. <host> can refer to an IP address or a DNS qualified hostname and <port> refers to the tcp port to connect to.

**[P] '@closeTcpSocketConnection|<identifier>':** Close the connection identified by <identifier>.

**[P] '@sendTcpSocketString|<identifier>|<string>|<optional:repeat time value>':** Send a string to an open socket connection identified by <identifier>. Use percent encoding to encode special characters (for example a carriage return can be encoded using %0d). Encode hex values with '\x'. The optional repeat time value specifies a repeat time in seconds.

**[P] '@registerTcpSocketString|<identifier>|<string>|<optional:event number to fire when feedback has been received>':** Register a string for feedback recognition. This enabled 2-way tcp socket communication.

**[P] '@aButtonString|<tag>|<identifier>|<repeat time>|...':** Add a socket string to the a control push identified by <tag>. Repeat time is a number that specifies the number of times per second to send the string (0=once). You can add multiple strings by separating them with |. This will enable toggle functionality. Using tFeedback|<tag>|0 will reset the toggle.

**[P] '@aButtonStringOnRelease|<tag>|<string>':** Add a socket string to a control release identified by <tag>.

**[P] '@aLevelString|<tag>|<identifier>|<string>|<\*value type conversion>':** Add a socket string to a level identified by <tag>.

- \*'1' = level will be encoded to ASCII.
- \*'2' = level will be encoded to ASCII HEX representation.
- \*'3' = level will be encoded to ASCII using 2 digit zero padding.
- \*'4' = level will be encoded as a raw byte.

## Miscellaneous commands (control system ⇒ AVcontrol)

<a href="#">@setBackgroundPortrait</a>	Set background image for portrait mode
<a href="#">@setBackgroundLandscape</a>	Set background image for landscape mode
<a href="#">@setBackgroundColor</a>	<b>[TE]</b> Set background color ( <i>widget only</i> )
<a href="#">@aColor</a>	<b>[P] [TE]</b> Add a custom color
<a href="#">@sColorsBrightness</a>	<b>[TE]</b> Set the brightness for the built in (standard) colors
<a href="#">@sCornerRadiusGlobal</a>	<b>[TE]</b> Set global corner radius
<a href="#">@pSound</a>	Play a sound
<a href="#">@Vibrate</a>	<b>[TE]</b> Vibrate device
<a href="#">@Keyboard</a>	Pop up the device keyboard
<a href="#">@bLevel?</a>	Request battery level
<a href="#">@bState?</a>	Request battery state
<a href="#">@sPageAnimationDuration</a>	Set sub & full page animation duration
<a href="#">@sControlAnimationDuration</a>	<b>[TE]</b> Set control animation duration
<a href="#">@addNextControlToMenu</a>	Add next control to the menu
<a href="#">@addNextControlToMainView</a>	Add next control to the main view
<a href="#">@sBlinkRate</a>	Set blink button blink rate
<a href="#">@accelerometerOn</a>	<b>[P]</b> Start accelerometer notifications
<a href="#">@accelerometerOff</a>	<b>[P]</b> Stop accelerometer notifications
<a href="#">@flushDynamicImageCache</a>	<b>[P]</b> Clear the dynamic image cache
<a href="#">@setScreenBrightness</a>	<b>[P]</b> Set screen brightness
<a href="#">@screenBrightness?</a>	<b>[P]</b> Request screen brightness
<a href="#">@showStatusBar</a>	Show the device status bar
<a href="#">@hideStatusBar</a>	Hide the device status bar
<a href="#">@setStatusbarStyle</a>	Set the status bar style
<a href="#">@delayCommand</a>	Send a delayed command
<a href="#">@cancelDelayCommand</a>	Cancel a previously sent delayed command
<a href="#">@sPingTimeout</a>	<b>[TE]</b> Set ping timeout value
<a href="#">@setIdleTimeout</a>	Set idle timeout value
<a href="#">@sAlertColor</a>	<b>[P]</b> Customize the connection bar alert colors
<a href="#">@launchApplication</a>	<b>[P] [TE]</b> Launch external app (URL scheme support)
<a href="#">@reloadResources</a>	Reload resources from controller
<a href="#">@composeMail</a>	<b>[P]</b> Launch mail composer dialog
<a href="#">@lockApp</a>	<b>[P]</b> Lock the app in ASAM mode
<a href="#">@unlockApp</a>	<b>[P]</b> Unlock the app from ASAM mode
<a href="#">@systemLanguage?</a>	Request the device language
<a href="#">@systemDate?</a>	Request the device date and time
<a href="#">@appVersion?</a>	Request application version
<a href="#">@deviceName?</a>	Request device name
<a href="#">@touchID?</a>	Check if the device supports touchID authentication
<a href="#">@faceID?</a>	Check if the device supports faceID authentication
<a href="#">@disableIdleTimer</a>	Disables device idle timer (prevents sleep mode)

**`@setBackgroundPortrait|<image name>`**: Set the background image for portrait mode. Substitute <image name> with a [color value](#) for a solid colored background.

**`@setBackgroundLandscape|<image name>`**: Set the background image for landscape mode. Substitute <image name> with a [color value](#) for a solid colored background.

**[TE] `@setBackgroundColor|<color value>`**: Set the background color for Today Extension (widget).

**[P] [TE] `@aColor|<red>|<green>|<blue>|<optional:alpha>`**: Add a custom color value to the shortcut [color list](#). The new color index starts at index 35. Valid values are float values ranging from 0.0 to 1.0. Alpha value is optional.

**[TE] `@sColorsBrightness|<off colors brightness value>|<optional:on colors brightness value>`**: Set the brightness for the built in (standard) colors. Valid values are float values ranging from 0.0 to 1.0.

**[TE] `@sCornerRadiusGlobal|<radius>`**: Set a global rounded corner (mask) value. All interface elements following this command will have the corner radius set to the <radius> value.

**`@pSound|<\*sound name or ID>`**: play a sound. You can specify either a resource name (for instance: `@pSound|click.wav` or use a sound ID for [built in sounds](#).

- \*Refer to [appendix A](#) for a list of supported file formats. Maximum recommended sound duration is 30 seconds.

**[TE] `@Vibrate`**: Causes the device to vibrate shortly (*if the device supports this feature*).

**`@Keyboard|<\*keyboard type>|<optional:initial keyboard text>|<optional:private option>`**: Will show the keyboard. Each user entered character will immediately be sent to the control system as feedback. When an initial keyboard text is supplied the keyboard will automatically attach a text box for the entered text. If <YES> is supplied for the private parameter the text box will display dots.

- \*`0` = Default keyboard.
- \*`1` = ASCII keyboard.
- \*`2` = Numbers and punctuation keyboard.
- \*`3` = URL keyboard.
- \*`10` = Sentence capitalization ascii keyboard (private option not available).
- \*`11`, `12`, `13`, `20` = same as above except keyboard appearance is dark.

**`@bLevel?`**: Request the current battery level.

**`@bState?`**: Request the current battery/device state.

**'@sPageAnimationDuration|<duration>|<\*optional:direction>'**: Set the global page transition animation duration value. Valid duration values are 0.0 to 5.0 (in seconds). Default duration is 1.0. For some animations a direction can be set.

- **\*'1'** = From right.
- **\*'2'** = From left.
- **\*'3'** = From top.
- **\*'4'** = From bottom.

**[TE] '@sControlAnimationDuration|<duration>'**: Set the global control animation duration value. Valid duration values are 0.0 to 5.0 (in seconds). Default duration is 1.0.

**'@addNextControlToMenu'**: Will target the menu for the first command received after this one. It enables you to add custom user interface elements to the scroll-menu. It will disable this mode automatically after the first UI command is received. AVcontrol also deletes any custom UI elements in the scroll-menu upon rotation of the device.

**'@addNextControlToMainView'**: Will target the main view (this is the master for for all subpages) for the first command received after this one. It enables you to add custom user interface elements to the subpage that ' float ' above the subview. It will disable this mode automatically after the first UI command is received.

**'@sBlinkRate|<duration>'**: Set the global blink button blink rate. Valid duration values are 0.0 to 5.0 (in seconds). Default duration is 0.75.

**[P] '@accelerometerOn|<number of notifications per second><accuracy>'**: Enable notifications for the device accelerometer. Valid number of notifications per second are 1 to 20. Accuracy refers to the number of decimals returned (maximum 3 decimals).

**[P] '@accelerometerOff'**: Disable device accelerometer notifications.

**[P] '@flushDynamicImageCache'**: Empty the dynamic image cache. Every application startup clears the cache automatically.

**[P] '@setScreenBrightness|<brightness>'**: Set the screen brightness. Range is 0.0 to 1.0

**[P] '@screenBrightness?'**: Query the current screen brightness.

**'@showStatusBar'**: Show the device status bar. Regardless of this setting AVcontrol will always hide the status bar in landscape mode on the iPhone.

**'@hideStatusBar'**: Hide the device status bar (default application startup behavior).

**'@setStatusBarStyle|<\*value>'**: Set the status bar style (iOS 7.0 or later)

- **\*'0'** = Status bar default
- **\*'1'** = Status bar light content.

**`@delayCommand|<delay>|<command>`**: Send a command that executes after the delay specified in seconds (float value).

**`@cancelDelayCommand|<command>`**: Cancel a delayCommand when it has not yet fired. <command> serves as the identifier.

**[TE] `@sPingTimeout|<timeout value>`**: Set AVcontrol ping timeout value. Default value is 10.0 (seconds). Valid range is from 5.0 to 60.0 seconds. Experiment with longer timeout values when processor intensive operations are being performed. Note: this value also sets the Bring-Your-Own-Device timeout value.

**`@setIdleTimeout|<timeout value>|<optional:event number when user becomes idle>|<optional:event number when user becomes active>`**: Set a timeout for touch events, specify <0> for disabling the idle timer, this is the default behavior.

**[TE] `@sAlertColor|<background color value>|<text color value>`**: Customize the [color value](#) for the connection bar and tcp alert colors.

**[P][TE] `@launchApplication|<URL scheme>`**: Open an external application on the device.

**`@reloadResources`**: Reload resources from controller.

**[TE] `@composeMail|<recipient>|<subject>|<message>`**: Launch the mail composer dialog.

**[TE] `@lockApp`**: Lock the application in ASAM ([Autonomous Single App Mode](#)). For this command to work the device needs to be managed with a MDM server and AVcontrol must be whitelisted for ASAM mode.

**[TE] `@unlockApp`**: This command will release the app from ASAM mode.

**`@systemLanguage?`**: Query the current device language.

**`@systemDate?`**: Query the current device date and time.

**`@appVersion?`**: Query the application version number.

**`@deviceName?`**: Query the device name

**`@touchID?`**: Check if the device supports touchID.

**`@faceID?`**: Check if the device supports faceID.

**'@disableIdleTimer'**: Disables device idle timer (prevents sleep mode).

**'@Pong'**: Reply for a ping request sent from AVcontrol.

## Command feedback (AVcontrol ⇒ control system)

<a href="#">#appVersion</a>	Reports the application version
<a href="#">#Ping</a>	<b>[TE]</b> Ping request
<a href="#">#Device</a>	Report the device model
<a href="#">#DeviceName</a>	Report the device name
<a href="#">#touchIDAvailable</a>	Reports the availability of touchID
<a href="#">#faceIDAvailable</a>	Reports the availability of faceID
<a href="#">#authenticateUser</a>	Reports the result of a user authentication request
<a href="#">#LandscapeLeft</a>	Report landscape left orientation
<a href="#">#LandscapeRight</a>	Report landscape right orientation
<a href="#">#Portrait</a>	Report portrait orientation
<a href="#">#PortraitUpsideDown</a>	Report portrait upside down orientation
<a href="#">#appPrefersStatusBarHidden</a>	Report the status bar hidden preference setting
<a href="#">#screenBrightness</a>	Report the current screen brightness
<a href="#">#systemLanguage</a>	Report the current device language
<a href="#">#systemDate</a>	Report the current device date and time
<a href="#">#tableViewRowSelected</a>	A table view row was selected
<a href="#">#actionSheetSelected</a>	An action sheet button was selected
<a href="#">#tableViewLongPress</a>	Table view row touch event was held
<a href="#">#swipeGesture</a>	Swipe gesture detected
<a href="#">#swipeGesture List</a>	Swipe gesture for table view detected
<a href="#">#tapGesture</a>	Tap gesture detected
<a href="#">#panGestureStarted</a>	Pan gesture started
<a href="#">#panGesture</a>	Pan gesture detected
<a href="#">#panGestureEnded</a>	Pan gesture ended
<a href="#">#sLevel</a>	<b>[TE]</b> Level change report
<a href="#">#tPush</a>	<b>[TE]</b> Control was pushed
<a href="#">#tRelease</a>	<b>[TE]</b> Control was released
<a href="#">#iText</a>	Text feedback
<a href="#">#Keyboard</a>	Keyboard feedback
<a href="#">#switchEvent</a>	<b>[TE]</b> Switch state was changed
<a href="#">#stepperValue</a>	<b>[TE]</b> Stepper value report
<a href="#">#segControlSelected</a>	<b>[TE]</b> Segment from segmented control was selected
<a href="#">#pViewRowSelected</a>	An element from a picker view was selected
<a href="#">#alertViewSelected</a>	Alert view was selected
<a href="#">#dImageLoadingDone</a>	Dynamic image has finished loading
<a href="#">#webViewFinishedLoading</a>	Web view has finished loading
<a href="#">#tableViewIndexingDone</a>	Table view has finished indexing content
<a href="#">#tableViewRefresh</a>	User requested a table view refresh
<a href="#">#tableViewRowDeleted</a>	User deleted a table view row
<a href="#">#bLevel</a>	Reports battery level
<a href="#">#bState</a>	Reports battery state

<a href="#">#accelerometerUpdate</a>	Reports accelerometer values
<a href="#">#colorPickerColorSelected</a>	Reports the selected color picker color values
<a href="#">#menuItemsAdded</a>	Reports that all menu items have been added
<a href="#">#subPageWillBeDismissed</a>	Reports when the subpage will be removed
<a href="#">#showPageAnimationFinished</a>	Reports a completed subpage animation
<a href="#">#showFullPageAnimationFinished</a>	Reports a completed full page animation
<a href="#">#WidgetSize</a>	<b>[TE]</b> Reports the widget size ( <i>widget only</i> )
<a href="#">#InterfaceStyle</a>	<b>[TE]</b> Reports the interface style ( <i>widget only</i> )
<a href="#">#userIsIdle</a>	Reports user inactivity
<a href="#">#userIsActive</a>	Reports user activity
<a href="#">#Error</a>	<b>[TE]</b> Reports an error

**`#appVersion=<version>`**: Reply to an application version inquiry.

**[TE] `#Ping`** : Ping request from AVcontrol to control system, reply with **`@Pong`**. A timeout of 10 seconds is implemented in AVcontrol where the app will automatically disconnect if no answer is received. Implementation of a timeout using the same timeout duration in the control system program is recommended.

**`#Device|<model>`**: Reports the device model (i.e. 'iPhone 12/13 mini').

**`#DeviceName|<name>`**: Reports the device name (i.e. 'John's iPhone'):

**`#touchIDAvailable|<YES/NO>`**: Reports if touchID is available on the device.

**`#faceIDAvailable|<YES/NO>`**: Reports if faceID is available on the device (>iOS11).

**`#authenticateUser|<Authenticated/Failed/Unavailable>`**: Reports the result of a user authentication request.

**`#LandscapeLeft`** : Device was rotated to landscape orientation (home button right).

**`#LandscapeRight`** : Device was rotated to landscape orientation (home button left).

**`#Portrait`** : Device was rotated to portrait orientation (home button below).

**`#PortraitUpsideDown`** : Device was rotated to portrait upside down orientation (home button above).

**`#appPrefersStatusBarHidden|<YES|NO>`** : Reports the status bar hidden preference setting.



**`#screenBrightness|<brightness>`**: Reply to a screen brightness inquiry. Value range is 0.0 (0%) to 1.0 (100%).

**`#systemLanguage|<language>`**: Reply to a device language inquiry.

**`#systemDate|<brightness>`**: Reply to a device date inquiry.

**`#tableViewRowSelected|<row>`**: Table (list) view row was selected by the user.

**`#actionSheetSelected|buttonIndex=<index>`**: An action sheet button was selected, indices are zero based.

**`#tableViewLongPress|section=<section>|index=<index>`**: Table (list) view row touch even was held. Section parameter will be sent if the table view was indexed.

**`#swipeGesture|<tag>|<\*direction>`**: A swipe gesture was executed on control identified by tag number.

- **\*'direction=left'** = User swiped left.
- **\*'direction=right'** = User swiped right.
- **\*'direction=up'** = User swiped up.
- **\*'direction=down'** = User swiped down.

**`#swipeGesture|tag=list|<\*direction>`**: A swipe gesture was executed on table (list) view.

- **\*'direction=left'** = User swiped left.
- **\*'direction=right'** = User swiped right.
- **\*'direction=up'** = User swiped up.
- **\*'direction=down'** = User swiped down.

**`#tapGesture|<tag>|<number of taps>`**: A tap gesture was executed on control identified by tag number.

**`#panGestureStarted|<tag>`**: A pan gesture was started on control identified by tag number.

**`#panGesture|<tag>|<x location>|<y location>|<x velocity>|<y velocity>`**: A pan gesture was detected. This gesture reports the x-y location and velocity of the gesture.

**`#panGestureEnded|<tag>`**: A pan gesture ended on control identified by tag number.

**[TE] `#sLevel|<tag>|<value>`**: User changed level on an active bar graph (slider).

**[TE] `#tPush|<tag>`**: Control identified by tag number was pushed.

**[TE] `#tRelease|<tag>`**: Control identified by tag number was released.

**`#iText|<tag>|<text>`**: User text input on control identified by tag number.

**`#Keyboard|<character>`**: User keyboard input, 1 character per message.

**[TE] `#switchEvent|<tag>|<value>`**: User changed switch value.

**[TE] `#stepperValue|<tag>|<value>`**: User changed a stepper value.

**[TE] `#segControlSelected|tag=<tag>|index=<index>`**: A segment from a segmented control identified by tag number was selected. <index> represents the specific segment that was selected and starts at 0 for the leftmost segment.

**`#pViewRowSelected|<tag>|<index>|contents=<content>`**: An element from a picker view identified by tag number was selected. <index> represents the specific element that was selected and starts at 0 for the first element. <content> represents the text in the row that was selected.

**`#alertViewSelected|tag=<tag>|buttonIndex=<button index>`**: User selected an alert button. Button indices start with 0 in the order that they were added.

**`#dImageLoadingDone|tag=<tag>`**: The application has finished loading the dynamic image on the control identified by tag.

**`#webViewFinishedLoading|tag=<tag>|url=<url>`**: The application has finished loading the web view on the control identified by tag.

**`#tableViewIndexingDone`**: The application has finished indexing the table view. Tables with a lot of content can take some time to analyze.

**`#tableViewRefresh`**: User requested a table view refresh. The control system should perform the necessary actions and also [end the table view refresh mode](#).

**`#tableViewRowDeleted|<tag>`**: User has deleted a row by using "slide to delete".

**`#bLevel|<bLevel>`**: Feedback on the current battery level (5% accuracy in iOS8.0 or lower, 1% accuracy in versions 8.1 or higher) where <bLevel> is a string representation of a float value (between 0 and 1).

**`#bState|<\*state>`**: Feedback on the current battery/device state.

- **\*'0'** = Battery/device state 'Unknown'.
- **\*'1'** = Battery/device state 'Unplugged'.
- **\*'2'** = Battery/device state 'Charging'.
- **\*'3'** = Battery/device state 'Full'.

**`#accelerometerUpdate|x=<X>|y=<Y>|z=<Z>`**: Current device accelerometer reading. X,Y and X values will be in the range from -1.0 to 1.0. Accuracy depends on the accuracy requested in the [accelerometerOn](#) command.

**`#colorPickerColorSelected|R=<r>|G=<g>|B=<b>|Alpha=<a>`** : Reports the color that the user selected using the sColorPicker command. Values are float values ranging from 0.0xxxx to 1.0xxxx. If the color picker was called without support for alpha the alpha value will default to 1.

**`#menuItemsAdded`** : Reports that all menu items have been added so that they are ready for further customization. This will be reported when an [initMenu](#) or [initUpdatedMenu](#) command has finished displaying the items.

**`#subPageWillBeDismissed`** : Reports when the subpage will be removed, for example when the user has touched outside the page to return to the updated menu.

**`#showPageAnimationFinished|originView=<tag>`** : Reports a completed subpage animation. <tag> refers to the button that initiated the animation.

**`#showFullPageAnimationFinished|originView=<tag>`** : Reports a completed full page animation. <tag> refers to the button that initiated the animation.

**[TE] `#WidgetSize|<Size>`** : Reports the widget size. Automatically reports upon successful connection (*widget only*).

- **\*'Small'** = iPhone portrait/landscape + iPad landscape = 320 x 220 (points).
- **\*'Normal'** = iPad portrait = 480 x 220 (points).

**[TE] `#InterfaceStyle|<Mode>`** : Reports the interface style. Automatically reports upon successful connection and when user select/deselects dark mode (*widget only*).

- **\*'Dark'** = Dark mode is enabled.
- **\*'Light'** = Dark mode is disabled.

**`#userIsIdle`** : Reports user inactivity. This feedback will be sent after the idle timer has ended. The timer value can be set user the [set idle timeout command](#).

**`#userIsActive`** : Reports user activity.

**[TE] `#Error|<\*error message>`** : Error message.

- **\*Error message** (will include the original command that generated the error).

## Watch API (control system ⇒ AVcontrol)

<a href="#">@Pong</a>	[P] Reply to a ping request
<a href="#">@initPage</a>	[P] Initialize a new subpage
<a href="#">@showPage</a>	[P] Show a new page
<a href="#">@addButton</a>	[P] Add a button row
<a href="#">@addDualButton</a>	[P] Add a dual button row
<a href="#">@addLabel</a>	[P] Add a text label row
<a href="#">@addSlider</a>	[P] Add a slider row
<a href="#">@sLevel</a>	[P] Set slider level
<a href="#">@addSwitch</a>	[P] Add a switch row
<a href="#">@sSwitch</a>	[P] Set switch state
<a href="#">@addImage</a>	[P] Add an image row
<a href="#">@addSeparator</a>	[P] Add a separator row
<a href="#">@addMenuItem</a>	[P] Add a menu item row
<a href="#">@sText</a>	[P] Set text for button, label, switch or menu item
<a href="#">@aView</a>	[P] Display an alert that presents options to the user
<a href="#">@rControl</a>	[P] Remove a row
<a href="#">@removeAllControls</a>	[P] Remove all rows
<a href="#">@sAlpha</a>	[P] Set control alpha
<a href="#">@tFeedback</a>	[P] Set feedback
<a href="#">@tFont</a>	[P] Set font
<a href="#">@aColor</a>	[P] Add a custom color
<a href="#">@sColorsBrightness</a>	[P] Set the brightness for the built in (standard) colors
<a href="#">@setTitle</a>	[P] Set the app title
<a href="#">@playHaptic</a>	[P] Play a haptic (vibrate) pattern
<a href="#">@sPingTimeout</a>	[P] Set ping timeout value

[P] **@Pong**: Reply for a ping request sent from AVcontrol.

[P] **@initPage**: Start a new subpage (put all UI control commands in between 'initPage' and 'showPage').

[P] **@showPage**: Show a previously initialized page with a fade in animation.

[P] **@addButton|<tag>|<title>|<\*image name off state>|<\*image name on state>|<\*\*optional:text colorValue>**: Add a channel (feedback from control system) button.

- \*Substitute <image name off state> and <image name on state> with a [color value](#) for a solid colored button.
- \*\*Default colorValue = whiteColor, refer to [appendix C](#) for a list of possible values.

[P] **@addDualButton|<tag-1>|<title-1>|<\*image name off state-1>|<\*image name on state-1>|<tag-2>|<title-2>|<\*image name off state-2>|<\*image name on state-2>**: Add a dual channel (feedback from control system) button. Dual buttons will be positioned next to each other.

- \*Substitute <image name off state> and <image name on state> with a [color value](#) for a solid colored button.

**[P] '@addLabel|<tag>|<title>|<\*text color>|<\*\*text options>|<\*\*\*optional:background color>':** add a text label.

- \*Refer to [appendix C](#) for a list of possible values.
- \*\*'0' = default..
- \*\*'1' = underlined.
- \*\*'2' = italic.
- \*\*'3' = underlined+italic.
- \*\*\*Default colorValue = clearColor, refer to [appendix C](#) for a list of possible values.

**[P] '@addSlider|<tag>|<number of steps from 0-100>|<initial value>|<\*optional:color value>':** Add an active slider.

- \*Refer to [appendix C](#) for a list of possible values.

**[P] '@sLevel|<tag>|<level>':** Set the level of a slider identified by tag number.

**[P] '@addSwitch|<tag>|<title>|<\*initial state>|<\*\*\*optional:color value>|<\*\*\*optional:text color value>':** Add a switch control.

- \*'0' = state off.
- \*'1' = state on.
- \*\*refer to [appendix C](#) for a list of possible values.

**[P] '@sSwitch|<tag>|<\*state>':** Set the state of a switch identified by tag number.

- \*'0' = state off.
- \*'1' = state on.

**[P] '@addImage|<tag>|<url>':** Add a dynamic image.

- Image gets scaled down (keeping aspect ratio) if the width is larger than the screen real estate.

**[P] '@addSeparator|<tag>|<color value>|<optional:row height>':** Add a (line) separator.

**[P] '@addMenuItem|<tag>|<\*text color>|<\*\*text options>|<\*\*\*optional:background color>|<optional:height>':** Add a menu item.

- \*Refer to [appendix C](#) for a list of possible values.
- \*\*'0' = default.
- \*\*'1' = underlined.
- \*\*'2' = italic.
- \*\*'3' = underlined+italic.
- \*\*\*Default colorValue = clearColor, refer to [appendix C](#) for a list of possible values.

**[P] '@sText|<tag>|<text>':** Set the text of the button, label, switch, or menu item identified by tag number.

**[P] '@aView|<tag>|<title>|<message>|<button 1 text>|<button 2 text>|etc...':** Display an alert that presents options to the user. Use as many button text parameters as needed. <tag> is limited from 1 up to 100.

**[P] '@rControl|<tag>|<optional:animated value>':** Remove control identified by tag number.

**[P] '@removeAllControls|<tag>|<optional:animated value>':** Remove all controls (effectively creating a new blank page).

**[P] '@sAlpha|<tag>|<value>':** Set the alpha value on a control identified by tag. Valid values are float values ranging from 0.0 to 1.0.

**[P] '@tFeedback|<tag>|<\*state>':** Set the button on/off state of a button identified by tag.

- \*'0' = state off.
- \*'1' = state on.

**[P] '@tFont|<tag>|<font name>|<optional:size>':** Set the font of the control identified by tag number. For v2.6 the only built in font for the Watch App is "Guifx V2 Transports". refer to [appendix D](#) for a detailed keyboard map.

**[P] '@aColor|<red>|<green>|<blue>|<optional:alpha>':** Add a custom color value to the shortcut [color list](#). The new color index starts at index 35. Valid values are float values ranging from 0.0 to 1.0. Alpha value is optional.

**[P] '@sColorsBrightness|<off colors brightness value>|<optional:on colors brightness value>':** Set the brightness for the built in (standard) colors. Valid values are float values ranging from 0.0 to 1.0.

**[P] '@setTitle|<title>':** Set the app title (shown in the top left corner of the Watch screen).

**[P] '@playHaptic|<pattern number>':** Play a haptic (vibrate) pattern. Pattern enumeration start with 0, refer to [this](#) page for a list of available haptic types.

**[P] '@sPingTimeout|<timeout value>':** Set AVcontrol ping timeout value. Default value is 10.0 (seconds). Valid range is from 5.0 to 60.0 seconds. Experiment with longer timeout values when processor intensive operations are being performed.

## Watch API Command feedback (AVcontrol ⇒ control system)

<a href="#">#Ping</a>	Ping request
<a href="#">#Device</a>	Report the device model
<a href="#">#tTap</a>	A control was tapped
<a href="#">#sLevel</a>	Level change report
<a href="#">#switchEvent</a>	Switch state was changed
<a href="#">#Error</a>	Reports an error

**`#Ping`** : Ping request from AVcontrol to control system, reply with **`@Pong`**. A timeout of 10 seconds is implemented in AVcontrol where the app will automatically disconnect if no answer is received. Implementation of a timeout using the same timeout duration in the control system's program is recommended.

**`#Device|<model>`**: Reports the Watch device model. Possible values are:

- 38mm.
- 40mm.
- 41mm.
- 42mm.
- 44mm.
- 45mm.
- 49mm.

**`#tTap|<row>`**: A control was tapped by the user. The Watch App does not support separate push and release events, this is a limitation of the Watch OS.

**`#sLevel|<tag>|<value>`**: User changed level on a slider.

**`#switchEvent|<tag>|<value>`**: User changed switch value.

**`#Error|<\*error message>`**: Error message.

- \*Error message (will include the original command that generated the error).

## Appendix A • Supported resource file formats

AVcontrol supports the following image and sound file formats;

- .tiff & .tif – Tagged Image File Format.
- .jpg & .jpeg – Joint Photographic Experts Group.
- .gif – Graphic Interchange Format.
- .png – Portable Network Graphic (**recommended image format**).
- .bmp & .f – Windows Bitmap Format.
- .ico – Windows Icon Format.
- .cur – Windows Cursor.
- .xbm – Xwindow Bitmap.
- .caf – Core Audio File.
- .aif – Audio Interchange File.
- .wav – Waveform Audio Fil (**recommended sound format**).



## Appendix B • background image resolutions

Portrait image name	Pixel width	Pixel height
bg_portrait@2x.png	640	960
bg_portrait~ipad.png	768	1024
bg_portrait@2x~ipad.png	1536	2048
bg_portrait-810h@2x.png	1620	2160
bg_portrait-834-1h@2x.png	1668	2224
bg_portrait-834-2h@2x.png	1668	2388
bg_portrait-1024h@2x.png	2048	2732
bg_portrait-568h@2x.png	640	1136
bg_portrait-667h@2x.png	750	1338
bg_portrait-736h@3x.png	1242	2208
bg_portrait-744h@2x.png	1488	2266
bg_portrait-812h@3x.png	1125	2436
bg_portrait-844h@3x.png	1170	2532
bg_portrait-852h@3x.png	1179	2556
bg_portrait-896h@2x.png	826	1792
bg_portrait-780h@3x.png	1080	2340
bg_portrait-896h@3x.png	1242	2688
bg_portrait-926h@3x.png	1284	2778
bg_portrait-932h@3x.png	1290	2796
Landscape image name	Pixel width	Pixel height
bg_landscape@2x.png	960	640
bg_landscape~ipad.png	1024	768
bg_landscape@2x~ipad.png	1024	1536
bg_landscape-810h@2x.png	2160	1620
bg_landscape-834-1h@2x.png	2224	1668
bg_landscape-834-2h@2x.png	2388	1668
bg_landscape-1024h@2x.png	2732	2048
bg_landscape-568h@2x~ipad.png	1136	640
bg_landscape-667h@2x~ipad.png	1338	750
bg_landscape-736h@3x.png	2208	1242
bg_landscape-744h@2x.png	2266	1488
bg_landscape-780h@3x.png	2340	1080
bg_landscape-812h@3x.png	2436	1125
bg_landscape-844h@3x.png	2532	1170
bg_landscape-852h@3x.png	2556	1179
bg_landscape-896h@2x.png	1792	826
bg_landscape-896h@3x.png	2688	1242
bg_landscape-926h@3x.png	2778	1284
bg_landscape-932h@3x.png	2796	1290






















































































Appendix C • color value table












































































0	1	2	3	4	5	6	7
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34 transparant	35					

Appendix D • symbol font keyboard map



## Appendix E • icon number reference

100 	117 	308 	401 	514 
101 	118 	309 	402 	515 
102 	119 	310 	403 	516 
103 	120 	311 	500 	517 
104 	200 	312 	501 	518 
105 	201 	313 	502 	519 
106 	202 	314 	503 	520 
107 	203 	315 	504 	521 
108 	204 	316 	505 	522 
109 	300 	317 	506 	600 
110 	301 	318 	507 	601 
111 	302 	319 	508 	602 
112 	303 	320 	509 	603 
113 	304 	321 	510 	604 
114 	305 	322 	511 	605 
115 	306 	323 	512 	606 
116 	307 	400 	513 	607 

608		707		901		919		937	
609		708		902		920		938	
610		709		903		921		939	
611		710		904		922		940	
612		711		905		923			
613		712		906		924			
614		713		907		925			
615		800		908		926			
616		801		909		927			
617		802		910		928			
618		803		911		929			
700		804		912		930			
701		805		913		931			
702		806		914		932			
703		807		915		933			
704		808		916		934			
705		809		917		935			
706		900		918		936	