ADVANCED
NETWORK DEVICES

Inline Text Message Markup

Version 1.5

5/6/2025

3820 NORTH VENTURA DR.

ARLINGTON HEIGHTS, IL 60004

U.S.A

PROPRIETARY NOTICE AND LIABILITY DISCLAIMER

The information disclosed in this document, including all designs and related materials, is the valuable property of Digital Advanced Network Devices and/or its licensors. Advanced Network Devices and/or its licensors, as appropriate, reserve all patent, copyright and other proprietary rights to this document, including all design, manufacturing, reproduction, use, and sales rights thereto, except to the extent said rights are expressly granted to others.

The Advanced Network Devices product(s) discussed in this document are warranted in accordance with the terms of the Warranty Statement accompanying each product. However, actual performance of each product is dependent upon factors such as system configuration, customer data, and operator control. Since implementation by customers of each product may vary, the suitability of specific product configurations and applications must be determined by the customer and is not warranted by Advanced Network Devices.

To allow for design and specification improvements, the information in this document is subject to change at any time, without notice. Reproduction of this document or portions thereof without prior written approval of Advanced Network Devices is prohibited.

Static Electric Warning

TROUBLESHOOTING AND ADDITIONAL RESOURCES

User Support:	https://www.anetd.com/user-support/
Technical Resources:	https://www.anetd.com/user-support/technical-resources/
AND Legal Disclaimer:	https://www.anetd.com/legal

OVERVIEW

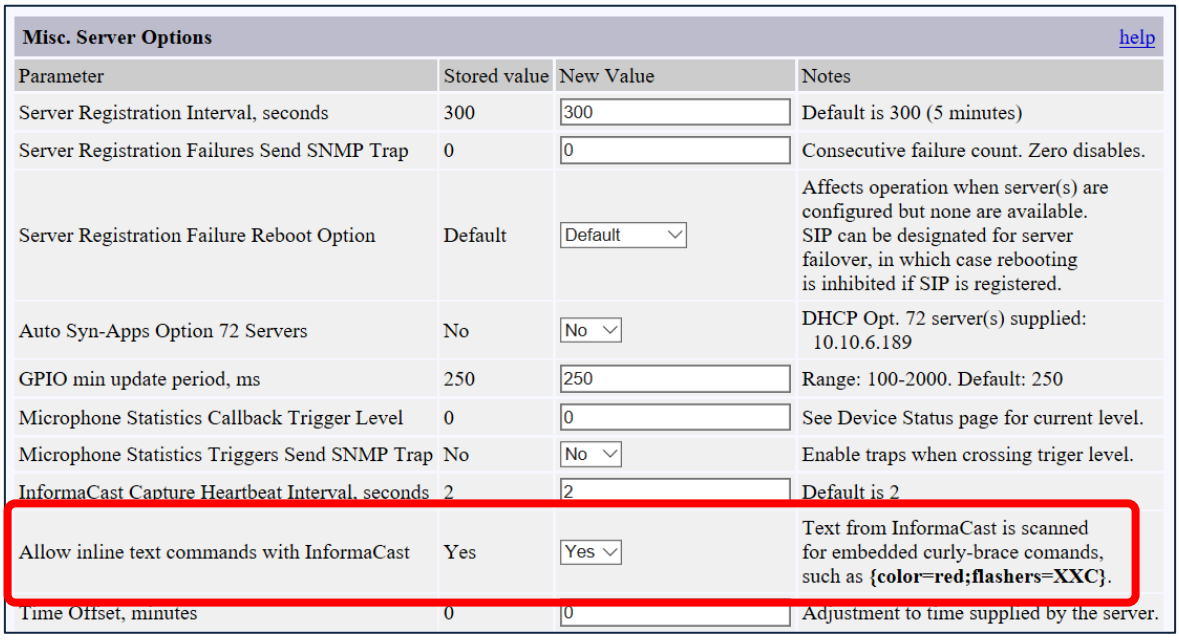
AND display devices support inline text message markup functionality, which makes it possible to define text and LED flasher behavior when setting up messages within InformaCast and other compatible third-party software. This capability allows the user to leverage more from AND devices, such as changing text fonts, colors, shadows, and flashing, as well as sending messages, images, and controlling accompanying LED flashers (if available on the device).

This document provides steps on how to setup the device, instructions to configure these parameters within the message text, as well as a list of supported parameters.

DEVICE SETUP

Before you can use inline markup with Singlewire's InformaCast Messages, you must first enable each device to allow inline text commands.

If using the device's web page interface, go to **Device Settings Servers** for the *Misc Server Options* section. Set the parameter *Allow inline text commands with InformaCast* to "Yes".



Misc. Server Options help

Parameter	Stored value	New Value	Notes
Server Registration Interval, seconds	300	300	Default is 300 (5 minutes)
Server Registration Failures Send SNMP Trap	0	0	Consecutive failure count. Zero disables.
Server Registration Failure Reboot Option	Default	Default	Affects operation when server(s) are configured but none are available. SIP can be designated for server failover, in which case rebooting is inhibited if SIP is registered.
Auto Syn-Apps Option 72 Servers	No	No	DHCP Opt. 72 server(s) supplied: 10.10.6.189
GPIO min update period, ms	250	250	Range: 100-2000. Default: 250
Microphone Statistics Callback Trigger Level	0	0	See Device Status page for current level.
Microphone Statistics Triggers Send SNMP Trap	No	No	Enable traps when crossing triger level.
InformaCast Capture Heartbeat Interval, seconds	2	2	Default is 2
Allow inline text commands with InformaCast	Yes	Yes	Text from InformaCast is scanned for embedded curly-brace comands, such as **{color=red;flashers=XXC}**.
Time Offset, minutes	0	0	Adjustment to time supplied by the server.

If using a configuration file, add the following option to the display tag:

```
<display
  allow_inline_commands_shorttext="1"
/>
```

SENDING TEXT TO AND DEVICES

Once the AND device or group of AND devices allows inline text markup, you can add parameters, in-line with the text message, to control the behavior of that text, display on-board pixmaps, images, or trigger any LED flashers on the targeted device(s). The device(s) will process these changes via one or more name-value pairs within curly braces "{}". In InformaCast applications, this markup should be added to the *Short Text* field of the InformaCast Message. See the **APPENDIX** for a full list of parameters and their function.

Syntax to add:

```
{parameter1=value;parameter2=value}
```

Example:

```
{color=green;bgcolor=red}Display red text with a green
background.
```

You can modify the font color and effects per sentence, phrase, word, or even individual characters, if desired. To return to default colors and effects within the text, set the parameter value(s) to a blank value, or specify *default*.

Example:

```
Display {color=red;bgcolor=green}red text. Now display the
{color=default;bgcolor=}default color.
```

Note: The following parameters will not work per-character. They only apply to the whole message. This functionality includes the LED flasher behavior, which will activate only for the duration of the displayed message.

flashers	*flash*
flashers_b	*flash_dc*
loops	*scroll*
speed	*splitting*
font	*autosplit*
shadloc	*still_ms*
imageX	

APPENDIX

Parameter	Default	Description
loops	1	The number of times to display the message. Zero means continual. Note: Available in firmware release 1.6.0002 or later.
speed	5	The scroll speed. Note: Available in firmware release 1.6.0002 or later. Range: 1 (slowest) to 10 (fastest).
pause	0	Enables a single text message to break into separate messages. Set {pause=x} within the text message to indicate where the text should break, and the display will pause x seconds before continuing with the next part of the text. Note the clock will be displayed during this pausing period. Example: `Display some text. {pause=3} Then display this text 3 seconds later.` Include {pause=} or {pause=0} to separate the two parts of the message by the width of the display, that is, the next part of the message will not display until the previous part scrolls off the display.
font	*n/a*	The message font type (from the following options): Arial Bold — arial_bold Arial (Larger size) — arial_huge Dotum — dotum Dotum Bold — dotum_bold Dotum (Larger size) — dotum_huge Dotum Bold (Larger) — dotum_bold_huge Small font* — and_8high Smaller font* — and_7high Tiny font* — and_5high ** These fonts support 2-line mode (clock + text).*

scroll	*horizontal*	Specifies the direction of scrolling: left, right, horizontal, up, down, or still. The default setting, horizontal, will scroll the message left or right based on the character set in use. For example, English characters will scroll left (from right to left), whereas Arabic and Hebrew characters will scroll right (from left to right). **Scrolling Description** — **Scroll string** Horizontal, left or right depending on character set — horizontal, horiz Left, from right to left — left Right, from left to right — right Up, from bottom to top — up Down, from top to bottom — down Still, no scrolling — still Note: Available in firmware release 1.6.0003 or later.
splitting	*0 (off)*	Specifies how to display static text. If this value is 0, the text is scrolled, normally. Other values display the text in pieces, statically on the screen. The *scroll* parameter should be un-specified or set to still. Valid non-zero values are: 1: displays a single line of text at once 2: displays two lines of text at once 3: displays three lines of text at once 32: displays two lines of text at once, with the top one smaller 23: displays two lines of text at once, with the bottom smaller 20: displays two lines of text at once, both using a small font. If this parameter is set to a valid non-zero value, it will split the message based on either using the pipe characters, "|", in the message as line separators, or automatically, if the *autosplit* parameter is set to 1. Note: Available in firmware release 1.6.0003 or later.
autosplit	*0 (off)*	Specifies whether the text message should be split automatically. If this is off, text is split with the pipe character, "|". If this is on, "1", the text is split automatically so that as many words as possible may be displayed on one line. This applies to the *still*, *up* and *down* scroll modes. Note: Available in firmware release 1.6.0003 or later.
still_ms	*1000*	In *still* scroll mode, this specifies the duration to display each part of the static text. Note: Available in firmware release 1.6.0003 or later.

<table>
<tr><td>color</td><td>n/a</td><td>The foreground message color (for dual-color displays). Specify a text string, or use a 4-bit hexadecimal number from the list below, such as <code>{color=green}</code> or <code>{color=c}</code>.

<table>
<tr><th>Hexadecimal</th><th>Color</th><th>Green Level</th><th>Red Level</th></tr>
<tr><td>0</td><td>Black</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>Cranberry</td><td>0</td><td>1</td></tr>
<tr><td>2</td><td>Cherry</td><td>0</td><td>2</td></tr>
<tr><td>3</td><td>Red</td><td>0</td><td>3</td></tr>
<tr><td>4</td><td>Hunter</td><td>1</td><td>0</td></tr>
<tr><td>5</td><td>Sienna</td><td>1</td><td>1</td></tr>
<tr><td>6</td><td>Terracotta</td><td>1</td><td>2</td></tr>
<tr><td>7</td><td>Vermillion</td><td>1</td><td>3</td></tr>
<tr><td>8</td><td>Olive</td><td>2</td><td>0</td></tr>
<tr><td>9</td><td>Tan</td><td>2</td><td>1</td></tr>
<tr><td>a</td><td>Ochre</td><td>2</td><td>2</td></tr>
<tr><td>b</td><td>Pumpkin</td><td>2</td><td>3</td></tr>
<tr><td>c</td><td>Green</td><td>3</td><td>0</td></tr>
<tr><td>d</td><td>Yellow</td><td>3</td><td>1</td></tr>
<tr><td>e</td><td>Gold</td><td>3</td><td>2</td></tr>
<tr><td>f</td><td>Orange</td><td>3</td><td>3</td></tr>
</table></td></tr>
<tr><td>bgcolor</td><td>black</td><td>The background message color (for dual-color displays). Specify a text string or the 4-bit hexadecimal number from the color chart, such as <code>{bgcolor=red}</code> or <code>{bgcolor=3}</code>.</td></tr>
<tr><td>shadcolor</td><td>(not drawn)</td><td>The font shadow color (for dual-color displays). Specify a text string or 4-bit hexadecimal number from the color chart above, such as <code>{shadcolor=black}</code> or <code>{shadcolor=0}</code>. Specify both shadcolor and shadloc, or no shadow will display.</td></tr>
<tr><td>shadloc</td><td>(no shadow)</td><td>A one- to four-character string that specifies the location of the text shadow, in relation to the foreground character. Typical values (not case-sensitive):

U — up
UR — up and to the right
R — right
DR — down and to the right
D — down
DL — down and to the left
L — left
UL — up and to the left
O or UDLR — fully outline the complete perimeter of the character

Note: Specify both shadcolor and shadloc, or no shadow will appear.</td></tr>
</table>

flash	0 (off)	The number of milliseconds of each flashing period. Special case: Values of 1-10 will flash the text the specified number of times per second, (e.g., flash=3 will flash the text 3 times per second).
flash_dc	50	The duty cycle of the flashing text. Range: 0 to 100.
flash_fg	*bgcolor*	The color of the foreground text during the flashing period. If no value is specified, it will match the *bgcolor* value.
flash_bg	*bgcolor*	The color of the background during the flashing period. If no value is specified, it will match the *bgcolor* value.
flash_shad	*bgcolor*	The color of the shadow during the flashing period. If no value specified, it will match the *bgcolor* value.
flashers	*n/a*	A three-character field, controlling the state of the left, middle, and right LED's respectively: 0 or O off S slow blink (200ms on, 800ms off) F fast blink (200ms on, 300ms off) 1 or C on x or X indicates not to change the state 2, 3, … 9 a 2-9 second flashing period The on-time remains 200ms for all cases. Note: Available in firmware release 1.6.0002 or later. Specific to IPCSHD-MB model: set color of LED following the state setting above with /color or /hex code: 1/red or 1/#FF0000 on/red F/red or F/#FF0000 fast blink/red S/red or S/#FF0000 slow blink/red
flashers_b	*1*	The brightness level of the flashers. Range: 0 (off) to 100 (brightest) This parameter can also accept the following text strings: "dim" equivalent to 50 "full" or "bright" equivalent to 100 Note: Available in firmware release 1.6.0002 or later.
messagename	*n/a*	Filename of an on-board message to display. See App Note 50: OnBoard Messages for more details. Note: Available in firmware release 1.6.0002 or later.

imageX	*n/a*	Filename of the image to display, supported on HD LCD devices. Image can be in the device's onboard file system or on a hosted server. Use the *Device Settings->Onboard* device menu to upload images.
		Onboard file example:
		`{image1=Sunrise.png}`
		External file example:
		`{image1=http://10.10.6.30:8081/InformaCast/resources/InformaCastHD.png}`

timer_time	*n/a*	The expiration time of a countdown timer to display expressed in ***your local time*** using the 24-hour format. Enter the time in this format: **Year_Month_Day_Hour_Minute_Second** For example, "2030_01_01_14_30_00" represents January 1st, 2030 at 2:30 PM in your local time.
timer_utctime	*n/a*	The expiration time of a countdown timer to display expressed in ***Coordinated Universal Time (UTC)*** using the 24-hour format. Enter the time in this format: **Year_Month_Day_Hour_Minute_Second** For example, "2030_01_01_14_30_00" represents January 1st, 2030 at 2:30 PM, UTC.
timer_secs	*n/a*	A countdown timer that starts from a specified duration (in seconds) and counts down to zero. For example, setting "timer_secs=30" will initiate a countdown from 30 seconds to 0.

timer_font	*n/a*	The timer font type (has the following options): Arial Bold — arial_bold Arial (Larger size) — arial_huge Dotum — dotum Dotum Bold — dotum_bold Dotum (Larger size) — dotum_huge Dotum Bold (Larger) — dotum_bold_huge Small font* — and_8high* Smaller font* — and_7high* Tiny font* — and_5high* ** These fonts support 2-line mode (clock + timer).*
timer_critsecs	*n/a*	Time string that specifies the duration from a countdown timer's expiration to enter the critical time period, during which the timer color can change, per the *timer_critcolor* parameter, if specified.
timer_lingerms	*3000*	The amount of time, in milliseconds, to display "00:00" at the end of a countdown timer before the display returns to the original clock time.
timer_prio	*50*	Specifies the timer priority (1-100). If the device is displaying a message, the timer will be displayed if its priority is less than or equal to the playing message's priority. If the incoming message has a priority greater than the playing message's priority, it will be discarded.
timer_title	*n/a*	Specifies static text to display above the countdown timer. Requires a *timer_font* that supports 2-line mode. ** Supported on LED display models only.*
timer_titlecolor	*black*	refers to the feature where the color of the timer's title text changes dynamically based on specific events or stages of the countdown, just like the color changes in **timer_color**.

timer_titlebgcolor	*black*	Timer Title background color changes based on the selection of a specified color, similar to how timer_color works for the countdown text itself.
timer_color	*black*	The timer color feature involves changing the color of the timer display from its default color to one of the specified colors as the countdown progresses or at a specific event (like time expiring). The available colors to choose from are: • Black • Cranberry • Cherry Red • Hunter • Sienna • Terracotta • Vermillion • Olive • Tan • Ochre • Pumpkin • Green • Yellow • Gold • Orange
timer_critcolor	*black*	Similar to time_color.