CONFIGURATION & PROGRAMMING MANUAL

VERSION 1.2

VARIA TOUCH PANELS

VARIA SL50	5.5” ULTRA-SLIM WALL MOUNT TOUCH PANEL
VARIA SL80	8” ULTRA-SLIM WALL MOUNT TOUCH PANEL
VARIA 80	8” TOUCH PANEL
VARIA 100	10.1” TOUCH PANEL
VARIA 100N	10.1” TOUCH PANEL (NO-COMM)
VARIA 150	15.6” TOUCH PANEL
VARIA 150N	15.6” TOUCH PANEL (NO-COMM)

AV FOR AN IT WORLD®

IMPORTANT SAFETY INSTRUCTIONS

1. READ these instructions.
2. KEEP these instructions.
3. HEED all warnings.
4. FOLLOW all instructions.
5. DO NOT use this apparatus near water.
6. CLEAN ONLY with dry cloth.
7. DO NOT block any ventilation openings. Install in accordance with the manufacturer's instructions.
8. DO NOT install near any heat sources such as radiators, heat registers, stoves, or other apparatus (including amplifiers) that produce heat.
9. ONLY USE attachments/accessories specified by the manufacturer.
10. UNPLUG this apparatus during lightning storms or when unused for long periods of time.
11. REFER all servicing to qualified service personnel. Servicing is required when the apparatus has been damaged in any way, such as liquid has been spilled or objects have fallen into the apparatus, the apparatus has been exposed to rain or moisture, does not operate normally, or has been dropped.
12. DO NOT expose this apparatus to dripping or splashing and ensure that no objects filled with liquids, such as vases, are placed on the apparatus.
13. To completely disconnect this apparatus from the AC Mains, disconnect the power supply cord plug from the AC receptacle or disconnect the PoE injector.
14. Where the mains plug or an appliance coupler is used as the disconnect device, the disconnect device shall remain readily operable.
15. DO NOT overload wall outlets or extension cords beyond their rated capacity as this can cause electric shock or fire.
16. The unit is to be connected only to PoE networks without routing to the outside plant.
17. DO NOT hang anything on the panel when it is mounted to a wall, glass, and other smooth surfaces. The equipment is only suitable for mounting at heights ≤ 2 m (6.56ft).

⚠ The exclamation point, within an equilateral triangle, is intended to alert the user to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the product.

The lightning flash with arrowhead symbol within an equilateral triangle is intended to alert the user to the presence of uninsulated “dangerous voltage” within the product’s enclosure that may be of sufficient magnitude to constitute a risk of electrical shock to persons.

ESD Warning: The icon to the left indicates text regarding potential danger associated with the discharge of static electricity from an outside source (such as human hands) into an integrated circuit, often resulting in damage to the circuit.

WARNING: To reduce the risk of fire or electrical shock, do not expose this apparatus to rain or moisture.

WARNING: No naked flame sources – such as lighted candles – should be placed on the product.

CAUTION: To be installed by instructed, or skilled, persons only.

WARNING: This product is intended to be operated ONLY from the voltages listed on the back panel or the recommended, or included power supply of the product. Operation from other voltages other than those indicated may cause irreversible damage to the product and void the products warranty. The use of AC Plug Adapters is cautioned because it can allow the product to be plugged into voltages in which the product was not designed to operate. If you are unsure of the correct operational voltage, please contact your local distributor and/or retailer. If the product is equipped with a detachable power cord, use only the type provided, or specified, by the manufacturer or your local distributor.

CAUTION
RISK OF ELECTRIC SHOCK
DO NOT OPEN

WARNING: Do Not Open! Risk of Electrical Shock. Voltages in this equipment are hazardous to life. No user-serviceable parts inside. Refer all servicing to qualified service personnel.

Place the equipment near a main power supply outlet and make sure that you can easily access the power breaker switch.

BATTERY INFORMATION:

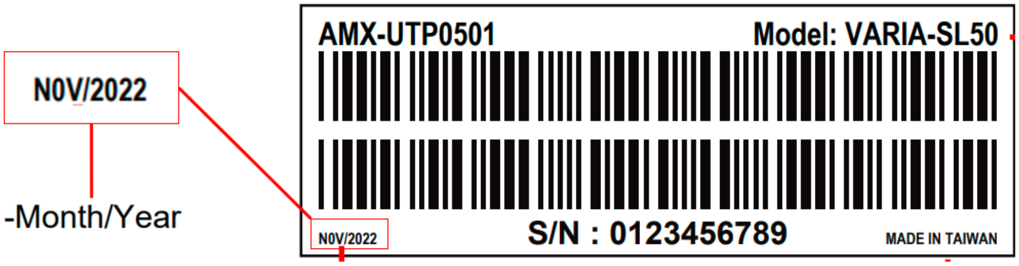
This product contains batteries that are covered under the 2006/66/EC European Directive, which cannot be disposed of with normal household waste. Please follow local regulations. Do not incinerate.

WARNING: **40 °C / 104 °F** is maximum ambient operating temperature. Avoid exposure to extreme heat or cold.

LASER SAFETY:

This product complies with FDA performance standards for laser products except for conformance with IEC 60825-1 Ed. 3., as described in Laser Notice No. 56, dated May 8, 2019.

Date of manufacture: the date code shown on the product label represents the manufacturing date of this laser product.



CAUTION: Invisible laser radiation when open. Avoid exposure to beam. Class 1 laser product. This system must be opened only by qualified technicians to prevent accidents caused by the laser beam.

CAUTION: Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

FCC AND CANADA EMC COMPLIANCE INFORMATION:

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

CAN ICES-3 (B)/NMB-3(B)

FCC SDOC SUPPLIER'S DECLARATION OF CONFORMITY:

HARMAN Professional, Inc. hereby declares that this equipment is in compliance with the FCC part 15 Subpart B.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Approved under the verification provision of FCC Part 15 as a Class B Digital Device.

Caution: Changes or modifications not expressly approved by the manufacturer could void the user's authority to operate this device.

WIRELESS TRANSMITTER COMPLIANCE INFORMATION:

The term “IC:” before the radio certification number only signifies that Industry Canada technical specifications were met.

Le terme «IC:» avant le numéro de certification radio signifie seulement que les specifications techniques d'Industrie Canada ont été respectées.

This device complies with part 15 of the FCC Rules and the applicable Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes : (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

This equipment complies with FCC and IC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20mm between the radiator and your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Cet appareil est conforme à FCC et IC l'exposition aux rayonnements limites fixées pour un environnement non contrôlé. Cet appareil doit être installé et utilize avec une distance minimale de 20mm entre le radiateur et votre corps. Cet transmetteur ne doit pas être co-situé ou operant en liaison avec toute autre antenne ou transmetteur.

EU COMPLIANCE INFORMATION:

Hereby, Harman Professional, Inc. declares that the radio equipment type AMX VARIA-SL80, VARIA-80, VARIA-100 and VARIA-150 are in compliance with directive 2014/53/EU. The full text of the EU declaration of conformity is available at the following internet address: https://www.amx.com.

This radio product operates in frequency of 13.56 MHz with maximum power 0.01mW.

WEEE NOTICE:

This appliance is labeled in accordance with European Directive 2012/19/EU concerning waste of electrical and electronic equipment (WEEE). This label indicates that this product should not be disposed of with household waste. It should be deposited at an appropriate facility to enable recovery and recycling.

RoHS Compliance:

This product is in compliance with Directive 2011/65/EU and (EU) 2015/863 of the European Parliament and of the Council of 31/03/2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

REACH Compliance:

REACH (Regulation No 1907/2006) addresses the production and use of chemical substances and their potential impacts on human health and the environment. Article 33 (1) of REACH Regulation requires suppliers to inform the recipients if an article contains more than 0.1% (per weight per article) of any substance(s) on the Substances of Very High Concern (SVHC) Candidate List (‘REACH candidate list’).

This product contains the substance ‘’lead’’ (CAS-No. 7439-92-1) in a concentration of more than 0.1% per weight.

At the time of release of this product, except for the lead substance, no other substances of REACH candidate list are contained in a concentration of more than 0.1% per weight in this product.

Note: on June 27, 2018, lead was added to the REACH candidate list. The inclusion of lead in the REACH candidate list does not mean that lead-containing materials pose an immediate risk or results in a restriction of permissibility of its use.

此标识适用于在中华人民共和国销售的电子信息产品. 标识中间的数字为环保实用期限的年数.

This logo applies to electronic information products sold in the People's Republic of China. The number in the middle of the logo is the number of years of environmental utility.

Manufacturer Information:
HARMAN Professional, Inc.
Address: 8500 Balboa Blvd. Northridge, CA 91329 USA

EU Regulatory Contact:
Harman Professional Denmark ApS
Olof Palmes Allé 44, 8200 Aarhus N, Denmark

UK Regulatory Contact:
Harman Professional Solutions
2 Westside, London Road, Apsley, Hemel Hempstead, HP3 9TD, UK

COPYRIGHT NOTICE

LIABILITY NOTICE

No patent liability is assumed with respect to the use of information contained herein. While every precaution has been taken in the preparation of this publication, AMX assumes no responsibility for error or omissions. No liability is assumed for damages resulting from the use of the information contained herein. Further, this publication and features described herein are subject to change without notice.

AMX WARRANTY AND RETURN POLICY

The AMX Warranty and Return Policy and related documents can be viewed/downloaded at www.amx.com.

Table of Contents

Varia Touch Panels Configuration & Programming

Overview

The AMX Varia touch panels redefine control with personas that fit any application. Any AMX Varia touch panel can be configured as any persona:

- **AMX G5 Control** that utilizes AMX G5 programming tools like TPDesign5 & Netlinx Studio
- **AMX Book Scheduling** that directly connects directly to popular scheduling software
- **Zoom Rooms Controller** to manage Zoom Rooms environments
- **Web Kiosk** that can display HTML5 & web content on a full-screen borderless browser
- **CloudworX Control** (future availability)

The VARIA touch panels covered in this document are listed below**:**

VARIA TOUCH PANELS		
AMX-UTP0501	VARIA-SL50	5.5” ULTRA-SLIM WALL MOUNT TOUCH PANEL, PORTRAIT
AMX-UTP0801	VARIA-SL80	8” ULTRA-SLIM WALL MOUNT TOUCH PANEL
AMX-UTP0811	VARIA-80	8” TOUCH PANEL
AMX-UTP1011	VARIA-100	10.1” TOUCH PANEL
AMX-UTP1011N	VARIA-100N	10.1” TOUCH PANEL (NO-COMM)
AMX-UTP1511	VARIA-150	15.6” TOUCH PANEL
AMX-UTP1511N	VARIA-150N	15.6” TOUCH PANEL (NO-COMM)

NO-COMM Touch Panels are used in highly secure environments where device peripherals are not allowed. NO-COMM panels omit the camera, microphone, and NFC reader. NO-COMM panels include speakers, ambient light sensor, and proximity sensor.

For NO-COMM panels, please disregard any section of this manual that refers to camera, microphones, or NFC.

To use the AMX G5 Control persona, please use the minimum required versions of the following AMX software:

- TPDesign5 version 1.5.0, Build 111, Border Version 25
- G5 Support Files 1.5.68
- NetLinx Studio 4.4.1915
- Resource Management Suite 4.8.3

Software can be downloaded from AMX.com.

Additional Documentation

- For instructions on using NetLinx Studio, refer to NetLinx Studio online help, or the NetLinx Studio v4 Instruction Manual.
- For instructions on using TPDesign5, refer to TPDesign5 online help, or the TPDesign5 Instruction Manual.
- For installation instructions for Varia panels, refer to the Varia Touch Panels’ Quick Start Guides.

Out of the Box Setup Wizard

Overview

AMX Varia touch panels include a step-by-step wizard to walk a user through basic setup requirements. This includes language selection, date & time, network, and touch panel persona.

The Setup Wizard

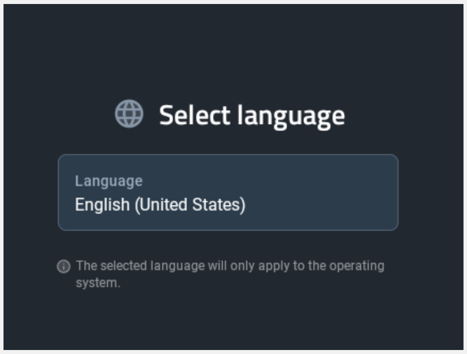
When a panel is first turned on (either out of the box, or after a factory reset operation), the setup wizard appears. Follow the steps to select your preferences. After making selections on each page, press [Continue] on the bottom-right of the screen to go to the next page. To go back, press [Back] on the bottom-left of the screen.

Language

Use the dropdown to select your language.

NOTE: The selected language will only apply to the operating system.

Additional language selections/decisions may be required based on the selected persona.



Language
English (United States)

The selected language will only apply to the operating system.

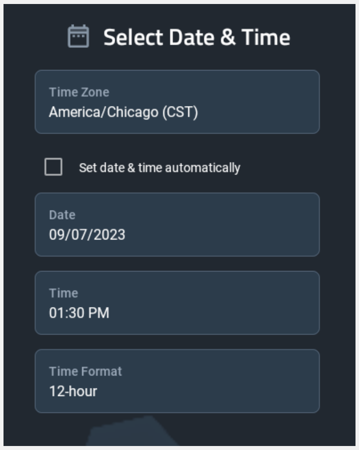
Date & Time

Use the dropdown to select your time zone.

You can select *Set date & time automatically* to utilize NTP.

Or, if required, adjust your date & time manually by using the dropdowns.

Select either 12-hour or 24-hour time format.



Time Zone
America/Chicago (CST)

□ Set date & time automatically

Date
09/07/2023

Time
01:30 PM

Time Format
12-hour

Network

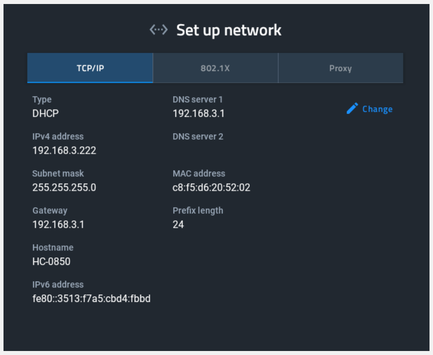
Select your network configuration

The current configuration will be shown. Default is DHCP with 802.1x disabled and no proxy.

A blue Change button is available to modify settings as needed.

Change

Make sure to press {Save] after making changes, or your newly entered network settings will not be saved.



Type
DHCP

IPv4 address
192.168.3.222

Subnet mask
255.255.255.0

Gateway
192.168.3.1

Hostname
HC-0850

IPv6 address
fe80::3513:f7a5:cbd4:fbbd

DNS server 1
192.168.3.1

DNS server 2

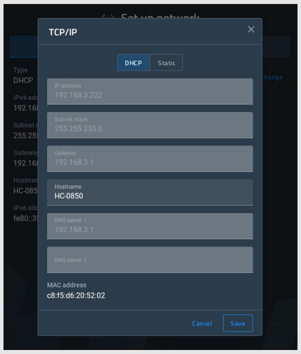
MAC address
c8:f5:d6:20:52:02

Prefix length
24

DHCP

Network DHCP server will automatically assign IP address, subnet mask, default gateway, and DNS server info.

In this mode, IP fields are read-only. Hostname is available to change as needed.



TCP/IP

IP address
192.168.3.222

Subnet mask
255.255.255.0

Gateway
192.168.3.1

Hostname
HC-0850

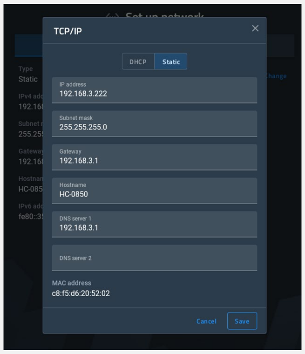
DNS server 1
192.168.3.1

DNS server 2

MAC address
c8:f5:d6:20:52:02

Static IP

IP address, subnet mask, default gateway, and DNS server info must be entered manually. Make sure to press [Save] after entering the information. Press [Cancel] to leave without saving any edits.



TCP/IP

IP address
192.168.3.222

Subnet mask
255.255.255.0

Gateway
192.168.3.1

Hostname
HC-0850

DNS server 1
192.168.3.1

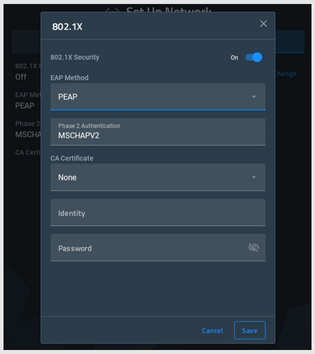
DNS server 2

MAC address
c8:f5:d6:20:52:02

802.1X

If 802.1X is available/required, turn it [ON], select your EAP method, and enter the required info for that method (eg. certificate, domain, identity, password, etc.).

For further assistance and detail on whether your network supports or requires 802.1X, please contact your or your client’s IT department.



802.1X

802.1X Security: On

EAP Method
PEAP

Phase 2 Authentication
MSCHAPV2

CA Certificate
None

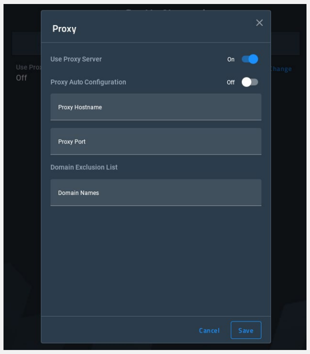
Identity

Password

Proxy

If a proxy is available/required, turn it [ON] and enter the appropriate configuration information.

For further assistance and detail on whether your network supports or requires 802.1X, please contact your or your client’s IT department.



Proxy

Use Proxy Server: On

Proxy Auto Configuration: Off

Proxy Hostname

Proxy Port

Domain Exclusion List

Domain Names

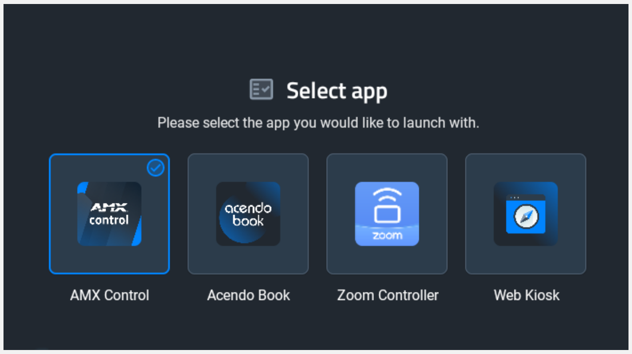
Persona Selection

Select the panel's persona, which is the primary app that will be used by the panel. Each subsequent reboot will launch only this persona.

Each persona is described in detail later in this manual.

Once selected, press [Continue].

NOTE: The panel persona may be changed at any time in Advanced Settings.



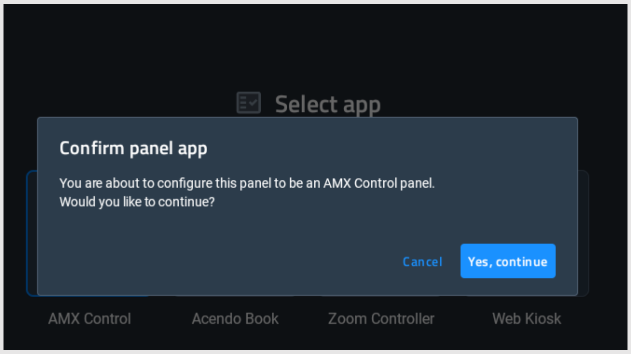
Select app

Please select the app you would like to launch with.

Confirm Panel App

Your selection will be confirmed in a pop-up. Press [Yes, continue] to complete the selection or [Cancel] to go back and select a different persona.

NOTE: Once confirmed, the panel may take up to one (1) minute to initially configure the persona.



Select app

Confirm panel app

You are about to configure this panel to be an AMX Control panel.
Would you like to continue?

Wizard - AMX Control

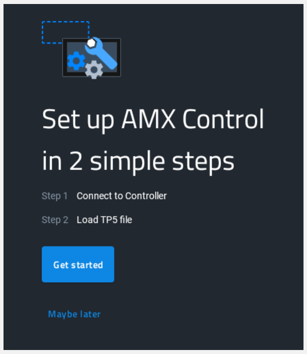
Overview

The AMX Control persona utilizes familiar G5 touch panel technology & features, used with Netlinx control systems. The touch panel GUI (graphical user interface) is programmed with TPDesign5 and is compatible with .TP5 files.

2 Simple Steps

When the AMX Control persona is selected & configured, the wizard continues to help guide through the setup of Netlinx controller and load the TPDesign5 (TP5) UI file.

Select [Get Started] to continue, or [Maybe Later] to exit the wizard.



Set up AMX Control in 2 simple steps

Step 1 Connect to Controller

Step 2 Load TP5 file

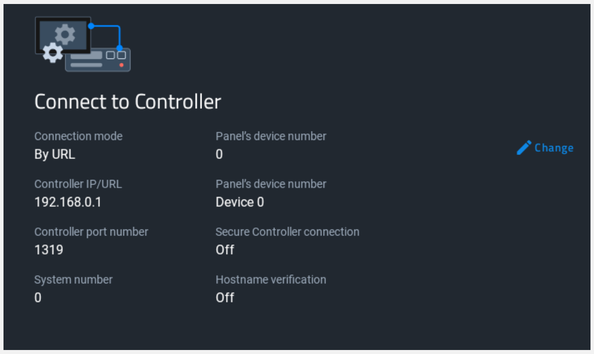
Netlinx Setup

Connect to your Netlinx controller. Start by pressing the blue [Change] button on the right.

Change

Select one of the three (3) available connection methods.

Each is described in detail below.



Connect to Controller

Connection mode: By URL

Panel's device number: 0

Controller IP/URL: 192.168.0.1

Panel's device number: Device 0

Controller port number: 1319

Secure Controller connection: Off

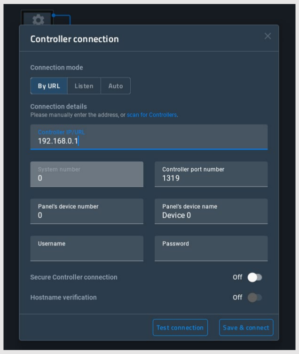
System number: 0

Hostname verification: Off

By URL

Enter the Controller’s IP/URL, port number (if not using default 1319), and username/password (if used). The System Number field is read-only; the panel obtains this information from the controller.

Assign your touch panel a Device ID and a Device Name for your panel.



Controller connection

Connection mode: By URL | Listen | Auto

Connection details

Please manually enter the address, or scan for Controllers.

Controller IP/URL: 192.168.0.1

System number: 0

Controller port number: 1319

Panel's device number: 0

Panel's device name: Device 0

Username

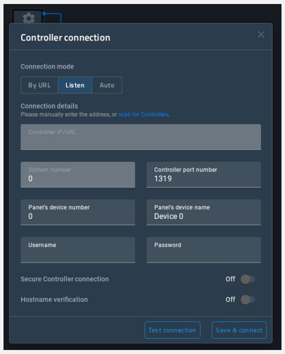
Password

Secure Controller connection: Off

Hostname verification: Off

Listen

Add the panel address into the URL List in NetLinx Studio and set the connection mode to Listen. This mode allows the panel to “listen” for the Controller’s communication signals. The System Number and Controller IP/URL fields are read-only.



Controller connection

Connection mode: By URL | **Listen** | Auto

Connection details
Please manually enter the address, or scan for Controllers.

Controller IP/URL:

System number: 0

Controller port number: 1319

Panel’s device number: 0

Panel’s device name: Device 0

Username:

Password:

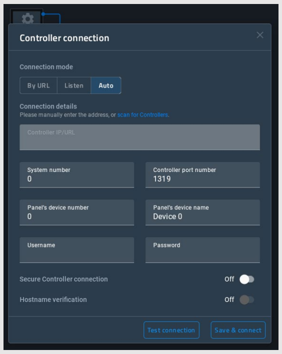
Secure Controller connection: Off

Hostname verification: Off

Auto

Enter the System Number and a username/password (if applicable).

Use this mode when both the panel & NetLinx Controller are on the same Subnet. The Controller IP/URL field is read-only.



Controller connection

Connection mode: By URL | Listen | **Auto**

Connection details
Please manually enter the address, or scan for Controllers.

Controller IP/URL:

System number: 0

Controller port number: 1319

Panel’s device number: 0

Panel’s device name: Device 0

Username:

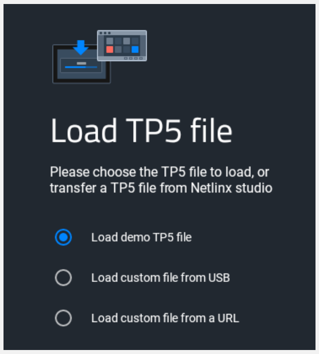
Password:

Secure Controller connection: Off

Hostname verification: Off

Load TP5 File

Choose one of the three (3) available options to load a .TP5 file to the panel.



Load TP5 file

Please choose the TP5 file to load, or transfer a TP5 file from Netlinx studio

● Load demo TP5 file

○ Load custom file from USB

○ Load custom file from a URL

Demo TP5 file will load a pre-loaded & pre-built panel file

Load from USB will allow a file to be loaded via a USB thumb drive.

Load from URL will allow the user to enter a URL from which the .TP5 file will be downloaded.

Alternatively, Cloudworx Manager Desktop software may be used to load a .TP5 file to one or more touch panels.

If you don’t wish to load a .TP5 file right now, the final option [Maybe Later] may be selected at the bottom of the screen

To use the AMX G5 Control persona, please use the minimum required versions of the following AMX software:

- TPDesign5 version 1.5.0, Build 111, Border Version 25
- G5 Support Files 1.5.68
- NetLinx Studio 4.4.1915
- Resource Management Suite 4.8.3

Software can be downloaded from AMX.com.

Wizard - AMX Book

Overview

AMX Book offers an out-of-the box room scheduling solution, connects directly to popular scheduling platforms and supports the latest authentication protocols like Microsoft Modern Authentication.

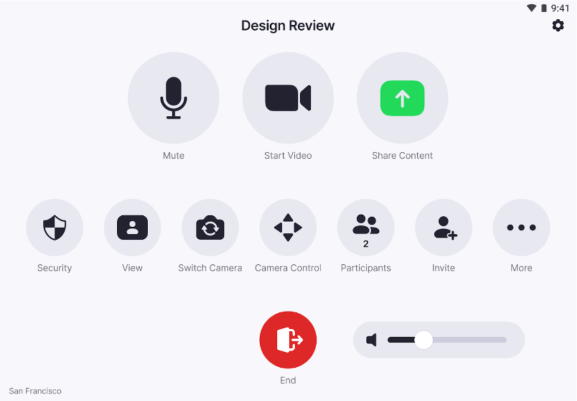
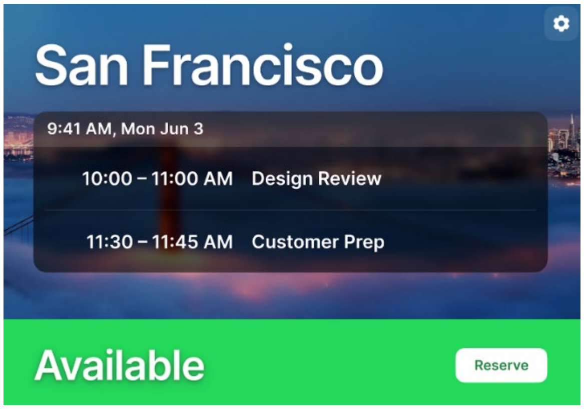
Setup

Complete setup instructions can be found later in this manual.

Wizard - Zoom Rooms Controller

Overview

The built-in Zooms Room Controller persona can be used to control & manage Zoom Rooms environments. Varia panels can be set up as a Zoom Rooms **Room Controller** or a Zoom Rooms **Scheduling Display**.

Room Controller	Scheduling Display
	

San Francisco
9:41 AM, Mon Jun 3
10:00 – 11:00 AM Design Review
11:30 – 11:45 AM Customer Prep
Available
Reserve

Setup

Setup for ZRC environments is completed within the ZRC app. More information on Zoom Rooms can be found here:

https://www.zoom.com/en/products/meeting-rooms/resources/configuration/

Please note that the ZRC app is very sensitive to correct date, time, and time zone. If an incorrect date/time is set on the panel, Zoom will refuse to connect.

Wizard - Web Kiosk

Overview

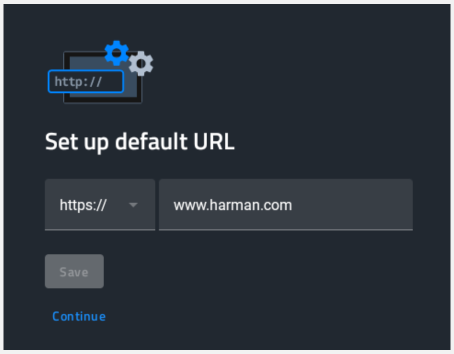
A full-screen borderless browser for HTML5, JavaScript, & web can be selected as a persona to turn your panel into a standards-based kiosk designed around web content.

Setup

Use the dropdown to select **https://** (preferred) or **http://**. Be careful not to repeat this in the URL field.

Enter the URL to be displayed in the field.

Press [Save], and then [Continue]. Your web content will load immediately and will also load on any subsequent panel reboot.



Set up default URL

https:// ▾ www.harman.com

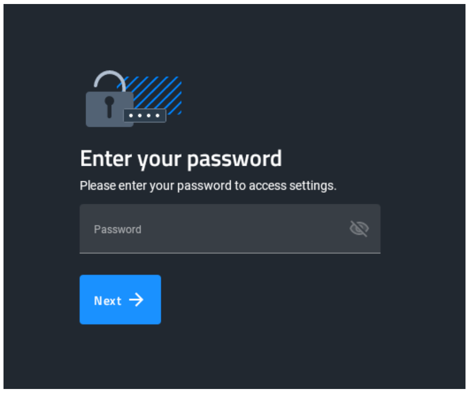
Panel Settings Menu

Overview

Panel Settings contain configuration info for the currently active persona, and advanced settings.

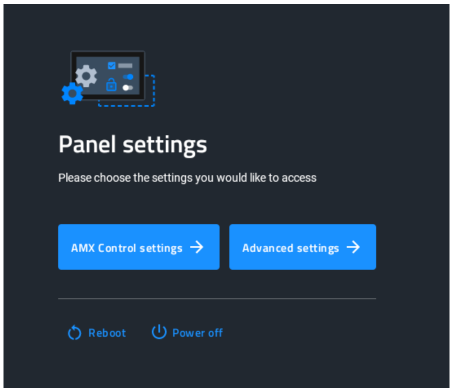
Accessing the Settings Menu

To access the Panel Settings menu, briefly press & release the Settings Button. The Settings Button is the pin-hole button on the top-right of the panel bezel, next to the rectangular Function button. The default password to access Panel Settings is 1988.



Enter your password

Please enter your password to access settings.



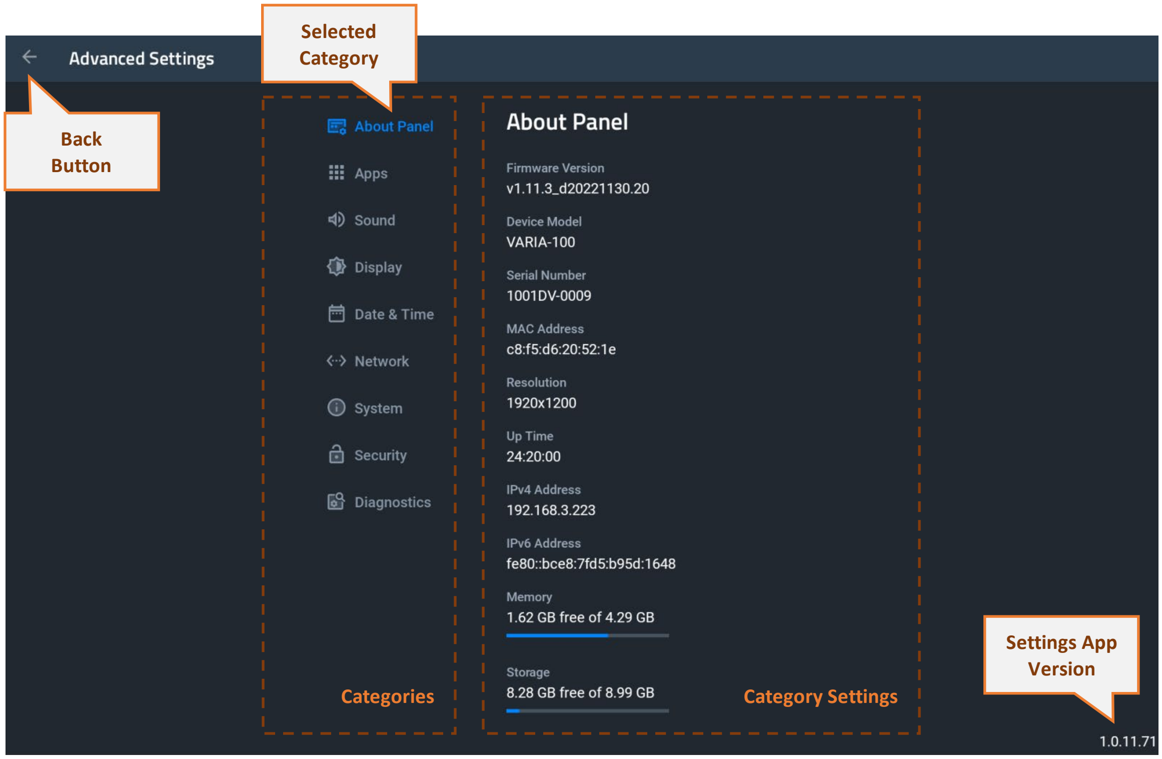
Panel settings

Please choose the settings you would like to access

NOTE: Do not confuse the Settings button (pin-hole button) with any of the microphone or sensor holes. The Settings button is the **larger hole located closest to the Function Button**. Irreparable damage may occur if a foreign object is pushed down the microphone hole or into one of the sensors.

Using the Settings Menu

When opened, the Settings menu appears in the center of the panel display. Please note that many of the pages in the menu may be longer than they initially appear. To reach additional functions on a given page, the page itself may be scrolled up & down to reveal those functions.



About Panel

Firmware Version
v1.11.3_d20221130.20

Device Model
VARIA-100

Serial Number
1001DV-0009

MAC Address
c8:f5:d6:20:52:1e

Resolution
1920x1200

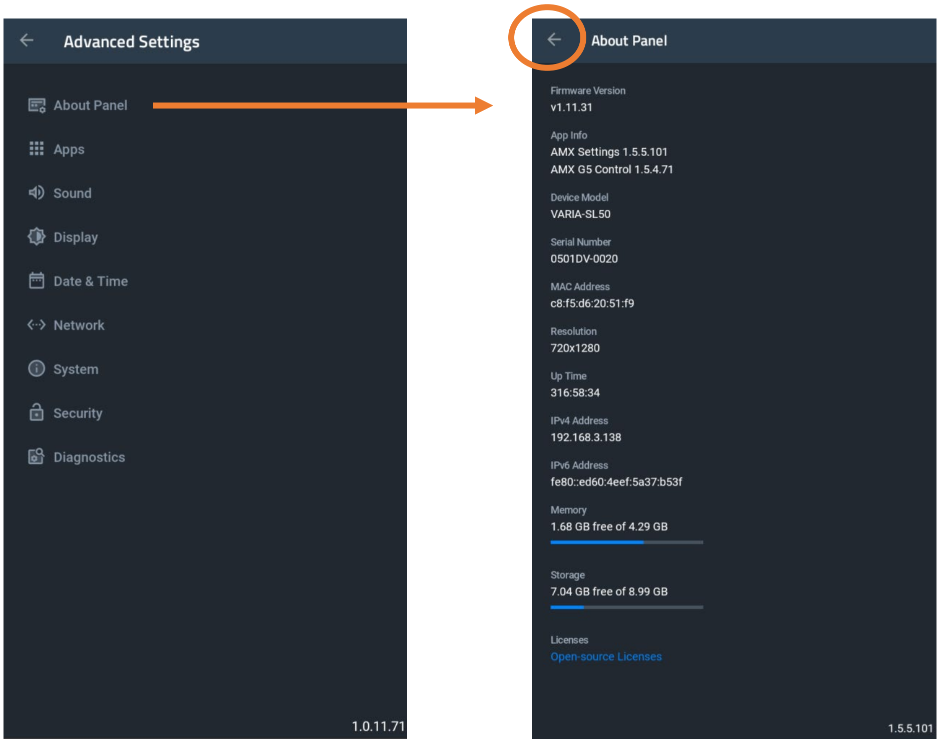
Up Time
24:20:00

IPv4 Address
192.168.3.223

IPv6 Address
fe80::bce8:7fd5:b95d:1648

Memory
1.62 GB free of 4.29 GB

Storage
8.28 GB free of 8.99 GB



About Panel

Firmware Version
v1.11.31

App Info
AMX Settings 1.5.5.101
AMX G5 Control 1.5.4.71

Device Model
VARIA-SL50

Serial Number
0501DV-0020

MAC Address
c8:f5:d6:20:51:f9

Resolution
720x1280

Up Time
316:58:34

IPv4 Address
192.168.3.138

IPv6 Address
fe80::ed60:4eef:5a37:b53f

Memory
1.68 GB free of 4.29 GB

Storage
7.04 GB free of 8.99 GB

Licenses
Open-source Licenses

1.5.5.101

Please note on the 5.5" panel, the Categories Menu is shown on the full screen and then flips to each Categories' settings page. Press the ← arrow on the top-left of the page to go back to the Categories Menu.

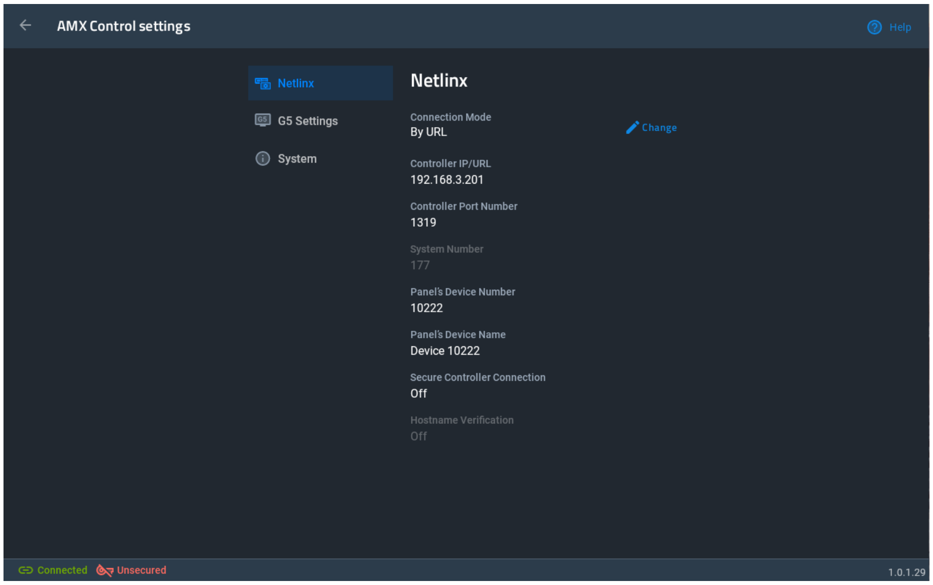
AMX Control

Overview

The Overview page presents a summary of the options available in an AMX NetLinx control system.

NetLinx

The NetLinx page controls the method of connecting to a NetLinx Controller.



Netlinx

Connection Mode
By URL

Controller IP/URL
192.168.3.201

Controller Port Number
1319

System Number
177

Panel's Device Number
10222

Panel's Device Name
Device 10222

Secure Controller Connection
Off

Hostname Verification
Off

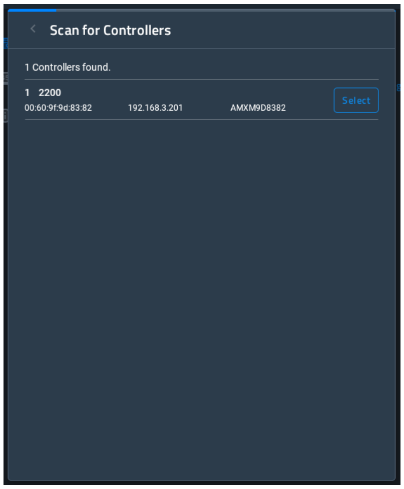
Netlinx	
Scan for Controllers	Press to scan for NetLinx controllers on the network, via the Controller Connection window. See Scanning for Controllers on the next page for details.
Mode	Cycles between the connection modes: URL, Listen, and Auto. **URL** - Enter the IP/URL, Controller Port Number, and username/password (if used) on the Controller. The System Number field is read-only - the panel obtains this information from the Controller. **Listen** - Add the panel address into the URL List in NetLinx Studio and set the connection mode to Listen. This mode allows the panel to "listen" for the Controller's communication signals. The System Number and Controller IP/URL fields are read-only. **Auto** - Enter the System Number and a username/password (if applicable). Use this mode when both the panel and the NetLinx Controller are on the same Subnet. The Controller IP/URL field is read-only.
System Number	Allows entry of a system number. Default value is 0 (zero). Note: Available in Auto Mode Only - disabled when URL or Listen is selected.
Controller IP/URL	Sets the Controller IP or URL of the NetLinx Controller. Note: Available in URL Only - disabled when Listen or Auto is selected.
Controller/Port Number	Allows entry of the port number used with the NetLinx Controller. Default = 1319.
Username	If the target Controller has been previously secured, enter the alpha-numeric string (into each field) assigned to a preconfigured user profile on the Controller. This profile should have the predefined level of access/configuration rights.
Password	If the target Controller has been previously secured, enter the alpha-numeric string (into each field) assigned to a preconfigured user profile on the Controller. This profile should have the predefined level of access/configuration rights.
Device Number	Displays the panel's device number and allows entry of a new one.
Device Name	Displays the panel's device name and allows entry of a new one.

Netlinx	
Scan for Controller Connection	Press to enable and configure a connection a secured NetLinx Controller, via options in the NetLinx window. Note: The secure connection is a TLS connection to a NetLinx Controller on port 1320.
Hostname Verification	Press to enable hostname verification of the NetLinx Controller via the Controller's device certificate. Note: This option is only available if the Secure Controller Connection option is enabled.
Connection Status	Displays the panel's connection status to the Controller

Scanning for Controllers

Use the Scan For Controllers feature to quickly and easily identify all of the available NetLinx Controllers on the network. The site survey on this page passively listens to network traffic and presents all the compatible Controllers for easy selection. Selecting the desired Controller automatically updates the NetLinx Controller section and establishes a connection.

1. In the NetLinx page, press **Scan For Controllers** to begin listening for NetLinx controllers and open the Controller Connection window (FIG. 64):Select the NetLinx Controller for this panel.



Scan for Controllers

1 Controllers found.

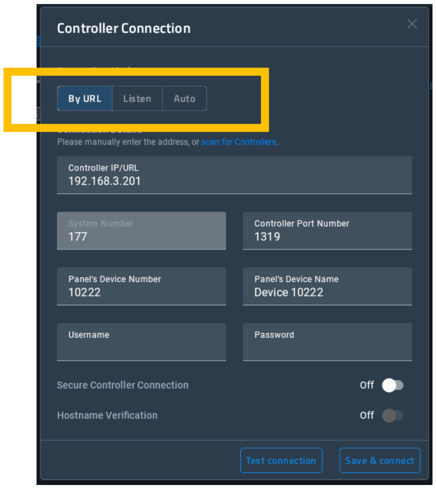
1 2200
00:60:9f:9d:83:82 192.168.3.201 AMXM9D8382

2. The NetLinx page will automatically display the connection information on the selected Controller

Changing the Controller Connection Mode

To select the Controller Connection mode (URL, Listen, or Auto):

1. In the NetLinx page, press **Mode** to open NetLinx Options window:



Controller Connection

Please manually enter the address, or scan for Controllers.

Controller IP/URL: 192.168.3.201

System Number: 177

Controller Port Number: 1319

Panel's Device Number: 10222

Panel's Device Name: Device 10222

Username

Password

Secure Controller Connection: Off

Hostname Verification: Off

2. Select the desired option from the **Mode** menu - URL (default setting), Listen or Auto.
3. When finished, press **OK** to return to the NetLinx page.

Changing the Controller IP/URL

To change the IP address or URL for the chosen Controller:

1. In the NetLinx page, press **Controller IP/URL** to open the NetLinx Options window and on-screen keyboard.
2. Enter the IP address or the URL.
3. Press the double-down arrow key at the bottom right of the on-screen keyboard to close the keyboard.
4. The new IP address/URL is now displayed in the Controller IP/URL field.
5. Press **OK** to save the changes and return to the NetLinx page.:

Changing the Controller Port Number

To change the Controller Port Number from its default:

1. In the NetLinx page, press **Controller Port Number** to open the NetLinx Options window and on-screen keyboard.
2. Enter the new Controller Port Number.
3. Press the double-down arrow key at the bottom right of the on-screen keyboard to close the keyboard.
4. The new Port Number is now displayed in the Controller Port Number field.
5. Press **OK** to save the changes and return to the NetLinx page.

Changing the Controller Username

1. In the NetLinx page, press the Username field to open the NetLinx Options window and on-screen keyboard.
2. Enter the new username.
3. Press the double-down arrow key at the bottom right of the on-screen keyboard to close the keyboard.
4. The new Username is now displayed in the Username field.
5. Press **OK** to save the changes and return to the NetLinx page

Changing the Controller Password

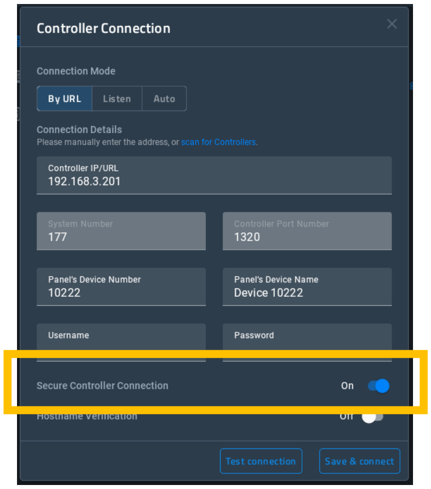
1. In the NetLinx page, press **Password** to open the NetLinx Options window and on-screen keyboard.
2. Enter the new password.
3. Press the double-down arrow key at the bottom right of the on-screen keyboard to close the keyboard.
4. The new Password is now displayed in the Password field.
5. Press **OK** to close the NetLinx window and return to the NetLinx

Changing the Device Number and Device Name

1. In the NetLinx page, press the Device Number field to open the NetLinx Options window and on-screen keypad.
2. Enter a new Device Number.
3. Press **Next**, to select Device Name in the NetLinx page and open the n-screen keyboard.
4. Enter a new Device Name.
5. Press **Done** to close the keypad and keyboard.
6. The new Device Number and Device Name are now displayed in the Device Number and Device Name fields.
7. Press **OK** to save changes and close the NetLinx window and return to the NetLinx page.

Enabling a Secure Controller Connection

1. In the NetLinx page, press the Secure Controller Connection field to open the NetLinx Options window.
2. Scroll to the bottom of the options list and toggle the Secure Controller Connection option ON:



Controller Connection

Connection Mode

By URL | Listen | Auto

Connection Details

Please manually enter the address, or scan for Controllers.

Controller IP/URL: 192.168.3.201

System Number: 177

Controller Port Number: 1320

Panel's Device Number: 10222

Panel's Device Name: Device 10222

Username

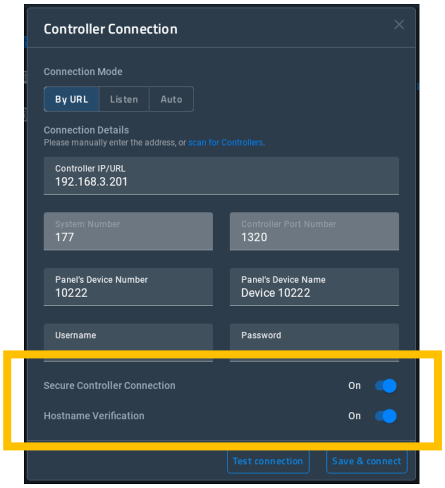
Password

Secure Controller Connection: On

Hostname Verification: Off

Enabling Hostname Verification

1. In the NetLinx page, press the Hostname Verification field to open the NetLinx Options window.
2. Scroll to the bottom of the options list and toggle the Hostname Verification option **ON** (FIG. 67):



Controller Connection

Connection Mode

By URL | Listen | Auto

Connection Details

Please manually enter the address, or scan for Controllers.

Controller IP/URL: 192.168.3.201

System Number: 177

Controller Port Number: 1320

Panel's Device Number: 10222

Panel's Device Name: Device 10222

Username

Password

Secure Controller Connection: On

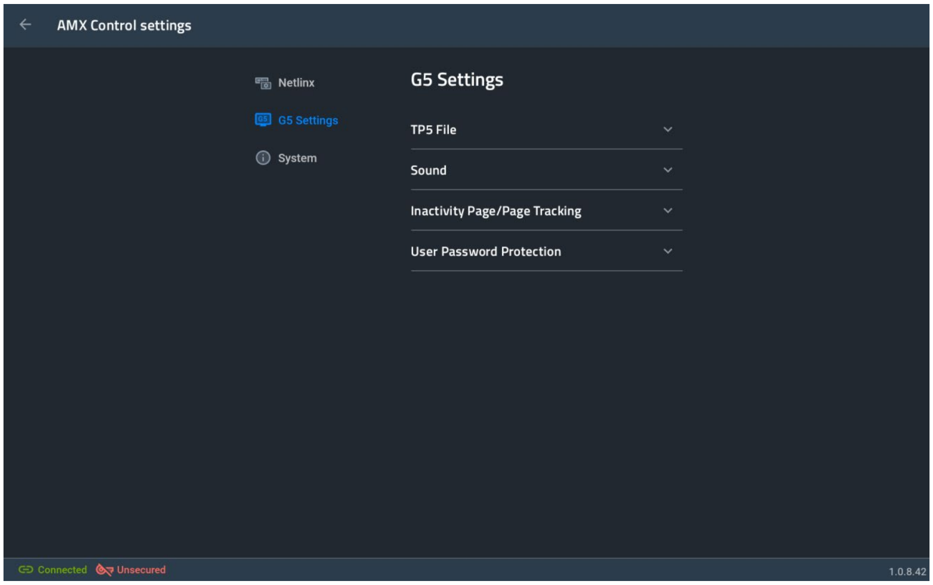
Hostname Verification: On

Test connection | Save & connect

Note that this option is available only if the Secure Controller Connection option is ON.

G5 Settings

The G5 Settings contains specific settings related to the G5 experience and to the TP5 project file loaded.



AMX Control settings

Netlinx

G5 Settings

System

G5 Settings

TP5 File

Sound

Inactivity Page/Page Tracking

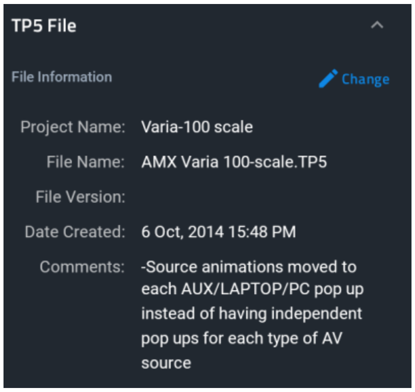
User Password Protection

Connected | Unsecured

1.0.8.42

G5 Settings Options	
TP5 File	This tab contains information related to your TP5 project file, and also gives the option to change the TP5 file loaded.

G5 Settings Options



TP5 File

File Information

Project Name: Varia-100 scale

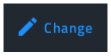
File Name: AMX Varia 100-scale.TP5

File Version:

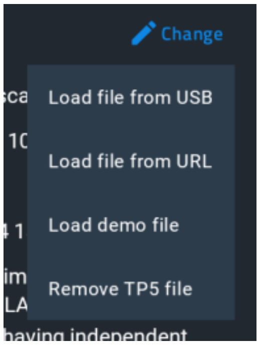
Date Created: 6 Oct, 2014 15:48 PM

Comments: -Source animations moved to each AUX/LAPTOP/PC pop up instead of having independent pop ups for each type of AV source

To change the TP5 project file, press [Change].



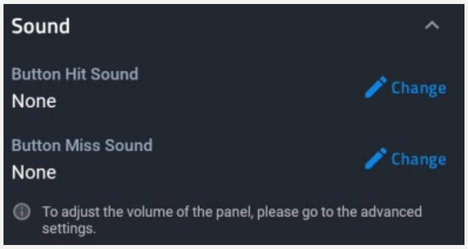
There are options to load from USB, URL, demo file, or remove the current TP5 project file



Note: USB drives should be formatted to FAT32 file system.

Sound

This tab contains sound settings for your touch panel. Dedicated sounds for *Button Hit* & *Button Miss* can be applied here.



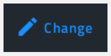
Sound

Button Hit Sound
None

Button Miss Sound
None

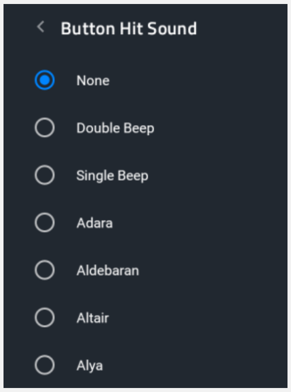
To adjust the volume of the panel, please go to the advanced settings.

To change either of these sounds, or to select no sound, press its respective [Change] button.



Scroll through and select a new sound. The options for *None* is at the very top.

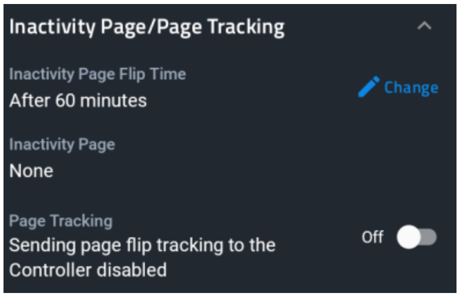
G5 Settings Options


Button Hit Sound

● None
○ Double Beep
○ Single Beep
○ Adara
○ Aldebaran
○ Altair
○ Alya

Inactivity Page / Page Tracking

This contains options to set up an Inactivity page Flip Time, which is a page flip made inside a TP5 project file after a specified period of inactivity on the panel (ie. no touches), and to enable or disable Page Tracking.

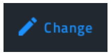

Inactivity Page/Page Tracking

Inactivity Page Flip Time
After 60 minutes
Change

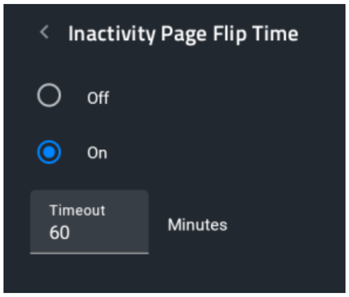
Inactivity Page
None

Page Tracking
Sending page flip tracking to the Controller disabled
Off

To set up, press [Change].


Change

Select On or Off. If On, select the desired amount of time.


Inactivity Page Flip Time

○ Off
● On

Timeout
60
Minutes

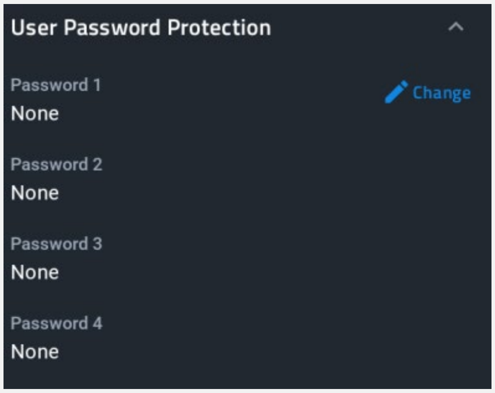
Inactivity Page is read-only, and dictated by the TP5 project file. See TPDesign5 for more information.

Page Tracking , if enabled, will send page flip tracking messages to the Netlinx controller.

User Password Protection

This provides the option of assigning passwords to the secured Settings pages. Users have the the ability to assign alphanumeric values to particular password sets.

G5 Settings Options



User Password Protection

Password 1
None

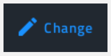
Change

Password 2
None

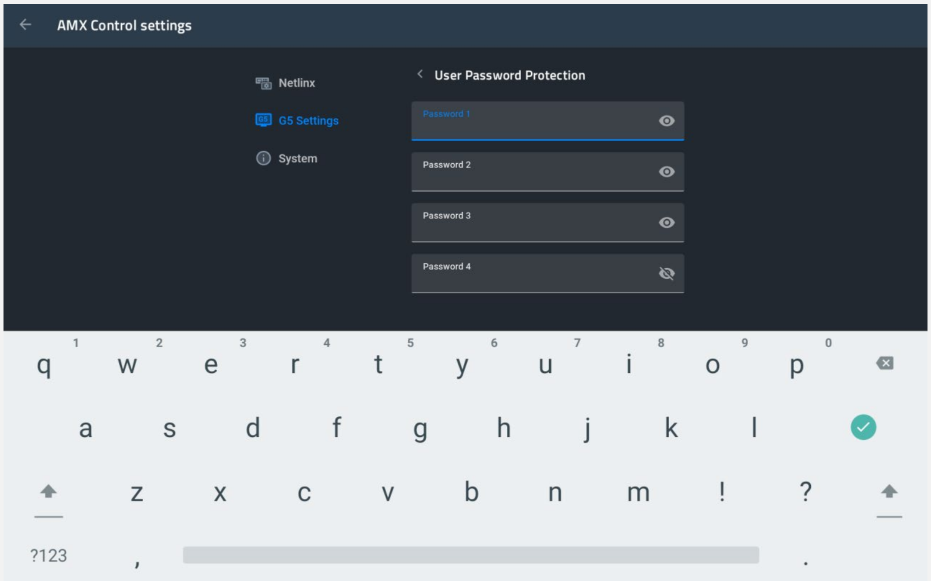
Password 3
None

Password 4
None

To assign, change, or delete a password, press [Change].



You may enter up to four (4) passwords.



AMX Control settings

Netlinx

G5 Settings

System

User Password Protection

Password 1

Password 2

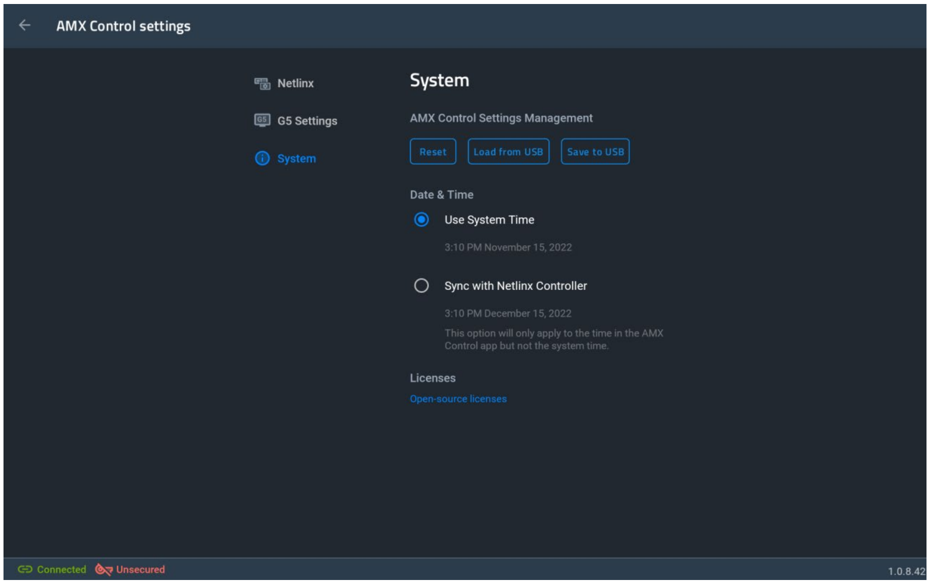
Password 3

Password 4

Note: This function does not overwrite the default settings password (1988).

System

The System settings contains options for the panel configuration, date & time, and date & time.



System

AMX Control Settings Management

Date & Time

● Use System Time

3:10 PM November 15, 2022

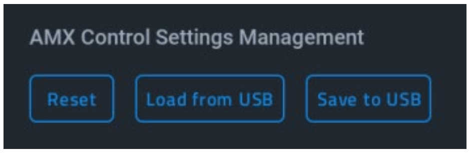
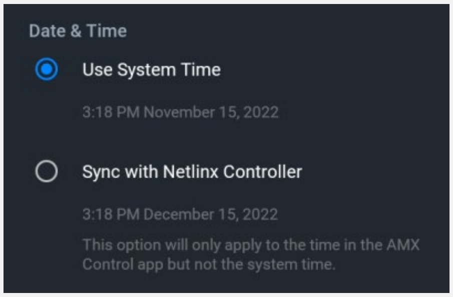
○ Sync with Netlinx Controller

3:10 PM December 15, 2022

This option will only apply to the time in the AMX Control app but not the system time.

Licenses

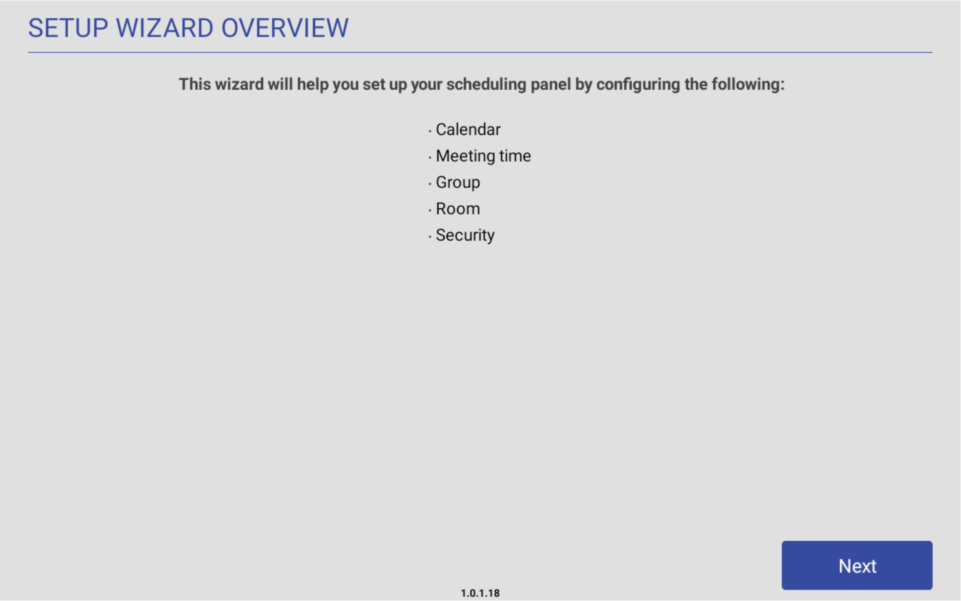
Open-source licenses

System	
AMX Control Settings Management	This allows management of the settings in the AMX G5 Control persona:  AMX Control Settings Management Reset Load from USB Save to USB **Reset** will revert all AMX Control settings to their default values. **Load from USB** will allow a panel config file to be loaded via connected storage device. **Save to USB** saves the current panel configuration to USB. **Note:** USB drives should be formatted to FAT32 file system.
Date & Time	Allows syncing with either the touch panel system date & time or the Netlinx controller date & time.  Date & Time ● Use System Time 3:18 PM November 15, 2022 ○ Sync with Netlinx Controller 3:18 PM December 15, 2022 This option will only apply to the time in the AMX Control app but not the system time. Note: this will only apply to the AMX Control persona.
Licenses	This displays all the open sources licenses being utilized, if applicable.

AMX Book

Overview

The Overview page presents a summary of the steps that entail the Scheduling Panel Setup Wizard. Press [Next] to start.



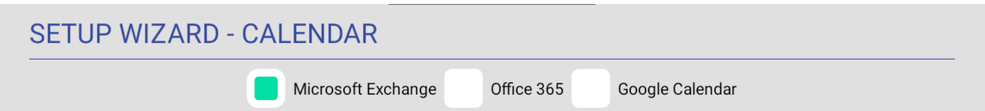
SETUP WIZARD OVERVIEW

This wizard will help you set up your scheduling panel by configuring the following:

- Calendar
- Meeting time
- Group
- Room
- Security

Calendar

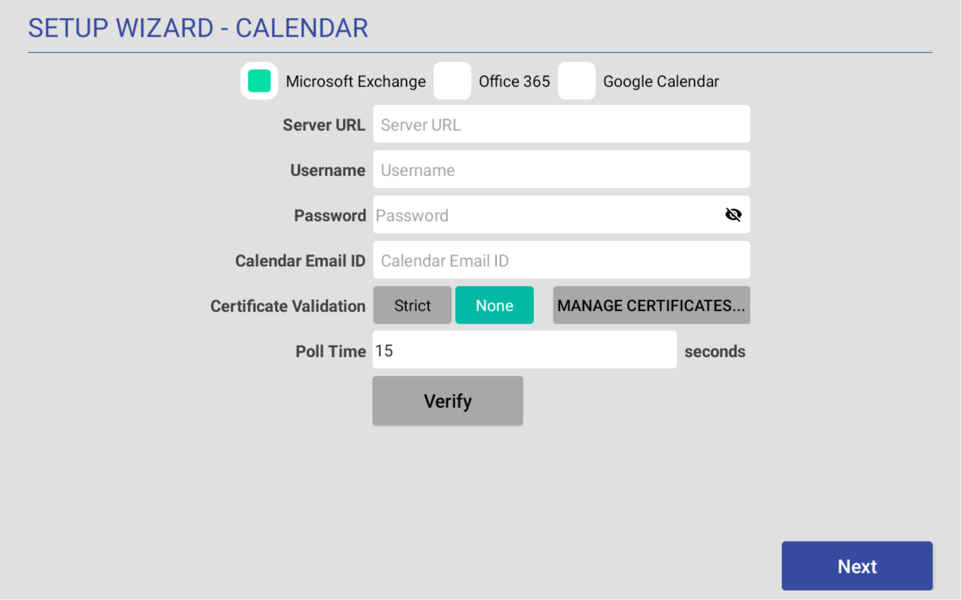
Use the options in the CALENDAR page to configure the scheduling panel's connection to the scheduling system.



SETUP WIZARD - CALENDAR

☑ Microsoft Exchange □ Office 365 □ Google Calendar

Microsoft Exchange



SETUP WIZARD - CALENDAR

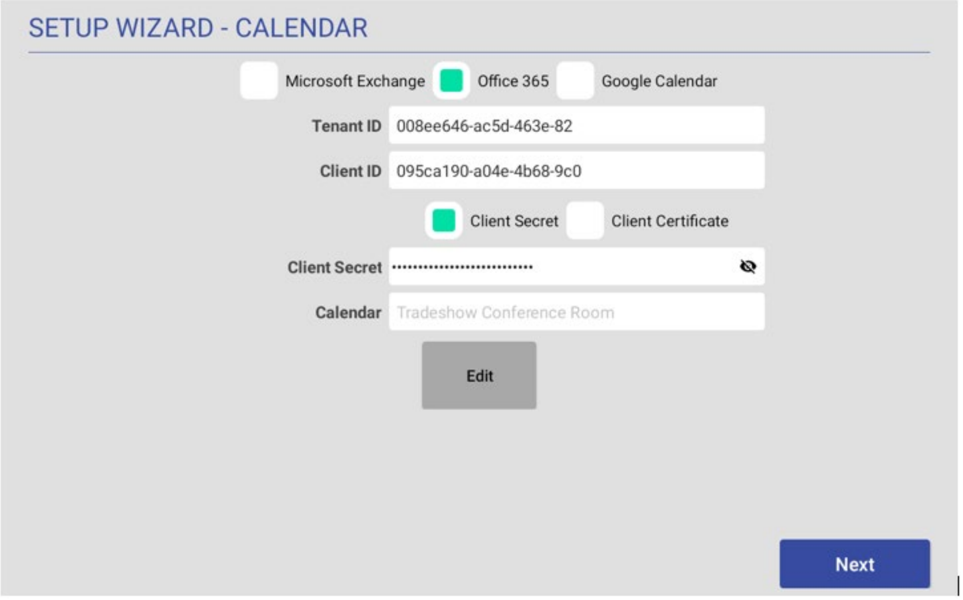
☑ Microsoft Exchange □ Office 365 □ Google Calendar

Server URL	Server URL
Username	Username
Password	Password
Calendar Email ID	Calendar Email ID
Certificate Validation	Strict [✓ None] MANAGE CERTIFICATES...
Poll Time	15 **seconds**

Microsoft Exchange Calendar Configuration Settings

Server URL	Enter the full URL for the scheduling server. Example syntax: https://<EXCHANGE_SERVER_HOSTNAME>/EWS/Exchange.asmx
Username	Enter the Username (including domain) required to login to the scheduling server (as required). Example: "JaneDoe@acme.onmicrosoft.com".
Password	Enter the Password required to login to the scheduling server (as required).
Calendar Email ID	Enter the Email ID (including domain) used by the scheduling service. Example: "ConfRoom1@acme.onmicrosoft.com"
Certificate Validation	• Select **Strict** to include hostname validation and apply certain certificate requirements - see *SSL Validation Schemes* (below). • Select **None** to disable validation schemes - see *SSL Validation Schemes* (below). • Select **Manage Certificates** to open the *Security* (Advanced Settings) page, which provides access to *Credential Storage* options, as described in the *Installing Certificates* section.
Poll Time	•

Office 365



SETUP WIZARD - CALENDAR

□ Microsoft Exchange ☑ Office 365 □ Google Calendar

Tenant ID: 008ee646-ac5d-463e-82

Client ID: 095ca190-a04e-4b68-9c0

☑ Client Secret □ Client Certificate

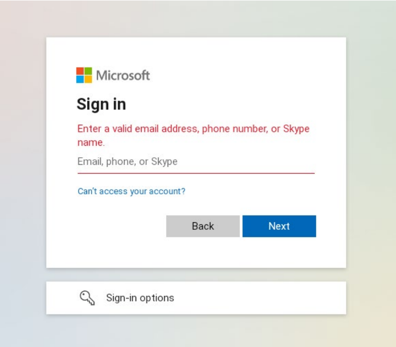
Client Secret: ••••••••••••••••••••••••••

Calendar: Tradeshow Conference Room

Office 365 Calendar Configuration Settings	
Tenant ID	Enter the required information regarding your O365 Tenant. Please refer to your Azure Portal https://portal.azure.com for more information and see Appendix F for steps to find this ID.
Client ID	Enter the required information regarding your O365 Client. Please refer to your Azure Portal https://portal.azure.com for more information and see Appendix F for steps to find this ID.
Client Secret	Enter the required information regarding your O365 Client Secret. Please refer to your Azure Portal https://portal.azure.com for more information and see Appendix F for steps to find this ID.
Certificate	You may select Client Certificate instead of using a Client Secret. Upload the required certificate here. Please refer to your Azure Portal https://portal.azure.com for more information and see Appendix F for steps to find this ID.
Verify	Once entered, the wizard will attempt a connection to your O365 server. You will be prompted to log in with your email & password as normal when accessing Microsoft services. Depending on your configurations and the calendars to which you have access, you may be prompted to select a calendar.

NOTE: Only Modern Authentication is supported. Support for Basic Authentication has ended and will not be supported.

Microsoft login & calendar selection:



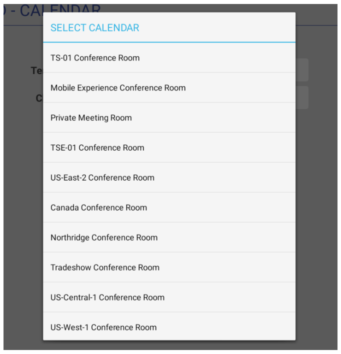
Microsoft

Sign in

Enter a valid email address, phone number, or Skype name.

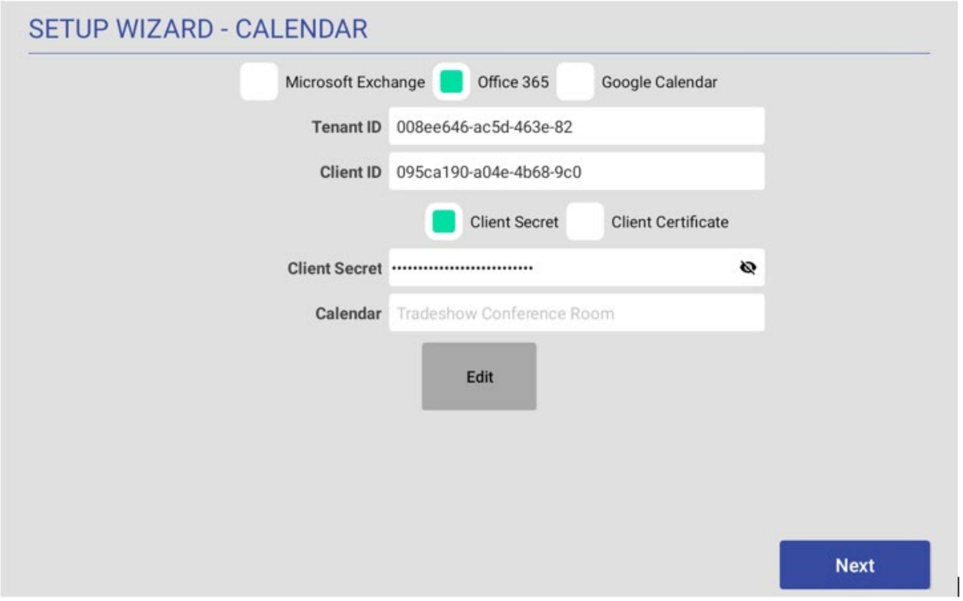
Email, phone, or Skype

→



SELECT CALENDAR
TS-01 Conference Room
Mobile Experience Conference Room
Private Meeting Room
TSE-01 Conference Room
US-East-2 Conference Room
Canada Conference Room
Northridge Conference Room
Tradeshow Conference Room
US-Central-1 Conference Room
US-West-1 Conference Room

A correctly setup mailbox will look like this:



SETUP WIZARD - CALENDAR

□ Microsoft Exchange ■ Office 365 □ Google Calendar

Tenant ID: 008ee646-ac5d-463e-82

Client ID: 095ca190-a04e-4b68-9c0

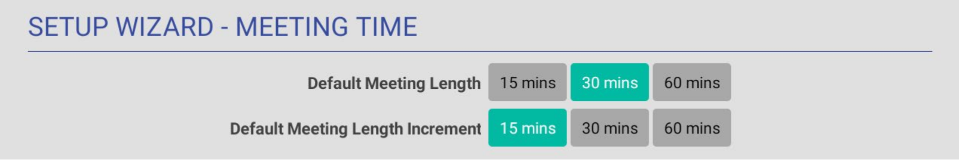
■ Client Secret □ Client Certificate

Client Secret: •••••••••••••••••••••••••••••

Calendar: Tradeshow Conference Room

Meeting Time

Use the options in the MEETING TIME page to customize default meeting & timing options.



SETUP WIZARD - MEETING TIME

Default Meeting Length: 15 mins [✓ 30 mins] 60 mins

Default Meeting Length Increment: [✓ 15 mins] 30 mins 60 mins

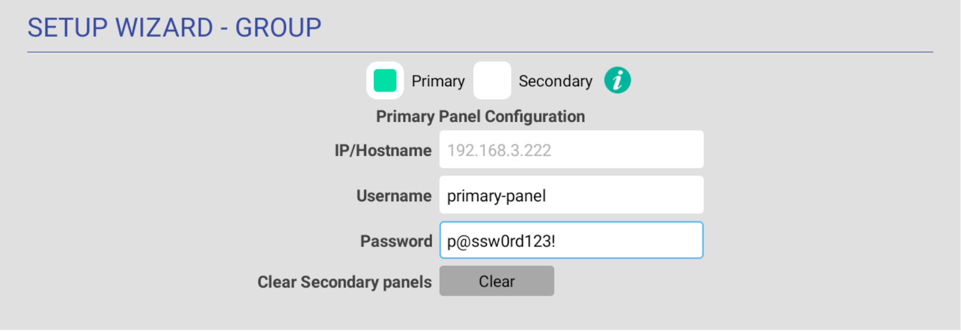
Setup Wizard - LOCALIZATION page options	
Default Meeting Length	This setting specifies how many times each hour this panel can indicate separate meetings. The default setting is *15 minutes*, in which case this panel can schedule up to 4 meetings in an hour. To change the default meeting increment setting, press either the **30 mins** or **60 mins** button. • With *30 mins* selected, this panel can schedule a meeting every 30 minutes. With *60 mins* selected, this panel can schedule a meeting every 60 minutes.
Default Meeting Increment	Indicates the current default meeting length setting for this panel (default = *30 minutes*). To change the default meeting length setting, press either the **15 mins** or **60 mins** button.

Group

AMX Book touch panels can be configured either as a Primary or Secondary, as it relates to a Scheduling Panel Group. Secondary panels communicate with the Primary panel, which enables panels to browse the schedules of panels in the same Group. This provides users with the ability to schedule a meeting in any room included in the Scheduling Panel Group, from any panel in the Group. For example, when a room is occupied, users can browse other rooms and schedule a meeting in any of the rooms in the Group.
Use the options in the GROUP page to configure this panel as either the Primary or Secondary of a Scheduling Panel Group. Note that there can be only one Primary panel for a Group. If there is only one stand-alone AMX Book panel, then it must be configured as a Primary.

NOTE: Up to 49 AMX Book panels can be configured as Members of a Primary panel's Group (for a total of 50 panels including the Primary). However, note that when configuring Scheduling Groups, smaller Room Groups typically have quicker responsiveness for "Browse Rooms" requests.

NOTE: Using Screen Timeout is not recommended when using Groups. Screen Timeout will cause panels in groups (both primary & secondary) to appear offline.



SETUP WIZARD - GROUP

☑ Primary □ Secondary

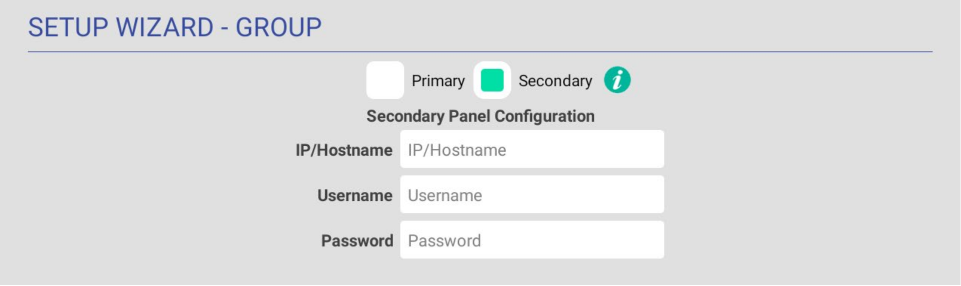
Primary Panel Configuration

IP/Hostname: 192.168.3.222

Username: primary-panel

Password: p@ssw0rd123!

Clear Secondary panels: Clear



SETUP WIZARD - GROUP

□ Primary ☑ Secondary

Secondary Panel Configuration

IP/Hostname: IP/Hostname

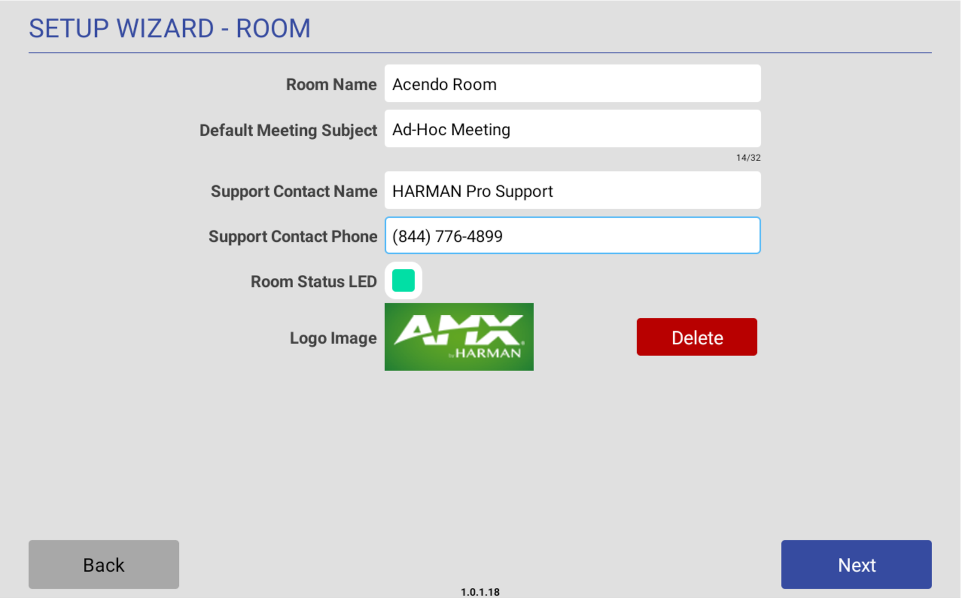
Username: Username

Password: Password

Setup Wizard - GROUP page options	
Primary/Secondary	Specify to use this panel either as a Primary panel or as a Secondary panel, as part of a Scheduling Panel Group (default = *Primary*).
Primary Configuration	With *Primary* selected, the following Primary Configuration options are presented:
IP/Hostname	This read-only field displays the Primary IP/Hostname currently assigned to this panel.
Username	Enter a Username to associate with this panel (required for Primary panels).
Password	Enter a Password to associate with this panel (required for Primary panels).
Clear Secondary Panels	Press to clear any and all members of the Group with which this Primary panel is associated.
Secondary Configuration	With *Secondary* selected, the following Secondary Configuration options are presented:
IP/Hostname	Enter the Primary IP/Hostname of the Primary panel for the Group to which this Secondary panel belongs.
Username	Enter the Username associated with the Primary panel for this panel's Group.
Password	Enter the Passwords associated with the Primary panel for this panel's Group.

Room

Use the options in the ROOM pages to set room-level options for this panel



SETUP WIZARD - ROOM

Room Name: Acendo Room

Default Meeting Subject: Ad-Hoc Meeting

14/32

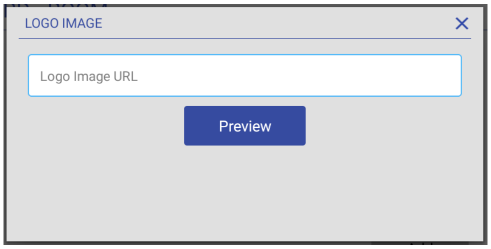
Support Contact Name: HARMAN Pro Support

Support Contact Phone: (844) 776-4899

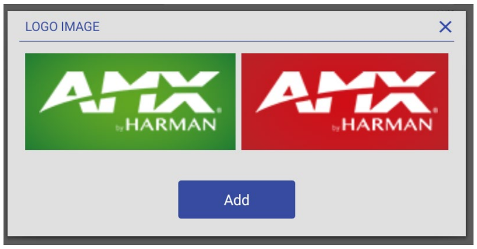
Room Status LED: 🟩

Logo Image

Setup Wizard - ROOM page options	
Room Name	Enter the room name as it will appear on this panel.
Default Meeting Subject	Enter a default meeting subject to be displayed on this panel. Note that this default subject can be over-written in the Room page (see the *Editing Meeting Details* section).
Support Contact Name	Enter the name of the support personnel as it will appear on this panel.
Support Contact Phone	Enter the phone number of the support personnel as it will appear on this panel.
Room Status LED	Press to toggle the Room Status LEDs on the panel, which light green when the Room is available, or red when the Room is occupied (default = *enabled*).
Logo Image	Indicates the current logo image, if applicable. Press **Add** to add a custom logo image. • Supported Formats: JPG, PNG • The custom logo image window is 375 x 165. AMX Book Panels will take the downloaded image and scale it down to fit in the window. However, to maximize the available space, the downloaded logo should have a similar aspect ratio (~2.2).

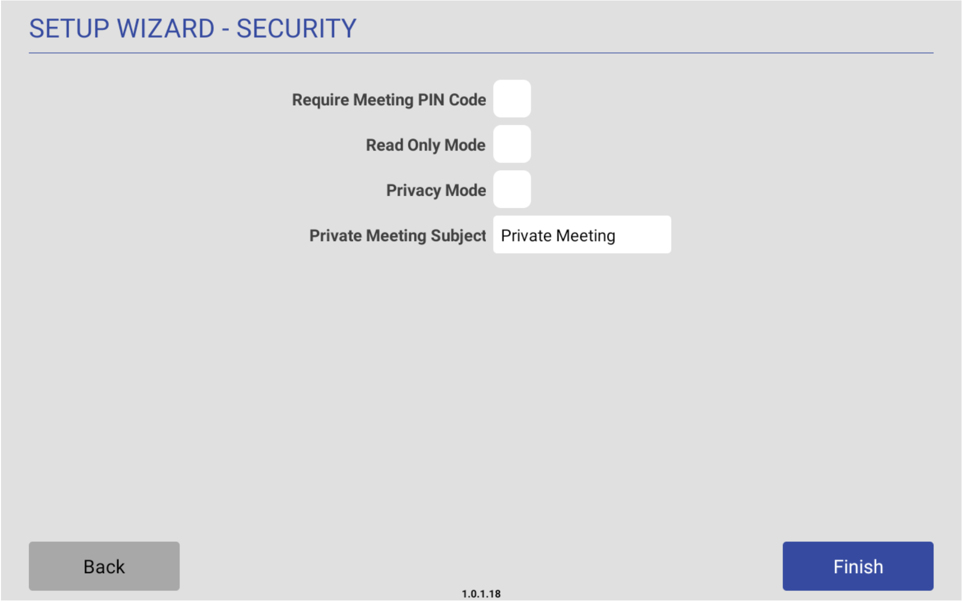


→



Security

Use the options in the SECURITY pages to set security & privacy options for this panel.



SETUP WIZARD - SECURITY

Require Meeting PIN Code □

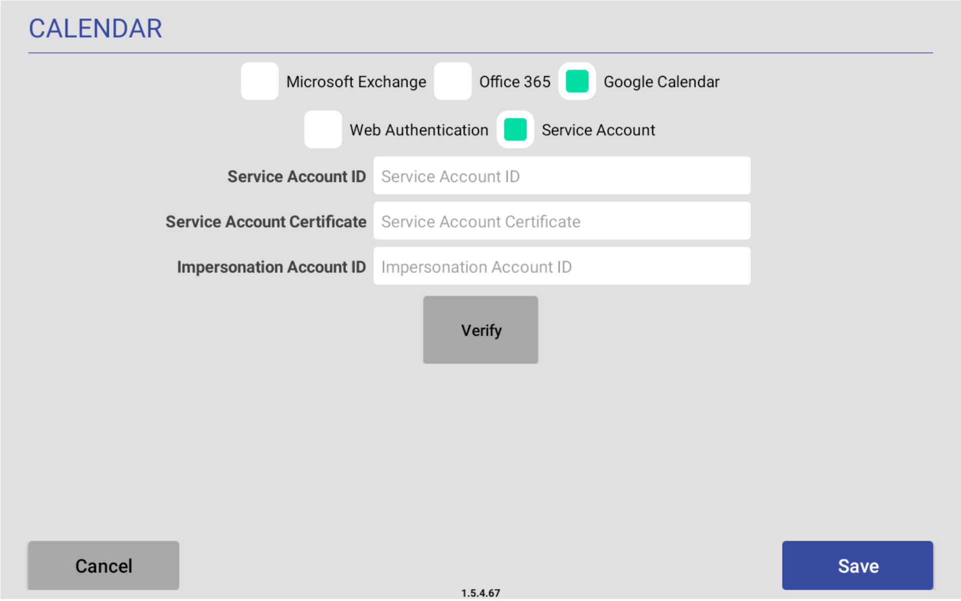
Read Only Mode □

Privacy Mode □

Private Meeting Subject: Private Meeting

Setup Wizard - SECURITY page options	
Require Meeting PIN Code	When enabled, the PIN code will be required to reserve a meeting, edit Meeting Time/Details and Delete Meeting. Press to enable this option (default = *disabled*).
Meeting PIN Code	Enter the 4-digit PIN code that will be required to reserve, edit or delete a meeting on this panel, only if the *Require Meeting PIN Code* option (above) is enabled.
Read Only Mode	Press to toggle this option (default = *disabled*). When enabled, all fields on the Room page are displayed, but are read-only. In this case, users will be able to see all meetings and reservation details, but will not be allowed to reserve, edit or delete meetings on this panel.
Privacy Mode	Press to toggle this option (default = *disabled*). When in Privacy Mode, information considered to be private is hidden and immutable on this panel. See the *Privacy Mode* section for details. *Note: If a meeting is booked outside of an AMX Book panel (i.e directly via the scheduling system software), then the subject line of the meeting will be replaced with the Private Meeting Subject text (default = "Private Meeting"). The default Private Meeting Subject text can be edited if desired.*
Private Meeting Subject	Press to enter the text that will be used as the meeting subject displayed for private meetings. The text entered here provides the meeting subject for all types of private meetings: • It will be used when *Privacy Mode* is enabled on the AMX Book panel. • Will be used when a meeting is marked as *Private* by the calendar server (Exchange/Office 365/Google).
Finish	Press to exit the Setup Wizard (see *Exiting the Scheduling Panel Setup Wizard*).

Google Calendar



CALENDAR

□ Microsoft Exchange □ Office 365 ☑ Google Calendar

□ Web Authentication ☑ Service Account

Service Account ID: Service Account ID

Service Account Certificate: Service Account Certificate

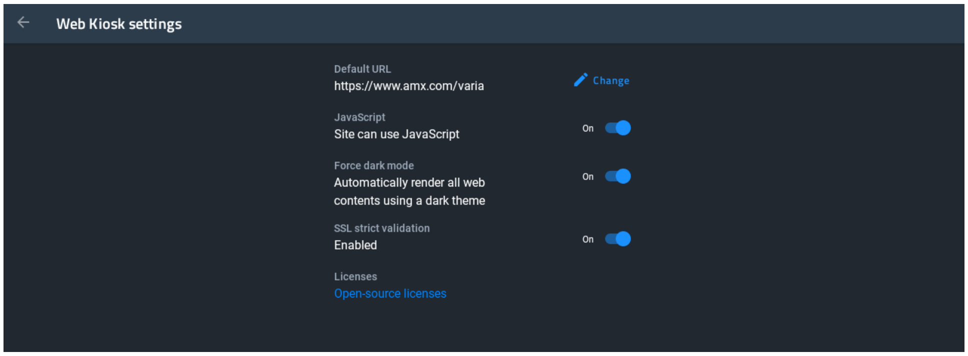
Impersonation Account ID: Impersonation Account ID

Google Calendar Configuration Settings	
Web Authentication	Log into your Google Account via web browser and assign a calendar resource.
Service Account	Use login credentials to assign a calendar resource. A Service Account is an account that belongs to an application instead of to an individual user. AMX Book will call Google APIs on behalf of the service account. Please also see Appendix G for more information.
Service Account ID	Enter your Service Account ID.
Service Account Certificate	Enter Service Account Certificate
Impersonation Account ID	Enter your Service Account Impersonation ID.

Web Kiosk

Overview

These are settings available for the Web Kiosk persona.



Web Kiosk settings

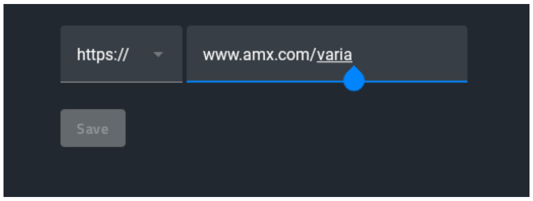
Default URL
https://www.amx.com/varia
Change

JavaScript
Site can use JavaScript
On

Force dark mode
Automatically render all web contents using a dark theme
On

SSL strict validation
Enabled
On

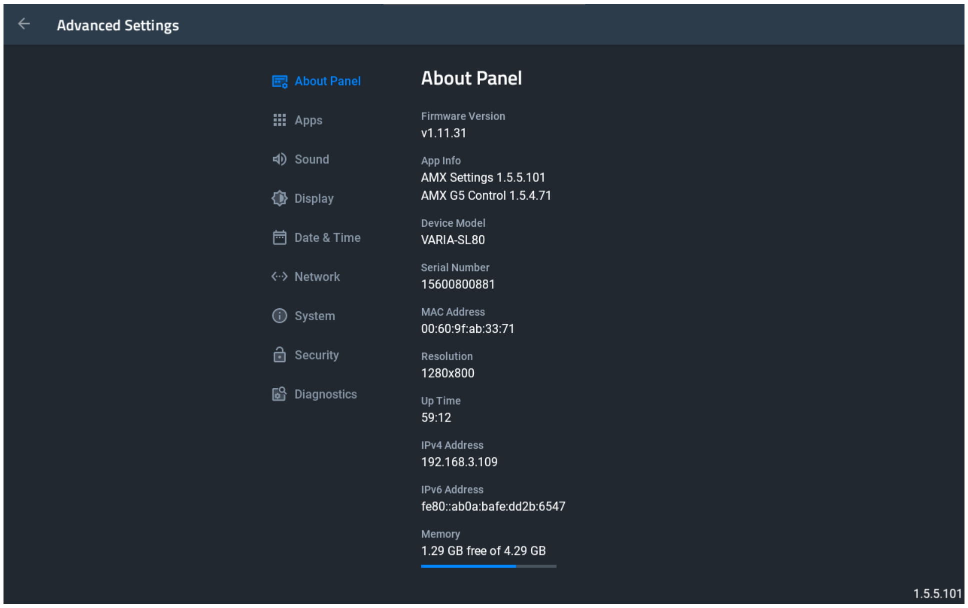
Licenses
Open-source licenses

Web Kiosk Settings	
Default URL	The URL that the Kiosk will access. To make adjustments, press [Change]. Change Use the dropdown to select **https://** (preferred) or **http://**. Be careful not to repeat this protocol in the URL field.  https:// www.amx.com/varia Save To save your changes, Press [Save]. To exit without saving, press the back arrow on the top-left of the screen.
JavaScript	Allow the kiosk to use JavaScript
Force Dark Mode	Automatically render all web content using a dark theme (website-dependent).
SSL Strict Validation	Enable or disable strict SSL certificate validation
Licenses	This displays all the open sources licenses being utilized, if applicable.

Advanced Settings

Overview

About Panel



About Panel

Firmware Version
v1.11.31

App Info
AMX Settings 1.5.5.101
AMX G5 Control 1.5.4.71

Device Model
VARIA-SL80

Serial Number
15600800881

MAC Address
00:60:9f:ab:33:71

Resolution
1280x800

Up Time
59:12

IPv4 Address
192.168.3.109

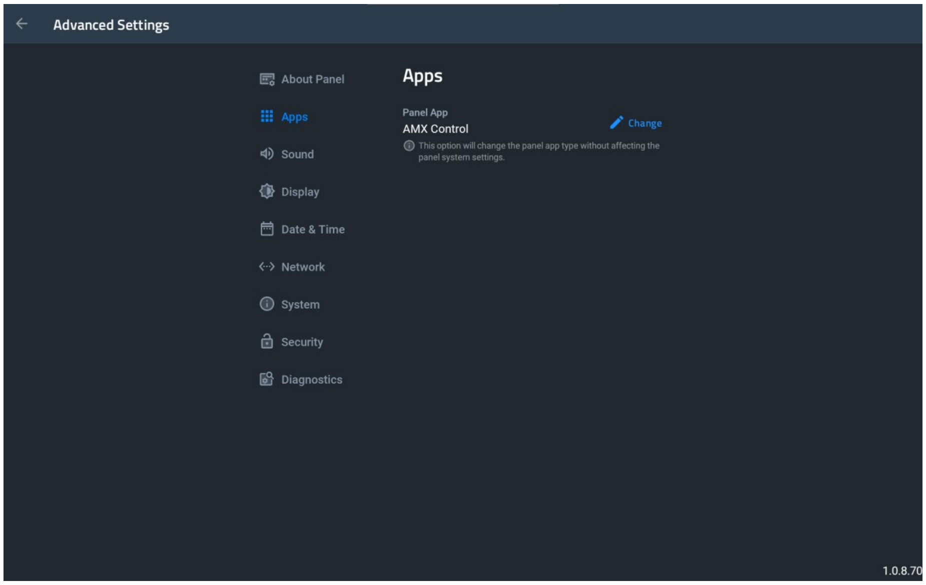
IPv6 Address
fe80::ab0a:bafe:dd2b:6547

Memory
1.29 GB free of 4.29 GB

1.5.5.101

About Panel	
Firmware Version	Displays the panel's FW version
App Info	Displays the app versions for Settings, and for the currently selected persona
Device Model	Displays the panel's model number.
Serial Number	Displays the specific serial number value assigned to the panel.
MAC Address	Displays the panel MAC address
Resolution	Displays the panel's screen width & height in pixels.
Up Time	Displays the time elapsed since the panel was last started.
IPv4 Address	Displays the panel's IP address (v4).
IPv6 Address	Displays the panel's IP address (v6).
Memory	Displays the panel's used & total memory (RAM).
Storage	Displays the panel's used & total storage (eMMC)
Licenses	This displays all the open sources licenses being utilized.

Apps



Apps

Panel App
AMX Control

This option will change the panel app type without affecting the panel system settings.

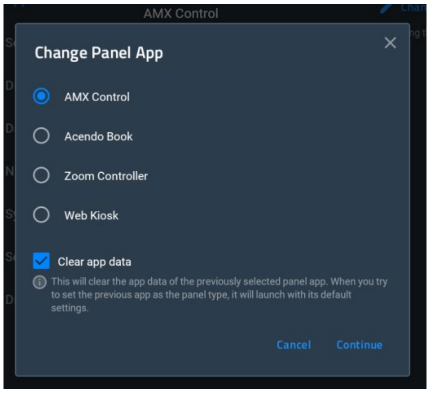
Apps

Panel App

Displays to panel's currently selected Persona, and gives the option to change to a new persona.

Change

By pressing [Change], you may choose a new panel persona. Select the desired Persona and press [Continue].



Change Panel App

● AMX Control
○ Acendo Book
○ Zoom Controller
○ Web Kiosk

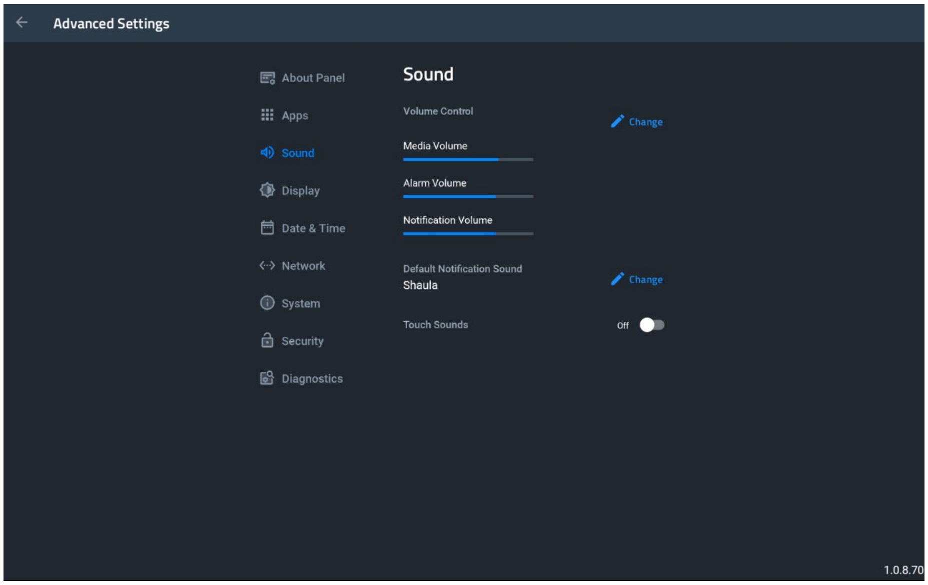
☑ Clear app data

This will clear the app data of the previously selected panel app. When you try to set the previous app as the panel type, it will launch with its default settings.

If you wish to exit without selecting a new persona, press [Cancel] or the [x] button.

Clear App Data will clear or preserve (checked or unchecked, respectively) the app data of the **current** persona, before switching to the new persona. This is helpful if you ever wish to return to the previous persona.

Sound



Sound

Volume Control

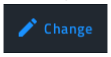
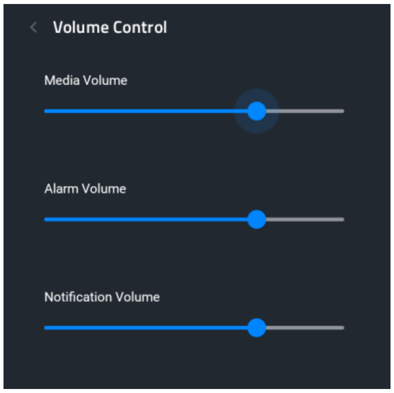
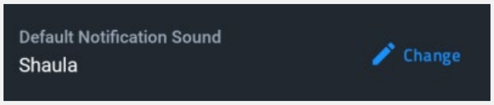
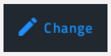
Media Volume

Alarm Volume

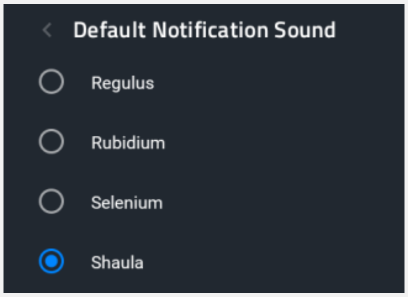
Notification Volume

Default Notification Sound
Shaula

Touch Sounds Off

Sound	
Volume Control	Displays the current volume level for media, alarm, & notifications. If adjustment is needed, press the [Change] button.  Change Press or drag your finger across the slider and let go at the desired volume level. A sound will play at that new volume level to confirm loudness.  Volume Control Media Volume Alarm Volume Notification Volume Press the Back arrow [<] to return.
Default Notification Sound	Changes the default notification sound on the panel.  Default Notification Sound Shaula Change To select a notification sound, press [Change].  Change

Sound



< Default Notification Sound

○ Regulus

○ Rubidium

○ Selenium

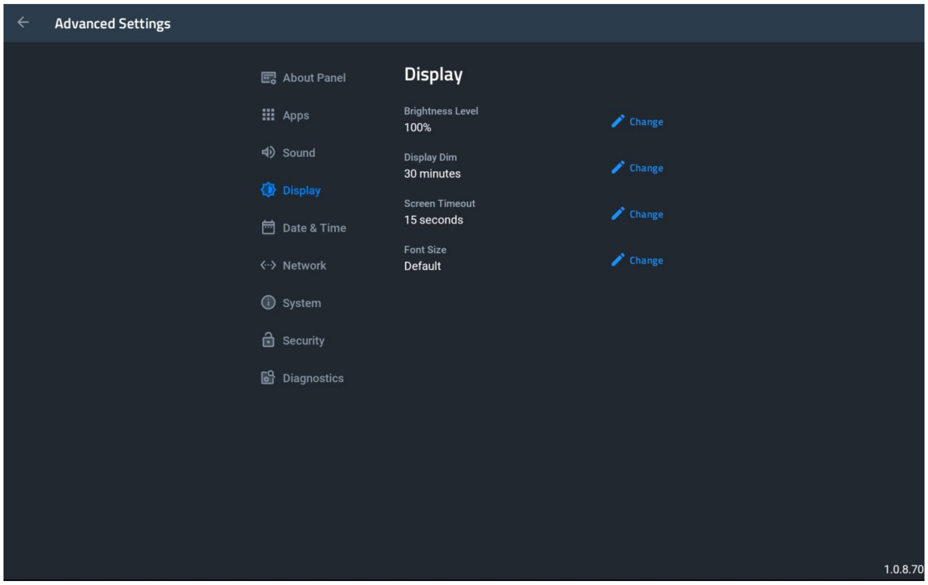
● Shaula

Press the Back arrow [<] to return.

Note: this is not the *Button Hit* or *Button Miss* sound for the AMX G5 Control persona. Button Hit & Button Miss can be found in G5 Settings.

Touch Sounds	Displays the specific serial number value assigned to the panel.

Display



Display

Brightness Level: 100% — Change

Display Dim: 30 minutes — Change

Screen Timeout: 15 seconds — Change

Font Size: Default — Change

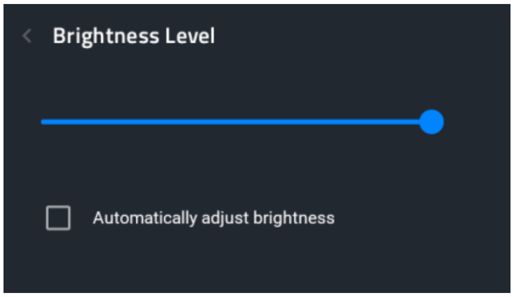
Display

Brightness

Contains settings for panel brightness & Auto Display Brightness If adjustment is needed, press the respective [Change] button.

Change

Press or drag your finger across the slider and let go at the desired brightness level. The panel will react to your changes when you lift your finger.



□ Automatically adjust brightness

When **Automatically Adjust Brightness** is enabled, the panel will automatically adjust its brightness relative to the light levels in the environment.

Press the Back arrow [<] to return.

Display Dim

Display dim will lower the brightness level after a period of inactivity.

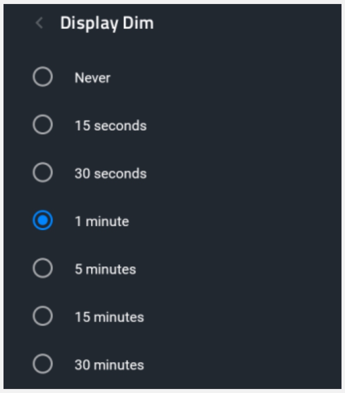
To adjust the time interval before dimming, press [Change].

Change

Select a new time, or select Never to keep the screen at full brightness.

Screen Timeout (below) starts after Display Dim. If Never is selected here, the panel will also never timeout (ie. display turn off).

Display



< Display Dim

○ Never

○ 15 seconds

○ 30 seconds

● 1 minute

○ 5 minutes

○ 15 minutes

○ 30 minutes

Press the Back arrow [<] to return.

.

Screen Timeout

Screen timeout will turn off the display after a period of inactivity. This time is **in addition to** the Display Dim value selected.

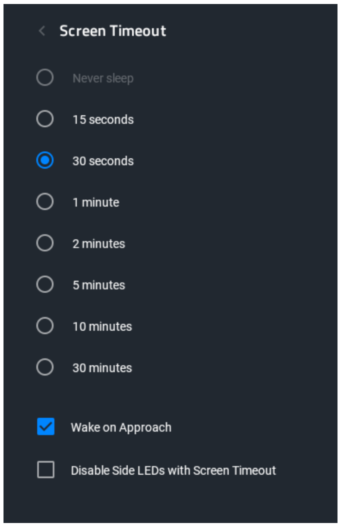
To adjust the time interval before timeout, press [Change].

Change

Select a new time, or select Never to keep the screen on at all times.

Wake On Approach will utilize the proximity sensor to wake the panel up when someone approaches it.

Disable LEDs will turn the side LEDs off with Screen Timeout.



< Screen Timeout

○ Never sleep

○ 15 seconds

● 30 seconds

○ 1 minute

○ 2 minutes

○ 5 minutes

○ 10 minutes

○ 30 minutes

☑ Wake on Approach

☐ Disable Side LEDs with Screen Timeout

Press the Back arrow [<] to return.

Note: Some apps may override these settings. For example, the Zoom Rooms Controller (ZRC) app ignores Screen Timeout values.

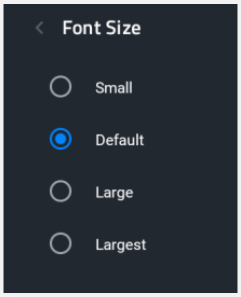
Font Size

The font size can be increased or decreased to your desired level. To adjust, press [Change].

Change

Select a new font size from the choices.

Display



< Font Size

○ Small

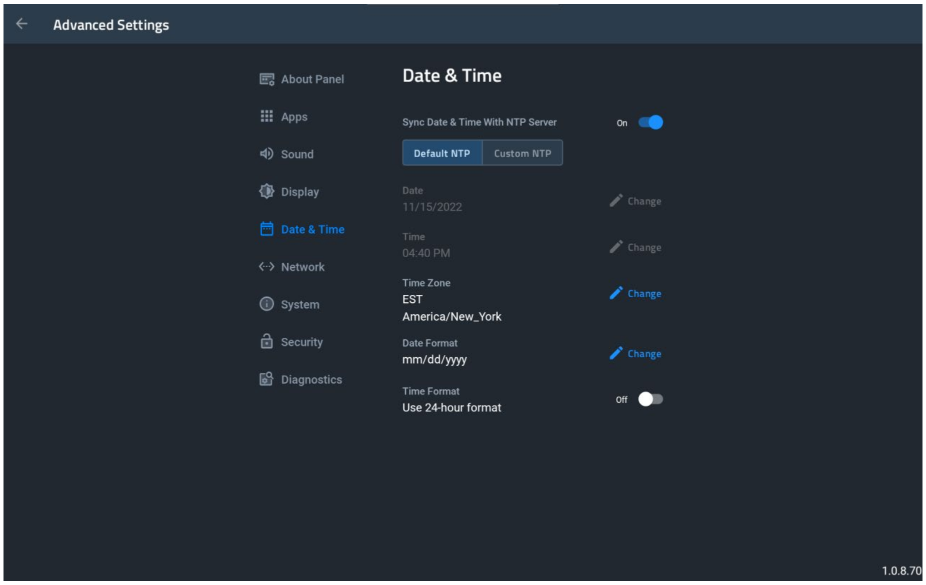
● Default

○ Large

○ Largest

Press the Back arrow [<] to return.

Date & Time



Date & Time

Sync Date & Time With NTP Server On

Default NTP | Custom NTP

Date 11/15/2022 Change

Time 04:40 PM Change

Time Zone EST America/New_York Change

Date Format mm/dd/yyyy Change

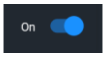
Time Format Use 24-hour format Off

Date & Time

Sync
Date & Time
with NTP Server

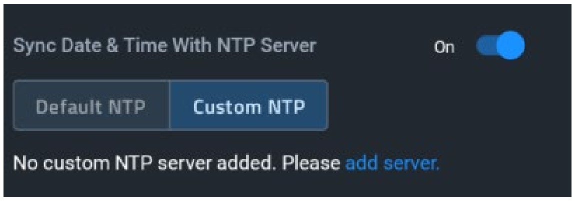
When enabled, the touch panel's date & time will be synced to either the default NTP (Network Time Protocol) server, or a custom NTP server (eg. *time.google.com*)

To enable, turn the radio button ON.



For Default NTP, there are no additional settings to enter. The panel will periodically check the NTP server for date & time. A reboot may be required if you want immediate synchronization. If your panel's date & time are drastically different than actual, it may take up to 24 hours or more for the correct time to be synchronized.

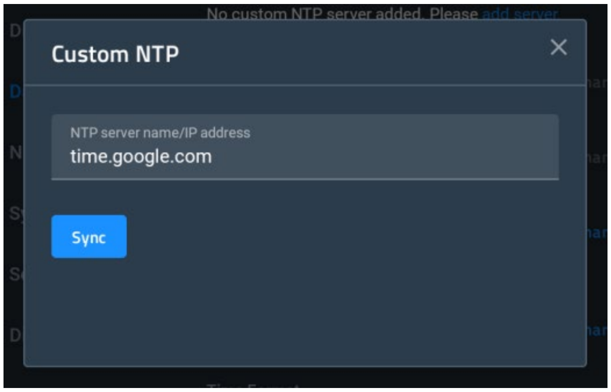
For Custom NTP, select this option and then press [add server].



Sync Date & Time With NTP Server On

No custom NTP server added. Please add server.

Enter the custom NTP server URL and press [Sync].



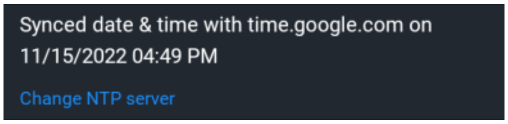
Custom NTP

NTP server name/IP address: time.google.com

Date & Time

The panel will attempt to connect and list the last successful sync.

If you need to change the NTP, press [Change NTP server].


Synced date & time with time.google.com on 11/15/2022 04:49 PM

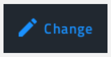
Change NTP server

To disable NTP and set the date & time manually, turn the radio button OFF.

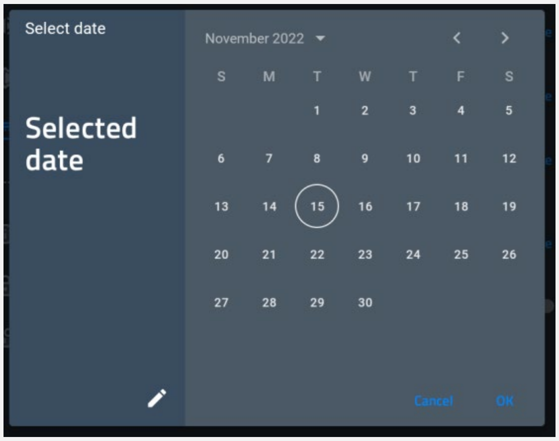
Date

When using NTP, the Date field is read-only.

With NTP disabled, adjust the date by pressing the [Change] button.



Use the calendar to select the date, either by using the calendar picker or by selecting the pencil icon to enter the numeric value.


Select date

Selected date

November 2022

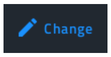
S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

To save your changes, press [OK]. To close without saving changes, press [Cancel].

Time

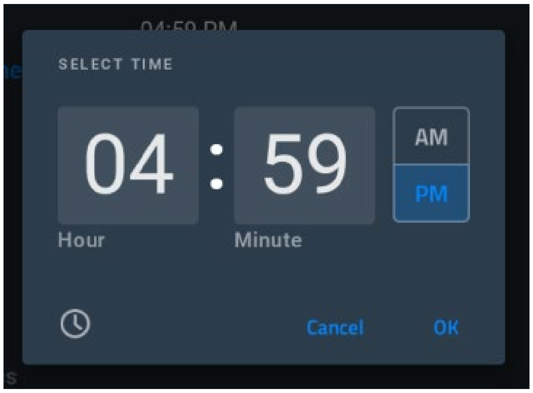
When using NTP, the Time field is read-only.

With NTP disabled, adjust the date by pressing the [Change] button.

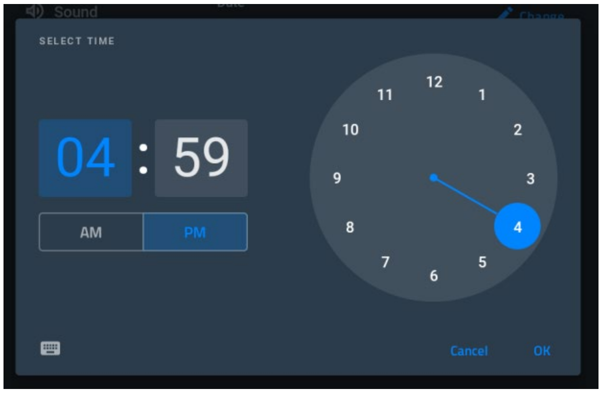


Press the Hours and Minutes fields to numerically enter the time, and appropriately select AM or PM.

Date & Time

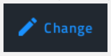


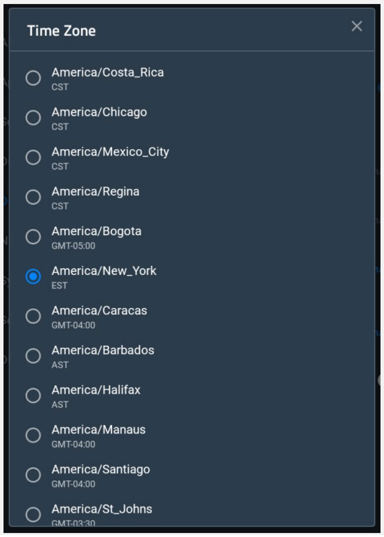
Or, press the clock icon on the lower-left to switch to the clock picker, and drag the clock hands to the correct hours & minutes, respectively. Then, appropriately select AM or PM.



To save your changes, press [OK]. To close without saving changes, press [Cancel].

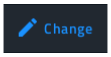
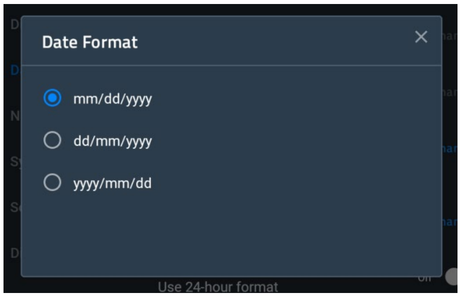
Time Zone	Whether using NTP or setting Date & Time manually, the Time Zone requires a selection. To adjust, press [Change]. Scroll through the list and select your time zone.



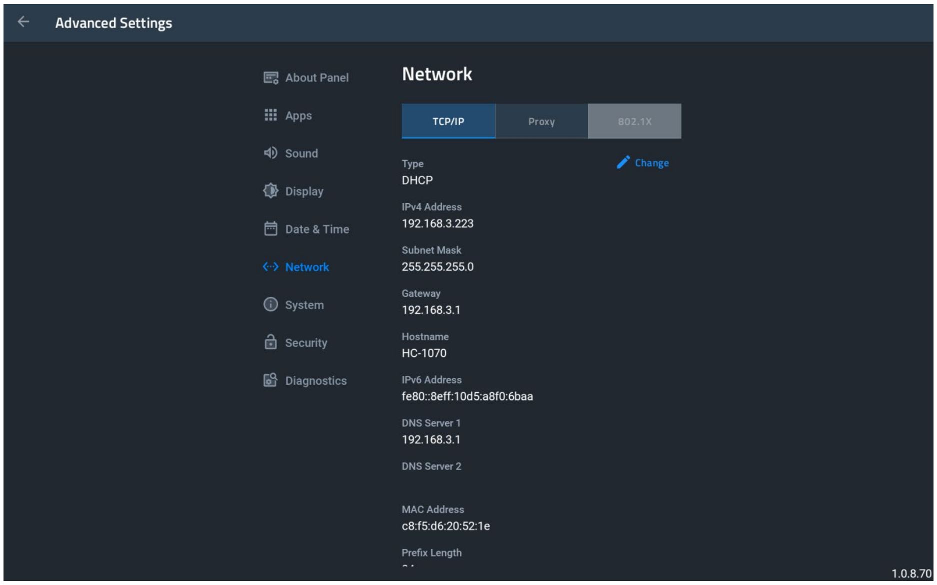


Time Zone ✕

○ America/Costa_Rica CST
○ America/Chicago CST
○ America/Mexico_City CST
○ America/Regina CST
○ America/Bogota GMT-05:00
● America/New_York EST
○ America/Caracas GMT-04:00
○ America/Barbados AST
○ America/Halifax AST
○ America/Manaus GMT-04:00
○ America/Santiago GMT-04:00
○ America/St_Johns GMT-03:30

Date & Time	
	Press the [x] to exit.
Date Format	Format the date to preferred arrangement. To make adjustments, press [Change]. Change Select the preferred date format. Date Format ● mm/dd/yyyy ○ dd/mm/yyyy ○ yyyy/mm/dd Use 24-hour format Press the [x] to exit.
Time Format	The time can be shown in either 12-hour or 24-hour format. Turn the radio button ON to enable 24-hour time format, or OFF to enable 12-hour time format.

Network



Advanced Settings

Network

TCP/IP | Proxy | 802.1X

Change

Type
DHCP

IPv4 Address
192.168.3.223

Subnet Mask
255.255.255.0

Gateway
192.168.3.1

Hostname
HC-1070

IPv6 Address
fe80::8eff:10d5:a8f0:6baa

DNS Server 1
192.168.3.1

DNS Server 2

MAC Address
c8:f5:d6:20:52:1e

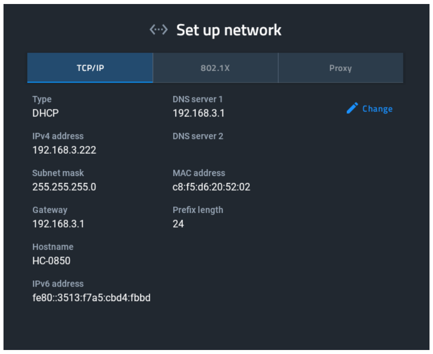
Prefix Length

1.0.8.70

Network

TCP/IP

IP address can be either IPv4 or IPv6, DHCP or static (IPv4 only).



Set up network

TCP/IP | 802.1X | Proxy

Change

Type
DHCP

IPv4 address
192.168.3.222

Subnet mask
255.255.255.0

Gateway
192.168.3.1

Hostname
HC-0850

IPv6 address
fe80::3513:f7a5:cbd4:fbbd

DNS server 1
192.168.3.1

DNS server 2

MAC address
c8:f5:d6:20:52:02

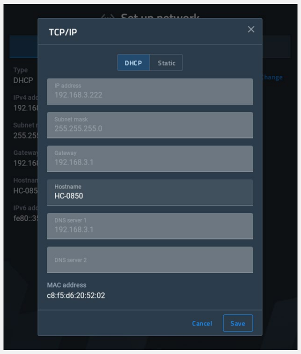
Prefix length
24

To make adjustments to IP address, press [Change].

Change

DHCP (preferred)

Network DHCP server will automatically assign IP address, subnet mask, default gateway, and DNS server info. In this mode, IP fields are read-only.



TCP/IP

DHCP | Static

IP address
192.168.3.222

Subnet mask
255.255.255.0

Gateway
192.168.3.1

Hostname
HC-0850

DNS server 1
192.168.3.1

DNS server 2

MAC address
c8:f5:d6:20:52:02

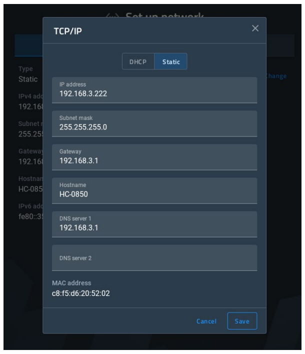
Cancel | Save

Press [Save] after making changes, or your newly entered network settings will not be saved.

Network

Static IP

IP address, subnet mask, default gateway, and DNS server info must be entered manually.



TCP/IP

DHCP | Static

IP address
192.168.3.222

Subnet mask
255.255.255.0

Gateway
192.168.3.1

Hostname
HC-0850

DNS server 1
192.168.3.1

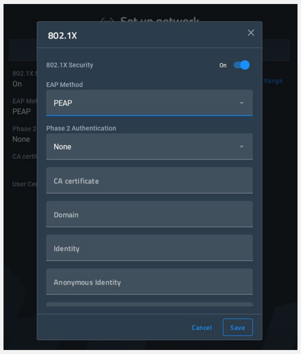
DNS server 2

MAC address
c8:f5:d6:20:52:02

Press [Save] after entering the information. Press [Cancel] to leave without saving any edits.

802.1X

If 802.1X is available/required, turn it [ON], select your EAP method, and enter the required info for that method (eg. certificate, domain, identity, password, etc.).



802.1X

802.1X Security: On

EAP Method
PEAP

Phase 2 Authentication
None

CA certificate

Domain

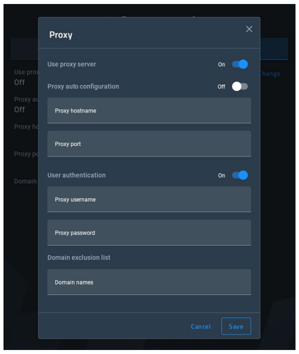
Identity

Anonymous Identity

For further assistance and detail on whether your network supports or requires 802.1X, please contact your (or your client’s) IT department.

Proxy

If a proxy is available/required, turn it [ON] and enter the appropriate configuration information.



Proxy

Use proxy server: On

Proxy auto configuration: Off

Proxy hostname

Proxy port

User authentication: On

Proxy username

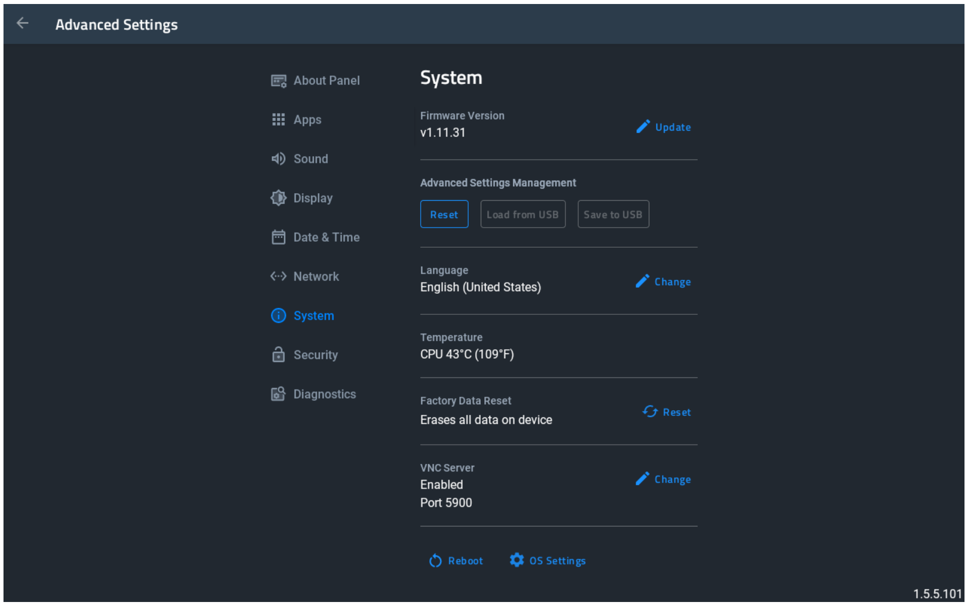
Proxy password

Domain exclusion list

Domain names

For further assistance and detail on whether your network supports or requires 802.1X, please contact your (or your client’s) IT department.

System



System

Firmware Version
v1.11.31 Update

Advanced Settings Management

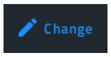
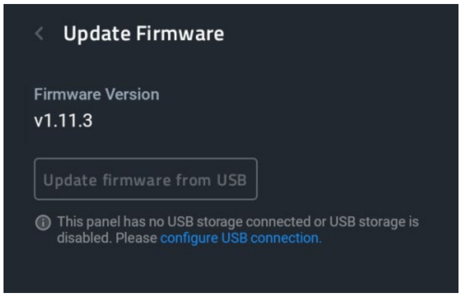
Language
English (United States) Change

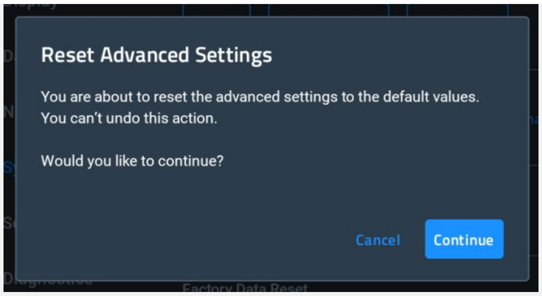
Temperature
CPU 43°C (109°F)

Factory Data Reset
Erases all data on device Reset

VNC Server
Enabled
Port 5900 Change

1.5.5.101

System	
Firmware	The currently loaded firmware version is shown. To update firmware via USB, press [Change].  Change Insert a USB drive and press [Update firmware from USB].  Update Firmware Firmware Version v1.11.3 Update firmware from USB This panel has no USB storage connected or USB storage is disabled. Please configure USB connection. The button will turn from gray to blue when the USB drive is inserted. Remember to Enable USB Storage in the System Settings menu. Update firmware from USB Follow the prompts to find the appropriate KIT file on the USB drive. Once completed, a reboot of the touch panel is recommended. Follow instructions in the USB Storage Settings section. **Note:** USB drives should be formatted to FAT32 file system.
Advanced Settings Management	The Advanced Settings can be reset to default values using the [Reset] button here.

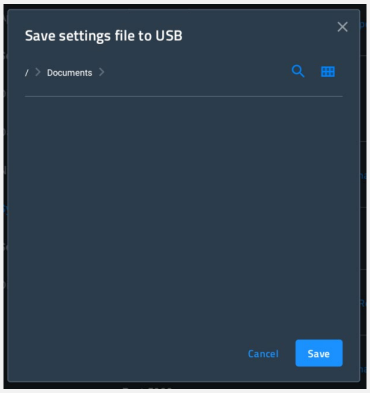


Reset Advanced Settings

You are about to reset the advanced settings to the default values. You can't undo this action.

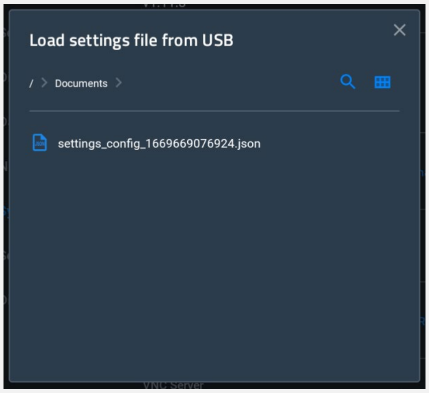
Would you like to continue?

To Save the current Advanced Settings configuration to USB, insert a USB Drive and press [Save to USB]. Select the desired folder location and press [Save].



Save settings file to USB

To load a previously saved settings configuration from USB, press [Load from USB]. Browse to the desired folder location and select the file.



Load settings file from USB

settings_config_1669669076924.json

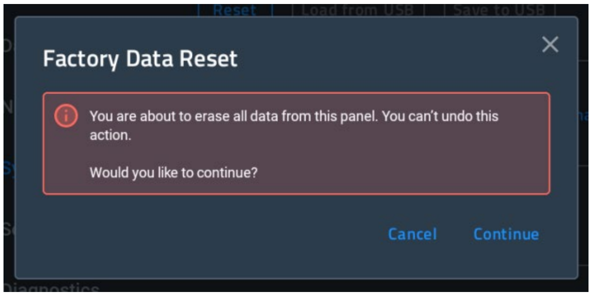
Once completed, a reboot of the touch panel is recommended. Follow instructions in the USB Storage Settings section.

Note: USB drives should be formatted to FAT32 file system.

Language	Select your desired language from the list. **Note**: this selection will only apply to the OS Settings and not to the selected Persona.
Temperature	CPU temperature can be monitored here. This field is read-only
Factory Data Reset	Pressing the [Reset] button will initiate a Factory Reset on the touch panel. Reset

This will revert the panel back to its default state, using the last firmware transferred.

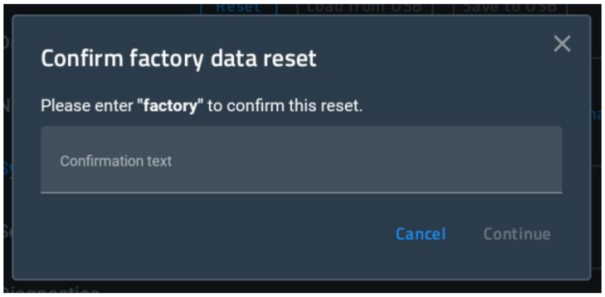
This cannot be undone. You will be prompted twice to confirm your choice.



Factory Data Reset

You are about to erase all data from this panel. You can't undo this action.

Would you like to continue?



Confirm factory data reset

Please enter "**factory**" to confirm this reset.

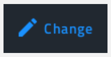
Confirmation text

To exit without resetting, press [Cancel].

VNC Server

The VNC Server is used to access the panel from another device, which is running a VNC client app.

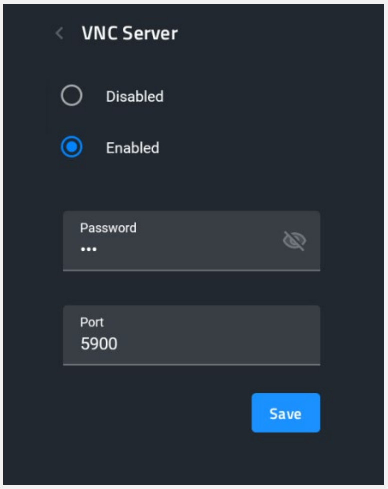
To make adjustments to the VNC settings, press [Change].



Enable or Disable the VNC Server by pressing the appropriate selection.

A password is optional, but recommended to secure your connection. You can use the “eye” icon to view or hide the password.

Port 5900 is defualt, but can be adjusted if required.



VNC Server

○ Disabled

● Enabled

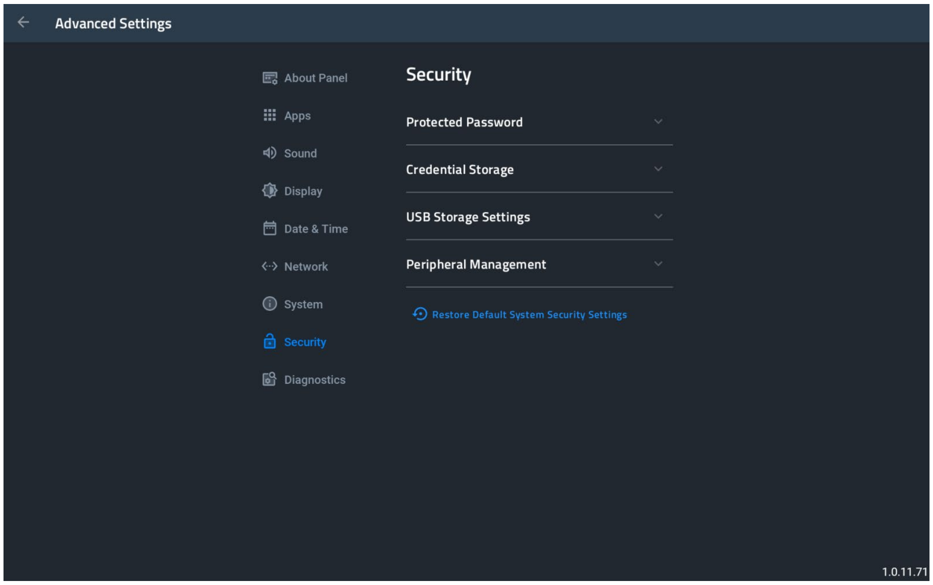
Password
•••

Port
5900

Changes are saved as they are made. To exit, press the back [<] button.
Note: During an active VNC session, the panel may not dim or enter Screen Timeout.

Reboot	Pressing [Reboot] will reboot the touch panel. This is the same as a press & hold of the pin-hole button for four (4) seocnds.
OS Settings	[OS Settings] will open the Android OS Settings menu. **Note:** Entering this area is for advanced use only, and not recommended or needed for normal operation. If you need assistance, please contact HARMAN Professional Service & Support: https://pro.harman.com/service

Security



Advanced Settings

- About Panel
- Apps
- Sound
- Display
- Date & Time
- Network
- System
- Security
- Diagnostics

Security

Protected Password

Credential Storage

USB Storage Settings

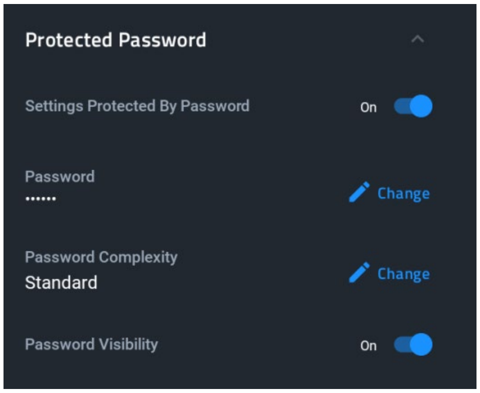
Peripheral Management

Restore Default System Security Settings

1.0.11.71

Security

Contains settings for panel's protected password, used to access the Advanced Settings menu.



Protected Password

Settings Protected By Password: On

Password: •••••• Change

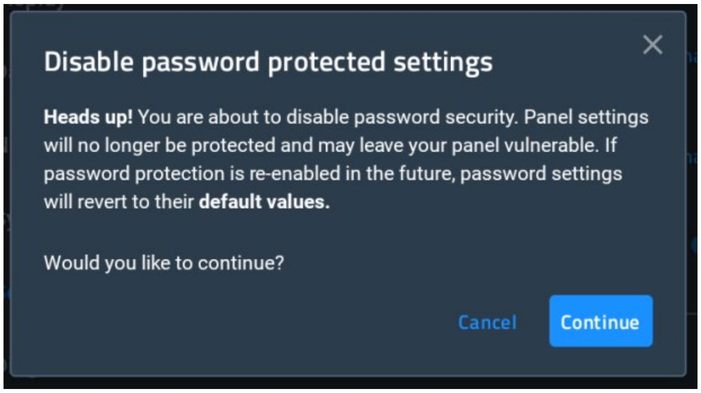
Password Complexity: Standard Change

Password Visibility: On

Protected Password

Settings Protected by Password will turn [ON] or [OFF] the password required to enter Advanced Settings. Use the button to select the preferred setting.

<u>**Note**</u>**:** Re-enabling the password after disabling it will revert to their default values.



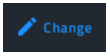
Disable password protected settings

Heads up! You are about to disable password security. Panel settings will no longer be protected and may leave your panel vulnerable. If password protection is re-enabled in the future, password settings will revert to their **default values.**

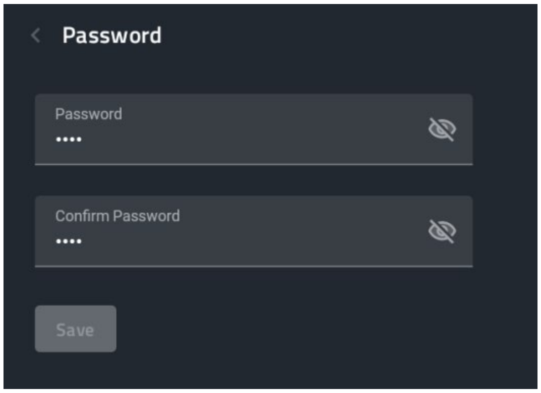
Would you like to continue?

The default PIN to access Panel Settings is 1988.

To change the password, press [Change] next to the current password value.

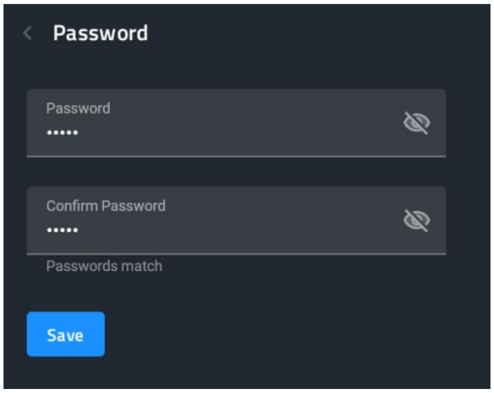


Enter a new password into both fields below. Requirements will vary based on the current *Password Complexity* (see below).



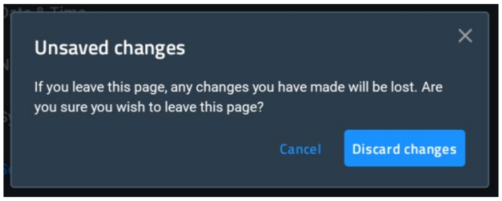
To reveal the passwords being entered, press the “eye.”

Passwords must be entered in both fields and must match for the [Save] button to become active (ie. blue).



Press [Save] to save the new password entry, or press the Back [<] button to exit/cancel.

You will need to confirm you want to discard your changes before exiting.



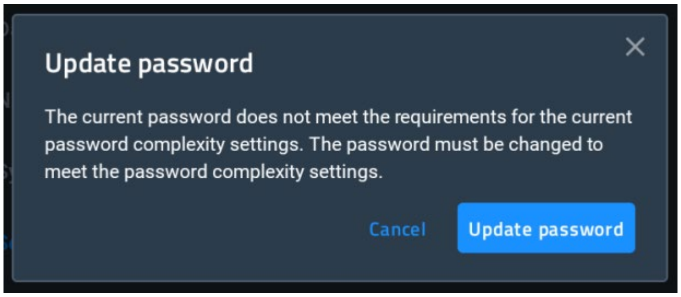
If you leave this page, any changes you have made will be lost. Are you sure you wish to leave this page?

Password Complexity dictates length, characters, and character usage rules.

- STANDARD - There are no complexity rules for a STANDARD complexity password. In this case, the password can be any length, including empty, and there are no minimum requirements for characters in the password.
- HIGH - HIGH complexity passwords must contain at least 15 characters such that the following conditions are met:

- The password must contain at least one uppercase alphabetic character.
- The password must contain at least one lowercase alphabetic character.
- The password must contain at least one numeric character.
- The password must contain at least one special character.
- The password must not contain more than three consecutive repeating characters.

Note: If the current password does not meet the high complexity password requirements, when this option is selected the panel will prompt you to change the current password to one that does meet the high complexity requirements.



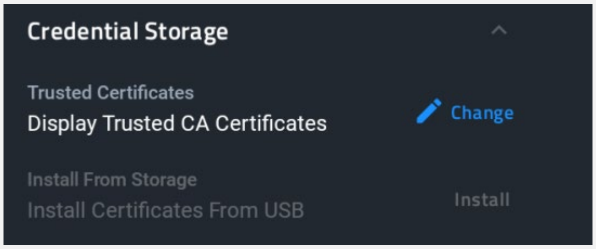
Update password

The current password does not meet the requirements for the current password complexity settings. The password must be changed to meet the password complexity settings.

Password Visibility allows you to see the number of characters in a password, and to see, briefly, the character just typed in clear text for verification. If this option is not selected, then characters are not displayed in the password text input field.

Credential Storage

This section helps manage or install certificates.

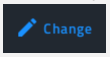


Credential Storage

Trusted Certificates
Display Trusted CA Certificates

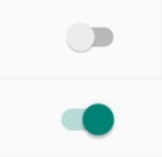
Install From Storage
Install Certificates From USB

To manage credentials, press [Change].



Navigate to *Encryption & Credentials* → *Trusted Credentials*

Credentials can be enabled or disabled with each of their respective buttons:



Disabled

Enabled

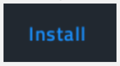
Note: Credentials are part of the OS Settings. Entering this area is for advanced use only, and not recommended or needed for normal operation.

If you need assistance, please contact HARMAN Professional Service & Support:
https://pro.harman.com/service

To install certificate from USB, insert a USB drive and press [Install].

Security

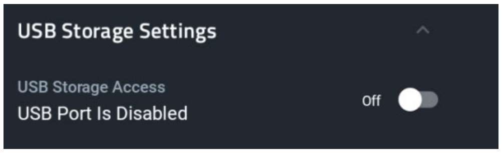
The button will turn from gray to blue when the USB drive is inserted. Remember to Enable USB Storage in the System Settings menu.



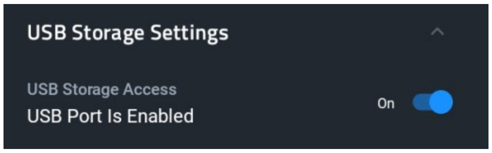
Browse to the desired folder location and select the appropriate certificate.

USB Storage Settings

The panel's USB ports can be enabled or disabled here. Use the button to turn USB ports [ON] or [OFF].


USB Storage Settings

USB Storage Access
USB Port Is Disabled
Off


USB Storage Settings

USB Storage Access
USB Port Is Enabled
On

After using USB for file transfers (eg. touch panel project files, firmware, logs, etc.), a reboot of the touch panel is recommended.

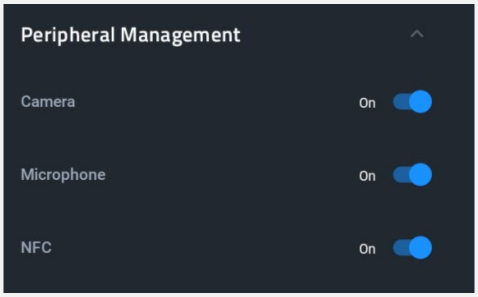
1. Disable USB by navigating to *Advanced Settings* → *Security* → *USB Peripheral*
2. Once disabled, remove your USB drive and any associate dongles, OTG adapters, etc.
3. Reboot the panel by either of these three methods:
   a. Select [Reboot] from the Panel Settings menu
   b. Press & hold the pin-hole button for four (4) seconds until the panel reboots, then release.
   c. Power cycle by unplugging the networking cable. Then wait 10 seconds and reinsert.

Failure to unmount the USB properly can cause an issue where the panel does not wake up from its Screen Timeout. If this happens, remove all USB drives, dongles, & OTG adapters and reboot the panel.

Note: USB drives should be formatted to FAT32 file system.

Peripheral Management

The device's communication & intelligence peripherals can be enabled or disabled here.


Peripheral Management

Camera On
Microphone On
NFC On

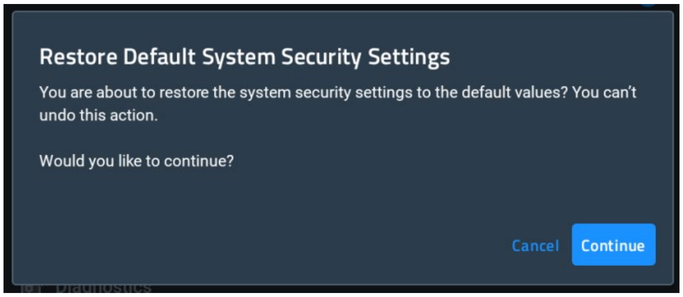
Use each button to turn [ON] or [OFF] the respective peripheral.

Note: Panels that do not include these peripherals (eg. VARIA-SL50 & No-Comm touch panels) will have some or all of these options grayed out.

Restore Default System Security Settings

Pressing this option will restore all System Security Settings to their default values.

You will be prompted to continue.



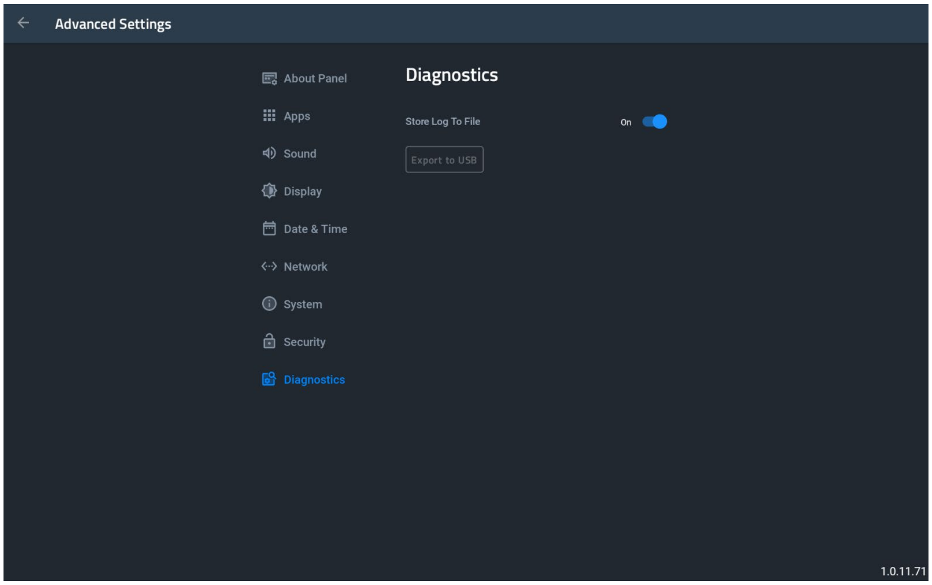
Restore Default System Security Settings

You are about to restore the system security settings to the default values? You can't undo this action.

Would you like to continue?

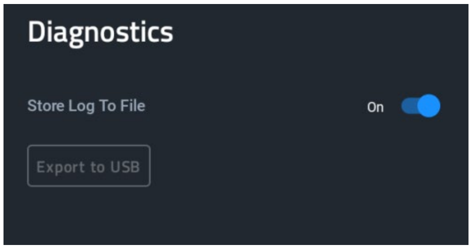
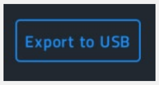
To exit without performing this action, press [Cancel].

Diagnostics

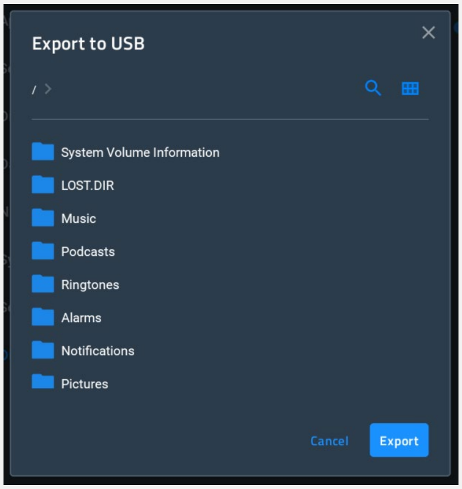


Diagnostics

Store Log To File On

Diagnostics	
Save Log to File	Turn this [ON] to record logs to file. That file can then be exported to USB for investigation by technicians or HARMAN Pro Tech Support.  Diagnostics Store Log To File On
Export to USB	To export the saved logs to USB, insert a USB drive and press [Export to USB]. The button will turn from gray to blue when the USB drive is inserted. Remember to Enable USB Storage in the System Settings menu.  Browse to the desired folder location and press [Export].

Diagnostics



A dialog window will show progress and a toast message will confirm completion.

Once completed, a reboot of the touch panel is recommended. Follow instructions in the USB Storage Settings section.

Note: USB drives should be formatted to FAT32 file system.

Gestures

Overview

You can program Varia touch panels using the commands in this section to perform a wide variety of operations using Send Commands and variable text commands.

A device must first be defined in the NetLinx programming language with values for the Device: Port: System (in all programming examples - Panel is used in place of these values and represents all Varia panels).

NOTE: Verify you are using the latest NetLinx Controller and Varia firmware, as well as the latest version of NetLinx Studio and TPD5.

NOTE: For more information on gestures and on designing touch panel pages, please refer to the TPDesign 5 online help, available at www.amx.com.

Touch Gesture Recognition

Gesturing refers to the act of moving a finger or stylus across the overlay and having the panel recognize and process this motion as a gesture. In G5, gesture events are assigned as individual buttons or pages. In addition, a gesture velocity is calculated and transmitted to the controller along with the gesture type itself in a custom event message.

NOTE: Nothing will be processed if the button associated with this gesture has no gesture event operations programmed, is disabled, or has no values programmed for address, channel, level, string output or command output. The custom event, however, is always transmitted.

The following gesture types are supported:

1. Swipe up
2. Swipe down
3. Swipe right
4. Swipe left
5. Double-tap
6. 2 Finger Swipe Up
7. 2 Finger Swipe Down
8. 2 Finger Swipe Right
9. 2 Finger Swipe Left

Gesture Velocity

A gesture “velocity” is calculated to represent the speed of the gesture. This is done by measuring the time from when the user first presses the screen until the user releases. The following simplified velocities are supported and transferred to the controller in the custom event message:

1. Fast
2. Normal
3. Slow

A precise velocity is sent in the custom event message which represents the velocity in terms of pixels per second for slides and circles. For a double tap, this value is the total time in milliseconds from the first press to the second release.

Gesture Prioritization

The following table describes the process used to determine what the user meant whenever a gesture operation is defined globally versus for the current page.

Gesture Prioritization	
The user presses outside of a button or slider and moves before releasing.	The firmware will always try to recognize a gesture as long as the user moves at least 20 pixels before the release occurs.
The user presses inside of a slider and moves before releasing.	This will always be processed as a slider operation and no attempt will be made to recognize a gesture.
The user moves a movable popup page.	This will always be processed as a popup page move and not a gesture.

The user presses on a button and then moves.	In this case, the press will not be sent for the first 0.15 second. If the user has moved at least 60 pixels by this time, then a button press/release will not be processed, but this will be processed as a gesture. At 0.15 second, the button press is processed and once the user releases, the release is processed and no gesture recognition is attempted. To be clear, it is not necessary for the user to move off of a button to be considered a gesture, but to move at least 60 pixels in that first 0.15 of a second.
The user double taps on a button or slider.	This will not be recognized as a gesture. This would be considered two quick press/release operations on the button or slider.
The user double taps outside of a button or slider.	This will be registered as a gesture.

Gesture VNC/Mouse Support

Gestures are recognized when the user is using a finger or stylus on the panel's screen overlay, a mouse on a VNC connection, or a mouse connected to the local USB port on the panel.

Gesture Custom Event

Whenever a gesture is recognized and processed a custom event is also sent to the controller. The following values describe this event:

```
CUSTOM_EVENT ADDRESS is 1

CUSTOM_EVENT EVENTID is 600

Custom.Value1 is the gesture number

Custom.Value2 is the simplified gesture velocity

Custom.Value3 is the precise gesture velocity
```

Gesture numbers and velocity values

Gesture Numbers and Velocity Values		
Gesture numbers		**Simplified gesture velocity values**
1- Swipe up **2-** Swipe down **3-** Swipe right **4-** Swipe left **5-** Circle (not implemented) **6-** CCW Circle (not implemented)	**7** Double-Tap **8** Two-Finger Swipe up **9** Two-Finger Swipe down **10** Two-Finger Swipe right **11** Two-Finger Swipe left.	**1** - Fast **2** - Normal **3** - Slow

Precise gesture velocity

For double taps, this is the time in milliseconds from the first press to the second release.

Enabling or Disabling the Gesture Custom Event

The **^GCE** Send Command sets whether or not the panel sends a custom event to the controller whenever a gesture is detected.

- The value sent is not retained - gesture custom events will be enabled each time the panel restarts.
- The default is to always NOT send the events.

Programming - Send Commands

Overview

You can program VARIA touch panels, using the commands in this section, to perform a wide variety of operations using Send Commands and variable text commands.

A device must first be defined in the NetLinx programming language with values for the Device: Port: System (in all programming examples - Panel is used in place of these values and represents all Varia panels).

- Verify you are using the latest NetLinx Controller and VARIA firmware, as well as the latest version of NetLinx Studio and TPDesign5.
- The Send Commands described in this document are case-insensitive.

Using the “Pipe” (|) Character

Previously, in G4, the pipe character (|) was used to create a new line.

G5 uses carriage return / line feed ($0d,$0a) instead.

The examples below illustrate indicating a new line (between the words “Hello” and “World”) in G4 and in G5 programming:

G4: `“’^TXT-200,0,Hello|World’”`

G5: `“’^TXT-200,0,Hello’,$0d,$0a,’World’”`

Panel Commands

Panel Commands	
^ABP **ABEEP**	Single Beep Command - Output a single beep. The ‘ABEEP’ command is implemented for G4 compatibility. • **Syntax:** “’^ABP’” or “’ABEEP’” • **Variables**: None • **Example**: SEND COMMAND Panel,”’^ABP’”
^ADB **ADBEEP**	Double Beep Command - Output a double beep. The ‘ADBEEP’ command is implemented for G4 compatibility. • **Syntax:** “’^ADB’” or “’ADBEEP’” • **Variables**: None • **Example**: SEND COMMAND Panel,”’^ADP’”
^AKB **@AKB** **AKEYB**	Show System Keyboard Command - Brings up system keyboard. When user presses the “Done” button, a string is returned to the controller with the user-entered value. The keyboard can be removed either by the Back button or the “^AKR” command. The ‘@AKB’ and ‘AKEYB’ commands are implemented for G4 compatibility. • **Syntax:** “’^AKB-[optional initial text];[optional prompt text];[optional hint text];[optional return prefix];[optional return port]’” or “’@AKB-[optional initial text];[optional prompt text];[optional hint text];[optional return prefix];[optional return port]’” or “‘AKEYB-[optional initial text];[optional prompt text];[optional hint text];[optional return prefix];[optional return port]’”

Panel Commands	
	• **Variables**: Initial text: Pre-populated text to appear on keyboard (i.e. default) Prompt text: Descriptive header to appear above keyboard text entry box Hint Text: Hint text to appear behind the keyboard text entry box Return prefix: Prefix to the send string returned to the controller. If not specified, the entered text will be preceded by “AKB-”. Return port: The port number to return the response on if different than the port to which the command is sent. • **Example**: `SEND_COMMAND Panel,”'^AKB-username;Enter user name;Enter the name of the user for this panel'”` Present a keyboard with a prompt of ‘Enter user name’, the initial text of ‘username’, and hint text of ‘Enter the name of the user for this panel’.

^AKP
@AKP
AKEYP
	Show System Keypad Command - Brings up system keypad. When user presses the “Done” button, a string is returned to the controller with the user-entered value. The keypad can be removed either by the Back button or the “^AKR” command. The ‘@AKP’ and ‘AKEYP’ commands are implemented for G4 compatibility.
	
	• **Syntax:**
	`“'^AKP-[optional initial text];[optional prompt text];[optional hint text];[optional return prefix];[optional return port]'”`
	or
	`“'@AKP-[optional initial text];[optional prompt text];[optional hint text];[optional return prefix];[optional return port]'”`
	or
	`“'AKEYP-[optional initial text];[optional prompt text];[optional hint text];[optional return prefix];[optional return port]'”`
	
	• **Variables**:
	Initial text: Pre-populated text to appear on keyboard (i.e. default)
	Prompt text: Descriptive header to appear above keyboard text entry box
	Hint Text: Hint text to appear behind the keyboard text entry box
	Return prefix: Prefix to the send string returned to the controller. If not specified, the entered text will be preceded by “AKP-”.
	Return port: The port number to return the response on if different than the port to which the command is sent.
	
	• **Example**:
	`SEND_COMMAND Panel,”'^APK-John Doe;Enter Username:;Enter the name for the user;AKP-username-;1'”`
	Opens a keyboard with the initial text as John Doe, the keyboard prompt as Enter Username:, the Hint text as Enter the name for the user, the return prefix as AKP-username-, and the return port as port 1.

^AKR
@AKR
AKEYR
	Remove Keyboard/Keypad Command - This command removes any keyboard or keypad that is currently displayed. If it is a non-virtual keyboard or keypad, it is essentially an Abort, because any user-entered text is lost. The ‘@AKR’ and ‘AKEYR’ commands are implemented for G4 compatibility.
	
	• **Syntax:**
	`“'^AKR'”`
	or `“'@AKR'”`
	or
	`“'AKEYR'”`
	
	• **Variables**: None:
	
	• **Examples**:
	`SEND COMMAND Panel,”'^AKR'”`
	Remove the displayed keyboard/keypad.

^APC	Automatic close application command - Setup alarm times to close all open applications. •**Syntax:** `“'^APC-<enable>,[optional alarm time],[optional alarm time]'”`

Panel Commands	

	• **Variables**:
	enable: 1 to enable alarms, 0 to disable alarms. Default is 1.
	Alarm time: Time of day to trigger alarm in HH:mm format. Format is 24 hour values. Up to six alarm times can be set each day. Valid HH formats are 00-23. Valid mm format is 00-59. Invalid formats and parameters will be disregarded. The default is one time set at 00:00 (midnight).
	• **Examples**:
	`SEND_COMMAND Panel, “’^APC-1,00:00, 08:00, 18:00’”`
	Enable the application close alarms at midnight (00:00), 8:00 AM (08:00), and 6:00 PM (18:00).
	`SEND_COMMAND Panel, “’^APC-0’”`
	Disable application close alarms.
	`SEND_COMMAND Panel, “’^APC-1’”`
	Enable alarms to close applications at previous alarm times.

?APC	Query application close alarms - Query the values of the close applications alarms. The response is a NetLinx DATA/Command event to the controller from the port the command was sent to in the format used in the ^APC command.
	•**Syntax:**
	`“’?APC’”`
	• **Variables**: None
	• **Example**:
	`SEND_COMMAND Panel, “’?APC’”`
	Response is a DATA/Command event to controller from the port the ?APC command was sent on in the format of:
	`^APC-<enable>,[optional alarm time],[optional alarm time]`
	If alarms are enabled and times set to midnight and noon, the response would be:
	`^APC-1,00:00,12:00`

^APP Launch application chooser	Launch application chooser command - Launch a dialog showing all available apps. • **Syntax:** `“’^APP’”` **Variables**: None

^APP- Launch application command	Launch application command - Launch the specified application (if not previously launched)
	• **Syntax:**
	`"'^APP-<AppPackageName>'"`
	• **Variables**:
	AppPackageName: The package name of the application to launch.
	• **Example**:
	`SEND_COMMAND Panel,”’^APP-us.zoom.zrc’”`
	Launch Zoom Rooms Controller app
	SEND_COMMAND Panel,"'^APP-http://www.amx.com'"
	Launch Chrome browser to amx.com website

^APP- Close a specific application	Close a specific application command - Close the specified application (if previously launched)
	•**Syntax:**
	`“’^APP-<app name>’”`
	• **Variables**:
	app name - The name of the application to close.
	• **Example**:
	`SEND_COMMAND Panel,”’^APP-Browser’”`
	Close the browser

Panel Commands	

^APP
Application
action
	Application action command - Performs a specified action on an application specified by app name.
	
	• **Syntax:**
	`“’^APP-<action>,<app name>[,<param list>]’”`
	
	• **Variables**: None action: The action to perform on the application. The available actions are: show: show an app, launch if not visible centered on the screen in a floating, moveable, resizable window. close: close a running app close_all: close all running apps
	app name: The name of the application to act upon. param list: The optional
	comma-separated list of parameter triplets as follows:
	`<param_1_name>,<param_1_type>,<param_1_value>,...,<param_N_name>,<param_N_type>,<param_N_value>`
	where: name: parameter name
	(e.g.”URI”)
	type: parameter type (e.g. “String”) - not case sensitive value:
	parameter value (e.g. http://www.amx.com)
	Note: The name, type and value are separated by a single comma. If there are additional parameters, a single comma should separate the previous parameter’s value and the next parameter’s name. Since comma is used to delimit the arameter fields, any comma appearing in the value of the element must be escaped with a backslash (‘\’). If a backslash itself appears in any element, it too must be escaped with another backslash. To access a file on an attached USB drive, the URI must be: file:///udisk/path_to_file. (Note there are three (3) forward slashes after the file: and you must specify udisk to point to the USB disk.)
	
	• **Example**:
	`SEND_COMMAND Panel,”’^APP-show,Browser’”`
	Show the browser centered on the screen in a floating, movable, resizable window.
	`SEND_COMMAND Panel,”’^APP-close,Browser’”`

?APP	Query available application command - Query all the available apps installed.
	
	• **Syntax:**
	`“’?APP’”`
	
	• **Variables**:None l
	•
	App names are sent through a custom event:
	`Custom Event   Property Value`
	`Port           port command was received on`
	`ID             1`
	`Type           4170`
	`Flag           0`
	`Value 1        App Number (0 - max number apps in no particular order)`
	`Value 2        Number of available apps`
	`Value 3        n/a`
	`Text           App Name (suitable for launching via ^APP,0,0,AppName)`

^BRT
@BRT
BRIT
	Panel Brightness Command - Set the panel brightness. The ‘@BRT’ and ‘BRIT’ commands are implemented for G4 compatibility.
	
	•**Syntax:**
	`“’^BRT-<brightness level>’”`
	or
	`“’@BRT-<brightness level>’”`
	or
	`“’BRIT-<brightness level>’”`
	
	• **Variables**:
	brightness level = 0 - 100.
	
	• **Example**:
	`SEND COMMAND Panel,”’^BRT-70’”`
	Sets the brightness level to 70

Panel Commands

?BRT

Query Brightness Command - Query panel brightness.

- **Syntax:**

```
“’?BRT’”
```

- **Variables**: None

- **Example:**

```
SEND_COMMAND Panel,”’?BRT’”
```

Gets the current brightness value.
The response returned is a custom event with the following properties:

```
Custom Event Property    Value
Port                     port command was received on
ID                       0
Type                     1303
Flag                     0
Value 1                  Brightness value 0-100
Value 2                  0
Value 3                  0
Text                     String that represents the brightness value
```

- **Example response:**

```
Custom Event Property    Value
Port                     port command was received on
ID                       0
Type                     1303
Flag                     0
Value 1                  70
Value 2                  0
Value 3                  0
Text                     70
```

^CPR

Cache Purge Command - Purge the image cache.

- **Syntax:**

```
“’^CPR’”
```

- **Variables**:None

- **Example**:

```
SEND COMMAND Panel,”’^CPR’”
```

Purge the image cache.

^DMM

Panel Streaming Audio Mute Command. Set the audio mute for a specified streaming URL.

- **Syntax:**

```
“’^DMM-<audio mute>,<video mute>,<url>’”
```

- **Variables**: l audio mute - mute/unmute the audio for <url> (0 = unmute, 1 = mute)

video mute - mute/unmute the video for <url> (0 = unmute, 1 = mute) (not implemented at this time) url - a valid ^SDM url that is already in the playing state.

- **Examples**:

```
SEND_COMMAND Panel, “’^DMM-1,0,udp://224.1.1.1:1234’”
```

Mute audio, unmute video for UDP stream server 224.1.1.1 port 1234.

```
SEND_COMMAND Panel, “’^DMM-0,0,udp://224.1.1.1:1234’”
```

Unmute audio, unmute video for UDP stream server 224.1.1.1 port 1234.

^EKP
@EKP

System Extended Keypad - Brings up system extended keypad. Currently, the ’system extended keypad’ and the ’system telephone keypad’ are the same, and have all the keys that the G4 extended keypad had except the “:” key. When the user presses the “Done” button, a string is returned to the controller with the user-entered value. The keypad can be removed either by the Back button or the “^AKR” command.
Note: The ‘@EKP’ command is implemented for G4 compatibility.

- **Syntax:**

Panel Commands	
	“’^EKP-[optional initial text];[optional prompt text];[optional hint text];[optional return prefix];[optional return port]’” or “’@EKP-[optional initial text];[optional prompt text];[optional hint text];[optional return prefix];[optional return port]’” • **Variables**: Initial text: Pre-populated text to appear on keypad (i.e. default) Prompt text: Descriptive header to appear above keypad text entry box Hint Text: Hint text to appear behind the keypad text entry box Return prefix: Prefix to the send string returned to the controller. If not specified, the entered text will be preceded by “EKP-”. Return port: The port number to return the response on if different than the port to which the command is sent.
^ENC	Set Text Encoding Method - Sets the text encoding method which is used for commands and strings sent from panel to controller (the default is UTF-8). • **Syntax:** “’^ENC-<Encoding>’” • **Variables**: Encoding: 0: UTF-8 (default), 1: Latin-1 (ISO 8859-1) • **Example:** SEND_COMMAND Panel,”’^ENC-1’” Sets the encoding method used for all strings to the Controller to Latin-1. Note: NetLinx Studio does not support UTF-8 at this time; therefore UTF-8-encoded characters cannot be copied from TPD5 and pasted in Studio. To use NetLinx Studio to send UTF-8 encoded text, byte values must be enumerated in the command. For example, the following command sends a UTF-8 string to the panel, consisting of ASCII, extended ASCII and Unicode (Chinese) characters: ”’^UTF-3,0,Hello’,$C3,$A2,$C3,$A3,$E5,$9C,$B0,$E7,$9B,$A4,$E3,$83,$87” Also note that in backwards compatibility mode (i.e. when the ^TXT command is sent or when the ^ENC-1 command has been sent), ISO-8859-1 is used for character encoding/decoding, since that is what G4 panels used. ISO-8859-1 is different from the Windows-1252 character set in that characters in the range 128-159 (decimal) are non-printable control characters. So in response to a ?TXT query, any characters in that range (assuming the ^ENC-1 was previously sent) will be returned as AMX Hex quad-encoded values with Custom Event Flag=1, whereas the remainder of the extended ASCII range (160-255) will be returned as Latin-1-encoded characters with Custom Event Flag=0 (see the ISO8859-1 Character Encoding/Decoding table).
?ENC	Get Text Encoding Method - Gets the current text encoding method which is used for commands and strings sent from panel to controller (the default is UTF-8). • **Syntax:** “’?ENC’” • **Variables**:None • **Example**: SEND_COMMAND Panel,”’?ENC’” Get the panel’s text encoding status. The response returned is a custom event with the following syntax: Custom Event Property Value Port port command was received on ID 0 Type 1331 Flag 0 Value 1 encoding (0 = UTF-8, 1 = ISO-8859-1) Value 2 0 Value 3 0 Text String that represents the encoding name • **Example response for encoding status:** Custom Event Property Value

Panel Commands	

	`Port                         port command was received on`
	`ID                           0`
	`Type                         1331`
	`Flag                         0`
	`Value 1                      0`
	`Value 2                      0`
	`Value 3                      0`
	`Text                         UTF-8`

^GCE	Set Gesture Custom Event - Sets whether or not the panel sends a custom event to the controller whenever a gesture is detected.
	• **Syntax:**
	`“’^GCE-<state>’”`
	• **Variables**:None l state: ON or OFF / 1 or 0 / on or off.
	Note: This setting is not retained and the default is to always NOT send the events. To enable sending the event, the value after the dash can be “on”, “ON”, or “1”. Anything else will disable sending custom events.
	• **Examples**:
	`SEND_COMMAND Panel, “’^GCE-on’”`
	Enables gesture custom event reporting to the controller.
	`SEND_COMMAND Panel, “’^GCE-0’”`
	Disables gesture custom event reporting to the controller.

LEVON	Level on command (generated by NetLinx controller) - Enable device to send level changes to the controller. By default, devices will not report level changes unless a LEVON command is received. The LEVON command is automatically sent by the controller to the device if: There is a LEVEL event for the DPS of the device. There is a CREATE_LEVEL defined in the NetLinx program for the DPS of the device. • **Syntax:** `“’LEVON’”` **Variables**: None
LEVOF	Level off command (generated by NetLinx controller) - Disable the device from sending level changes to the controller. By default, devices will not report level changes unless a LEVON command is received. The LEVON command is automatically sent by the controller to the device if: There is a LEVEL event for the DPS of the device. There is a CREATE_LEVEL defined in the NetLinx program for the DPS of the device. • **Syntax:** `“’LEVOF’”` **Variables**:None

?MAC	Query Panel MAC Address - Query the MAC Address of the panel.
	• **Syntax:**
	`“’?MAC’”`
	• **Variables**:None
	•
	• **Example**:
	`SEND_COMMAND Panel,”’?MAC’”`
	Get the panel’s MAC Address. The response returned is a custom event with the following syntax:
	`Custom Event Property        Value`
	`Port                         port command was received on`
	`ID                           0`
	`Type                         1315`
	`Flag                         0`
	`Value 1                      0`

Panel Commands	
	Value 2 0 Value 3 0 Text String that represents the the MAC Address • **Example response:** Custom Event Property Value Port port command was received on ID 0 Type 1315 Flag 0 Value 1 0 Value 2 0 Value 3 0 Text 00:60:9f:90:00:01
^MSG	Message Dialog Command - A generic message dialog that has displayed content defined from the ^MSG command. • **Syntax:** '^MSG-dialog_id[:dialog_theme],dialog_type[-input_option][:dialog_image_name], timeout,custom_event_type, custom_event_id, title_text, message_text, positive_ button_text,negative_button_text, neutral_button_text, cancel_text, timeout_text' • **Variables**: None **dialog_id**: Unique id to reference the dialog. Used to track IDs to displayed dialogs. **dialog_theme**: Optional theme of the dialog is set by appending the theme to the dialog_id number with ':' and the theme. Valid themes are light and dark (default) dialog_type. The type of dialog to display: std - standard dialog. By default, no image is displayed in the title area. warn - warning dialog. The built-in warning image is used in the title area. error - error dialog. The built-in error image is used in the title area. quest - question dialog. The built-in question image is used in the title area. list - list of items to choose. By default, no image is displayed in the title area. List items are put in the message_ text field and are separated by colons (':'). input - input entry. By default, no image is displayed in the title area. Optional input_options follow a dash ('-') and are: no option present - alphanumeric input num - numeric input (no alphabetic input) phone- phone pad presented uri - URI keyboard presented email - Email keyboard presented name - Keyboard presented and capital words are used. date - Date pad presented time - Time pad presented datetime - Date/Time pad presented The message_text is 'System is busy'. The positive_button_text is 'OK' SEND_COMMAND Panel,'^MSG-1,list:question-flat-48x48.png,30000,32001,10, Select item,"item 1:item 2:item 3:item 4:item 5",,"Cancel"' Display dialog ID 1 as a list dialog. The image 'question-flat-48x48.png' is used as the image in the title area. The timeout is 30s. The custom_event_type to use is 32001. The custom_event_id to use is 10. The title_text is 'Select Item'. The message_text is list of 5 items (item 1, item 2, item 3, item 4, item5). The positive_button_text is empty. The negative_button_text is empty. The neutral_button_text is 'Cancel'. pass - password entry. By default, no image is displayed in the title area. Optional input_options follow a dash ('-') and are: no option present - alphanumeric input num - numeric input (no alphabetic input) **dialog_image_name**: It is optional to override any type with a custom image or dynamic image from the TP5 file to be displayed in the title area. The image used is set by appending a ':' and image file/resource name to the **dialog_type-input_option** (e.g. std:number.png or warn:mywarningimage.jpg). timeout: Timeout is in milliseconds. If timeout is 0, message does not timeout and is considered modal. **custom_event_type**: The custom event type value to use for result custom events. **custom_event_id**: The custom event ID value to use for result custom events. **title_text**: Text that is displayed in the dialog title. If this field is empty, no title is displayed on the dialog. **message_text**: In most cases, the contents of this field is displayed in the message of the dialog. There are a few exceptions based on dialog_type: list - In a list dialog type, the message_text contains the list items. List items are separated by a colon (':'). input - In a input dialog type, the message_text contains the initial value of the text entry field of the dialog. pass In a pass dialog type, the message_text contains the initial value of the text entry field of the dialog. **positive_button_text**: Text to display on the positive button (e.g. Yes, OK, Enter, etc.) In most cases, if the positive button is selected, this text is sent to controller in the custom.text field. Note: If this field is empty, the positive button is not displayed in the MessageDialog.

Panel Commands	

	Note: Text fields can be put into quotations (“ ”) so that commas can be used in text. Like the CSV parser, if a “ is needed in the text, the “ can be escaped by a prepended another “ (e.g. “”).
	Note: The use of text params in command instead of preset definitions for button text is so that the language of text can be set in code. Unicode quads for text are supported by using the command ‘^MSGU-’ command. Legacy ISO-8859-1 (like ^TXT) text is supported by using the ‘^MSGT-’ command.
	There is a LEVEL event for the DPS of the device.
	There is a CREATE_LEVEL defined in the NetLinx program for the DPS of the device.
	
	• **Response Data:**
	The response to the MessageDialog is sent to the controller via a Custom Event. Some of the custom event values are set in the ^MSG command, and others are generated as a result of the dialog action.
	Result Custom Events data: custom.type: The value set in the custom_event_type field custom.id: The value set in the custom_event_id field. custom.flag: value has the result. In most cases, it indicates which button was selected, or cancel, or timeout:
	`-1 = timeout`
	`0 = cancel`
	`1 = positive button`
	`2 = negative button`
	`3 = neutral button`
	In a list dialog type, when an item is selected, the custom.flag field will be set to 1 (positive button).
	custom.value1 The dialog_id value set in the command
	custom.value2 In a list dialog type, this field has the index of the selected list item. If the first item was selected then value2==1, second item selected then value2==2, etc. If the dialog_type is not a list, then value2 is unused and is set to 0.
	custom.value3: Unused. Set to 0. custom.text: The text of the resulting button selected, or cancel_text if dialog was canceled, or timeout_text if timed out. In list mode, the selected list item text value is sent in this field. In input or pass, the entered value is sent in this field.
	Note: Custom events are returned on the port the command was sent to from the controller.
	
	• **Examples:**
	`SEND_COMMAND Panel,’^MSG-1,std,60000,32001,1,Please Wait,”System is busy”,OK’;`
	Display dialog ID 1 as a standard dialog.
	The timeout is 60s.
	The custom_event_type to use is 32001.
	The custom_event_id to use is 1.
	The title_text is ‘Please Wait’.
	
	`SEND_COMMAND Panel,”’^MSGT-1:light,error,30000,32001,32002,”Error Title”,”Lorem ipsum dolor sit amet, consectetur adipiscing elit,sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex eacommodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore ND_COMMAND Panel,”’^MSGT-1:lig ht,error,30000,32001,32002,”Error Title”,”Lorem ipsum dolor sit amet, consectetur adipiscing elit,sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. “”the end””,”Positive”,”Negative”, “Neutral”,”Cancel”,”Timeout”’”`
	`^MSGT` –The dialog text is encoded in the ISO-8859-1 (Latin-1) format (like what is expected by ^TXT command). Display dialog ID 1 with a light theme as an error dialog. The default error image is used as the image in the title area.
	The timeout is 30s.
	The custom_event_type to use is 32001.
	The custom_event_id to use is 32001.
	The title_text is ‘Error Title’.
	The message_text is a variation of ‘Lorem ipsum…’.
	The positive_button_text is ‘Positive’.
	The negative_button_text is ‘Negative’.
	The neutral_button_text is ‘Neutral’.
	The cancel_text is ‘Cancel’.
	The timeout_text is ‘Timeout’.

^MUT	Panel Volume Mute - Mute or unmute a panel volume.

Panel Commands	

	• **Syntax:**
	`“’^MUT-<mute value>’”`
	
	• **Variables**:
	mute value: 0 for not muted, 1 for muted.
	
	• **Examples:**
	`SEND_COMMAND Panel,”’^MUT-1’”`
	Mute the controller volume.
	`SEND_COMMAND Panel,”’^MUT-0’”` Unmute the controller volume.

?MUT	Query Panel Mute Status - Query the mute status of the panel.
	
	• **Syntax:**
	`“’?MUT’”`
	
	• **Variables**: None
	
	• **Example:**
	`SEND_COMMAND Panel,”’?MUT’”`
	Get the panel’s mute status. The response returned is a custom event with the following syntax:
	`Custom Event Property    Value`
	`Port                     port command was received on`
	`ID                       0`
	`Type                     1305`
	`Flag                     0`
	`Value 1                  mute status (0 unmuted or 1 for muted)`
	`Value 2                  0`
	`Value 3                  0`
	`Text                     String that represents the mute status (0 or 1)`
	
	• **Example response for muted status:**
	`Custom Event Property    Value`
	`Port                     port command was received on`
	`ID                       0`
	`Type                     1305`
	`Flag                     0`
	`Value 1                  1`
	`Value 2                  0`
	`Value 3                  0`
	`Text                     1`

^NOT	Popup Note Command - A generic popup note message that can be used to display information for a short duration on the display. • **Syntax:** `‘^NOT-note_text, duration, location, text_size’` • **Variables**: None note_text - The text to displayed in the popup note. duration - The time in milliseconds to display the popup note location - Where to display the popup note. Options are ‘c’ for CENTERED on display, ‘t’ for TOP CENTER on display, and ‘b’ for BOTTOM CENTER on display. Any other value will be displayed as CENTER. text_size - The size value to display the popup note text. Default is 18. Note: The note text field can be put into quotations (“ ”) so that commas can be used in text. Like the CSV parser, if a “ is needed in the text, the “ can be escaped by a perpending another “ (e.g. “”). Note text is assumed to be UTF-8 encoded.
^PKB **@PKB** **PKEYB**	Show System Private Keyboard Command - Brings up system private keyboard (the same as the system keyboard, with typed text hidden with the ‘*’ character). When user presses the “Done” button, a string is returned to the controller with the user-entered value. The keyboard can be removed either by the Back button or the “^AKR” command. The ‘@PKB’ and ‘PKEYB’ commands are implemented for G4 compatibility. • **Syntax:**

Panel Commands	
	“’^PKB-<initial text>;<prompt text>;<hint text>;<return prefix>;<return port>’” or “’@PKB-<initial text>;<prompt text>;<hint text>;<return prefix>;<return port>’” or “’PKEYB-<initial text>;<prompt text>;<hint text>;<return prefix>;<return port>’ • **Variables**: Initial text - Pre-populated text to appear on keyboard (i.e. default). Note that for the private keyboard, this text will be hidden. Prompt text - Descriptive header to appear above keyboard text entry box Hint Text - Hint text to appear behind the keyboard text entry box Return prefix - Prefix to the send string returned to the controller. If not specified, the entered text will be preceded by “PKB-”. Return port - The port number to return the response on if different than the port to which the command is sent.
^PKP **@PKP** **PKEYP**	Show System Private Keypad Command - Brings up system private keypad (the same as the system keypad, with typed text hidden with the ‘*’ character). When user presses the “Done” button, a string is returned to the controller with the userentered value. The keypad can be removed either by the Back button or the “^AKR” command. The ‘@PKP’ and ‘PKEYP’ commands are implemented for G4 compatibility. • **Syntax:** “’PKEYP-[optional initial text];[optional prompt text];[optional hint text];[optionalreturn prefix];[optional return port]’” • **Variables**: Initial text: Pre-populated text to appear on keypad (1 - 50 ASCII characters). Note that for the private keypad, this text will be hidden. Prompt text: Descriptive header to appear above keypad text entry box Hint Text: Hint text to appear behind the keypad text entry box Return prefix: Prefix to the send string returned to the controller. If not specified, the entered text will be preceded by “PKP-”. Return port: The port number to return the response on if different than the port to which the command is sent. • **Examples:** SEND COMMAND Panel,”’PKEYP-123456789’” Pops up the Keypad and initializes the text string ‘123456789’ in ‘*’.
^RPP	Reset protected password command - This command is used to reset the protected setup password to the factory default value. • **Syntax:** “’^RPP’” • **Variables**: None • **Example:** SEND_COMMAND Panel,”’^RPP’” Reset the panel protected password to the factory default.
^RSS	Reset System Settings Command - Reset Settings to factory default. • **Syntax:** “’^RSS’” • **Variables**: None • • **Example:** SEND_COMMAND Panel, “’^RSS’” Reset the panel to factory default settings.

Panel Commands	
RXON	Send string on command (generated by NetLinx controller) - Enable device to send STRING changes to the controller. By default, devices will not report STRING changes unless a RXON command is received. The RXON command is automatically sent by the controller to the device if: There is a DATA/STRING event for the DPS of the device. There is a CREATE_BUFFER defined in the NetLinx program for the DPS of the device. • **Syntax:** `“‘RXON’”` **Variables**: None
RXOF	Send string off command (generated by NetLinx controller) - Disable the device from sending STRING changes to the controller. By default, devices will not report STRING changes unless a RXON command is received. The RXON command is automatically sent by the controller to the device if: There is a DATA/STRING event for the DPS of the device. There is a CREATE_BUFFER defined in the NetLinx program for the DPS of the device. • **Syntax:** `“‘RXOF’”` **Variables**: None
SHUTDOWN	Power Off the Panel Command - Receipt of this command will cause the panel to power off. • **Syntax:** `“‘SHUTDOWN’”` **Variables**: None
^SCO	Session clear out command - Clears session data for some applications (Browser, Firefox, Gallery, Skype, Dropbox, VNC server, PlanMaker, TextMaker, and Presentations). • **Syntax:** `‘^SCO’` **Variables**: None

^SLP
SLEEP
	Panel Sleep Command - Place the panel in sleep state. Sleep state turns the display off. The ‘SLEEP’ command is implemented for G4 compatibility.
	Syntax:
	`“‘^SLP’”`
	or
	`“‘SLEEP’”`
	• **Variables**:None
	• **Example:**
	`SEND COMMAND Panel,”‘^SLP’”`
	Sends the panel to the sleep (display off)
	The response returned is a custom event with the following syntax:
	`Custom Event Property    Value`
	`Port                     port command was received on`
	`ID                       1`
	`Type                     1701`
	`Flag                     3`
	`Value 1                  0`
	`Value 2                  0`
	`Value 3                  0`
	`Text                     com.amx.touchpanel.Gx/com.amx.touchpanel.GxActivity`

Panel Commands

^SOU
@SOU
	Play Sound Command - Plays a specified sound file. The ‘@SOU’ command is implemented for G4 compatibility.
	Syntax:
	`“’^SOU-<sound name>’”` or `“’SLEEP’”`
	• **Variables**:
	sound name: Name of the sound file. Supported sound file formats are: WAV & MP3.
	• **Example:**
	`SEND COMMAND Panel,”’^SOU-Music.wav’”`
	Plays the ‘Music.wav’ file.

^SSL
@SSL
	Set the Sleep String Command - Set the content of the string that is sent to the controller when the panel goes to sleep (display off). The ‘@SSL’ command is implemented for G4 compatibility.
	Syntax:
	`“’^SSL-<sleep string>’”`
	or
	`“’@SSL-<sleep string>’”`
	• **Variables**:
	Sleep string: The string sent to the controller when the panel goes to sleep.
	• **Example:**
	`SEND COMMAND Panel,”’^SSL-Sleeping…’”`
	Sets the sleep string to ’Sleeping…’.

^STP
SETUP
	Settings application command - Open the Settings Applications. The ‘SETUP’ command is implemented for G4 compatibility.
	• **Syntax:**
	`“’^STP’”`
	or
	`“’SETUP’”`
	• **Variables**:None
	• **Example:**
	`SEND COMMAND Panel,”’^STP’”`
	Opens the Settings application.

^SWK
@SWK
	Set the Wake String Command - Set the content of the string that is sent to the controller when the panel wakes up from sleep (display on). The ‘@SWK’ command is implemented for G4 compatibility.
	Syntax:
	`“’^SWK-<wake string>’”`
	or
	`“’@SWK-<wake string>’”`
	• **Variables**:
	Wake string: The string sent to the controller when the panel wakes up from sleep.
	• **Example:**
	`SEND COMMAND Panel,”’^SWK-Wakeing Up…’”`
	Sets the sleep string to ‘Waking Up…’.

^TKP **@TKP**	Brings up system telephone keypad - Currently, these keypads are the same, and have all the keys that the G4 extended keypad had except the “:” key. When user presses the “Done” button, a string is returned to the controller with the user

Panel Commands	

	entered value. The keypad can be removed either by the Back button or the “^AKR” command. The ‘@TKP command is implemented for G4 compatibility.
	
	• **Syntax:**
	`“^TKP-[optional initial text];[optional prompt text];[optional hint text];`
	`[optional return prefix];[optional return port]’”`
	
	• **Variables**:
	Initial text: Pre-populated text to appear on keypad (i.e. default)
	Prompt text: Descriptive header to appear above keypad text entry box
	Hint Text: Hint text to appear behind the keypad text entry box
	Return prefix: Prefix to the send string returned to the controller. If not specified, the entered text will be preceded by “TKP-”.
	Return port: The port number to return the response on if different than the port to which the command is sent. Note: See also - ^EKP (system telephone keypad).

^TPF
TPAGEOF
	Turn Off Page Tracking Command. The ‘TPAGEOF’ command is implemented for G4 compatibility.
	
	• **Syntax:**
	`“’^TPF’”`
	or
	`“’^TPF’”`
	
	• **Variables**: None
	
	• **Example:**
	`SEND COMMAND Panel,”’^TPF’”`

^TPN
TPAGEON
	Turn On Page Tracking Command - This command turns on page tracking, whereby when the page or popups change, a string is sent to the Controller. This string may be captured with a CREATE_BUFFER command for one panel and sent directly to another panel. The ‘TPAGEON’ command is implemented for G4 compatibility.
	
	• **Syntax:**
	`“’^TPN’”`
	or
	`“’TPAGEON’”`
	
	• **Variables**: None
	
	• **Example:**
	`SEND COMMAND Panel,”’^TPN’”`

^VKB **@VKB**	Show Virtual Keyboard Command - Brings up system virtual keyboard, which is the keyboard without a designated text entry area. A Text Input button must be in focus; if not, the keyboard will not appear. The type of keyboard is determined by the Text Area currently in focus. When user presses the “Done” button, a string is returned to the controller with the userentered value. The keyboard can be removed either by the Back button or the “^AKR” command. The ‘@VKB’ command is implemented for G4 compatibility. **Syntax:** `“’^VKB’”` • **Variables**: None
^VKP **@VKP**	Show Virtual Keypad Command - Brings up system virtual keypad, which is the keypad without a designated text entry area. A Text Input button must be in focus; if not, the keypad will not appear. The type of keypad is determined by the Text Area currently in focus. When user presses the “Done” button, a string is returned to the controller with the user-entered value. The keypad can be removed either by the Back button or the “^AKR” command. The ‘@VKP’ command is implemented for G4 compatibility. • **Syntax:** `“’^VKP’”`

Panel Commands	
	Variables: None

^VKS	Virtual Key Stroke Command - Sends a Virtual Key Stroke to the Varia touch panel.
	Note: this command does not function in the same way as with G4 touch panels.
	
	• **Syntax:**
	`"'^VKS-<keycode>'"`
	or
	`"'^TPF'"`
	
	• **Variables**: None keycode: Android key code decimal value. Note that these are not the same as in G4.
	
	Note: For the key code values, please refer to the Virtual Keystroke Commands table.

^VOL	Set Volume Command - Set the [specified] volume.
	
	• **Syntax:**
	`"'^VOL,<level>,[optional type]'"`
	
	• **Variables**:
	Level: the volume level from 0-100. The level will be scaled according to the platforms abilities.
	Type (option): Change the volume of the given type
	0 = Controller volume (change all volumes simultaneously). Used by default if no type is specified. This is not a real volume, but instead is a virtual value that changes all other volume type concurrently. 10 = Alarm Volume
	11 = Call Volume
	12 = Media Volume
	13 = Notification Volume
	44 = Display the volume dialog (level is ignored)
	Note: the platform dialog sliders will NOT update if they are displayed when the command is received. They are accurate, however, if displayed after receiving the command.
	
	• **Examples:**
	`SEND_COMMAND Panel,"'^VOL,50'"`
	Sets the controller volume to 50.
	`SEND_COMMAND Panel,"'^VOL,50,0'"`
	Sets the controller volume to 50.

?VOL	Query Volume Command - Query the volume.Note: Allow 10-15 minutes for update to complete before sending another ^UPD command.
	
	• **Syntax:**
	`"'?VOL,[optional type]'"`
	
	• **Variables**:
	Type (option) Get the volume of the given type
	0 = Controller volume. Used by default if no type is specified. Since Controller volume is not a real volume, the value returned will actually be the Media Volume Value.
	10 = Alarm Volume
	11 = Call Volume
	12 = Media Volume
	13 = Notification Volume
	The response returned is a custom event with the following syntax:
	`Custom Event Property    Value`
	`Port                     port command was received on`
	`ID                       0`
	`Type                     1306`
	`Flag                     0`
	`Value 1                  volume level`
	`Value 2                  volume type`
	`Value 3                  0`
	`Text                     String containing 'type=level'`

<table>
<tr><th colspan="2">Panel Commands</th></tr>
<tr><td></td><td>

• **Examples:**
<pre>
SEND_COMMAND Panel,”’?VOL’”
</pre>
Query the Controller volume. Response would be similar to:
<pre>
Custom Event Property   Value
Port   port command was received on
ID   0
Type   1306
Flag   0
Value 1   80
Value 2   0
Value 3   0
Text   Controller=80
SEND_COMMAND Panel,”’?VOL,10’”
</pre>
Query the Alarm volume. Response would be similar to:
<pre>
Custom Event Property   Value
Port   port command was received on
ID   0
Type   1306
Flag   0
Value 1   20
Value 2   10
Value 3   0
Text   Media=72
</pre>
</td></tr>
<tr><td><b>^WCN</b></td><td>
Web Control Name (Panel to Controller) - Report the Web Control (VNC) name to the controller.
This is originated in the panel and sent to the controller if VNC is enabled.
</td></tr>
<tr><td><b>WEBU</b></td><td>

Update Firmware from URL - This command tells the panel to retrieve a firmware kit file from the included URL and update to the firmware included in that kit file.

• **Syntax:**
<pre>
“’WEBU-<url>’”
</pre>

• **Variables**:

url: URL to the kit file. Support protocols are HTTP only at this time.

• **Example:**
<pre>
SEND_COMMAND PANEL,”’WEBU,http://file.server/VARIA-firmware.kit’”
</pre>
Download and install the VARIA-firmware.kit file from the HTTP server file.server.
</td></tr>
<tr><td><b>^WKE</b>
<b>WAKE</b></td><td>

Panel Wakeup Command - Place the panel in wake state. Wake state turns the display on. The ‘WAKE’ command is implemented for G4 compatibility.

• **Syntax:**
<pre>
“’^WKE’”
</pre>

• **Variables**: None

• **Example:**
<pre>
SEND COMMAND Panel,”’^WKE’”
</pre>
Wakes the panel from sleep (turn display on)

The response returned is a custom event with the following syntax:
<pre>
Custom Event Property   Value
Port   port command was received on
ID   1
Type   1701
Flag   4
Value 1   0
Value 2   0
Value 3   0
Text   com.amx.touchpanel.Gx/com.amx.touchpanel.GxActivity
</pre>
</td></tr>
</table>

Panel Commands

^WLD	Side LED Command - turn on/off the side LEDs. LEDs can be turned on any RGB color. **Syntax:** "'^WLD-<command>,<action>'" **Variables:** COMMAND Indicates the LED ID number or Brightness. • 0 - All LED status • 1 - Red • 2 - Green • 3 - Blue • 10 - All LED brightness • 11 - Red brightness • 12 - Green brightness • 13 - Blue brightness ACTION For LED on/off commands 0, 1, 2, 3: • 0 - Turns LED Off • 1 - Turns LED On For LED Brightness commands: 0 - 100 : Percentage of LED brightness when LED is on. **Examples:** SEND_COMMAND Panel,"'^WLD-1,1'" // Turn on the Red LED SEND_COMMAND Panel,"'^WLD-10,50'" // Set the Red, Blue and Green LED brightness to 50%.

Page Commands

Page Commands are case in-sensitive

Page Commands

^AFP

Flip to specified page using the named animation.

- **Syntax**:
```
“’^AFP-<page name>,<animation>,<origin>,<duration>’”
```

- **Variables**:
  Page Name: If the page name is blank, flip the to the previous page *Animation*: If blank/invalid, the default animation is *Fade*.

Animation Name	Command Snytax* (see note below)	Origin(s)	Default Origin
Center Door Fade	cntrdrfade, centerdoorfade, or center door fade	top(2), bottom(3), left(4), right(5)	right(5)
Door Fade	doorfade, door fade, or door	top(2), bottom(3), left(4), right(5)	right(5)
Fade	fade	center(1)	center(1)
Slide	slide	top(2), bottom(3), left(4), right(5)	right(5)
Slide Bounce	sldbouce, slidebounce, or slide bounce	top(2), bottom(3), left(4), right(5)	right(5)
Spin In	spinin or spin in	center(1)	center(1)
Spin Out	spinout or spin out	center(1)	center(1)
Zoom In	zoomin or zoom in	center(1)	center(1)
Zoom Out	zoomout or zoom out	center(1)	center(1)

Note: Multiple aliases for the transition name command syntax are allowed to maintain backwards compatibility with G4.

Duration: Transition time in 10ths of a second. Range is 3-30 with 15 (1.5 seconds) as the default

- **Examples**:
```
SEND_COMMAND Panel,”’^AFP-NextPage,slide,4,5’”
```
  Flip to NextPage sliding from the left for half a second.
```
SEND_COMMAND Panel,”’^AFP-,centerdoorfade,2,10’”
```
  Flip to NextPage center door fade from the top for a second.

^PCL

Collapse Collapsible Popup Command - Moves the named closeable popup to the collapsed position.

- **Syntax**:
```
“’^PCL-<popup name>;[optional target page]’”
```

- **Variables**:
  Popup name: the name of the popup to collapse
  Target page: name of the page hosting the popup to affect the change upon. If target page is not specified, the command is applied to the current page.

- **Examples**:
```
SEND_COMMAND Panel,”’^PCL-Contacts’”
```
  Collapse the Contacts popup on the current page.
```
SEND_COMMAND Panel,”’^PCL-Contacts;Teleconference Control’”
```
  Collapse the *Contacts* popup on the Teleconference Control pages

^PCT

Collapsible Popup Custom Toggle Command - This is an advanced “toggle” command for collapsible popups, working with a comma-separated list of commands. This list is parsed and a command table is created. Based on the current state of the collapsible popup, the correct command is executed.
Note: The previously parsed list is saved and is only parsed again if the command string differs for this popup.

- **Syntax**:
```
“’^PCT-<popup>,<custom toggle commands>;[optional target page]’”
```

Page Commands	
	• **Variables**: *Popup*: popup name *Custom toggle commands*: a comma separated list of commands. This list is parsed and a command table is created. The state letters are as follows: o - open c - collapsed d - dynamic, followed by an integer indicating the offset. * - wildcard, always last in the list Before and after states are separated by -> characters. *Target page*: name of the page hosting the popup to affect the change upon. If target page is not specified, the command is applied to the current page. • **Example**: `SEND_COMMAND Panel,”’^PCT-RightSlider,c->o,o->d100,*->c’”` The popup named *RightSlider* opens if collapsed, move to d100 if open, and collapse otherwise.

^PDO	Collapsed Popup Dynamic Offset Command - Moves the collapsible popup to a specific offset position relative to the collapsed direction configured for the popup. This allows other positions besides open and collapsed.
	• **Syntax**:
	`“’^PDO-<popup name>,<offset>;[optional target page]’”`
	• **Variables**:
	Popup name: name of the popup to affect
	offset: number of pixels to offset (hide). <offset> is constrained as follows: 0 <= offset <= collapsed offset *Target page*: name of the page hosting the popup to affect the change upon. If target page is not specified, the command is applied to the current page.
	• **Examples**:
	`“’^PDO-RightSlider,66’”`
	Move popup named *RightSlider* to an offset position of 66 on the current page.
	`“’^PDO-RightSlider,66;Media Controls’”`
	Move popup named *RightSlider* to an offset position of 66 on the Media Controls page.

^PGE
PAGE
	Page Flip Command - Flips to a page with a specified page name. If the page is currently active, it will not redraw the page. The ‘PAGE’ command is implemented for G4 compatibility.
	• **Syntax**:
	`“’^PGE-<page name>’”`
	or
	`“’PAGE-<page name>’”`
	• **Variables**: *page name*: Name of the page to be displayed. If left blank, the page flips back to the previous page.
	•
	• **Examples**:
	`SEND_COMMAND Panel,”’^PGE-Page1’”`
	Flips to page1.
	`SEND_COMMAND Panel,”’^PGE-’”`
	Flips to the previous page.

^POP	Open Collapsible Popup Command - Moves the named collapsible popup to the open position.
	• **Syntax**:
	`“’^POP-<popup>;[optional target page]’”`
	• **Variables**:
	Popup: the name of the popup to collapse
	Target page: name of the page hosting the popup to affect the change upon. If target page is not specified, the command is applied to the current page.
	• **Example**:
	`SEND_COMMAND Panel, “’^POP-Contacts’”`
	Open the *Contacts* popup on the current page.
	`SEND_COMMAND Panel, “’^POP-Contacts;Teleconference Control’”`
	Open the *Contacts* popup on the Teleconference Control page.

^PPA **@PPA**	Close All Popups Command - Close all popups on a specified page. The ‘@PPA’ command is implemented for G4 compatibility. • **Syntax**:

Page Commands	
	“’^PPA-<page name>’” or “’@PPA-<page name>’” - **Variables**: *Page name*: Name of the page to close all popups on. If no name is specified, then the current page will have all popups closed. - **Example**: SEND_COMMAND Panel,”’^PPA-Page1’” Close all pop-ups on Page1.
^PPF **@PPF** **PPOF**	Popup Page Off Command - Detach a popup from a page. If the page name is empty, the current page is used. If the popup page is part of a group, the whole group is deactivated. This command works in the same way as the ‘Hide Popup’ command in TPDesign 5. The ‘@PPF’ and ‘PPOF’ commands are implemented for G4 compatibility. - **Syntax**: “’^PPF-<popup page name>;[optional page name]’” or “’@PPF-<popup page name>;[optional page name]’” or “’PPOF-<popup page name>;[optional page name]’” - **Variables**: *Popup page name*: name of the popup page. *page name*: name of the page the popup is displayed On. If not specified the popup is detached from the current page. - **Examples**: SEND_COMMAND Panel,”’^PPF-Popup1;Main’” Detach the popup ‘Popup1’ from page ‘Main’. SEND_COMMAND Panel,”’^PPF-Popup1’” Detach the popup page ‘Popup1’ from the current page.
^PPG **@PPG** **PPOG**	Toggle a Popup Page - Toggle a specific popup page. If the page name is empty, the current page is used. Toggling refers to the activating/deactivating (On/Off) of a popup page. This command works in the same way as the ‘Toggle Popup’ command in TPDesign. The ‘@PPG’ and ‘PPOG’ commands are implemented for G4 compatibility. - **Syntax**: “’^PPG-<popup page name>;[optional page name]’” or “’@PPG-<popup page name>;[optional page name]’” or “’PPOG-<popup page name>;[optional page name]’” - **Variables**: *Popup page name*: the name of the popup page. *Page name*: name of the page the popup is toggled on. If not specified the popup is toggled on the current page. - **Examples**: SEND_COMMAND Panel,”’^PPG-Popup1;Main’” Toggles the popup page ‘Popup1’ on the ‘Main’ page from one state to another (On/Off). SEND_COMMAND Panel,”’^PPG-Popup1’” Toggles the popup page ‘Popup1’ on the current page from one state to another (On/Off).
^PPK **@PPK**	Kill Popup Page Command - Kill a specific popup page from all pages. Kill refers to the deactivating (Off) of a popup window from all pages. If the pop-up page is part of a group, the whole group is deactivated. This command works in the same way as the ‘Clear Group’ command in TPDesign. The ‘@PPK’ command is implemented for G4 compatibility. - **Syntax**: “’^PPK-<popup page name>’” or “’@PPK-<popup page name>’” - **Variables**: *Popup page name*: name of the popup page. - **Example**: SEND_COMMAND Panel,”’^PPK-Popup1’” Kills the popup page ‘Popup1’ on all pages.

Page Commands	

^PPM
@PPM
	Popup modal command - Set whether a popup is modal or not modal. The ‘@PPM’ command is implemented for G4 compatibility.
	- **Syntax**:
	`“'^PPM-<popup page name>;<modal 1|0>'”`
	or
	`“'@PPM-<popup page name>;<modal mode 1|0>'”`
	- **Variables**:
	Popup page name: Name of the popup page.
	Modal mode: 1 if modal, 0 if non-modal.
	- **Example**:
	`SEND_COMMAND Panel,"'^PPM-Popup1;1'"`
	Set the popup page named Popup1 to modal mode.

^PPN
@PPN
PPON
	Attach a popup on a page - Attach a specific popup page to launch on either a specified page or the current page. If the page name is empty, the current page is used. If the popup page is already on, do not re-draw it. This command works in the same way as the ‘Show Popup’ command in TPDesign5. The ‘@PPN’ and ‘PPON’ commands are implemented for G4 compatibility.
	- **Syntax**:
	`“'^PPN-<popup page name>;[optional page name]'”`
	or
	`“'@PPN-<popup page name>;[optional page name]'”`
	or
	`“'PPON-<popup page name>;[optional page name]'”`
	- **Variables**:
	Popup page name: name of the popup page.
	page name: name of the page the popup is displayed On. If the page name is not specified the current page is used.
	- **Examples**:
	`SEND_COMMAND Panel,"'^PPN-Popup1;Main'"`
	Activates ‘Popup1’ on the ‘Main’ page.
	`SEND_COMMAND Panel,"'^PPN-Popup1'"`
	Activates the popup page *‘Popup1’* on the current page.

^PPT
@PPT
	Popup Timeout Command - Set the popup to close after timeout. The ‘@PPT’ command is implemented for G4 compatibility.
	- **Syntax**:
	`“'^PPT-<popup page name>;<timeout>'”`
	or
	`“'@PPT-<popup page name>;<timeout>'”`
	- **Variables**:
	Popup page name: the name of the popup to apply the timeout to. Popup must be visible on screen in order to apply timeout.
	Timeout: the time in tenths of seconds (10 = 1 second) or 0 to cancel timeout.
	Note: Successive calls to timeout will reset the timeout. A timeout of 0 cancels the timeout and the popup stays open.
	- **Examples**:
	`SEND_COMMAND Panel,'^PPT-MyPopup;150'`
	Close *MyPopup* after 15 seconds.

^PPX
@PPX
	Close All Popup Pages Command - Close all popups on all pages. This command works in the same way as the ‘Clear All’ command in TPDesign5. The ‘@PPX’ command is implemented for G4 compatibility.
	- **Syntax**:
	`“'^PPX'”`
	or
	`“'@PPX'”`
	- **Variables**: None
	-
	- **Example**:
	`SEND_COMMAND Panel,"'^PPX'"`
	Close all popups on all pages.

Page Commands	

^PTC	Toggle Collapsible Popup Collapsed Command - Toggles the named collapsible popup between the open and collapsed positions. More specifically, if the popup is not fully collapsed, it is collapsed.
	- **Syntax**:
	“’^PTC-<popup>;[optional target page]’”
	- **Variables**:
	Popup: the name of the popup to toggle
	Target page: name of the page hosting the popup to affect the change upon. If target page is not specified, the command is applied to the current page.
	- **Examples**:
	`SEND_COMMAND Panel,”’^PTC-Contacts’”`
	Toggle the Contacts popup collapsed on the current page.
	`SEND_COMMAND Panel,”’^PTC-Contacts;Teleconference Control’”`
	Toggle the Contacts popup collapsed on the Teleconference Control page.
	Note: Collapsible popup send commands do not automatically show the popup on the target page. The popup must be first shown with a standard show command. This applies even when the collapsible popup is a member of a popup group. For all of these commands, if the target page is blank, the current page is used. If the named popup is not collapsible, the commands are ignored.

^PTO	Toggle Collapsed Popup Open Command - Toggles the named collapsible popup between the open and collapsed positions. More specifically, if the popup is not fully open, it is opened.
	- **Syntax**:
	“’^PTO-<popup>;[optional target page]’”
	- **Variables**:
	Popup: the name of the popup to toggle
	Target page: name of the page hosting the popup to affect the change upon. If target page is not specified, the command is applied to the current page.
	- **Examples**:
	`SEND_COMMAND Panel,’^PTO-Contacts’`
	Toggle the Contacts popup open on the current page.
	`SEND_COMMAND Panel,’^PTO-Contacts;Teleconference Control’`
	Toggle the Contacts popup open on the Teleconference Control page.
	Note: Collapsible popup send commands do not automatically show the popup on the target page. The popup must be first shown with a standard show command. This applies even when the collapsible popup is a member of a popup group. For all of these commands, if the target page is blank, the current page is used. If the named popup is not collapsible, the commands are ignored.

Button Commands

<table>
<tr><th colspan="2">Button Commands</th></tr>
<tr><td><b>^ANI</b></td><td>Multistate Button Animation Command - Commands a multistate button to animate from a starting state to an ending state.
• <b>Syntax</b>:
<code>^ANI-&lt;addr range>,<start state>,<end state>,<time&gt;</code>
• <b>Variables</b>: <i>address range</i>: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. <i>start state</i>: Beginning of button state (0= current state). <i>end state</i>: End of button state. <i>time</i>: In 1/10 second intervals.
• <b>Example</b>:
<code>SEND_COMMAND Panel,”’^ANI-1,1,10,50’”</code>
Command button with Address 1 to animate from state 1 to state 10 over 5 seconds.</td></tr>
<tr><td><b>^APF</b></td><td>Add page flip action - Add page flip action to a button. This command installs a page flip command to the Button Release event action.
• <b>Syntax</b>:
<code>”’^APF-&lt;addr range>,<page flip action>,<page name> [,<animation>,[origin],[duration]]’”</code>
• <b>Variables</b>: <i>address range</i>: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. <i>page flip action</i>: (see the following): Stan[dardPage] - flip to standard page
StanAni - flip to standard page with animation
PrevAni - flip to previous page with animation
Prev[iousPage] - flip to previous page
Show[Popup] - Show popup page
Hide[Popup] - Hide popup page
Togg[lePopup] - toggle popup state
ClearG[roup] - clear popup page group from all pages
ClearP[age] - clear all popup pages from a page with the specified page name
ClearA[ll] - Clear all popup pages from all pages
<i>Page Name</i>: the name of the page to flip to, or name of popup to show/hide/toggle Animation: If animated flip, the animation to perform.
<i>Origin</i>: If animated flip, the origin of the animation.
<i>Duration</i>: Transition time in 10ths of a second. Range is 3-30 with 15 (1.5 seconds) as the default
<table>
<tr><th>Animation Name</th><th>Command Snytax* (see note below)</th><th>Origin(s)</th><th>Default Origin</th></tr>
<tr><td>Center Door Fade</td><td>cntrdrfade, centerdoorfade, or center door fade</td><td>top(2), bottom(3), left(4), right(5)</td><td>right(5)</td></tr>
<tr><td>Door Fade</td><td>doorfade, door fade, or door</td><td>top(2), bottom(3), left(4), right(5)</td><td>right(5)</td></tr>
<tr><td>Fade</td><td>fade</td><td>center(1)</td><td>center(1)</td></tr>
<tr><td>Slide</td><td>slide</td><td>top(2), bottom(3), left(4), right(5)</td><td>right(5)</td></tr>
<tr><td>Slide Bounce</td><td>sldbouce, slidebounce, or slide bounce</td><td>top(2), bottom(3), left(4), right(5)</td><td>right(5)</td></tr>
<tr><td>Spin In</td><td>spinin or spin in</td><td>center(1)</td><td>center(1)</td></tr>
<tr><td>Spin Out</td><td>spinout or spin out</td><td>center(1)</td><td>center(1)</td></tr>
<tr><td>Zoom In</td><td>zoomin or zoom in</td><td>center(1)</td><td>center(1)</td></tr>
<tr><td>Zoom Out</td><td>zoomout or zoom out</td><td>center(1)</td><td>center(1)</td></tr>
<tr><td colspan="4"><i>Note: Multiple aliases for the transition name command syntax are allowed to maintain backwards compatibility with G4.</i></td></tr>
</table>
• <b>Example</b>:
<code>SEND COMMAND Panel,”’APF-400,StanAni,Main Page,ZoomIn,30’”</code>
Add animated page flip action to button 400 to flip to Main Page using zoom in for 3 seconds.</td></tr>
<tr><td><b>^BAF</b></td><td>Append UTF-8 Text to State Command - append non-unicode text.
• <b>Syntax</b>:
<code>“’^BAF-&lt;addr range>,<button states range>,<new text&gt;’”</code></td></tr>
</table>

Button Commands	
	- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for general buttons 1 = Off state and 2 = On state). *new text*: UTF-8 encoded characters. - **Examples**: `SEND_COMMAND Panel,”’^BAF-520,1,ξεσκεπάζω τὴν ψυχοφθόρα βδελυγμία’”` Appends the UTF-8 text ‘ξεσκεπάζω τ?ν ψυχοφθόρα βδελυγμία’ to the button’s OFF state

^BAT	Append Text to State Command - Append non-unicode text.
	- **Syntax**:
	`“’^BAT-<addr range>,<button states range>,<new text>’”`
	- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	button states range: 1 - 256 for multi-state buttons (0 = All states, for general buttons 1 = Off state and 2 = On state).
	new text: ISO-8859-1 encoded characters
	- **Examples**:
	`SEND_COMMAND Panel,”’^BAT-520,1,Enter City’”`
	Appends the text ‘Enter City’ to the button’s OFF state.

^BAU	Append Unicode Text to State Command - Append unicode text. Same format as ^UNI.
	- **Syntax**:
	`“’^BAU-<addr range>,<button states range>,<unicode text>’”`
	- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state).
	unicode text: Unicode characters must be entered in Hex format.
	- **Example**:
	`SEND_COMMAND Panel,”’^BAU-520,1,00770062’”`
	Appends Unicode text “00770062” (‘wb’) to the button’s OFF state.

^BCB	Set Border Color Command - Set the border color to the specified color. Only if the specified border color is not the same as the current color.
	- **Syntax**:
	`“’^BCB-<addr range>,<button states range>,<color value>’”`
	- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	button states range: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
	color value: See color table for more information.
	Note: Colors can be set by Color Numbers, Color name, RGB alpha colors (RRGGBBAA) or RGB colors values (RRGGBB).
	RGBA and RGB color are given in HEX ASCII prepended by a ‘#’.
	- **Examples**:
	`SEND_COMMAND Panel,”’^BCB-500.504&510,1,12’”`
	Sets the Off state border color to 12 (Yellow).
	`SEND_COMMAND Panel,”’^BCB-520,2,#FF000080’”`
	Set the ON state border color to RED with opacity at 128 ($80 / 0x80).

?BCB	Get Border Color Command - Get the current border color. - **Syntax**: `“’?BCB-<addr range>,<button states range>’”` - **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between address range: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).

Button Commands

Value is returned in a custom event with the following properties:

```
Custom Event Property        Value
Port                         port command was received on
ID                           Address code of the button responding
Type                         1011
Flag                         0
Value 1                      Button state number
Value 2                      Actual length of string (should be 9)
Value 3                      0
Text                         Hex encoded color value (ex: #000000FF
```

- **Examples**:

```
SEND COMMAND Panel,"'?BCB-529,1'"
```

Gets the button 'OFF state' border color. information. The result sent to the Controller would be:

```
Custom Event Property        Value
Port                         port command was received on
ID                           529
Type                         1011
Flag                         0
Value 1                      1
Value 2                      9
Value 3                      0
Text                         #222222FF
```

^BCF

Background Color Fill Command - Set the background color fill to specified color in state(s).

- **Syntax**:

```
"'^BCF-<addr range>,<button state range>,<color value>'"
```

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
  color value: See the color table for details.

Note: Colors can be set by Color Numbers, Color name, RGB alpha colors (RRGGBBAA) or RGB colors values (RRGGBB).
RGBA and RGB color are given in HEX ASCII prepended by a '#'

- **Examples**:

```
SEND_COMMAND Panel,"'^BCF-500.504&510.515,1,Blue'"
```

Sets the OFF state background color fill for the buttons with variable text ranges of 500-504 & 510-515 to Blue.

?BCF

Get Fill Color Command - Get the current fill color.

- **Syntax**:

```
"'?BCF-<addr range>,<button states range>'"
```

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address.

  button states range: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
  Value is returned in a custom event with the following properties:

```
Custom Event Property        Value
Port                         port command was received on
ID                           Address code of the button responding
Type                         1012
Flag                         0
Value 1                      Button state number
Value 2                      Actual length of string (should be 9)
Value 3                      0
Text                         Hex encoded color value (ex: #000000FF)
```

- **Examples**:

```
SEND COMMAND Panel,"'?BCF-529,1'"
```

Gets the button 'OFF state' fill color. information. The result sent to the Controller would be:

```
Custom Event Property        Value
Port                         port command was received on
ID                           529
Type                         1012
```

Button Commands	
	Flag 0 Value 1 1 Value 2 9 Value 3 0 Text #FF8000FF
^BCT	Set Text Color Command - Set the text color to the specified color. • **Syntax**: "'^BCT-<addr range>,<button states range>,<color value>'" • **Variables**: *address range:* Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range:* 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *color value:* See the color table for details. *Note: Color can be assigned by color name (without spaces), number or R,G,B value (RRGGBB or RRGGBBAA).* • **Examples**: SEND_COMMAND Panel,"'^BCT-500.504&510,1,12'" Sets the OFF state text color to 12 (Very Light Yellow).
?BCT	Get Text Color Command - Get the current text color. • **Syntax**: "'?BCT-<addr range>,<button states range>'" • **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *Value* is returned in a custom event with the following properties: Custom Event Property Value Port port command was received on ID Address code of the button responding Type 1013 Flag 0 Value 1 Button state number Value 2 Actual length of string (should be 9) Value 3 0 Text Hex encoded color value (ex: #000000FF) • **Examples**: SEND COMMAND Panel,"'?BCT-529,1'" Gets the button 'OFF state' text color. information. The result sent to the Controller would be: Custom Event Property Value Port 529 ID Address code of the button responding Type 1013 Flag 0 Value 1 1 Value 2 9 Value 3 0 Text #FFFFEFF
^BDC	Button Drag and Drop Custom Event Command - This command configures Drag and Drop custom events. This command can be used to enable or disable the transmission of custom events to the controller whenever certain operations occur. For example, the system programmer may want to be notified whenever a drag button enters an acceptable target. The notification mechanism is a custom event. The ^BDC command takes the form of a comma separated list of custom event numbers. If the number is 0 or blank for a given event type then no custom event will be transmitted when that event occurs. If a number is specified, then it is used as the EVENT TYPE value for the custom event. The range of 32001 to 65535 has been reserved in the panel for user custom event numbers. A different value could be used but might collide with other AMX event numbers. Event configuration is not permanent and all event numbers revert to the default of 0 when the panel restarts. **Syntax**: "'^BDC-<drag start event number>,<enter valid drop target event number>, <exit valid drop target event number>,<drop event number>,<drag cancel event number>, <enter invalid drop target event number>,<exit invalid drop target event number>

Button Commands

Variables:

- *drag start event number: Value of a drag start event.*
- *enter valid drop target event number: Value of an enter valid drop target event.*
- *enter valid drop target event number: Value of an enter valid drop target event.*
- *exit valid drop target event number: Value of an exit valid drop target event.*
- *drop event number: Value of a drop event*
- *drag cancel event number: Value of a drag cancel event*
- *enter invalid drop target event number: Value of an enter invalid drop target event.*
- *exit invalid drop target event number: Value of an exit invalid drop target event.* By default the ^BDC command is enabled, the default values are:
- *DragStartedEvent =* **1410**
- *ValidDropEnterEvent =* **1411**
- *ValidDropExitEvent =* **1412**
- *DropEvent =* **1413**
- *DragCancelEvent =* **1414**
- *InvalidDropEnterEvent =* **1415**;
- *InvalidDropExitEvent =* **1416**

To disable the ^BDC command send: **^BDC-0,0,0,0,0,0,0** The events are:

- *DragStarted - a draggable button has initiated a drag*
- *ValidDropEntered - a draggable button has entered a valid target*
- *ValidDropExited - a draggable button has exited a valid target*
- *Drop - a draggable button has been dropped on a valid target*
- *DragCancel - a draggable button has been dropped outside of a valid target*
- *InvalidDropEntered - a draggable button has entered an invalid target*
- *InvalidDropExited - a draggable button has exited an invalid target*

In response to any or all of the above events, the panel will create a custom event which is then sent to the controller.

The format of **START** custom events transmitted to the controller are as follows:

```
  CUSTOM.TYPE = the specified drag event custom event type
(started)
  CUSTOM.ID     = the address of the viewer button which generated the event
  CUSTOM.FLAG   = 0
  CUSTOM.VALUE1 = the button address of the draggable
  CUSTOM.VALUE2 = 0
  CUSTOM.VALUE3 = 0
  CUSTOM.TEXT   =

'dr{ch=<channelPort>,<channel>:ad=<addressPort>,<address>:gp=<groupName>:nm=<buttonName>}

    dt{vl=<dropTargetValid
1=valid,0=invalid>:ch=<channelPort>,<channel>:ad=<addressPort>,
    <address>:nm=<buttonName>}…

    dt{vl=<dropTargetValid
1=valid,0=invalid>:ch=<channelPort>,<channel>:ad=<addressPort>,
<address>:nm=<buttonName>}'
```

The CUSTOM.TEXT provides data sets that represent the draggable's info (dr). The draggable's info included is the drag channel port, the drag channel code, the drag address port, the drag address code, the drag group name, and the drag button name. Drag target info is also presented, with a data set for each drag target visible at that time. The drag targets info (dt) includes the target validity to accept the drop, the drop target channel port, the drop target channel code, the drop target address port, the drop target address code, and the drop target button name.

- Buttons are identified as dr (draggable) or dt (drop target)
- Button properties are contained between open brace ({) and close brace (})
- Button properties are represented by key=value pairs (KVP).
- Keys are two letters followed by equal (=) by convention but the two letter keys are not a requirement.
- Property KVPs are separated by colon (:).

Each Button's data sets are on a separate line (i.e. the close brace is followed by a \n).

Key values.

- *dr* = draggable
- *ch* = channel (port,channel)
- *ad* = address (port,address)
- *gp* = group name
- *nm* = button name
- *dt* = drop target
- *vl* = validity of drop target (valid=1, invalid=0)

Button Commands

- *ch* = channel (port,channel)
- *ad* = address (port,address)
- *nm* = button nameBy default the ^BDC command is enabled, the default values are:

Example texts:

```
  dr{ch=1,31:ad=1,31:gp=:nm=Drag1}
dt{vl=1:ch=1,101:ad=1,101:nm=Tgt1}
dt{vl=1:ch=3,103:ad=3,103:nm=Tgt3}
dt{vl=1:ch=3,103:ad=3,103:nm=Tgt3}
dt{vl=0:ch=1,11:ad=1,11:nm=Grp1 Tgt1}
dt{vl=0:ch=1,12:ad=1,12:nm=Grp1 Tgt2}
dt{vl=0:ch=2,11:ad=2,11:nm=Grp2 Tgt1}
dt{vl=0:ch=1,15:ad=1,15:nm=Grp1 Tgt5}
dt{vl=0:ch=1,16:ad=1,16:nm=Grp1 Tgt6}
dt{vl=0:ch=2,13:ad=2,13:nm=Grp2 Tgt3}
dt{vl=0:ch=1,15:ad=1,15:nm=Grp1 Tgt5}
dt{vl=0:ch=1,16:ad=1,16:nm=Grp1 Tgt6}
dt{vl=0:ch=2,13:ad=2,13:nm=Grp2 Tgt3}

  dr{ch=2,4:ad=2,4:gp=Group1+2:nm=Drag2_4}
dt{vl=1:ch=1,11:ad=1,11:nm=Grp1 Tgt1}
dt{vl=1:ch=1,12:ad=1,12:nm=Grp1 Tgt2}
dt{vl=1:ch=2,11:ad=2,11:nm=Grp2 Tgt1}
dt{vl=1:ch=1,15:ad=1,15:nm=Grp1 Tgt5}
dt{vl=1:ch=1,16:ad=1,16:nm=Grp1 Tgt6}
dt{vl=1:ch=2,13:ad=2,13:nm=Grp2 Tgt3}
dt{vl=1:ch=1,15:ad=1,15:nm=Grp1 Tgt5}
dt{vl=1:ch=1,16:ad=1,16:nm=Grp1 Tgt6}
dt{vl=1:ch=2,13:ad=2,13:nm=Grp2 Tgt3}
dt{vl=0:ch=1,101:ad=1,101:nm=Tgt1}
dt{vl=0:ch=3,103:ad=3,103:nm=Tgt3}
dt{vl=0:ch=3,103:ad=3,103:nm=Tgt3}
```

The format of **VALIDENTER/VALIDEXIT/CANCEL** custom events transmitted to the controller are as follows:

```
  CUSTOM.TYPE   = the specified drag event (validEntered/validExited/drop/cancel)
  CUSTOM.ID     = the address of the drag/drop button which generated the event
  CUSTOM.FLAG   = 0 // 0 specifies valid
  CUSTOM.VALUE1 = the button address of the draggable
  CUSTOM.VALUE2 = 0
  CUSTOM.VALUE3 = 0
  CUSTOM.TEXT   = “”
```

The format of **INVALIDENTER/INVALIDEXIT** custom events transmitted to the controller are as follows:

```
  CUSTOM.TYPE   = the specified drag event (invalidEntered/invalidExited)
  CUSTOM.ID     = the address of the drag/drop button which generated the event
  CUSTOM.FLAG   = 65535 (-1) // -1 specifies invalid target
  CUSTOM.VALUE1 = the button address of the draggable
  CUSTOM.VALUE2 = 0
  CUSTOM.VALUE3 = 0
  CUSTOM.TEXT   = “”
```

If the **VALIDENTER** and **INVALIDENTER** events are set to the same event number, the flag value indicates whether the targets are valid or not. 0 == valid, 65535 (-1) == invalid.

If the **VALIDEXIT** and **INVALIDEXIT** events are set to the same event number, the flag value indicates whether the targets are valid or not. 0 == valid, 65535 (-1) == invalid.

The format of the **DROP** custom event transmitted to the controller is as follows:

```
  CUSTOM.TYPE   = the specified drag event (started/entered/exited/drop/cancel) the
address of the viewer button which generated the event
  CUSTOM.ID     = the address of the viewer button which generated the event
  CUSTOM.FLAG   = 0
  CUSTOM.VALUE1 = the button address of the draggable
  CUSTOM.VALUE2 = the button address of the dropTarget
  CUSTOM.VALUE3 = 0
  CUSTOM.TEXT   = group name to which the dropTarget belongs
```

Example:

```
  SEND_COMMAND panel,”’^BDC-32001,32002,32003,32004,32005’”
```

After the users sends this command to the panel, if the user then drags a button addressed 9 and then proceeds to drop that draggable button on a dropTarget button addressed 10, the following event would be transmitted to the controller.

```
  CUSTOM.ID     = 10 (the dropTarget receives the drop event)
  CUSTOM.TYPE   = 32004  (this our drop event)
  CUSTOM.FLAG   = 0
  CUSTOM.VALUE1 = 9  (the button we dragged over the target & dropped)
  CUSTOM.VALUE2 = 10  (the dropTarget that the draggable was dropped on)
  CUSTOM.VALUE3 = 0
```

Button Commands	
	CUSTOM.TEXT = “” (a name we had given to the group the target was assigned,since the target was not assigned to a group we’ll receive an empty string)

?BDC	Query Button Drag and Drop Custom Event Command - Get the drag and drop custom event values.
	• **Syntax:**
	“’?BDC’”
	• **Variables:** None
	The response returned is a custom event with the following syntax:
	CUSTOM.TYPE = 0
	CUSTOM.ID = 1332
	CUSTOM.FLAG = 0
	CUSTOM.VALUE1 = 0
	CUSTOM.VALUE2 = 0
	CUSTOM.VALUE3 = 0
	CUSTOM.TEXT = String containing a comma separated list of Button Drag & Drop Custom Event values ‘[StartEventNum],[ValidEnterEventNum],[ValidExitEventNum],[DropEventNum],[CancelEventNum],[InvalidEnterEventNum],[InvalidExitEventNum]’
	• **Example:**
	SEND_COMMAND Panel,”’?BDC’”
	Query the Controller Button Drag and Drop Custom Event values. Response would be similar to:
	Custom.ID = 0
	Custom.Type = 1332
	Custom.Flag = 0
	Custom.Value1 = 0
	Custom.Value2 = 0
	Custom.Value3 = 0
	Custom.Text = ‘1410,1411,1412,1413,1414,1415,1416’

^BFB	Button set feedback command - Set the feedback type of the button. ONLY works on General-type buttons. • **Syntax**: “’^BFB-<addr range>,<feedback type>’” • **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *feedback type*: None, Channel, Invert, On (Always on), Momentary. • **Example**: SEND_COMMAND Panel,”’^BFB-500,Momentary’” Sets the Feedback type of the button to ‘Momentary’.
^BIM	Button set input mask command - Set the input mask for the specified address. • **Syntax**: “’^BIM-<addr range>,<input mask>’” • **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *input mask*: Refer to Appendix C: Text Formatting for character types. • **Example**: SEND_COMMAND Panel,”’^BIM-500,AAAAAAAAAA’” Sets the input mask to ten ’A’ characters, that are required, to either a letter or digit (entry is required).
^BIT	Button Input Type Command - Modifies the keyboard type of the text input button(s) with given address(es). If this is sent to a button that is not a Text Input button, it has no effect. • **Syntax**: “’^BIT-<address range>,<Input Type>,<return port>’” • **Variables**:

<table>
<tr><th colspan="2">Button Commands</th></tr>
<tr><td></td><td><i>Address Range</i>: range of addresses that this command applies to
<i>Input Type</i>: Input Type to Change to, as specified here: http://developer.android.com/reference/android/text/InputType.html
1: Text
2: Number (standard keypad)
3: Telephone
4: Date/Time

<i>Return port</i>: The port number to return the response on if different than the port to which the command is sent.</td></tr>
<tr><td><b>^BMC</b></td><td>Button copy command - Copy attributes of the source button to all the destination buttons. Note that the source is a single button state. Each state must be copied as a separate command. The <codes> section represents what attributes will be copied. All codes are 2 char pairs that can be separated by comma, space, percent or just ran together.

• <b>Syntax:</b>
<code>“’^BMC-<addr range>,<button states range>,<source port>,<source address>,<sourcestate>,<codes>’”</code>

• <b>Variables:</b>
<i>address range</i>: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. <i>button states range</i>: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). <i>source port</i>: port number of button to copy from.
<i>source address</i>: address number of button to copy from.
<i>source state</i>: state number of button to copy from. <i>codes</i>:
BM - Picture/Bitmap
BR - Border
CB - Border Color
CF - Fill Color
CT - Text Color
EC - Text effect color
EF - Text effect
FT - Font
JB - Bitmap alignment
JT - Text alignment
OP - Opacity
SO - Button Sound
TX - Text
WW - Word wrap on/off

• <b>Examples:</b>
<code>SEND_COMMAND Panel,”’^BMC-425,1,1,500,1,BR’”</code>
or
<code>SEND_COMMAND Panel,”’^BMC-425,1,1,500,1,%BR’”</code>
Copies the OFF state border of button with a variable text address of 500 onto the OFF state border of button with a variable text address of 425.
<code>SEND_COMMAND Panel,”’^BMC-150,1,1,315,1,%BR%FT%TX%BM%CF%CT’”</code>
Copies the OFF state border, font, Text, bitmap, fill color and text color of the button with a variable text address of 315 onto the OFF state border, font, Text, bitmap, fill color and text color of the button with a variable text address of 150.

<i>Note: Use this command if you are using the panel’s default color palette. For custom color palettes, use ^BMF instead.</i></td></tr>
<tr><td><b>^BMF</b></td><td>Button Modify Command - Set any/all button parameters by sending embedded codes and data.

• <b>Syntax</b>:
<code>“’^BMF-<addr range>,<button states range>,<data>’”</code>
<i>Note: Many subcommands do not use button state information. Refer to the subcommand for details</i> • <b>Variables</b>: <i>address range</i>: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. <i>button states range:</i> 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
<i>data:</i>
<table>
<tr><td><code>‘%B<border style>’</code></td><td>Set the border style name. (No support for states.)
<i>Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.</i></td></tr>
<tr><td><code>‘%CB<on border color>’</code></td><td>Set Border Color.</td></tr>
<tr><td><code>‘%CF<on fill color>’</code></td><td>Set Fill Color.</td></tr>
<tr><td><code>‘%CT<on text color>’</code></td><td>Set Text Color.</td></tr>
<tr><td><code>‘%EC<text effect color>’</code></td><td>Set the text effect color.</td></tr>
</table></td></tr>
</table>

Button Commands		
	`‘%EF<text effect name>’`	Set the text effect. *Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.*
	`‘%EN<1 or 0>’`	Enable/disable a button.
	`‘%F’<primary_font_filename: primary_font_size>, <alternate_font_filename: alternate_font_size’`	Set the font filename and optional font size for the primary font and/or the alternate font.
	`‘%GC<bargraph slider color>’`	Set the bargraph slider color
	`‘%GD<bargraph ramp down>’`	Set the bargraph ramp down time in 1/10 second.
	`‘%GG<bargraph drag increment>’`	Set the bargraph drag increment. Refer to the ^GDI command for more information.
	`‘%GH<bargraph hi>’`	Set the bargraph upper limit.
	`‘%GI<bargraph invert>’`	Set the bargraph invert/non-invert.
	`‘%GL<bargraph low>’`	Set the bargraph lower limit.
	`‘%GN<bargraph slider name>’`	Set the bargraph slider name/Joystick cursor name. *Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.*
	`‘%GR<repeat interval’`	Set bargraph repeat interval.
	`‘%GU<bargraph ramp up>’`	Set the bargraph ramp up time in intervals of 1/10 second.
	`‘%GV<bargraph value>’`	Set the bargraph value.
	`‘%J’,<set text alignment 0-10>’`	As shown in the Justification Values table, BUT the 0 (zero) is absolute and followed by ’,<left>,<top>’
	`‘%JB<alignment of bitmap 0-10>’`	As shown in the Justification Values table BUT the 0 (zero) is absolute and followed by ’,<left>,<top>’
	`‘%JT<alignment of text 0-9>’`	As shown in the Justification Values table BUT the 0 (zero) is absolute and followed by ’,<left>,<top>’
	`‘%MI<mask image>’`	Set the mask image. Refer to the ^BMI command for more information. *Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.*
	`‘%MK<input mask>’`	Set the input mask of a text area. See the text input mask area for more information. *Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.*
	`‘%ML<max length>’`	Set the maximum length of a text area.
	`‘%MI<mask image>’`	Set the mask image. Refer to the ^BMI command for more information. *Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.*
	`‘%OP<0-255>’`	Set the button opacity to either Invisible (value=0) or Opaque (value=255).
	`‘%OP#<00-FF>’`	Set the button opacity to either Invisible (value=00) or Opaque (value=FF).
	`‘%OT<feedback type>’`	Set the Feedback (Output) Type to one of the following: None, Channel,Invert, ON (Always ON), or Momentary. *Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.*
	`‘%P<bitmap,bitmap_index, justification>’`	Set the picture/bitmap filename (empty is clear). *Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.*
	`‘%R<l,t,r,b’`	Sets button location and also resizes the button. For more information, please refer to the ^BSP command.
	`‘%OP<0-255>’`	Set the button opacity to either Invisible (value=0) or Opaque (value=255).
	`‘%SC<1 or 0>’`	Set the bitmap scale to fit.
	`‘%SF<1 or 0>’`	Set the focus for text area button. (No support for states.)
	`‘%SM’`	Submit a text for text area button. (No support for states.)
	`‘%SP<spacing>’`	Set subpage viewer subpage spacing. (No support for states.)
	`‘%SO<sound>’`	Set the button sound. *Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.*
	`’%SW<1 or 0>’`	Show/hide a button. (No support for states.)
	`‘%T<text >’`	Set the text using ASCII characters (empty is clear). Note: This parameter should be always used in its own BMF command, and should not be combined with other BMF subcommands.

Button Commands		
	‘%UN<Unicode text>’	Set the Unicode text. See ^UNI for the text format.
	‘%UT<UTF-8 text>’	Set the Unicode text. See ^UTF for the text format.
	‘%WW<1 or 0>’	Word wrap ON/OFF.
	For some of these commands and values, refer to the RGB Values for all 88 Basic Colors table. • **Example**: SEND_COMMAND Panel,”’^BMF-500,1,%B10%CFRed%CB Blue %CTBlack%Ptest.png’” Sets the button OFF state as well as the Border, Fill Color, Border Color, Text Color, and Bitmap. *Note: Use this command if you are using custom color palette for your panel. If you intend to use the default color palette, use ^BMC instead.* *Note: To accept unspecified parameters, use either ,, or ,-1. If left or top is unspecified, then the current values for the button will be used. If right or bottom is unspecified, the current width and height is used to maintain the button size.* *This effectively creates a button “move” command (also works with ^BSP).*	
^BMI	Set state mask image command - Assign a Chameleon mask image to those buttons with a defined address and state range. • **Syntax**: “’^BMI-<addr range>,<button states range>,<name of mask image>’” • **Variables**: address range: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. button states range: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). name of mask image: The filename of the mask image in the TPD5 file to use. • **Example**: SEND_COMMAND Panel,”’^BMI-500.504&510.515,1,mask.png’” Sets the OFF state mask image for the buttons with address ranges of 500-504 & 510-515 to mask.png.	
^BML	Set text input max length command - Set the maximum length of the text area button. If this value is set to zero (0), the text area has no max length. This is only for a Text area input button and not for a Text area input masking button. • **Syntax** “’^BML-<addr range>,<max length>’” • **Variables**: address range: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. max length: The maximum length in characters of a text input area. (0=no max length) • • **Example**: SEND_COMMAND Panel,”’^BML-500,20’” Sets the maximum length of the text area input button to 20 characters.	
^BMP	Set State Bitmap Command - Assign a picture to those buttons with a defined address range. • **Syntax**: “’^BMP-<addr range>,<button states range>,<name of bitmap/picture>,[bitmap index],[optional justification]’” • **Variables**: *variable text address range*: 1 - 4000. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). *name of bitmap/picture* : ASCII characters. *Optional bitmap index*: 0 - 5, the state bitmap index to assign the bitmap. If not present, will place the referenced bitmap in index 1. The indexes are defined as: 0 - Chameleon Image (if present) 1 - Bitmap 1 2 - Bitmap 2 3 - Bitmap 3 4 - Bitmap 4 5 - Bitmap 5 *Optional justification*: 0-10 where: 0 - Absolute position: If absolute justification is set, the next two parameters are the X and Y offset of the bitmap for the referenced index. 1 - top left 2 - top center 3 - top right	

Button Commands

4 - middle left
5 - middle center
6 - middle right
7 - bottom left
8 - bottom center
9 - bottom right
10 - scale to fit
11 - scale-maintain-aspect-ratio

If no justification is specified, the current justification is used.

- **Example**:
```
SEND_COMMAND Panel,”’^BMP-500.504&510.515,1,bitmap.png’”
```
Sets the OFF state picture for the buttons with variable text ranges of 500-504 & 510-515.

?BMP

Query State Bitmap Command - Get the current bitmap name.

- **Syntax**:
```
“’?BMP-<addr range>,<button states range>,[index]’”
```

- **Variables**:
variable text address range: 1 - 4000. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state).
Optional index: 0-5, the state bitmap index to assign the bitmap. If not present, will place the referenced bitmap in index 1. The indexes are defined as:
0 - Chameleon Image (if present)
1 - Bitmap 1
2 - Bitmap 2
3 - Bitmap 3
4 - Bitmap 4 5 - Bitmap 5
The response returned is a custom event with the following properties:
```
Custom Event Property        Value
  Port                       port command was received on
  ID                         address code of button
  Type                       1002
  Flag                       0
  Value 1                    state number
  Value 2                    length of text
  Value 3                    bitmap index
  Text                       bitmap name
```

- **Example**:
```
SEND_COMMAND Panel,”’?BMP-529,1’”
```
Gets the button “OFF state” bitmap information (index 1 since index is unspecified). Example response:
```
Custom Event Property        Value
  Port                       port command was received on
  ID                         529
  Type                       1002
  Flag                       0
  Value 1                    1
  Value 2                    9
  Value 3                    1
  Text                       Buggs.png
```

^BMX

Set State Bitmap Extended Command - Assign a picture with justifications to those buttons with a defined address range.

- **Syntax**:
```
“’^BMX-<addr range>,<button states range>,<name of
bitmap/picture/resource,index,justification>;
<name of
bitmap/picture/resource,index,justification>; <name
of bitmap/picture/resource,index, justification>’”
```

- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). *name of bitmap*: The filename of the bitmap in the TPD5 file to use.
Optional bitmap index: 0 - 5, the state bitmap index to assign the bitmap. If not present, will place the referenced bitmap in index 1. The indexes are defined as:

Button Commands	
	0 - Chameleon Image (if present) 1 - Bitmap 1 2 - Bitmap 2 3 - Bitmap 3 4 - Bitmap 4 5 - Bitmap 5 *Optional justification*: 0-11 where: 0 - Absolute position: If absolute justification is set, the next two parameters are the X and Y offset of the bitmap for the referenced index. 1 - top left 2 - top center 3 - top right 4 - middle left 5 - middle center 6 - middle right 7 - bottom left 8 - bottom center 9 - bottom right 10 - scale to fit 11 - scale-maintain-aspect-ratio If no justification is specified, the current justification is retained. • **Example**: `SEND_COMMAND Panel,”’^BMX-500.504&510.515,1,bitmap.png,1,5;bitmap2.png,2,0,100,50;bitmap3.png,3,1’”` Sets the OFF state pictures for the buttons with address ranges of 500-504 & 510-515 as follows: bitmap.png is assigned to index 1 and is middle center justified. bitmap2.png is assigned to index 2 and is absolute justified with an X offset of 100 and a Y offset of 50. bitmap3.png is assigned to index 3 and is top left justified.

?BMX	Query State Bitmap Extended Command - Get the current bitmap name and justification for one or all indexes.
	• **Syntax**:
	`“’?BMX-<addr range>,<button states range>,[index]’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). *bitmap index*: 0 - 5, the state bitmap index to assign the bitmap. If not present, will place the referenced bitmap in index 1. The indexes are defined as:
	0 - Chameleon Image (if present)
	1 - Bitmap 1
	2 - Bitmap 2
	3 - Bitmap 3
	4 - Bitmap 4 5 - Bitmap 5
	The response returned is a series of custom events (one for each valid index) with the following syntax:
	`Custom Event Property    Value`
	`Port                     Button Address code`
	`ID                       address code of button`
	`Type                     1018`
	`Flag                     0`
	`Value 1                  Button state number`
	`Value 2                  Length of Custom.Text`
	`Value 3                  Index of bitmap (0-5)`
	`Text                     String that describes the bitmap name/justification.`
	`The text looks like: “bitmapname,justification”`
	`If absolute justification is set, then the X and Y offset are appended to the description.`
	• **Example**:
	`SEND_COMMAND Panel,”’?BMX-529,1’”`
	Gets the button ‘OFF state’ bitmap information (all index with a bitmap since index is unspecified). Example response:
	Custom Event 1:
	`Custom.ID     = 529`
	`Custom.Type   = 1018`
	`Custom.Flag   = 0`
	`Custom.Value1 = 1`
	`Custom.Value2 = 34`
	`Custom.Value3 = 1`

Button Commands	

	`Custom.Text   = button-background.png,scale-to-fit`
	Custom Event 2:
	`Custom.ID     = 529`
	`Custom.Type   = 1018`
	`Custom.Flag   = 0`
	`Custom.Value1 = 1`
	`Custom.Value2 = 26`
	`Custom.Value3 = 2`
	`Custom.Text   = arrow.png absolute,200,100`
	Custom Event 3:
	`Custom.ID     = 529`
	`Custom.Type   = 1018`
	`Custom.Flag   = 0`
	`Custom.Value1 = 1`
	`Custom.Value2 = 22`
	`Custom.Value3 = 3`
	`Custom.Text   = img_Varia,middle-center`
	For this case, 3 bitmaps are defined and 3 custom event s are sent as a response.

^BOP	Button Opacity Command - Set the button opacity in the selected state(s).
	• **Syntax**:
	`“’^BOP-<addr range>,<button state range>,<opacity>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). *opacity*: An integer value from 0-255 where 0 is fully transparent and 255 is fully opaque, or #XX where the value after the # is a HEX number between 0 and FF.
	•
	• **Example**:
	`SEND_COMMAND Panel,”’^BOP-500.504&510.515,1,200’”`
	Sets the OFF state opacity for the buttons with address ranges of 500-504 & 510-515 to 200. `SEND_COMMAND Panel,”’^BOP-500.504&510.515,1,#C8’”`
	Sets the OFF state opacity for the buttons with address ranges of 500-504 & 510-515 to 200 (0xC8).

?BOP	Get button opacity command - Get the overall button opacity.
	• **Syntax**
	`“’?BOP-<addr range>,<button states range>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	button states range: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). The response returned is a series of custom events (one for each valid index) with the following syntax:
	`Custom Event Property   Value`
	`Port                    port command was received on`
	`ID                      address code of button`
	`Type                    1015`
	`Flag                    0`
	`Value 1                 state number`
	`Value 2                 opacity`
	`Value 3                 0`
	`Text`
	• **Examples**:
	`SEND COMMAND Panel,”’?BOP-529,1’”`
	Gets the button ‘OFF state’ opacity information. The result sent to the Controller would be:
	`Custom Event Property   Value`
	`Port                    port command was received on`
	`ID                      529`
	`Type                    1015`
	`Flag                    0`
	`Value 1                 1`
	`Value 2                 200`
	`Value 3                 0`
	`Text`

Button Commands	

^BOS	Button State Video Fill Command - Sets the button state to display either a Video or Non-Video window.
	• **Syntax**
	`“’^BOS-<addr range>,<button states range>,<video state>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *video state*: Video Off = 0, URL Video On = 1, MPL Video On = 101.
	• **Example**:
	`SEND_COMMAND Panel,”’^BOS-500,1,1’”`
	Sets the button to display video.

?BOS	Query Button State Video Fill Command - get the current button state video fill.
	• **Syntax**:
	`“’?BOS-<addr range>,<button states range>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state).
	The response returned is a custom event with the following syntax:
	`Custom Event Property   Value`
	`  Port                    port command was received on`
	`  ID                      address code of button`
	`  Type                    1017`
	`  Flag                    0`
	`  Value 1                 state number`
	`  Value 2                 video state`
	`video state values:`
	`                            0 = no video fill`
	`100 = video fill`
	`101 = MPL video fill`
	`  Value 3                 0`
	`  Text                       video URL (or empty if no`
	`video)`
	• **Example**:
	`SEND_COMMAND Panel,”’?BOS-560,1’”`
	Gets the button “OFF state” video fill. Example response:
	`Custom Event Property   Value`
	`  Port                    port command was received on`
	`  ID                      560`
	`  Type                    1017`
	`  Flag                    0`
	`  Value 1                 1`
	`  Value 2                 1`
	`  Value 3                 0`
	`  Text                       1`

^BRD	Button state border command - Set the border of a button state/states.
	• **Syntax**
	`“’^BRD-<addr range>,<button states range>,<border name>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). *border name*: Refer to the Border Styles
	•
	• **Examples**:
	`SEND_COMMAND Panel,”’^BRD-500.504&510.515,1&2,Double Line’”`
	Sets the border by name (Double Line) to those buttons with the variable text range of 500-504 & 510-515.

?BRD	Get border name command - Get the current border name. • **Syntax**: `“’?BRD-<addr range>,<button states range>’”`

Button Commands	

	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state).
	The response returned is a custom event with the following syntax:
	`Custom Event Property   Value`
	`  Port                    port command was received on`
	`  ID                      address code of button`
	`  Type                    1014`
	`  Flag                    0`
	`  Value 1                 state number`
	`  Value 2                 text length`
	`  Value 3                 0`
	`  Text                       borner`
	`name`
	
	• **Example**:
	`SEND COMMAND Panel,”’?BRD-529,1’”`
	Gets the button “OFF state’ border information. The result sent to the Controller would be:
	`Custom Event Property   Value`
	`  Port                    port command was received on`
	`  ID                      529`
	`  Type                    1014`
	`  Flag                    0`
	`  Value 1                 1`
	`  Value 2                 22`
	`  Value 3                 11`
	`  Text                       Double Line`

^BSF	Button Focus Command - Set the focus to the text area.
	Note: Select one button at a time (single variable text address). Do not assign a variable text address range to set focus to multiple buttons. Only one variable text address can be in focus at a time.
	
	• **Syntax**:
	`“’^BSF-<addr range>,<selection value>’”`
	
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	selection value: Unselect = 0 and select = 1.
	
	• **Example**:
	`SEND_COMMAND Panel,”’^BSF-500,1’”`
	Sets the focus to the text area of the button.

^BSM	Button Submit Text Command - This command causes the text areas to send their text as strings to the NetLinx Controller.
	
	• **Syntax**
	`“’^BSM-<addr range>’”`
	
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	
	• **Example**:
	`SEND_COMMAND Panel,”’^BSM-500’”`
	Returns a String of format `“’<button name>-<text>’”`. The string is returned on the port a ^BIT command was received on, or if that has not occurred, is sent on the address port.

^BSO	Button state sound - Set the sound played when a button is pressed. If the sound name is blank, the sound is then cleared. If the sound name is not matched, the button sound is not changed. • **Syntax**: `“’^BSO-<addr range>,<button states range>,<sound name>’”` • **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *sound name*: Sound file name. If blank or file not found the sound is cleared.

Button Commands	
	• **Example**: `SEND_COMMAND Panel,”’^BSO-500,1&2,music.wav’”` Assigns the sound ‘music.wav’ to the button Off/On states.

^BSP	Set Button Size and Position Command - Set the button size and its position on the page.
	
	• **Syntax**:
	`“’^BSP-<addr range>,<left>,<top>,<right>,<bottom>’”`
	
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	left: position of left edge of the button on the panel
	top: position of the top edge of the button on the panel
	right: position of right edge of the button on the panel
	bottom: position of the bottom edge of the button on the panel
	
	• **Example**:
	`SEND_COMMAND Panel,”’^BSP-530,20,100,50,130’”`
	Makes the button with variable text address 530 appear at (20,100) and be 30px by 30px
	
	This command supports default parameters to simplify operations such as button moves, where you don’t want to calculate a right and bottom. To specify a default parameter it is recommended to simply use -1.
	
	The meaning of a given default parameter is as follows:
	left: use the current left position
	top: use the current top position
	right: calculate a new right position which is the left position plus the width
	bottom: calculate a new bottom position which is the top position plus the height
	
	Note: If left or top is unspecified, then the current values for the button will be used. If right or bottom is unspecified, the current width and height is used to maintain the button size. This effectively creates a button “move” command (also works with %R in ^BMF).
	
	• **Example** *(An easy button move of button 530 to position LEFT 20, TOP 100 while maintaining the button size):*
	`SEND_COMMAND Panel,”’^BSP-530,20,100,-1,-1’”`

^BWW	Button State Word Wrap Enable/Disable - Set the button word wrap feature to those buttons with a defined address range. By default, word-wrap is Off.
	
	• **Syntax**:
	`“’^BWW-<addr range>,<button states range>,<word wrap>’”`
	
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
	word wrap: 0=Off and 1=On. Default is Off.
	
	• **Example**:
	`SEND_COMMAND Panel,”’^BWW-500,1,1’”`
	Sets the word wrap on for the button’s Off state.

?BWW	Get Button State Word Wrap - Get the current word wrap flag status.
	
	• **Syntax**:
	`“’?BWW-<addr range>,<button states range>’”`
	
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state).
	Response is a custom event with the following properties:
	`Custom Event Property   Value`
	`  Port                    port command was received on`
	`  ID                      address of the button`
	`  Type                    1010`
	`  Flag                    0`
	`  Value 1                 state number`

Button Commands	

	`Value 2                  0 = no word wrap, 1 = word wrap`
	`Value 3                  0`
	`Text`
	• **Example**:
	`SEND COMMAND Panel,"'?BWW-529,1'"`
	Gets the button 'OFF state' word wrap information. The result sent to the Controller would be:
	`Custom Event Property   Value`
	`Port                    port command was received on`
	`ID                      529`
	`Type                    1010`
	`Flag                    0`
	`Value 1                 1`
	`Value 2                 1`
	`Value 3                 0`
	`Text`

^CPF	Clear Page Flip Command - Clear all page flips from a button. This only clears PageFlip actions from the Button Release event action.
	• **Syntax**:
	`"'^CPF-<address range>'"`
	• **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address.
	• **Example**:
	`SEND_COMMAND Panel,"'^CPF-500'"`
	Clear all page flip actions from button address 500 RELEASE event action list.

^DPF	Delete Page Flips Command - Delete page flips from a button release event if it already exists.
	• **Syntax**
	`"'^DFP-<addr range>,<actions>,<page name>'"`
	• **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *actions*:
	Stan[dardPage] - Flip to standard page
	Prev[iousPage] - Flip to previous page
	Show[Popup] - Show Popup page
	Hide[Popup] - Hide Popup page
	Togg[lePopup] - Toggle popup state
	ClearG[roup] - Clear popup page group from all pages
	ClearP[age] - Clear all popup pages from a page with the specified page name
	ClearA[ll] - Clear all popup pages from all pages *page name*: name of page or popup to affect.
	• **Example**:
	`SEND COMMAND Panel,"'^DPF-409,Prev'"`
	Deletes the assignment of a button from flipping to a previous page.

^ENA	Button Enable Command - Enable or disable buttons with a set variable text range.
	• **Syntax**:
	`"'^ENA-<addr range>,<command value>'"`
	• **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *command value*: 0 = disable, 1 = enable
	•
	• **Example**:
	`SEND_COMMAND Panel,"'^ENA-500.504&510.515,0'"`
	Disables buttons with variable text range 500-504 & 510-515.

^FON	Button state set font command - Set a font to a specific font filename and size for those buttons with a defined address range. • **Syntax**

Button Commands

```
"'^FON-<addr range>,<button states range>,<font filename>[:font size],[alternate font
filename] [:alternate font size]'"
```

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
  font filename: The filename of the font to display in the state. This is used as the primary font file for all button states *font size (optional)*: The size of the font to use. *alternate font filename*: The filename of the alternate font to display in the state. This is used as the alternate font file for a Listview button *font size (optional)*: The size of the alternate font to use in a Listview button.

- **Examples**:

```
SEND_COMMAND Panel,"'^FON-500.504&510.515,1&2,arialb.ttf:48'"
```

Sets the font file to arial bold (arialb.ttf) for the On and Off states of buttons with the address range of 500-504 & 510-515. Set the font size to 48.

```
SEND_COMMAND Panel,"'^FON-505,1&2,arialb.ttf:48,arial.ttf:24'"
```

Sets the primary font file to arial bold (arialb.ttf) for the selected (2) and unselected (1) states of Listview buttons with the address range of 505. Set the primary font size to 48. Sets the alternate font file to arial (arial.ttf) and the alternate font size to 24.

?FON

Get button state font command - Get the current font filename and size.

- **Syntax**:

```
"'?FON-<addr range>,<button states range>'"
```

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). Response is a custom event with the following properties:

```
Custom Event Property   Value
  Port                    port command was received on
  ID                      address of the button
  Type                    1007
  Flag                    0
  Value 1                 state number
  Value 2                 font index
Value 3                 font size
  Text                    font filename
```

If the button is a Listview, an additional custom event with the following properties are sent as well.

```
Custom Event Property   Value
  Port                    port command was received on
  ID                      address of the button
  Type                    1019
  Flag                    0
  Value 1                 state number
  Value 2                 0
  Value 3                 alternate font size
  Text                    alternate font filename
```

- **Example**:

```
SEND COMMAND Panel,"'?FON-529,1'"
```

Gets the button 'OFF state' font information. The result sent to the Controller would be:

```
Custom Event Property   Value
  Port                    port command was received on
  ID                      529
  Type                    1007
  Flag                    0
  Value 1                 1
  Value 2                 1
  Value 3                 48
  Text                    arialb.ttf
```

^GDI

Bargraph drag increment command - Change the bargraph drag increment.

- **Syntax**:

```
"'^GDI-<addr range>,<bargraph drag increment>'"
```

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address.
  bargraph drag increment: The amount to change the level on a drag. The default drag increment is 256.

Button Commands	
	• **Example**: `SEND_COMMAND Panel,”’^GDI-7,128’”` Sets the bargraph with address code 7 to a drag increment of 128.

^GIV	Bargraph invert command - Invert the bargraph to move in the opposite direction.
	• **Syntax**
	`“’^GIV-<addr range>,<invert=1, non-inverted=0>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *invert flag*: For a bargraph 1 = Invert, 0 = Non Invert
	• **Example**:
	`SEND_COMMAND Panel,”’^GIV-500,1’”`
	Invert the bargraph.

^GLH	Set Bargraph High Range Command - Sets the bargraph max range to <bargraph hi>. This does NOT affect the LEVEL value (if any) associated with this bargraph.
	• **Syntax**:
	`“’^GLH-<addr range>,<bargraph hi>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	bargraph hi: The new high value. It must be larger than the current low value.
	• **Example**:
	`SEND_COMMAND Panel,”’^GLH-100,128’”`
	Set the max bargraph value to 128.

^GLL	Set Bargraph Low Range Command - Sets the bargraph min range to <bargraph low>. This does NOT affect the LEVEL value (if any) associated with this bargraph.
	• **Syntax**:
	`“’^GLL-<addr range>,<bargraph low>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	bargraph low: The new low value. It must be smaller than the current high value.
	• **Example**:
	`SEND_COMMAND Panel,”’^GLL-100,64’”`
	Set the min bargraph value to 64.

^GRD	Bargraph set ramp down time command - Change the bargraph ramp-down time in 1/10th of a second increments.
	• **Syntax**
	`“’^GRD-<addr range>,<bargraph ramp down time>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *bargraph ramp down time*: Time to ramp down the entire range in 1/10th of a second intervals
	• **Example**:
	`SEND_COMMAND Panel,”’^GRD-500,200’”`
	Changes the bargraph ramp down time to 20 seconds.

^GRU	Bargraph set ran up time command - Change the bargraph ramp-up time in 1/10th of a second increments.
	• **Syntax**:
	`“’^GRU-<addr range>,<bargraph ramp up time>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *bargraph ramp up time*: Time to ramp down the entire range in 1/10th of a second intervals
	•
	• **Example**:
	`SEND_COMMAND Panel,”’^GRU-500,100’”`
	Changes the bargraph ramp up time to 10 seconds.

Button Commands	

^GSC	Bargraph set slider color command - Change the bargraph slider color. A user can also assign the color by name or R,G,B value RRGGBB or RRGGBBAA).
	- **Syntax**:
	`“’^GSC-<addr range>,<color value>’”`
	- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	color value: See the color table for more information.
	Note: Colors can be set by Color Numbers, Color name, RGB alpha colors (RRGGBBAA) or RGB colors values (RRGGBB). RGBA and RGB color are given in HEX ASCII prepended by a ‘#’.
	- **Example**:
	`SEND_COMMAND Panel,”’^GSC-500,12’”`
	Changes the bargraph slider color to Very Light Yellow.

^GSD	Bargraph slider display type command - Sets the display type for a slider. In G5, the default bargraph display type is to allow the center of the slider to move to the end of the bargraph and will be clipped visually. In G4 (legacy), the bargraph display type is to allow only the end of the slider to move to the end of the bargraph and the slider is not clipped visually.
	This command allows the bargraph slider display type to be changed from the G5 (default) type to the G4 type.
	- **Syntax**:
	`“’^GSD-<addr range>,<display type (g4 or g5)>’”`
	- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *display type*: Set the slider display type. A value of g4 will set the display to the G4 type, anything else will set to the G5 (default) type.
	- **Example**:
	`SEND_COMMAND Panel,”’^GSD-10,g4’”`
	Set the display type of the bargraph with address code 10 to the g4 (legacy) type.
	`SEND_COMMAND Panel,”’^GSD-10,g5’”`
	Set the display type of the bargraph with address code 10 to the g5 (default) type.

^GSN	Bargraph set slider name command - Change the bargraph slider name. Slider names can be found in the TPDesign5 slider name drop-down list.
	- **Syntax**:
	`“’^GSN-<addr range>,<bargraph slider name>’”`
	- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *bargraph slider name*: Name of valid sliders. At this point, the valid names are none, Circle -L, Circle -M, Circle -S, Precision, Rectangle -L, Rectangle -M, and Rectangle -S.
	- **Example**:
	`SEND_COMMAND Panel,”’^GSN-500,Rectangle -S’”`
	Changes the bargraph slider name to ’Rectangle -S’.

^JSB	Set button state bitmap alignment command - Set bitmap/picture alignment using a numeric keypad layout for those buttons with a defined address range. The alignment of 0 is followed by ‘,<left>,<top>’. The left and top coordinates are relative to the upper left corner of the button.
	- **Syntax**
	`“’^JSB-<addr range>,<button states range>,<new alignment>’”`
	- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). new alignment: Value of 0- 11 (see Justification Values).
	- **Example**:
	`SEND_COMMAND Panel,”’^JSB-500.504&510.515,1&2,1’”`
	Sets the off/on state bitmap alignment to upper left corner for those buttons with address ranges of 500-504 & 510-515.

?JSB	Get button state bitmap alignment value - Get the current bitmap alignment. - **Syntax**: `“’?JSB-<addr range>,<button states range>’”`

Button Commands

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). *index*: The bitmap index to get the value of.
  Response is a custom event with the following properties:

```
Custom Event Property   Value
  Port                    port command was received on
  ID                      address of the button
  Type                    1005
  Flag                    0
  Value 1                 state number
  Value 2                 alignment value 0-10
  Value 3                 bitmap index
  Text                    alignment description
```

The alignments description will be one of the following: *absolute, top-left, top-center, top-right, middle-left, middle-center, middle-right, bottom-left, bottom-center, bottom-right, scale-to-fit, scale-maintain-aspect-ratio*. If the alignment is *absolute*, the X and Y offsets will be specified in the text as well: *absolute,xoffset,yoffset*

- **Example**:

```
SEND COMMAND Panel,"'?JSB-529,1,2'"
```

Gets the button 'OFF state' bitmap justification information for bitmap at index 2. The result sent to the Controller would be:

```
Custom Event Property   Value
  Port                    port command was received on
  ID                      address of the button
  Type                    1005
  Flag                    0
  Value 1                 state number
  Value 2                 5
  Value 3                 2
  Text                       middle-center
```

^JST

Set button state text alignment command - Set text alignment for those buttons with a defined address range. The alignment of 0 is followed by ',<left>,<top>'. The left and top coordinates are relative to the upper left corner of the button.

- **Syntax**:

```
"'^JST-<addr range>,<button states range>,<new alignment>'"
```

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address.
  button states range: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
  new alignment: Value of 0- 11 (see Justification Values).

- **Example**:

```
SEND_COMMAND Panel,"'^JST-500.504&510.515,1&2,5'"
```

Sets the off/on state text alignment to middle-center for those buttons with address ranges of 500-504 & 510-515.

?JST

Get button state bitmap alignment value.

- **Syntax**:

```
"'?JST-<addr range>,<button states range>'"
```

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). Response is a custom event with the following properties:

```
Custom Event Property   Value
  Port                    port command was received on
  ID                      address of the button
  Type                    1004
  Flag                    0
  Value 1                 state number
  Value 2                 alignment value 0-10
  Value 3                 0
  Text                    alignment description
```

The alignments description will be one of the following: *absolute, top-left, top-center, top-right, middle-left, middle-center, middle-right, bottom-left, bottom-center, bottom-right, scale-to-fit*.
If the alignment is *absolute*, the X and Y offsets will be specified in the description as well: *absolute,xoffset,yoffset* • **Example**:

```
SEND COMMAND Panel,"'?JST-529,1,2'"
```

Gets the button 'OFF state' text justification information. The result sent to the Controller would be:

```
Custom Event Property   Value
  Port                    port command was received on
```

<table>
<tr><th colspan="2">Button Commands</th></tr>
<tr><td></td><td><code> ID   address of the button</code>
<code> Type   1004</code>
<code> Flag   0</code>
<code> Value 1   state number</code>
<code> Value 2   0</code>
<code> Value 3   0</code>
<code> Text   absolute,10,10</code></td></tr>
<tr><td><b>^SAD</b></td><td>Subpage add command - Adds a subpage to a viewer button without changing the anchor subpage.
If the named subpage is not present in the set it will be added in the specified position. If no position parameter is supplied the subpage is added to the end of the set. The anchor subpage will not be changed.
If the named subpage is already present, it will be hidden from the set and re-added in the specified position. The anchor subpage will not be changed, unless the named subpage is currently the anchor. In that case, the next appropriate subpage will become the anchor and the named subpage will be added at the appropriate position.
If no subpages are in the set, this command is effectively a Subpage Show command (^SSH).
• <b>Syntax</b>:
<code>"'^SAD-<addr range>,<name>,<optional position>,<optional time>'"</code>
• <b>Variables</b>: <i>address range</i>: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. <i>name</i>: Specifies the name of the subpage to be shown or added. <i>position</i>: Specifies where to add the named subpage in the set with 0 representing the beginning of the set. If this value is left out (or set to 65535) then the new subpage is placed at the end of the list. <i>time</i>: Can range from 0 to 30 and represents tenths of a second. This is the amount of time used to move the subpages around when subpages are added or removed from a button.
• <b>Example</b>:
<code>SEND_COMMAND Panel,"'^SAD-400,media1'"</code>
Add the media1 subpage at the end of the set.</td></tr>
<tr><td><b>^SCE</b></td><td>Subpage custom event command - Configure subpage custom events. This command can be used to enable or disable the transmission of custom events to the controller whenever certain operations occur. For example, the system programmer may want to be notified whenever a subpage enters the anchor position. The notification mechanism is a custom event. The ^SCE command takes the form of a addr range specifying one or more subpage viewer buttons followed by a comma separated list of custom event numbers. If the number is 0 or blank for a given event type then no custom event will be transmitted when that event occurs. If a number is specified, then it is used as the EVENTID value for the custom event. The range of 32001 to 65535 has been reserved in the panel for user custom event numbers. A different value could be used but might collide with other AMX event numbers. Event configuration is not permanent and all event numbers revert to the default of 0 when the panel restarts.
• <b>Syntax</b>:
<code>"'^SCE-<addr range>,<optional anchor event num>,<optional onscreen event num>,</code>
<code><optional offscreen event num>,<optional reorder event num>'"</code>
• <b>Variables</b>: <i>address range</i>: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. <i>anchor event number</i>: 0 for no event or a value from 32001 to 65535. <i>onscreen event number</i>: 0 for no event or a value from 32001 to 65535. <i>offscreen event number</i>: 0 for no event or a value from 32001 to 65535. <i>reorder event number</i>: 0 for no event or a value from 32001 to 65535.
The events are:
• <i>anchor</i> - a new subpage has docked in the anchor position.
• <i>onscreen</i> - a docking operation has been completed and the subpages in the list are now onscreen. This list will include the anchor along with any subpages that may be partially onscreen.
• <i>offscreen</i> - a docking operation has been completed and the subpages in the list are now offscreen.
• <i>reorder</i> - the user has reordered the subpages in the set and the list contains all subpages in the new order without regard to onscreen or offscreen state.
In response to any or all of the above events, the panel will create a string which is a list of subpage names separated by a pipe (|) character. The string for the anchor event is a single subpage name. If this string is too long to be transmitted in a single custom event, then multiple custom events will be created and transmitted. If defined, the events are sent in this order when a docking operation completes on a given viewer button: anchor, onscreen, offscreen. If reorder is defined and occurs, it is sent first: reorder, anchor, onscreen, offscreen.
The format of the custom event transmitted to the controller is as follows:
<code>Custom Event Property   Value</code>
<code> Port   port command was received on</code>
<code> ID   address of the button generating the event</code>
<code> Type   the non-zero event number in the ^SCE command</code>
<code> Flag   0</code>
<code> Value 1   which one of possible multiple events this is (1 based)</code>
<code> Value 2   total number of events needed to send the entire string</code>
<code> Value 3   the total size of the original string in bytes</code>
<code>Text   pipe character separated list of subpage names</code>
• <b>Example</b>:
<code>SEND_COMMAND Panel, "'^SCE-200,32001,0,0,0'"</code>
If the subpage named TV_Favorite_SyFy enters the anchor position on a subpage viewer button with an address of 200, the following event would be transmitted to the controller when the user had sent this command to the panel:
<code>Custom Event Property   Value</code>
<code> Port   port command was received on</code>
<code> ID   200</code>
<code> Type   32001</code>
<code> Flag   0</code></td></tr>
</table>

Button Commands	

	`Value 1                 1`
	`Value 2                 1`
	`Value 3                 16`
	`Text                       TV_Favorite_SyFy`

?SCE	Query Subpage Custom Event Numbers Command - Query the assigned subpage custom event numbers for a subpage viewer button. A series of custom events for the subpage viewer button may be sent as a response.
	• **Syntax**:
	`"'?SCE-<addr range>'"`
	• **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address.
	The format of the custom event transmitted to the controller is as follows:
	`Custom Event Property   Value`
	`Port                    port command was received on`
	`ID                      address of the button generating the event`
	`Type                    the non-zero event number in the ^SCE command`
	`Flag                    0`
	`Value 1                 which one of possible multiple events this is (1 based)`
	`Value 2                 total number of events needed to send the entire string`
	`Value 3                 the total size of the original string in bytes`
	`Text                    pipe character separated list of subpage names`
	• **Example** (Assuming the previous command, '^SCE-200,32001,0,0,0', has been sent…):
	`SEND_COMMAND Panel, "'?SCE-200'"`
	If the subpage named TV_Favorite_SyFy enters is in the anchor position on a subpage viewer button with an address of 200, the following event would be transmitted to the controller when the user had sent this command to the panel:
	`Custom Event Property   Value`
	`Port                    port command was received on`
	`ID                      200`
	`Type                    32001`
	`Flag                    0`
	`Value 1                 1`
	`Value 2                 1`
	`Value 3                 16`
	`Text                       TV_Favorite_SyFy`

^SDL	Streaming digital video loop count - This command allows a button state that has video fill to a streaming URL to set a number of times to play a video. This applies to local file video streams primarily.
	• **Syntax**:
	`"'^SDL-<Address range>,<State range>,<loop count>'"`
	• **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *loop count*: number of times to loop a completed video. 0 = loop indefinitely (default), >0 = number of times to loop.
	• **Example:**
	`SEND_COMMAND Panel,"'^SDL-10,1&2,1'"`
	Set the loop count to 1 for address 10 on and off states.

^SDM	Button State Streaming Digital Media Command - Starts or stops a streaming session. Stream starts if a valid URL is specified and stops if server URL string is empty or invalid. To use this command, the current page should have one visible streaming button.
	• **Syntax**:
	`"'^SDM-<address range>,<button states range>,<URL>'"`
	• **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
	URL: <protocol://><host name or host ip><:video port><:optional audio port>
	Protocol could have the following values: udp = MPEG2 transport stream over UDP http = Motion JPEG (JFIF format over MIME Multipart) [Varia Panels Only] rtpmpeg2 = MPEG2 elementary stream over RTP/RTCP [Varia Panels do not support] rtpmpeg4 = MPEG4 elementary stream over RTP/RTCP [Varia Panels do not support] If the optional audio port is not specified, video port + 2 is used for audio.
	Playing a video f ile stored on a USB drive attached to the panel
	Enter the path of the video file on the attached USB drive with **"file:///udisk/"** as the prefix:
	`"'^SDM-<Address range>,<State range>,file:///udisk/path_to_video_file_on_usb_drive'"`
	Note: There are three slashes after "file:", not two as in a standard URL. If there aren't three slashes, the video file won't be found to be played.
	For example, for a video file named "*test-video.mp4*" in a directory named "*videos*" on the USB drive, enter:
	"*file:///udisk/videos/test-video.mp4*" **Playing a video file stored on the panel**
	Enter the filename of the video file with "**amxdir:///**" as the prefix.
	`"'^SDM-<Address range>,<State range>,amxdir:///video_file'"`
	Note: There are three slashes after "amxdir:", not two as in a standard URL. If there aren't three slashes, the video file won't be found.

Button Commands	

	For example, for a video file named “*test-video.mp4*”, enter:
	“*amxdir:///test-video.mp4*”
	To change the video using the ^SDM command to a different video (that has been transfered to the panel), use the same URL scheme as the prefix (amxdir:///).
	Note that any files that are transfered to the amxdir:/// directory are not cleared by a panel file transfer or via “Remove User Pages”. The only way to clear the file is to do a Factory Data Reset, or to upload an empty file with the same filename.
	To get around this, you can specify the file to be in “amxdir:///AMXPanel/images/filename” instead.
	To do this using NetLinx Studio File Transfer, set the “Controller Directory” to \AMXPanel\images\ in the device mapping. This will put the file in the panel file images directory. A TP5 file transfer will not remove the file, but a “Remove User Pages” will. The Streaming Source value in the TP5 file will have to correspond to the same path.
	Refer to the Streaming a Video File Saved on the Panel via Custom URL Scheme section for an example workflow for playing a video file in the G5 panel’s internal storage.
	• **Examples**:
	`SEND_COMMAND Panel,”’^SDM-400,1,file:///udisk/Video-Clip.mp4’”`
	Set the OFF state to play the video file Video-Clip.mp4 located on an attached USB disk.
	`SEND_COMMAND 10001:2:0,”’^SDM-10,2,udp://234.4.0.4:5500’”`
	Sets ON state to play video on multicast address.
	`SEND_COMMAND 10001:2:0,”’^SDM-10,1,stop’”`
	Stop playing the current video.
	`SEND_COMMAND 10001:2:0,”’^SDM-10,1,’”`
	Stop playing the current video.
	`SEND_COMMAND 10001:1:0,”’^SDM-10,1,udp://169.254.11.12:5700’”`
	Start playing the current video.
	Note: When using the variable “udp,” this must be in lower case.

^SDR	Enabling subpage dynamic reordering command - This command can be used to enable or disable dynamic reordering for a given viewer button or set of viewer buttons. It can also be used to set the amount of time to wait before initiating the single finger reorder time. • **Syntax**: `“’^SDR-<addr range>,<enable state>,<optional hold time>’”` **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *enable state*: This value can be either “on” or “ON” or “1” to enable dynamic reordering for the specified viewer button(s). Any other value will disable dynamic reordering for the specified viewer button(s). *hold time*: This value is in tenths of a second. The value will be rounded up to the next highest quarter of a second. This is the amount of time that the user must press and hold a subpage with a single finger to trigger a dynamic reordering operation.

^SHA	Subpage Hide All Command - Hide all subpages in a subpage viewer button.
	• **Syntax**:
	`“’^SHA-<addr range>’”`
	• **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	• **Example:**
	`SEND_COMMAND Panel,”’^SHA-200’”`
	Remove all subpages from subpage viewer button with address 200.

^SHD	Subpage Hide Command - This command will hide the named subpage and relocate the surrounding subpages as necessary to close the gap. If the subpage to be hidden is currently offscreen then it is removed without any other motion on the subpage viewer button.
	• **Syntax**:
	`“’^SHD-<addr range>,<name>,<optional time>’”`
	• **Variables**:
	address range: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *name*: name of subpage to hide. If name is __all, then all subpages are hidden.
	time: Can range from 0 to 30 and represents tenths of a second. This is the amount of time used to move the subpages around when subpages are hidden from a button.
	• **Example:**
	`SEND_COMMAND Panel,”’^SHD-200,menu1,10’”`
	Remove the menu1 subpage from subpage viewer button with address 200 over one second.

^SHO	Button Show/Hide Command. Show or hide a button.
	• **Syntax**:
	`“’^SHO-<addr range>,<command value>’”`
	• **Variables**:
	address range: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	command value: 0 = hide, 1 = show
	• **Example:**
	`SEND_COMMAND Panel,”’^SHO-500.504&510.515,0’”`
	Hides buttons with variable text address range 500-504 & 510-515.

Button Commands	
^SPD	Subpage Padding Command - Set the padding between subpages on a subpage viewer button. • **Syntax**: “’^SPD-<addr range>,<padding>’” • **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *padding*: percentage from 0 to 100 of the first subpage in a set to set as a padding between subpages. For a horizontal subpage viewer button it is a percentage of the width and for a vertical subpage viewer button it is a percentage of the height. • **Example:** SEND_COMMAND Panel, “’^SPD-400,10’” Set the padding between subpages in the set to 10% of the dimension of the first subpage in the set.
^SSH	Subpage Show Command - This command will perform one of three different operations based on the following conditions: a) If the named subpage is hidden in the set associated with the viewer button it will be shown in the anchor position. b) If the named subpage is not present in the set it will be added to the set and shown in the anchor position. c) If the named subpage is already present in the set and is not hidden, then the viewer button will move it to the anchor position. The anchor position is the location on the subpage viewer button specified by its weighting. This will either be left, center or right for horizontal subpage viewer buttons or top, center or bottom for vertical subpage viewer buttons. Surrounding subpages are relocated on the viewer button as needed to accommodate the described operations. • **Syntax**: “’^SSH-<addr range>,<name>,<optional position>,<optional time>’” • **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *name*: Specifies the name of the subpage to be shown or added. *position*: Specifies where to add (or show) the named subpage in the set with 0 representing the beginning of the set. If this value is left out (or set to 65535) then the weighting value for the viewer button is used to place the new subpage, i.e. left/top, center or right/bottom. When using the weighting locations, set insertion positions can vary based on the current onscreen locations of existing subpages. *time*: Can range from 0 to 30 and represents tenths of a second. This is the amount of time used to move the subpages around when subpages are added or removed from a button. • **Example:** SEND_COMMAND Panel,”’^SSH-400,media1,0,10’” Add or show the media1 subpage in the anchor position over one second.
^STG	Subpage Toggle Command - If the named subpage is hidden, then this command activates a subpage show command. If the named subpage is present, then a subpage hide command is activated. • **Syntax**: “’^STG-<addr range>,<name>,[optional position],[optional time]’” • **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *name*: Specifies the name of the subpage to be shown or added. *position*: Specifies where to show the named subpage in the set with 0 representing the beginning of the set. If this value is left out (or set to 65535) then the weighting value for the viewer button is used to place the new subpage, i.e. left/ top, center or right/bottom. When using the weighting locations, set insertion positions can vary based on the current onscreen locations of existing subpages. If the subpage is being hidden this parameter is ignored. *time*: Can range from 0 to 30 and represents tenths of a second. This is the amount of time used to move the subpages around when subpages are added or removed from a button. • **Example:** SEND_COMMAND Panel,”’^STG-400,media1,0,10’” Show or hide the media1 subpage over one second.
^TEC	Set text effect color command - Set the text effect color for the specified addresses/states to the specified color. The Text Effect is specified by name and can be found in TPD5. You can also assign the color by name or RGB value (RRGGBB or RRGGBBAA). • **Syntax**: “’^TEC-<addr range>,<button states range>,<color value>’” • **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *color value*: See color table for more information. *Note: Colors can be set by Color Numbers, Color name, RGB alpha colors (RRGGBBAA) or RGB colors values (RRGGBB).* *RGBA and RGB color are given in HEX ASCII prepended by a ‘#’.* • **Example:** SEND_COMMAND Panel,”’^TEC-500.504&510.515,1&2,12’” Sets the text effect color to Very Light Yellow on buttons with variable text 500-504 and 510-515.
?TEC	Get text effect color command - Get the current text effect color. • **Syntax**: “’?TEC-<addr range>,<button states range>’” • **Variables**:

Button Commands

address range: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). The format of the custom event transmitted to the controller is as follows:

```
  Custom Event Property   Value
    Port                    port command was received on
    ID                      address of the button generating the event
    Type                    1009
    Flag                    0
    Value 1                 button state number
    Value 2                 actual length of string
    Value 3                 0
    Text                    Hex encoded color value (ex: #000000FF)
```

- **Example:**

```
  SEND COMMAND Panel,”’?TEC-529,1’”
```

Gets the button ‘OFF state’ text effect color information. The result sent to the Controller would be:

```
  Custom Event Property   Value
    Port                    port command was received on
    ID                      address of the button generating the event
    Type                    1009
    Flag                    0
Value 1                 1
    Value 2                 9
    Value 3                 0
    Text                    #5088F2AE
```

^TEF

Set the current text effect command - Set the current text effect.

- **Syntax**:

```
  “’^TEF-<addr range>,<button states range>,<text effect name/number>’”
```

- **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *text effect name/number*: See the Text Effect Name/Numbers table for text effect names and numbers.
- **Example:**

```
  SEND_COMMAND Panel,”’^TEF-500.504&510.515,1&2,Soft Drop Shadow 3’”
```

Sets the text effect to Soft Drop Shadow 3 for the button with variable text range 500-504 and 510-515.

?TEF

Get the current text effect command - Get the current text effect.

- **Syntax**:

```
  “’?TEF-<addr range>,<button states range>’”
```

- **Variables**:

address range: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.

button states range: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).

The format of the custom event transmitted to the controller is as follows:

```
  Custom Event Property   Value
    Port                    port command was received on
    ID                      address of the button generating the event
    Type                    1008
    Flag                    0
    Value 1                 button state number
    Value 2                 actual length of string
    Value 3                 text effect number
Text                    text effect name
```

- **Example:**

```
  SEND COMMAND Panel,”’?TEF-529,1’”
```

Gets the button ‘OFF state’ text effect name information. The result sent to the Controller would be:

```
  Custom Event Property   Value
    Port                    port command was received on
    ID                      529
    Type                    1008
    Flag                    0
Value 1                 1
    Value 2                 18
    Value 3                 27
    Text                    Hard Drop Shadow 3
```

^TXT

Set button state text command - Assign a Non-Unicode, non-UTF-8 text string to those buttons with a defined address range. Note that this command has been replaced by ^UTF, but is being kept for backwards compatibility. It supports ASCII characters, but extended ASCII (i.e. characters from 128-255) are interpreted according to the Latin-1 character set (ISO 8859-1). Unicode (i.e. characters > 255) are not supported

- **Syntax**:

Button Commands

```
"'^TXT-<addr range>,<button states range>,<new text>'"
```

- **Variables**:
  address range: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *new text*: new text as ASCII characters.
- **Example:**

```
SEND_COMMAND Panel,"'^TXT-500.504&510.515,1&2,Test Only'"
```

Sets the On and Off state text for buttons with the variable text ranges of 500-504 & 510-515.

?TXT

Query button state text command - Get the text of a button state.

- **Syntax**:

```
"'?TXT-<addr range>,<button states range>[,<optional index>]'"
```

- **Variables**: *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state). *optional index*: This is used if a string was too long to get back in one command. The reply will start at this index. The response returned is a custom event with the following syntax:

```
  Custom Event Property   Value
    Port                    port command was received on
    ID                      address of the button generating the event
    Type                    1001
    Flag                    0: Legacy Latin-1 (ISO-8859-1) encoded characters
                                 (^ENC must have previously been sent to change default
encoding method)
                              1: Legacy AMX Hex Quad encoded Unicode characters
                              2: UTF-8 encoded Characters (default encoding; ASCII-
compatible)
    Value 1                 button state number
    Value 2                 actual length of string
    Value 3                 optional index
    Text                    text from the button, encoded with the method specified by
Flag
```

- **Example:**

```
SEND_COMMAND Panel,"'?TXT-529,1'"
```

Gets the button 'OFF state' text information. Example Response:

```
  Custom Event Property   Value
    Port                    port command was received on
    ID                      529
    Type                    1001
    Flag                    2
Value 1                 1
    Value 2                 14
Value 3                 0
    Text                    This is a test
```

^UNI

Set button state legacy unicode text command - Set Unicode text in the legacy G4 format. For the ^UNI command, the Unicode text is sent as ASCII-HEX nibbles.

Note: In the legacy format, Unicode text is always represented in a HEX value. TPD generates (through the Text Enter Box dialog) Unicode HEX values. Refer to the TPDesign Instruction Manual for more information. This command has been replaced by ^UTF, but is being kept for backwards compatibility.

- **Syntax**:

```
"'^UNI-<addr range>,<button states range>,<unicode text>'"
```

- **Variables**:
  address range: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
  unicode text: Unicode HEX value.
- **Example:**

```
SEND_COMMAND Panel,"'^UNI-500,1,0041'"
```

Sets the button's unicode character to 'A'.

```
SEND_COMMAND TP,"'^UNI-1,0,0041'"
```

Send the variable text 'A' in unicode to all states of the variable text button 1, (for which the character code is 0041 Hex).

^UTF

Set button state text using UTF-8 text command - Set State Text Command using UTF-8 (replaces the ^TXT and ^UNI commands). Assign a text string encoded with UTF-8 (which is ASCII-compatible) to those buttons with a defined address range.

Note: This command replaces the legacy ^TXT command and the legacy ^UNI command, but text must be encoded with UTF-8. While UTF-8 is ASCII compatible, extended ASCII characters in the range 128-255 will be encoded differently based on UTF-8. his command also supports Unicode characters using UTF-8 (which is the encoding method used in >80% of web servers), making the old AMX Hex quad Unicode encoding obsolete (though the ^UNI command is still supported for backwards compatibility).

- **Syntax**:

```
"'^UTF-<vt addr range>,<button states range>,<new text>'"
```

- **Variables**:

Button Commands	
	variable text address range: 1 - 4000. *Button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons 1 = Off state and 2 = On state). *unicode text*: Unicode UTF-8 text. • **Example:** `SEND_COMMAND Panel,”’^UTF-500.504&510.515,1&2, ASCII ExtendedASCIIÇüéâäàåç Unicode 動き始めました’”` Sets the On and Off state text for buttons with the variable text ranges of 500-504 & 510-515.

Text Effect Name/Numbers

Text Effect Name/Numbers			
Number	**Name**	**Number**	**Name**
0	None	30	Hard Drop Shadow 6
1	Outline -S	31	Hard Drop Shadow 7
2	Outline -M	32	Hard Drop Shadow 8
3	Outline -L	33	Soft Drop Shadow 1 with Outline
4	Outline -X	34	Soft Drop Shadow 2 with Outline
5	Glow -S	35	Soft Drop Shadow 3 with Outline
6	Glow -M	36	Soft Drop Shadow 4 with Outline
7	Glow -L	37	Soft Drop Shadow 5 with Outline
8	Glow -X	38	Soft Drop Shadow 6 with Outline
9	Soft Drop Shadow 1	39	Soft Drop Shadow 7 with Outline
10	Soft Drop Shadow 2	40	Soft Drop Shadow 8 with Outline
11	Soft Drop Shadow 3	41	Medium Drop Shadow 1 with Outline
12	Soft Drop Shadow 4	42	Medium Drop Shadow 2 with Outline
13	Soft Drop Shadow 5	43	Medium Drop Shadow 3 with Outline
14	Soft Drop Shadow 6	44	Medium Drop Shadow 4 with Outline
15	Soft Drop Shadow 7	45	Medium Drop Shadow 5 with Outline
16	Soft Drop Shadow 8	46	Medium Drop Shadow 6 with Outline
17	Med Drop Shadow 1	47	Medium Drop Shadow 7 with Outline
18	Med Drop Shadow 2	48	Medium Drop Shadow 8 with Outline
19	Med Drop Shadow 3	49	Hard Drop Shadow 1 with Outline
20	Med Drop Shadow 4	50	Hard Drop Shadow 2 with Outline
21	Med Drop Shadow 5	51	Hard Drop Shadow 3 with Outline
22	Med Drop Shadow 6	52	Hard Drop Shadow 4 with Outline
23	Med Drop Shadow 7	53	Hard Drop Shadow 5 with Outline
24	Med Drop Shadow 8	54	Hard Drop Shadow 6 with Outline
25	Hard Drop Shadow 1	55	Hard Drop Shadow 7 with Outline
26	Hard Drop Shadow 2	56	Hard Drop Shadow 8 with Outline
27	Hard Drop Shadow 3		
28	Hard Drop Shadow 4		
29	Hard Drop Shadow 5		

Dynamic Image Commands

Dynamic Image Commands	

^BBR	Button State Bitmap Resource Command - Assign a resource to those buttons with a defined address range.
	• **Syntax**:
	`"'^BBR-<vt addr range>,<button states range>,<resource name>,[optional bitmap index], [optional justification]'"`
	• **Variables**:
	address range: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. *button states range*: 1 - 256 for multi-state buttons (0 = All states, for General buttons, 1 = Off state and 2 = On state).
	resource name: name of resource
	Optional bitmap index: 1 - 5, the state bitmap index to assign the resource. If not present, will place the referenced resource in index 1. The indexes are defined as:
	0 - Chameleon Image (if present)
	1 - Bitmap 1
	2 - Bitmap 2
	3 - Bitmap 3
	4 - Bitmap 4 5 - Bitmap 5
	Optional justification: 0-11 (see Justification Values). If absolute justification (0) is set, the next two parameters are the X and Y offset of the bitmap for the referenced index. If no justification is specified, the current justification is used.
	• **Example:**
	`SEND_COMMAND Panel,"'^BBR-500.504&510.515,1,image_xray'"`
	Sets the OFF state picture for the buttons with variable text ranges of 500-504 & 510-515 to the resource named *image_xray*.

^RAF	Resource Add Command - Add new resources. Adds any and all resource parameters by sending embedded codes and data. Since the embedded codes are preceded by a '%' character, any '%' character contained in the URL must be escaped with a second '%' character (see example). The file name field (indicated by a %F embedded code) may contain special escape sequences as shown in the ^RAF, ^RMF - Embedded Codes table. Note: For server authentication to occur, the %U (username) and %S (password) Embedded Codes must be included, and they must match the credentials required by the server.
	• **Syntax**:
	`"'^RAF-<resource name>,<data>'"`
	• **Variables**:
	resource name: name of the resource to add. *data*: Refers to the embedded codes, see the ^RAF, ^RMF - Embedded Codes.
	Note: The %P, %U, %S, %H, %A, and %F values can be entered In a single string .
	• **Example:**
	`SEND_COMMAND Panel,"'^RAF-New Image, %P0%HAMX.COM%ALab/Test%%5Ffile%Ftest.jpg'"`
	Adds a new resource.
	The resource name is 'New Image'
	%P (protocol) is 0 for an HTTP connection
	%H (host name) is AMX.COM
	%A (file path) is Lab/Test_file %F (file name) is test.jpg.
	Note: the %%5F in the file path is actually encoded as %5F.

^RFR	Refresh Resource Command - Force a refresh of the given resource. The command will refresh when the resource is visible onscreen. If it is not onscreen, it will be deferred until it is visible to do the refresh. An optional notification option can be set to receive a custom event from the panel when the resource refresh is complete. Optional width and height parameters can be specified to refresh the image at a specific resolution. If width and height parameters are not specified, the resource will be refreshed at the resolution(s) of any active buttons to which it is assigned. If there are no active buttons currently assigned that resource, it will be refreshed at its native resolution adjusted by any project scale factor.
	• **Syntax**:
	`"'^RFR-<resource name>,[notification option],[width],[height]'"`
	• **Variables**:
	Resource name: name of the resource to refresh
	Notification option: An optional notification option at the end of the command with the following possible values: On - notifications are sent whenever the named dynamic image resource is loaded/refreshed.
	Off - notifications are not sent (default).
	Once - notifications are sent one time whenever the named dynamic image resource is loaded/refreshed. Notifications are not sent on subsequent loads/refreshes.
	width: Specifies the width at which the resource should be refreshed (the image will be scaled as needed).
	height: Specifies the height at which the resource should be refreshed (the image will be scaled as needed).
	• **Example:**
	`SEND_COMMAND Panel,"'^RFR-Sports_Image,on'"`

	Force a refresh on ‘Sport_Image’ when the resource is visible onscreen and enable completion notifications. SEND_COMMAND Panel,”’^RFR-Sports_Image,off’” Force a refresh on ‘Sport_Image’ when the resource is visible onscreen and disable completion notifications. SEND_COMMAND Panel,”’^RFR-Sports_Image,once’” Force a refresh on ‘Sport_Image’ when the resource is visible onscreen and enable a onetime completion notification. SEND_COMMAND Panel,”’^RFR-Sports_Image,once,800,600’” Force a refresh on ‘Sport_Image’ at the resolution 800x600 when the resource is visible onscreen and enable a onetime completion notification.
^RFRP	Resource Refresh Prefetch Command - Force a refresh of the given resource. The command will “prefetch” the resource even if it is not currently visible. • **Syntax**: “’^RFRP-<resource name>,[notification option],[width],[height]’” • **Variables**: *Resource name*: name of the resource to refresh *Notification option*: An optional notification option at the end of the command with the following possible values: On - notifications are sent whenever the named dynamic image resource is loaded/refreshed. Off - notifications are not sent (default). Once - notifications are sent one time whenever the named dynamic image resource is loaded/refreshed. Notifications are not sent on subsequent loads/refreshes. *width*: Specifies the width at which the resource should be refreshed (the image will be scaled as needed). *height*: Specifies the height at which the resource should be refreshed (the image will be scaled as needed). • **Example:** SEND_COMMAND Panel,”’^RFRP-Sports_Image,on’” Force a refresh on ‘Sport_Image’ immediately and enable completion notifications. SEND_COMMAND Panel,”’^RFRP-Sports_Image,off’” Force a refresh on ‘Sport_Image’ immediately and disable completion notifications. SEND_COMMAND Panel,”’^RFRP-Sports_Image,once’” Force a refresh on ‘Sport_Image’ immediately and enable a one-time completion notification. SEND_COMMAND Panel,”’^RFRP-Sports_Image,once,800,600’” Force a refresh on ‘Sport_Image’ immediately at the resolution 800x600 and enable a onetime completion notification.
^RMF	Resource Modify Command - Modifies any and all resource parameters by sending embedded codes and data. Since the embedded codes are preceded by a ‘%’ character, any ‘%’ character contained in the URL must be escaped with a second ‘%’ character (see example). The file name field (indicated by a %F embedded code) may contain special escape sequences as shown in the ^RAF, ^RMF - Embedded Codes table. *Note: For server authentication to occur, the %U (username) and %S (password) Embedded Codes must be included, and they must match the credentials required by the server.* • **Syntax**: “’^RMF-<resource name>,<data>’” • **Variables**: *resource name*: name of the resource to modify *data*: Refers to the embedded codes, see the ^RAF, ^RMF - Embedded Codes. *Note: The %P, %U, %S, %H, %A, and %F values can be entered In a single string.* • **Example:** SEND_COMMAND Panel,”’^RMF-Sports_Image, %ALab%%5FTest/Images%Ftest.jpg’” Changes the resource ‘Sports_Image’ file name to ‘test.jpg’ and the path to ‘Lab_Test/Images’. *Note: the %%5F in the file path is actually encoded as %5F.*
^RSR	Resource Rate Command - Change the refresh rate for a given resource. • **Syntax**: “’^RSR-<resource name>,<refresh rate>’” • **Variables**: *Resource name*: name of the resource to set the refresh rate *refresh rate*: Measured in seconds. • **Example:** SEND_COMMAND Panel,”’^RSR-Sports_Image,5’” Sets the refresh rate to 5 seconds for the given resource (‘Sports_Image’).
^RAF, ^RMF - Embedded Codes	The ^RAF and ^RMF commands add and modify any and all resource parameters by sending embedded codes and data: “’^RAF-<resource name>,<data>’” “’^RMF-<resource name>,<data>’” The <data> variable uses the embedded codes described in the ^RAF and ^RMF Embedded Codes table.
^RAF, ^RMF - Escape Sequences	The ^RAF and ^RMF commands support the replacement of any special escape sequences in the filename (specified by the %F embedded code) with the corresponding data obtained from the system as outlined in the^RAF and ^RMF Escape Sequences table.

^RAF and ^RMF Embedded Codes / Escape Sequences

NOTE: *The %P, %U, %S, %H, %A, and %F values can be entered In a single string.*

^RAF and ^RMF Embedded Codes

Parameter	Embedded Code	Code Description
protocol	*%P<0|1|2>*	Set protocol: Either HTTP (0) or FTP (1), or HTTPS(2) Notes: • *FTP is not supported at this time.* • *HTTPS (%P2) is supported in G5 panel firmware v1.4.9 and higher.*
user	%U <user>	Set Username for authentication.
password	%S <password>	Set Password for authentication.
host	%H <host>	Set Host Name (fully qualified DNS or IP address).
path	%A <path>	Set directory path. The path must be a valid HTTP URL minus the protocol, host, and filename. The only exception to this is the inclusion of special escape sequences and in the case of the FTP protocol, regular expressions.
file	%F <file>	The file or program that will return the resource. The file must be a valid HTTP URL minus the protocol, host, and path. The only exception to this is the inclusion of special escape sequences and in the case of the FTP protocol, regular expressions.
refresh	%R <refresh 1-65535>	The number of seconds between refreshes in which the resource is downloaded again. Refreshing a resource causes the button displaying that resource to refresh also. The default value is 0, which means to only download the resource once for each time it comes into view (or if preserve is set, only once period). *Note: For Motion JPEGs, the Refresh interval should always be 0.*
preserve	%V <0-1>	Set the value of the preserve flag. A value of 0 (the default) means the resource should be reloaded each time it comes into view. A value of 1 means the resource should be preserved in cache after the first time it is loaded, and not reloaded each time it comes into view. This value is ignored if the Refresh interval is greater than 0.
dynamo	%D	Enable/disable Fast Dynamo. Panel will attempt to accelerate this resource in hardware. *Note: Fast Dynamo is not yet supported.*
notification	%C<on,off,once>	Indicates whether a notification is required when a Dynamic Image is loaded/ refreshed. The string following the %C can be: 1. on - notifications are sent whenever the named dynamic image resource is loaded/refreshed. 2. off - notifications are not sent (default). 3. once - notifications are sent one time whenever the named dynamic image resource is loaded/refreshed. Notifications are not sent on subsequent loads/refreshes. If the %C code is not sent as part of a ^RAF command, the notifications are set to off. If the %C code is not sent as part of a ^RMF command, the notifications are not changed from the current setting.

URL	%L <URL>	Set the complete URL as a single value. URL is in the format set in RFC 2396.
		Code Block http://username:password@host:port/directory/file?query#fragment
		Note: The %P, %U, %S, %H, %A, and %F values can be entered In a single string. Note: If the URL is the first part of the resource data, then the %L is assumed and need not be included. See example below.
		Example:
		The following send commands are equivalent. All examples set the resource Image1 to a URL of http://server/folder1/image.jpg with a username of username, password of password, notifications on, and refresh time of 30 seconds:
		`SEND_COMMAND Panel,’^RMF-Image1,%Lhttp://username:password@server/folder1/image.jpg%Con%R30’`
		`SEND_COMMAND Panel,’^RMF-Image1,%P0%Uusername%Spassword%Hserver%Afolder1%Fimage.jpg%Con%R30’`
		`SEND_COMMAND Panel,’^RMF-Image1,http://username:password@server/folder1/image.jpg%Con%R30’`
		`SEND_COMMAND Panel,’^RMF-Image1,http://server/folder1/image.jpg%Con%R30%Uusername%Spassword’`

^RAF and ^RMF Escape Sequences			
Sequence	**Panel Information**	**Sequence**	**Panel Information**
$DV	Device Number	$AP	Address port
$SY	System Number	$CC	Channel code
$IP	IP Address	$CP	Channel port
$HN	Host Name	$LC	Level code
$MC	Mac Address	$LP	Level port
$PX	X resolution of current panel mode/file	$BX	X Resolution of Current button
$PY	Y resolution of current panel mode/file	$BY	Y Resolution of Current button
$ST	Current state	$BN	Name of Button
$AC	Address code		

Listview (Data Access) Commands

The Data Access commands described in the following table represent a set of Button (^) Send Commands that support the use of dynamic data for Listview buttons in NetLinx code. Note that the address range indicated in the syntax examples represents the address of the Listview button, and works the same as it does for all other (^) Button Send Commands.

Many Listview Send Commands take a boolean parameter. Any of the following values can be used:

Will resolve to true	Will resolve to false
true	false
TRUE	FALSE
on	off
ON	OFF
1	0
	(empty)

Terminology

The NetLinx Data Access Send Commands use the following terminology:

Netlinx Data Access Send Commands - Terminology	
Name	**Description**
DataFeed	A DataFeed is a descriptor with a unique name used to publish data records. A DataFeed can be created by a NetLinx program and then published to the NetLinx web server for external consumption by devices like the G5 touch panel for use with Listview buttons. DataFeeds can also be sourced from a server running the AMX XPort software.
DataRecord	A DataRecord represents a container of data fields and the index/ordinal position of the row in the recordset. A DataRecord may contain metadata and/or content fields.
DataField	SA DataField represents the value that stores the actual data elements. All raw data in the NetLinx data access APIs are stored and managed as values and (one or more) attributes.

Listview Commands	

^LVC	Listview Cache Configure - This command configures the image cache used by the Listview.
	• **Syntax**:
	`"'^LVC-<configuration_option=configuration_value>'"`
	• **Variables:** a comma separated list of one or more configuration parameters followed by an equal sign and the configuration setting.
	• **Configuration Options:**
	clear: Clear the current memory and disk cache used for Listview image loading. *mem_size*: The size of the memory cache, either as a percentage of the available application memory or as total size. Percentages are specified as floating point. Percentage values are 2% (0.02) to 20% (0.20) and totals are 16 to 256 MB. The default is 10%.(0.10) *disk_size*: The size of the disk cache. Valid values are 16 to 500 MB The default is 200.
	• **Example**:
	`SEND_COMMAND Panel,"'^LVC-clear'"`
	Clear the Listview cache.

^LVD	Set Listview Data Source - This command sets the data source to drive the Listview entries. Note that this command only configures the data source it does not actually cause the data to be fetched. The ^LVR refresh command must be issued to load the data. • **Syntax**: `"'^LVD-<addr range>,<URL to data source or Dynamic Data Resource name>, <configuration_option=configuration_value>'"` • **Variable:** *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. Data source URL/Dynamic Data Resource name (required): If the suffix of the URL is .csv or .CSV then the URL will be assumed to point to a csv file. Otherwise the type is assumed to be the XPort amxstandard.xml format. Supported URL schemes are HTTP, HTTPS, and FILE. Data Source URL Notes:

Listview Commands

HTTPS is supported in G5 panel firmware version v1.4.9 and higher.

HTTPS is not supported by TPDesign5 dynamic image resources at this time.

A file on the panel's local filesystem can be specified using the file:/// option. *There must be three forward slashes after 'file:'.*

An FTP URL scheme is not supported.

Refer to *Notes on Using Image URLs With Listview Buttons* for additional details. *option list*: a optional comma separated list of one or more configuration parameters followed by an equal sign and the configuration setting.

- **Configuration Options:**

user - The user name to use for authenticating to the web server when retrieving the feed data source file. If specified when URL is a Dynamic Data Resource, this value will override the username inside the Dynamic Data Resource.

Note: For server authentication to occur, the Username (user) and Password (pass) must be included in the ^LVD command, and they must match the credentials required by the server. pass - The password to use for authenticating to the web server when retrieving the feed data source file. If specified when URL is a Dynamic Data Resource, this value will override the password inside the Dynamic Data Resource.

Note: For server authentication to occur, the Username (user) and Password (pass) must be included in the ^LVD command, and they must match the credentials required by the server. csv - a boolean indicating whether or not to parse the data source as a CSV file.

If not present, defaults to false.

has_headers - a boolean indicating that the first line of the CSV file has column headers which will be used to name the content fields for each data record.

If true it automatically implies that csv is also true.

If this option is not present then the default for a CSV file is false.

In the absence of headers, the content fields will be named using the following convention: column1, column2, column3... (CSV files only, since XML always has field names specified within the file).

- **Example**:

```
SEND_COMMAND Panel,"'^LVD-42,http://192.168.220.231/public/lv42data.csv,
has_headers=1'"
```

Configures the Listview button to use the CSV file at the URL as its data source. The first line of the CSV file should be parsed as field names and not as Listview entry record data.

^LVE

Set ListView custom event number - This command sets the custom event number reported by Listview refresh operations.

- **Syntax**:

```
"'^LVE-<addr range>,<Listview custom event number>'"
```

- **Variable:** *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address. event number: The custom event number to report Listview events. At this time, only refresh events are reported. A value of 0 turns off custom event reporting, A value > 0 assigns the value to the Listview custom event number for that address. The default value is 1401 (custom events reported).

When enabled, the custom event format reported is:

```
  Custom Event Property   Value
    Port                     port command was received on
    ID                       address of the button
    Type                     button event number set by ^LVE
    Flag                     StartRefresh = 1; FinishRefresh = 2; Error = 0xffff
($FFFF);     Value 1                   If flag is StartRefresh (1) or FinishRefresh (2):
                               InitRefresh = 0; (refresh by dynamic resource)
                               ManualRefresh = 1; (refresh by send command
                               TimedRefresh = 2; (refresh by timer)
If flag is Error:
                               Error = -1; (some form of error, see custom.text for
description)                                InvalidUrl = -2; (URL is null, should
never happen)
                               LoginFailed = -3; (could not authenticate to web
server).     Value 2                 data load id. Every data load is assigned a
unique id that                                      counts up from 0. This is
used to correlate StartRefresh/                          FinishRefresh/Error
events on particular addresses.
    Value 3                  When Custom.flag == FinishRefresh, this is the number of
records in list. Otherwise is 0.
    Text                     feed URL string, or error message if flag is Error
```

- **Example**:

```
SEND_COMMAND Panel,"'^LVE-42,1401'"
```

Configures the Listview widget to send Listview custom events on event 1401.

^LVF

Listview Filter - This command can be used to programmatically change the filter contents of the Listview widget. When the filter contents is changed, the filter will be applied to the current Listview data which can change the number of items displayed based on those that meet the filter sequence. The filter changes immediately, and the filter can be set or cleared with this command.

- **Syntax**:

```
"'^LVF-<addr range>,<filter character sequence>'"
```

- **Variable:** *address range*: Address codes of buttons to affect. A '.' between addresses includes the range, and & between addresses includes each address.

  filter character sequence: All characters including white space characters will be applied to the filter.

- **Example**:

<table>
<tr><th colspan="3">Listview Commands</th></tr>
<tr><td></td><td colspan="2">SEND_COMMAND Panel,”’^LVF-42,amx’”
Sets the filter sequence to amx. Only items in the data set that contain the sequence amx will be displayed.
SEND_COMMAND Panel,”’^LVF-42,’”
Clears the filter sequence. All items in the data set can be viewed in the Listview.</td></tr>
<tr><td rowspan="25">^LVL</td><td colspan="2">Listview Layout - This command sets the layout configuration to configure the visual representation of the Listview entries.
• Syntax:
“’^LVL-<vt addr range>,<layout_option=layout_value>’”
• Variables:
Variable text address range = 1 - 4000.
A comma separated list of one or more layout configuration parameters followed by an equal sign and the configuration setting.
• Layout Options: columns - Number of columns parameter. An integer that represents the number of columns to display. The number must be at least 1 and a value that exceeds the minimum cell width will truncate to the maximum.
Note: Valid tags for the columns parameter are columns=, nc=, and numcol=.
comp - Component parameter. An integer that is a value which determines which graphical components are present in the cell. When the component values are bitwise or’d together, it creates the encoding for the cell components that are populated. If a configuration parameter is not in the current command, the last value for the configuration parameter is used.
Note: Valid tags for the comp parameter are c= and comp=.</td></tr>
<tr><td>Component Value</td><td>Description</td></tr>
<tr><td>1</td><td>The image (i) is used in the cell.</td></tr>
<tr><td>2</td><td>The primary text field (t1) is used in the cell.</td></tr>
<tr><td>4</td><td>The secondary text field (t2) is used in the cell</td></tr>
<tr><td colspan="2">Not all variations of component values are valid. To have the secondary text field present, the primary text field must be preset as well.</td></tr>
<tr><td>Component Combinations</td><td>Description</td></tr>
<tr><td>0</td><td>Invalid. No component displayed.</td></tr>
<tr><td>1</td><td>The image (i) is the only component displayed.</td></tr>
<tr><td>2</td><td>The primary text field (t1) is the only component displayed.</td></tr>
<tr><td>3</td><td>The image (i) and the primary text field (t1) are displayed.</td></tr>
<tr><td>4</td><td>Secondary text (t2) only. Invalid. Secondary text (t2) cannot be displayed without the primary text (t1).</td></tr>
<tr><td>5</td><td>Secondary text (t2) and image (i). Invalid. Secondary text (t2) cannot be displayed without the primary text (t1).</td></tr>
<tr><td>6</td><td>The primary text (t1) and secondary text (t2) are displayed.</td></tr>
<tr><td>7</td><td>The image (i), primary text (t1), and secondary text (t2) are displayed</td></tr>
<tr><td colspan="2">cellheight - An integer or percentage that sets the height of a cell. The value can be an integer >= the minimum cell height (48), or a percentage of the list height (5% up to 95%). To specify a percentage, append a ‘%’ to the end of the value.
Note: Valid tags for the cellheight param are ch= and cellheight=.</td></tr>
<tr><td colspan="2">layout - An integer that sets the layout configuration of each cell. Note: valid tags for the layout parameter are l= and layout=.</td></tr>
<tr><td>Layout Value</td><td>Description</td></tr>
<tr><td>1</td><td>Horizontal layout with image on the left and text(s) on the right.
If multiple texts are selected then the texts are stacked vertically.</td></tr>
<tr><td>2</td><td>Horizontal layout with image on the right and text(s) on the left.
If multiple texts are selected then the texts are stacked vertically.</td></tr>
<tr><td>3</td><td>Horizontal layout with text1 on the left, image in the center, and text2 on the right. If multiple texts are selected then the texts are stacked vertically.</td></tr>
<tr><td>4</td><td>Vertical layout with the image on the top and text(s) below the image.
If multiple texts are selected then text1 is below the image and text2 is below text1.</td></tr>
<tr><td>5</td><td>Vertical layout with the image on the bottom and text(s) above the image.
If multiple texts are selected then text1 is on top, text2 is below text1, and the image is below text2.</td></tr>
<tr><td>6</td><td>Vertical layout with text1 on top, the image below text1, and text2 below the image.</td></tr>
<tr><td colspan="2">p1 - layout percentage 1. Sets the boundaries between cell components in different layouts. An integer between 10 and 90 that sets the boundary between components as a percentage of the cell dimension. The percentage can be specified as a number between 5-95 with an optional percentage sign ‘%’ at the end. p2 - layout percentage 2. Sets the boundaries between cell components in different layouts.An integer between 10 and 90 that sets the boundary between components as a percentage of the cell dimension. The percentage can be specified as a number between 5-95 with an optional percentage sign ‘%’ at the end. filter - Enable or disable the search filter on the Listview. To enable set to ‘true’, ‘on’, or ‘1’. To disable set to ‘false’, ‘off’, or ‘0’.</td></tr>
</table>

<table>
<tr><th colspan="2">Listview Commands</th></tr>
<tr><td></td><td><i>Note: Valid tags for the filter parameter are f= and filter=.</i> <b>filterheight</b> - An integer or percentage that sets the height of the filter in the Listview. The value can be an integer >= the minimum filter height (24), or a percentage of the list height (5% to 25%). To specify a percentage, append a ‘%’ to the end of the value.
<i>Note: Valid tags for the filterheight param is fh= and filterheight=.</i> <b>alphascroll</b> - Enable or disable the alpha scroll on the Listview. To enable set to ‘true’, ‘on’, or ‘1’. To disable set to ‘false’, ‘off’, or ‘0’.
<i>Note: Valid tags for the alphascroll parameter are as= and alphascroll=.</i>
• <b>Examples:</b>
<code>SEND_COMMAND Panel,”’^LVL-42, layout=1, comp=7, columns=1, cellheight=120, p1=40%, p2=66%’”</code>
Sets the Listview configuration display an image and 2 text fields (comp=7), in a layout 1 configuration (layout=1 horizontal layout of the image on left and text1 and text2 to the right of the image). There is 1 column (columns=1) and the cell is 120 pixels high (h=120). The image width will be 40% of the cell width (p1=40%) with text1 and text2 having a width of 60% of the cell width. The height of text1 will be 66% of the cell height (p2=66%) with text2 height of 34% of the cell height.
<code>SEND_COMMAND Panel,”’^LVL-42,l=4, c=3, ch=150, nc=4, p1=70’”</code>
Sets the Listview configuration display an image and 1 text fields (c=4), in a layout 4 configuration (l=4 vertical layout of the image on top and text1 below the image). There are 4 columns (nc=4) and the cell is 150 pixels high (ch=150). The image height will be 70% of the cell height (p1=70) with text1 having a height of 30% of the cell height.
<code>SEND_COMMAND Panel,”’^LVL-42,layout=3, comp=6, ch=100, numcol=1, p1=50’”</code>
Sets the Listview configuration display 2 text fields (comp=6), in a layout 3 configuration (layout=2 horizontal layout of text1 on the left and text2 on the right). There is 1 column (numcol=1) and the cell is 100 pixels high (ch=100). The text1 width will be 50% of the cell width (p1=50) with text2 having a width of 50% of the cell width.
<code>SEND_COMMAND Panel,”’^LVL-42,filter=1, fh=10%, as=false’”</code>
Sets the Listview search filter enabled (filter=1), the search filter textview height to 10% of the Listview height (fh=10%), and disables the alphascroller on the Listview.</td></tr>
<tr><td><b>^LVM</b></td><td>Listview Map Fields - This command maps the fields from the data source to the display elements of a Listview entry. Each list entry corresponds to a record if the data came from the NetLinx data access API or XPort. If the data source is a csv file, then each list entry corresponds to a row in the file. A list entry can have up to two lines of text and a URL that points to an image. Each display element for a list entry has to be mapped to a field in the record. If no mapping is specified, then a default mapping is used which is simply to map the fields in order based on the screen layout of the list entry. So, if the list type was an image and two lines of text, the first content field in the record would be interpreted as the URL to the image, the next field would be the first line of text and the next field would be the second line of text. To override this default behavior, the ^LVM command should be used to specify the correct mapping.
• <b>Syntax</b>:
<code>“’^LVM-<addr range>,<display_element=field_expression|<display_element= field_expression>|...’”</code>
• <b>Variable:</b> <i>address range</i>: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
<i>display element list</i>: A pipe character “|” separated list of mapping expressions. A pipe is used because typical field expressions may use more common characters such as the comma or semicolon. Display Elements: t1 - the first text element t2 - the second text element i1 - the first image future display types may support more text and image elements which will follow the same convention: t3... i2...
• <b>Field Expressions</b>:
An expression that can be used to map field values to display elements. Any time a field name is used, it follows the form <b>${field_name}</b>. Other text characters can be used to construct a more complex string using multiple fields.
• <b>Examples</b>:
<code>SEND_COMMAND Panel,”’^LVM-42,i1=${image}’”</code>
Configures the Listview widget to map an image field to the image display element. In this example, the Listview type is assumed to be a single image only.
<code>SEND_COMMAND Panel,”’^LVM-42,i1=${image}|t1=${lname}, ${fname}|t2=${number}’”</code>
The Listview widget is the type that has an image and two lines of text. The top line will consolidate two different fields in the form of last name, first name. The second line of text will be the phone number.
<code>SEND_COMMAND Panel,”’^LVM-42,t1=${column2}, ${column1}|t2=${column3}|i1=${column4}’”</code>
This is the same example as the one above it but the source of the data</td></tr>
<tr><td><b>^LVN</b></td><td>Listview Navigate - This command can be used to move the Listview widget. Navigation commands will be range checked. The command will attempt to position the specified list entry on the top line of the Listview widget. When navigating at the end of the list, however, the widget will position the last item in the list on the bottom line and will not leave blank lines at the bottom. The only exception to this case will be when the Listview has fewer entries than the number of displayable entries. If the optional select boolean is present, and the navigation command used support the select option, the item at the destination will be selected and a item selected custom event will be initiated.
• <b>Syntax</b>:
<code>“’^LVN-<addr range>,<navigation_command>[,<boolean_select_param>]’”</code>
• <b>Variables</b>: <i>address range</i>: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. <i>navigation command. optional select boolean</i> <b>Navigation Commands</b>: <b>t</b> or <b>T</b> - move to the top of the list (supports an optional select boolean). <b>b</b> or <b>B</b> - move to the bottom of the list (supports an optional select boolean). <b>d</b> or <b>D</b> - page down (DOES NOT support the optional select boolean. A select boolean will be ignored if present). <b>n</b> - move to a specific list entry number at position n. n is a zero based index. (supports an optional select boolean). (Note: If n is < 0 and select is true then the current selected item is deselected.) <b>u</b> or <b>U</b> - page up (DOES NOT support the optional select boolean. A select boolean will be ignored if present).
• <b>Examples</b>:
<code>SEND_COMMAND Panel,”’^LVN-42,B’”</code>
Move to the bottom of the list.
<code>SEND_COMMAND Panel,”’^LVN-42,d’”</code></td></tr>
</table>

Listview Commands	
	Move the list down a page. `SEND_COMMAND Panel,”’^LVN-42,3,1’”` Move the list to position 3 in the list and select the item at position 3.

^LVR	Listview Refresh Data - This command has two different functions. If it is sent without any parameters, it causes the Listview widget to load data from its configured data source. If optional parameters are included with the command, then the automatic data refresh options are configured. The typical behavior for auto refresh is that the last modified time of the data source is tracked. At the refresh interval, the last modified time of the data source is compared against the stored value. If the data is newer, then it is reloaded and the Listview widget is refreshed with the updated data. If the data is unchanged, then it is not reloaded. The default for auto refresh is off.
	• **Syntax**:
	`“’^LVR-<addr range>[,<refresh_interval>,<force_reload>]’”`
	• **Variable:** *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address.
	refresh_interval - the optional interval (in seconds) at which to check for newer data. 0 (the default) means auto refresh is off. Minimum is 5 seconds. If not specified, the current refresh interval is retained.
	force_reload - the optional parameter to force the Listview to ignore and data file timestamps and to force a clear on image caches for refreshed Listview images. Not specified or 0 will not force a reload, 1 will force a reload of data file and images associated with data file.
	Note: This can cause the images in a Listview to flicker upon the reload. This is the expected behavior due to the images being reloaded from the server.
	• **Examples**:
	`SEND_COMMAND Panel,”’^LVR-42’”`
	Commands the Listview widget to load the data from the data source and populate the Listview display widget.
	`SEND_COMMAND Panel,”’^LVR-42,15’”`
	Commands the Listview widget to check for an updated data source every 15 seconds.
	`SEND_COMMAND Panel,”’^LVR-42,600,1’”`
	Commands the Listview widget to check for an updated data source every hour, and to force a reload of the data and the images.

^LVS	Listview Sort Data - This command sets the columns that are used for sorting of lists, as well as the type of sorting that is done. The multiple columns are allowed in the sort procedure. The order of the columns in the command determine the order of the sorting. The first column is the primary sorting data, the second would be used for sorting with rows of data that are equal in the primary columns, and so on for however many columns are used for sorting. If no columns are listed in the command, then the current sorting columns are used if they have been previously defined. The type of sort is an optional part of the command and follows the sort columns. Initially, there are four different sort types available. *None* (**n**) - No sorting is performed. *Ascending* (**a**) - Ascending sort using localized character weighting. *Descending* (**d**) - Descending sort using localized character weighting. *Override* (*****) - Override sort syntax portion of command determines sorting. The override sort syntax allows for complex SQLite ORDER BY syntax for sorting. When override is selected, the sort columns that were set in the command or previously are ignored and the entire sorting statement must be in the override sort syntax. The words ORDER BY should not be in the syntax. They are inserted by the firmware. • **Syntax**: `“’^LVS-<addr range>,<primary sort column name, secondary sort column name,..., final sort column name>[;<sort type>[;<override sort syntax>]]’”` • **Variables**: *address range*: Address codes of buttons to affect. A ‘.’ between addresses includes the range, and & between addresses includes each address. *Sort columns* - comma separated list of sort columns in the order of sort priority. Sort columns can be specified using the ${column name} syntax that is used in the ^LVM command. Columns can be Content Fields or Metadata Fields in the controller Datafeed XML file generated by the controller. Metadata fields are prepended with “meta” in front of the “label” attribute of the field. *Sort Type* - A character indicating the sorting algorithm to use. ‘a’ - ascending ‘d’ - descending ‘*’ - override. Sort command syntax must follow in the next part of the command. ‘n’ - none (default). Any character that is not a,d, or * will set sort to none. *Override sort syntax* - A SQLite ORDER BY statement to use as the sort.

^LVS
(Cont.)
	• **Examples**:
	`SEND_COMMAND Panel,”’^LVS-42, ${artist name},${title};a ‘”`
	Commands the Listview widget to sort the data source by the artist name and then title in an ascending order. Equates to “artistname, title COLLATE LOCALIZED ASC” override syntax.
	`SEND_COMMAND Panel,”’^LVS-42, ${artist name},${title};d ‘”`
	Commands the Listview widget to sort the data source by the artist name and then title in an descending order. Equates to “artistname COLLATE LOCALIZED DESC, title COLLATE LOCALIZED DESC” override syntax.
	`SEND_COMMAND Panel,”’^LVS-42,;n’”`
	Commands the Listview widget to not sort the current data.
	`SEND_COMMAND Panel,”’^LVS-150,${user name},${text};*;meta${Record timestamp} ASC’”`
	Commands the panel to sort by the meta data field Record timestamp in ASCENDING order. The username and test fields are ignored.

Listview Commands	

	`SEND_COMMAND Panel,”’^LVS-150,;*;meta${Record timestamp} ASC’”`
	Commands the panel to sort by the meta data field “Record timestamp” in ASCENDING order. The username and test columns are ignored.
	`SEND_COMMAND Panel,”’^LVS-150,;*;LENGTH(${description}),${description} ASC’”`
	Command the panel to sort by the number of characters in the description field, and then by the contents of the description field in ASCENDING order.

Note: *Refer to Appendix B: Using NetLinx to Define a Data Source (Listview Buttons) for information on using NetLinx Code to define a data source for Listview buttons.*

Notes on Using Image URLs With Listview Buttons

Since a Listview button can retrieve images to display as part of the Listview, the column in the data table that sets the image URL will require the server’s username and password be included as part of the image URL.

The following example represents the contents of a .CSV file that has image URLs as part of the data. The URL Path column has some URLs with using http and no authentication credentials, some using http and user/password credentials, and one using https and user/password credentials:

```
File Type,    No,      URL Path
      GIF,     1,      http://www.w3schools.com/images/compatible_chrome.gif
      GIF,     2,      http://www.w3schools.com/images/compatible_ie.gif
      GIF,     3,      http://www.w3schools.com/images/compatible_firefox.gif
      PNG,     4,      http://user:password@controller-ni3100/xsimple_green.png
      PNG,     5,      https://user:password@controller-nx1200/__AMX__Varia-mute-
off.png       PNG,     6,      http://user:password@controller-nx1200/__AMX__Varia-
mute-on.png
```

NOTE: *HTTPS is supported in G5 panel firmware version v1.4.9 and higher.*

NOTE: *HTTPS is not supported by TPDesign5 dynamic image resources at this time.*

Programming Numbers

Color Table

Index No.	Name	Red	Green	Blue		Index No.	Name	Red	Green	Blue
0	Very Light Red	255	0	0		45	Medium Aqua	0	80	159
1	Light Red	223	0	0		46	Dark Aqua	0	64	127
2	Red	191	0	0		47	Very Dark Aqua	0	48	95
3	Medium Red	159	0	0		48	Very Light Blue	0	0	255
4	Dark Red	127	0	0		49	Light Blue	0	0	223
5	Very Dark Red	95	0	0		50	Blue	0	0	191
6	Very Light Orange	255	128	0		51	Medium Blue	0	0	159
7	Light Orange	223	112	0		52	Dark Blue	0	0	127
8	Orange	191	96	0		53	Very Dark Blue	0	0	95
9	Medium Orange	159	80	0		54	Very Light Purple	128	0	255
10	Dark Orange	127	64	0		55	Light Purple	112	0	223
11	Very Dark Orange	95	48	0		56	Purple	96	0	191
12	Very Light Yellow	255	255	0		57	Medium Purple	80	0	159
13	Light Yellow	223	223	0		58	Dark Purple	64	0	127
14	Yellow	191	191	0		59	Very Dark Purple	48	0	95
15	Medium Yellow	159	159	0		60	Very Light Magenta	255	0	255
16	Dark Yellow	127	127	0		61	Light Magenta	223	0	223
17	Very Dark Yellow	95	95	0		62	Magenta	191	0	191
18	Very Light Lime	128	255	0		63	Medium Magenta	159	0	159
19	Light Lime	112	223	0		64	Dark Magenta	127	0	127
20	Lime	96	191	0		65	Very Dark Magenta	95	0	95
21	Medium Lime	80	159	0		66	Very Light Pink	255	0	128
22	Dark Lime	64	127	0		67	Light Pink	223	0	112
23	Very Dark Lime	48	95	0		68	Pink	191	0	96
24	Very Light Green	0	255	0		69	Medium Pink	159	0	80
25	Light Green	0	223	0		70	Dark Pink	127	0	64
26	Green	0	191	0		71	Very Dark Pink	95	0	48
27	Medium Green	0	159	0		72	White	255	255	255

Color Table

28	Dark Green	0	127	0		73	Grey1	238	238	238
29	Very Dark Green	0	95	0		74	Grey3	204	204	204
30	Very Light Mint	0	255	128		75	Grey5	170	170	170
31	Light Mint	0	223	112		76	Grey7	136	136	136
32	Mint	0	191	96		77	Grey9	102	102	102
33	Medium Mint	0	159	80		78	Grey4	187	187	187
34	Dark Mint	0	127	64		79	Grey6	153	153	153
35	Very Dark Mint	0	95	48		80	Grey8	119	119	119
36	Very Light Cyan	0	255	255		81	Grey10	85	85	85
37	Light Cyan	0	223	223		82	Grey12	51	51	51
38	Cyan	0	191	191		83	Grey13	34	34	34
39	Medium Cyan	0	159	159		84	Grey2	221	221	221
40	Dark Cyan	0	127	127		85	Grey11	68	68	68
41	Very Dark Cyan	0	95	95		86	Grey14	17	17	17
42	Very Light Aqua	0	128	255		87	Black	0	0	0
43	Light Aqua	0	112	223		255	TRANSPARENT	99	53	99
44	Aqua	0	96	191						

Justification Values

Button State Number Justification Value

Justification	Justification Value	Justification Parameters
Absolute	0	0,<x offset,y offset>
top-left	1	none
top-middle	2	none
top-right	3	none
center-left	4	none
center-middle	5	none
center-right	6	none
bottom-left	7	none
bottom-center	8	none
bottom-right	9	none
scaled-to-fit	10	none
scale-maintain-aspect-ratio	11	none

Border Styles

Border Styles

#	Border Style	#	Border Style	#	Border Style	#	Border Style
1	None	41	Diamond 65	81	Menu Btm Rounded 25	121	Menu Rt Rounded 45
2	AMX Elite -L	42	Diamond 75	82	Menu Btm Rounded 35	122	Menu Rt Rounded 55
3	AMX Elite -M	43	Diamond 85	83	Menu Btm Rounded 45	123	Menu Rt Rounded 65
4	AMX Elite -S	44	Diamond 95	84	Menu Btm Rounded 55	124	Menu Rt Rounded 75
5	Bevel -L	45	Diamond 105	85	Menu Btm Rounded 65	125	Menu Rt Rounded 85
6	Bevel -M	46	Diamond 115	86	Menu Btm Rounded 75	126	Menu Rt Rounded 95
7	Bevel -S	47	Diamond 125	87	Menu Btm Rounded 85	127	Menu Rt Rounded 105
8	Circle 15	48	Diamond 135	88	Menu Btm Rounded 95	128	Menu Rt Rounded 115
9	Circle 25	49	Diamond 145	89	Menu Btm Rounded 105	129	Menu Rt Rounded 125
10	Circle 35	50	Diamond 155	90	Menu Btm Rounded 115	130	Menu Rt Rounded 135
11	Circle 45	51	Diamond 165	91	Menu Btm Rounded 125	131	Menu Rt Rounded 145
12	Circle 55	52	Diamond 175	92	Menu Btm Rounded 135	132	Menu Rt Rounded 155
13	Circle 65	53	Diamond 185	93	Menu Btm Rounded 145	133	Menu Rt Rounded 165
14	Circle 75	54	Diamond 195	94	Menu Btm Rounded 155	134	Menu Rt Rounded 175
15	Circle 85	55	Double Bezel -L	95	Menu Btm Rounded 165	135	Menu Rt Rounded 185
16	Circle 95	56	Double Bezel -M	96	Menu Btm Rounded 175	136	Menu Rt Rounded 195
17	Circle 105	57	Double Bezel -S	97	Menu Btm Rounded 185	137	Menu Lt Rounded 15
18	Circle 115	58	Double Line	98	Menu Btm Rounded 195	138	Menu Lt Rounded 25
19	Circle 125	59	Fuzzy	99	Menu Top Rounded 15	139	Menu Lt Rounded 35
20	Circle 135	60	Glow -L	100	Menu Top Rounded 25	140	Menu Lt Rounded 45
21	Circle 145	61	Glow -M	101	Menu Top Rounded 35	141	Menu Lt Rounded 55
22	Circle 155	62	Glow -S	102	Menu Top Rounded 45	142	Menu Lt Rounded 65

Border Styles

23	Circle 165	63	Help Down	103	Menu Top Rounded 55	143	Menu Lt Rounded 75
24	Circle 175	64	Neon Active -L	104	Menu Top Rounded 65	144	Menu Lt Rounded 85
25	Circle 185	65	Neon Active -S	105	Menu Top Rounded 75	145	Menu Lt Rounded 95
26	Circle 195	66	Neon Inactive -L	106	Menu Top Rounded 85	146	Menu Lt Rounded 105
27	Cursor Bottom	67	Neon Inactive -S	107	Menu Top Rounded 95	147	Menu Lt Rounded 115
28	Cursor Bottom w/hole	68	Oval H 60x30	108	Menu Top Rounded 105	148	Menu Lt Rounded 125
29	Cursor Top	69	Oval H 100x50	109	Menu Top Rounded 115	149	Menu Lt Rounded 135
30	Cursor Top w/hole	70	Oval H 150x75	110	Menu Top Rounded 125	150	Menu Lt Rounded 145
31	Cursor Left	71	Oval V 30x60	111	Menu Top Rounded 135	151	Menu Lt Rounded 155
32	Cursor Left w/hole	72	Oval V 50x100	112	Menu Top Rounded 145	152	Menu Lt Rounded 165
33	Cursor Right	73	Oval V 75x150	113	Menu Top Rounded 155	153	Menu Lt Rounded 175
34	Cursor Right w/hole	74	Oval V 100x200	114	Menu Top Rounded 165	154	Menu Lt Rounded 185
35	Custom Frame	75	Picture Frame	115	Menu Top Rounded 175	155	Menu Lt Rounded 195
36	Diamond 15	76	Quad Line	116	Menu Top Rounded 185		
37	Diamond 25	77	Single Line	117	Menu Top Rounded 195		
38	Diamond 35	78	Windows Style Popup	118	Menu Rt Rounded 15		
39	Diamond 45	79	Windows Style Popup (status bar)	119	Menu Rt Rounded 25		
40	Diamond 55	80	Menu Btm Rounded 15	120	Menu Rt Rounded 35		

ISO-8859-1 Character Encoding/Decoding table

ISO-8859-1 Character Encoding/Decoding

	Character value (decimal)	**Character value (hex)**	**^TXT and ^UTF interchangeable**	**?TXT Response Flag in Backwards Compatibility Mode (^ENC-1 was sent)**	**?TXT Response Flag in default (UTF-8) Mode**
ASCII	0-127	0x00-0x7F	Yes	0 (Latin-1)	2 (UTF-8)
Latin-1 (Windows-1252 remap range)	128-159	0x80-0x9F	No	1 (Hex-quad)	2 (UTF-8)
Latin-1	160-255	0xA0-0xFF	No	0 (Latin-1)	2 (UTF-8)
Unicode	>255	>0xFF	No	1 (Hex-quad)	2 (UTF-8)

Resource Escape Codes

Resource Escape Codes

Sequence	**Panel Information**	**Sequence**	**Panel Information**
$DV	Device number	$AP	Address port
$SY	System number	$CC	Channel code
$IP	IP address	$CP	Channel port
$HN	Host name	$LC	Level code
$MC	MAC address	$LP	Level port
$PX	X resolution of current panel mode/file	$BX	X resolution of current button

Resource Escape Codes

$PY	Y resolution of current panel mode/file	$BY	Y resolution of current button
$ST	Current state	$BN	Name of button
$AC	Address code		

Virtual Keystroke Commands

Virtual Keystroke Commands

Keycode	Key	Keycode	Key	Keycode	Key
1	Soft-L	74	;	147	Numpad 3
2	Soft-R	75	Apostrophe	148	Numpad 4
3	Home	76	/	149	Numpad 5
4	Back	77	@	150	Numpad 6
5	Call	78	Num	151	Numpad 7
6	End Call	79	Headset Hook	152	Numpad 8
7	0	80	Focus	153	Numpad 9
8	1	81	+	154	Numpad /
9	2	82	Menu	155	Numpad *
10	3	83	Notification	156	Numpad -
11	4	84	Search	157	Numpad +
12	5	85	Media Play/Pause	158	Numpad .
13	6	86	Media Stop	159	Numpad ,
14	7	87	Media Next	160	Numpad Enter
15	8	88	Media Prev	161	Numpad =
16	9	89	Media Rew	162	Numpad (
17	*	90	Media FF	163	Numpad)
18	#	91	Mute	164	Volume Mute
19	DPad-U	92	Page Up	165	Info
20	DPad-D	93	Page Down	166	Chan Up
21	DPad-L	94	Pict Symbols	167	Chan Down
22	DPad-R	95	Switch Charset	168	Zoom In
23	DPad-Center	96	Button A	169	Zoom Out
24	Vol Up	97	Button B	170	TV
25	Vol Dn	98	Button C	171	Window
26	Power	99	Button X	172	Guide
27	n/a	100	Button Y	173	DVR
28	Clear	101	Button Z	174	Bookmark
29	A	102	Button L1	175	Captions
30	B	103	Button R1	176	Settings
31	C	104	Button L2	177	TV Power
32	D	105	Button R2	178	TV Input
33	E	106	Button Thumb L	179	STB Power
34	F	107	Button Thumb R	180	STB Input
35	G	108	Button Start	181	AVR Power
36	H	109	Button Select	182	AVR Input
37	I	110	Button Mode	183	Prog Red
38	J	111	Escape	184	Prog Green
39	K	112	Forward Delete	185	Prog Yellow
40	L	113	Ctrl-L	186	Prog Blue
41	M	114	Ctrl-R	187	App Switch
42	N	115	Caps Lock	188	Button 1
43	O	116	Scroll Lock	189	Button 2
44	P	117	Meta L	190	Button 3

Virtual Keystroke Commands

45	Q	118	Meta R	191	Button 4
46	R	119	Function	192	Button 5
47	S	120	SysReq / Print Screen	193	Button 6
48	T	121	Break	194	Button 7
49	U	122	Move Home	195	Button 8
50	V	123	Move End	196	Button 9
51	W	124	Insert	197	Button 10
52	X	125	Forward	198	Button 11
53	Y	126	Media Play	199	Button 12
54	Z	127	Media Pause	200	Button 13
55	,	128	Media Close	201	Button 14
56	.	129	Media Eject	202	Button 15
57	Alt-L	130	Media Record	203	Button 16
58	Alt-R	131	F1	204	Language Switch
59	Shift-L	132	F2	205	Manner Mode
60	Shift-R	133	F3	206	3D Mode
61	TAB	134	F4	207	Contacts
62	Space	135	F5	208	Calendar
63	Sym	136	F6	209	Music
64	Explorer	137	F7	210	Calculator
65	Envelope	138	F8	211	Zenkaku Hankaku
66	Enter	139	F9	212	Eisu
67	Delete	140	F10	213	Mhenkan
68	Grave	141	F11	214	Henkan
69	-	142	F12	215	Katakana Hiragana
70	_	143	Num Lock	216	Yen
71	[	144	Numpad 0	217	Ro
72	]	145	Numpad 1	218	Kana
73	\	146	Numpad 1	219	Assist

Appendix A: Upgrading Firmware via NetLinx Studio

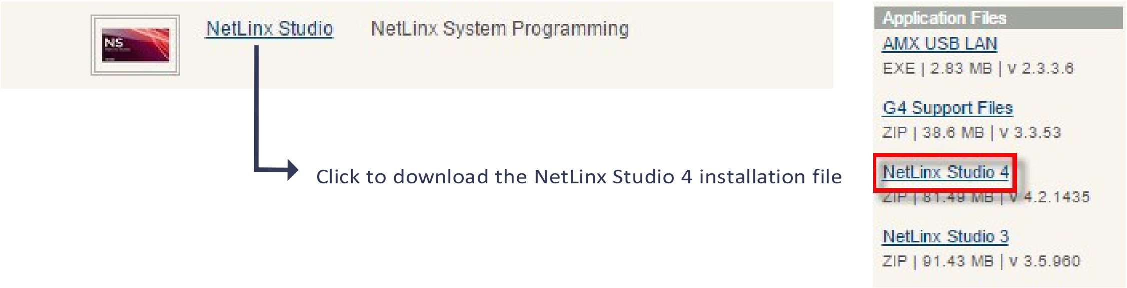
Overview

The latest firmware (*.kit) file for each panel is available to download from www.amx.com. To download firmware files, go to the catalog page for your panel type, and click the link under "Firmware Files" on the right side of the catalog page. The ZIP file that is downloaded via this link contains the firmware (*.kit) file that can be loaded on the panel, as well as release notes and any relevant programming instructions.

NetLinx Studio 4

The latest version (4.x) of the NetLinx Studio software program is available to download from www.amx.com:

1. Go to **Products > Integration Software > Development Tools** and click on NetLinx Studio to open the **NetLinx Studio** catalog page.
2. Click the *NetLinx Studio 4* link download the installation file (FIG. 156):



NetLinx Studio NetLinx System Programming

Application Files

AMX USB LAN
EXE | 2.83 MB | v 2.3.3.6

G4 Support Files
ZIP | 38.6 MB | v 3.3.53

NetLinx Studio 4
ZIP | 81.49 MB | v 4.2.1435

NetLinx Studio 3
ZIP | 91.43 MB | v 3.5.960

Click to download the NetLinx Studio 4 installation file

FIG. 156 NetLinx Studio v4 download links on www.amx.com

NOTE: *The following instructions assume that the G5 touch panel is connected and communicating with a NetLinx Controller, and that communication with the controller has been established in NetLinx Studio. Refer to NetLinx Studio online help and the NetLinx Studio 4 Instruction Manual for instructions on using NetLinx Studio.*

Upgrading Firmware via NetLinx Studio (v4 or Higher)

G5 touch panels use an Ethernet connection for programming, firmware updates, and touch panel file transfer via NetLinx Studio. If you have access to the panel's network, you may transfer files directly to the panel through NetLinx Studio.

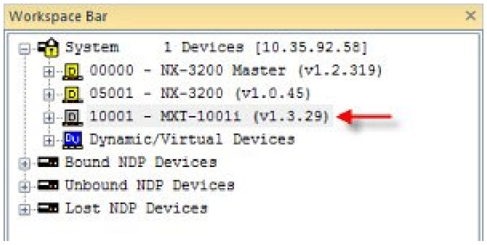
NetLinx Studio features the ability to transfer G5 firmware files directly to a G5 touch panel via HTTP (via a stand-alone web server). This feature is provided to shorten the amount of time required for transferring a G5 *.kit file by removing the NetLinx Controller from the transfer path.

*.kit files for G5 panels contain a token to signify to NetLinx Studio that a web server file transfer can take place, as indicated in the file information window of the Send To NetLinx Device dialog:

Look for "`**** HTTP File Transfer Capable ****`" at the end of the file.

When NetLinx Studio detects that the file is a G5 *.kit file, it will automatically attempt to send the file via HTTP (using the standalone web server that is started by NetLinx Studio).

1. In NetLinx Studio, open the *Online Tree* tab of the Workspace bar.
2. Under *System*, select a G5 panel for the firmware update (FIG. 157):



```
System     1 Devices [10.35.92.58]
  00000 - NX-3200 Master (v1.2.319)
  05001 - NX-3200 (v1.0.45)
  10001 - MXT-1001i (v1.3.29)
  Dynamic/Virtual Devices
Bound NDP Devices
Unbound NDP Devices
Lost NDP Devices
```

FIG. 157 NetLinx Studio Online Tree (MXT-1001 selected)

3. Right-Click on the G5 panel, and select **Firmware Transfer** from the context menu (FIG. 158):

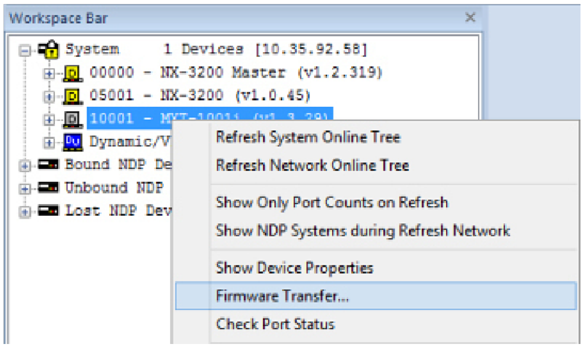

Workspace Bar
System 1 Devices [10.35.92.58]
00000 - NX-3200 Master (v1.2.319)
05001 - NX-3200 (v1.0.45)
10001 - MXT-10011 (v1.3.29)
Dynamic/V
Bound NDP De
Unbound NDP
Lost NDP Dev
Refresh System Online Tree
Refresh Network Online Tree
Show Only Port Counts on Refresh
Show NDP Systems during Refresh Network
Show Device Properties
Firmware Transfer...
Check Port Status

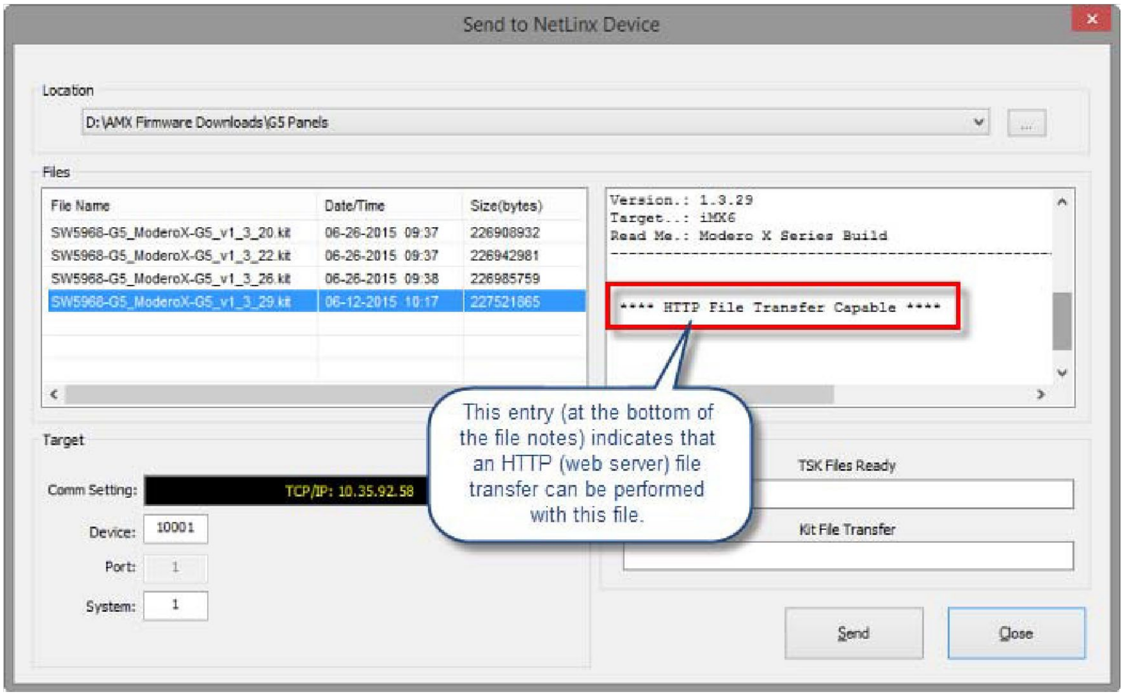
FIG. 158 NetLinx Studio Online context menu (Firmware Transfer selected) This

invokes the Send To *NetLinx Device* dialog.

4. Under Location. click the Browse (...) button to locate and select the directory containing the G5 firmware (*.kit) file that will be transferred, in the Browse For Folder dialog.

5. Click **OK** to close the Browse For Folder dialog and populate the Files window with a listing of *.kit files found in the selected folder.

6. In the Files window, click to select the G5 *.kit file to transfer (FIG. 159):



Location: D:\AMX Firmware Downloads\G5 Panels

File Name	Date/Time	Size(bytes)
SW5968-G5_ModeroX-G5_v1_3_20.kit	06-26-2015 09:37	226908932
SW5968-G5_ModeroX-G5_v1_3_22.kit	06-26-2015 09:37	226942981
SW5968-G5_ModeroX-G5_v1_3_26.kit	06-26-2015 09:38	226985759
SW5968-G5_ModeroX-G5_v1_3_29.kit	06-12-2015 10:17	227521865

```
Version.: 1.3.29
Target..: iMX6
Read Me.: Modero X Series Build
------------------------------------------------

**** HTTP File Transfer Capable ****
```

This entry (at the bottom of the file notes) indicates that an HTTP (web server) file transfer can be performed with this file.

Comm Setting: TCP/IP: 10.35.92.58

Device: 10001

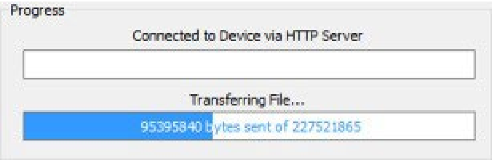
Port: 1

System: 1

FIG. 159 NetLinx Studio - Send to NetLinx Device dialog This

invokes the Send To *NetLinx Device* dialog.

7. Click **Send** to initiate the firmware file transfer. The progress of the transfer is indicated in the progress bars (FIG. 160):



Connected to Device via HTTP Server

Transferring File...

95395840 bytes sent of 227521865

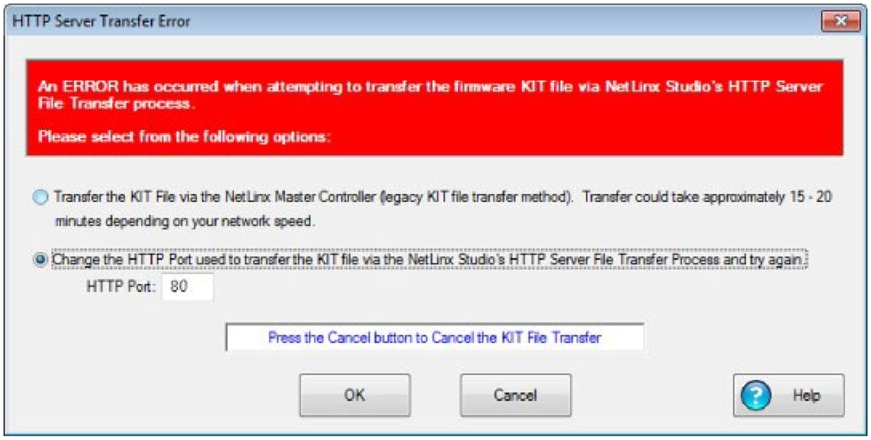
FIG. 160 NetLinx Studio - Send to NetLinx Device dialog (Progress bars indicating an active firmware file transfer)

8. The Panel will display the Message “*Updating System Files*”, then restart itself.

9. The Installing *System Update* page will be displayed on the panel until the firmware upgrade process is complete. At this point, the panel will reboot and open it’s home page.

HTTP Server Transfer Error

If an error occurs during this type of transfer, then the HTTP Server Transfer Error dialog is invoked (FIG. 161):



HTTP Server Transfer Error

An ERROR has occurred when attempting to transfer the firmware KIT file via NetLinx Studio's HTTP Server File Transfer process.

Please select from the following options:

○ Transfer the KIT File via the NetLinx Master Controller (legacy KIT file transfer method). Transfer could take approximately 15 - 20 minutes depending on your network speed.

● Change the HTTP Port used to transfer the KIT file via the NetLinx Studio's HTTP Server File Transfer Process and try again.

HTTP Port: 80

Press the Cancel button to Cancel the KIT File Transfer

FIG. 161 NetLinx Studio v3.4 or higher - HTTP Server Error dialog

In this case, there are two options for proceeding with the firmware transfer:

- Select **Transfer the KIT File via the NetLinx Controller Controller (legacy KIT file transfer method)**... to proceed using the standard (non-HTTP) method used for other NetLinx Devices (via the controller controller) when OK is clicked. Note that depending on network speed and the size of the *.kit file, this method could take up to 20-30 minutes to complete. More specifically, timed tests indicate that it takes approximately 60 seconds per 9.5MB of a *.kit file to transfer.

  The following table indicates the approximate length of time to send a *.kit file via the legacy file transfer method:

File Size	Time Required to Complete Transfer (legacy file transfer method)
0-150MB	10 - 15 minutes
150-200MB	15 - 20 minutes
200-250MB	20 - 25 minutes
250-300MB	25 - 30 minutes
300-350MB	30 - 35 minutes
>350MB	> 35 minutes

- By default, **Change the HTTP Port used to transfer the KIT file**... is selected. Use this option to change the HTTP port assignment, in cases where the IP port (default = 80) is in conflict or blocked on the PC. This option will restart the web server with a different HTTP port assignment and restart the file transfer when **OK** is clicked.
- Select the appropriate option and click **OK** to restart the file transfer.
- Click **Cancel** to cancel the current file transfer.

Appendix B: Using NetLinx to Define a Data Source (Listview Buttons)

Example Listview Workflow - NetLinx Data Source

The following section describes an example workflow for implementing a Listview button that uses NetLinx code as the data source. The use case for this example is that of a contact list for a SIP phone system. In this case, the user finds and presses a contact on the screen to initiate the call.

The workflow in this example describes each step required to implement a data source for a Listview button via NetLinx Code:

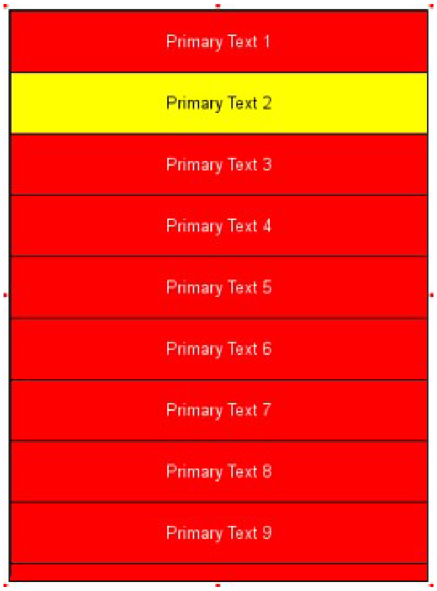
1. Creating a Listview button on a G5 panel page and set button properties
2. Creating a data source in NetLinx code
3. Conf iguring and populating the Listview
4. Configuring a response to a user selection

1) Create the Listview Button and Set Button Properties

Create a Listview button in TPDesign5 and configure the display characteristics for the default and selected states.

Although not currently being rendered correctly in the screenshot below, this Listview has two lines of text and an image Varia on the left for each Listview entry.

1. In TPDesign5 (v1.0.2 or greater), use the Button Draw Tool to draw a new button.
2. In the General tab of the Properties window, select Listview as the Type (FIG. 162):



Primary Text 1

Primary Text 2

Primary Text 3

Primary Text 4

Primary Text 5

Primary Text 6

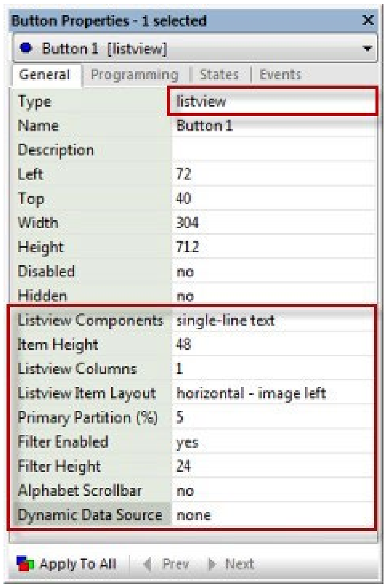
Primary Text 7

Primary Text 8

Primary Text 9

FIG. 162 TPDesign5 - Listview button

3. Use the TPD5 Properties window to set General, Programming, States and Events properties to configure the list items and the display characteristics for the Default and Selected states, as well as provide the Listview button with an Address code assignment. Note that Listview buttons use standard button properties, as well as several new properties that are specific to Listview buttons:
   a. In the General tab, set properties to specify basic display characteristics for the selected Listview button (FIG. 163).



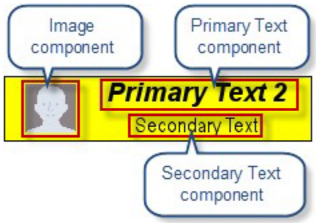
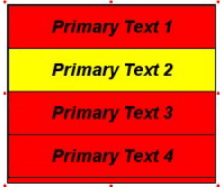
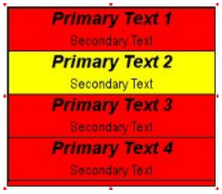
Button Properties - 1 selected

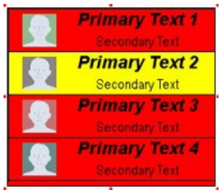
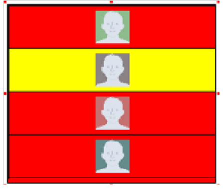
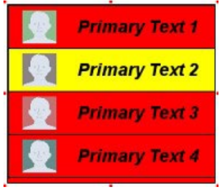
● Button 1 [listview]

Type	listview
Name	Button 1
Description	
Left	72
Top	40
Width	304
Height	712
Disabled	no
Hidden	no
Listview Components	single-line text
Item Height	48
Listview Columns	1
Listview Item Layout	horizontal - image left
Primary Partition (%)	5
Filter Enabled	yes
Filter Height	24
Alphabet Scrollbar	no
Dynamic Data Source	none

FIG. 163 TPDesign5 - General Properties for Listview buttons

General button properties that are specific to Listview buttons include:

Listview Buttons - General Properties	
List View Components	This property controls which components (*Primary Text, Secondary Text and Image*) will be displayed on the selected Listview button. With a Listview button selected in the Design View, click the browse (...) button on the Listview Components (General) property to open the *Edit Listview Components* dialog. Use this dialog to specify which components (Primary Text, Secondary Text and Image) will be displayed on the selected Listview button.  Image component Primary Text component Primary Text 2 Secondary Text Secondary Text component If only **Primary Text** is selected in the *Edit Listview Components* dialog (the default setting for new Listview buttons), each list item is represented with a single line of text using center-middle justification and the font face and size specified by the *Text Color, Font* and *Font Size* (State) properties (as well as *Text Effect* and *Text Effect Color* if desired).  Primary Text 1 Primary Text 2 Primary Text 3 Primary Text 4 • The *List View Components* (General) Property will indicate **single-line text**. If **Primary Text** and **Secondary Text** are selected, each list item is represented with a two lines of text.  Primary Text 1 Secondary Text Primary Text 2 Secondary Text Primary Text 3 Secondary Text Primary Text 4 Secondary Text • The two lines of text are stacked vertically, with each line centered horizontally. • The font face and size are specified by the Secondary Font and Secondary Font Size (State) properties. • The text is rendered within a two-pixel margin of the button boundary. • Note that the *Secondary Text* option is only enabled if *Primary Text* is selected. • Secondary Text uses the same Text Color settings as the Primary Text. • The *List View Components* (General) Property will indicate **two-line text**.

Listview Buttons - General Properties	
List View Components (Cont.)	If **Primary Text, Secondary Text** and **Image** are selected, each list item is represented with two lines of text and an image on the left side.  Primary Text 1 Secondary Text Primary Text 2 Secondary Text Primary Text 3 Secondary Text Primary Text 4 Secondary Text - The image is left-justified within a six-pixel margin of the top, bottom, and left item boundaries, and is scaled-to-fit within a square region. - The two lines of text are stacked vertically and centered horizontally in the remaining item region. The top line (Primary Text) is rendered using the font face and size specified by the Font and Font Size (State) properties. The bottom line (Secondary Text) is rendered using the font face and size specified by the *Secondary Font* and *Secondary Font Size* (State) properties. The text is rendered within a two-pixel margin of the button boundary. - The *List View Components* (General) Property will indicate **two-line text w/ Image**. If only **Image** is selected in the *Edit Listview Components* dialog, each list item is represented with a single image centered horizontally within the item region, within a six-pixel margin of the item region.  - The *List View Components* (General) Property will indicate **image only**. - If **Primary Text** and **Image** are selected in the *Edit Listview Components* dialog, each list item is represented with a single line of text and an image on the left side.  Primary Text 1 Primary Text 2 Primary Text 3 Primary Text 4 - The image is left-justified within a six-pixel margin of the top, bottom, and left item boundaries, and is scaled-to-fit within a square region. - The text is center-middle justified in the remaining portion of the item region within a two-pixel margin, using the font and font size specified by the *Font* and *Font Siz*e (States) properties. - The *List View Components* (General) Property will indicate **single-line text w/ Image**.
Item Height	This property controls the height for the list view items (in pixels).
List View Columns	This property controls the number of columns to display. By default, this value is set to 1. This property provides the ability to present a “grid view” on the Listview button, if desired.
List View Item Layout	This property controls the layout of the components (*Primary Text, Secondary Text* and *Image*) specified to display on the list view items in the selected Listview button. Listview components are selected via the *List View Components* (General) property. Click in this field to select from a drop-down of layout options for list items (horizontal - image left, horizontal - image right and vertical - image top).
Primary Partition (%)	This property sets the position of the separation between the Image and the Primary/Secondary Text components.
Secondary Partition (%)	If the *List View Item Layout* property is set to is set to horizontal - image left (the default setting), the Secondary Partition (%) sets the position of the separation between the Primary Text and the Image as a percentage of cell height (allowed range = 5%-95%). • If the List View Item Layout property is set to is set to horizontal - image right, the Secondary Partition (%)sets the position of the separation between the Primary Text and the Image as a percentage of cell height (allowed range = 5%-95%): • If the List View Item Layout property is set to vertical- image top, the Secondary Partition represents thearea used by the Image. In this case, Secondary Partition (%) sets the position of the separation between the Image and the Primary Text as a percentage of cell height (allowed range = 5%-95%).
Filter Enabled	Use this property to enable/disable the filter (Search) feature on the selected Listview button. By default, this property is set to no (disabled). To enable this feature, select yes from the drop-down menu. If enabled, a search window will be rendered at the top of the Listview button, with a height specified by the Filter Height property. The remaining area of the Listview button will be available for the display of list items:

Listview Buttons - General Properties	
Filter Height	• • Use this property to specify the height of the filter entry box for a Listview button (in pixels).Note that this property is available only if Filter Enabled is set to Yes.The minimum allowed value (and the default setting) is 24 pixels.
Alphabet Scrollbar	This property enables/disables the alphabet scrollbar feature for Listview buttons.
Dynamic Data Source	This property specifies the data source (CSV or XML) to use as the source for content that will be displayed on the selected Listview button.

b. In the Programming tab, assign a unique Address Port and Address Code to the selected Listview button:

Listview Buttons - Programming Properties	
Address Code	Select or enter the address code sent to the controller on the specified Address Port. The options available to the Address Code property depend on the Address Port selection: • If 1 is selected as the Address Port, then the options for Address Code are None and Auto-Assign. Select None to leave the Address Code unspecified. Select Auto-Assign to automatically assign the next available Address Code to the selected TPD5 element. • If 0-Setup Port is selected as the Address Port, then the options for Address Code are Advanced Codes or Basic Codes. By default, the Basic Address Codes are displayed: Click on Date Display to select from a list of date display formats. Click on Time Display to select from a list of time display formats. Click Advanced Codes to view the Advanced Channel Code options: Click on None to leave the Address Code unspecified. Click on Panel Setup to select Connection Status. This option will display the panel's current connection status on the selected element.
Address Port	Select or enter the port to which the selected element's Address Code will be associated. The options are "1" (the default setting) and "0-setup port": • If 1 is selected as the Address Port, then the options for the Address Code property are None and Auto-Assign. • If 0-Setup Port is selected as the Address Port, then the options for Address Code are Advanced Codes or Basic Codes. By default, the Basic Address Codes are displayed.
Note that Listview buttons do not use Channel Port and Channel Code assignments. The combination of Address Port and Address Code must be unique. See Address Codes (Basic and Advanced) in the TPD5 online help for details.	

c. In the States tab, set (font) properties to specify font display characteristics for the Default and Selected states for the selected Listview button. States properties that are specific to Listview buttons include:

- Secondary Font
- Secondary Font Size

d. In the Events tab, set event properties for the selected Listview button. Listview button support three Events properties that are specific to Listview buttons. However, these Events support the same actions as existing events:

- Item Selected
- Scrollbar Begin
- Scrollbar End

NOTE: Refer to the TPDesign5 online help for descriptions of all button properties.

2) Create the Data Source

Follow the example NetLinx code (below) to create a data source in NetLinx and publish the data source to the NetLinx Controller's internal web server.

The "*Data_PublishFeed*()" function (see *NetLinx.axi*) will return a URL for the published data.

NetLinx Usage Example - ASCII

```
PROGRAM_NAME='Listview Example'

DEFINE_DEVICE
dvTP = 10001:1:0

DEFINE_CONSTANT
// Listview button address
INTEGER btnListview = 11

DEFINE_VARIABLE
CHAR publishedURL[DATA_MAX_VALUE_LENGTH] CHAR recordsetID[DATA_MAX_ID_LENGTH]
```

```
DEFINE_FUNCTION CreateDataFeed()
{
    STACK_VAR DATA_FEED datafeed
    STACK_VAR DATA_RECORD record

// --------------------------------------------------
// CREATE A NEW DATA FEED
// ------------------------------------------------
- datafeed.name = 'phonelist'  datafeed.description
= 'Employees'
 datafeed.source = 'netlinx Listview Example code'
DATA_CREATE_FEED(datafeed)

 // A recordset id is required for adding records to the feed recordsetID = 'phonelist'

// --------------------------------------------------
// DEFINE AND POPULATE THE DATA FIELDS
// This example will have 10 names in a phone list
// --------------------------------------------------
 // Records can have metadata fields and content fields. In this
 // example we won't use any metadata
 SET_LENGTH_ARRAY(record.metadata, 0)

 // We will have 3 content fields per record: photo, name and phone number
SET_LENGTH_ARRAY(record.content, 3)

 // Initialize the field attributes that will be the same for every record
 // the first field in a record will be the image
 record.content[1].id = 'photo';
record.content[1].type = DATA_TYPE_IMAGE;
record.content[1].format = DATA_FORMAT_URL;

 // The label can be something different from the id but in our case we'll
    // keep them the same
record.content[1].label = 'photo';

 // The second field in a record will be the name
record.content[2].id = 'name';  record.content[2].type
= DATA_TYPE_STRING;  record.content[2].format = '';
record.content[2].label = 'name';

 // The third field will be the phone number
record.content[3].id = 'number';  record.content[3].type
= DATA_TYPE_STRING;  record.content[3].format =
DATA_FORMAT_PHONE;  record.content[3].label = 'number';

 // The next step is to put in the actual values for the 3 fields
 // Do this for the first record
 record.content[1].value = 'http://192.168.222.333/ftp/listview/hunter.jpg'
record.content[2].value = 'Hunter Pence'  record.content[3].value = '888-
555-1111'

 // Add the record to the feed
 DATA_ADD_RECORD(datafeed.name, recordsetID, record)
// The same record can be reused for the rest of the list // Just
change the relevant values and add the record to the feed
record.content[1].value =
'http://192.168.222.333/ftp/listview/pablo.jpg'
record.content[2].value = 'Pablo Sandoval' record.content[3].value =
'888-555-2222' DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
'http://192.168.222.333/ftp/listview/buster.jpg'
record.content[2].value = 'Buster Posey' record.content[3].value =
'888-555-3333' DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
'http://192.168.222.333/ftp/listview/angel.jpg'
record.content[2].value = 'Angel Pagan' record.content[3].value =
'888-555-4444' DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
'http://192.168.222.333/ftp/listview/jeremy.jpg'
record.content[2].value = 'Jeremy Affeldt' record.content[3].value =
'888-555-5555' DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
'http://192.168.222.333/ftp/listview/madison.jpg'
record.content[2].value = ‘Madison Bumgarner’ record.content[3].value =
‘888-555-6666’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/timh.jpg’
record.content[2].value = ‘Tim Hudson’ record.content[3].value =
‘4888-555-7777’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/timl.jpg’
record.content[2].value = ‘Tim Lincecum’ record.content[3].value =
‘888-555-8888’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/javier.jpg’
record.content[2].value = ‘Javier Lopez’ record.content[3].value =
‘888-555-9999’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/jake.jpg’
record.content[2].value = ‘Jake Peavy’ record.content[3].value =
‘888-555-1010’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/sergio.jpg’
record.content[2].value = ‘Sergio Romo’ record.content[3].value = ‘888-
555-1020’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/ryan.jpg’
record.content[2].value = ‘Ryan Vogelsong’ record.content[3].value =
‘888-555-1030’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/brandon.jpg’
record.content[2].value = ‘Brandon Belt’ record.content[3].value = ‘888-
555-1040’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/andrew.jpg’
record.content[2].value = ‘Andrew Susac’ record.content[3].value =
‘888-555-1050’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/gregor.jpg’
record.content[2].value = ‘Gregor Blanco’ record.content[3].value =
‘888-555-1060’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)

record.content[1].value =
‘http://192.168.222.333/ftp/listview/michael.jpg’
record.content[2].value = ‘Michael Morse’ record.content[3].value =
‘888-555-1070’ DATA_ADD_RECORD(datafeed.name, recordsetID, record)
 // The final step is to publish the feed
 publishedURL = DATA_PUBLISH_FEED(datafeed.name)
 }

DEFINE_START
     CreateDataFeed()

DEFINE_EVENT
DATA_EVENT[dvTP]
{

 ONLINE:
 {
 // Set the URL for the data source for the listviewer in the panel
 SEND_COMMAND dvTP,”’^LVD-’,ITOA(btnListview),’,’,publishedURL”
 // Map the fields in the listviewer to the columns
 SEND_COMMAND dvTP,”’^LVM-’,ITOA(btnListview),’,i1=${photo}|t1=${name}|t2=${number}’”
 // Sort by name
 SEND_COMMAND dvTP,”’^LVS-’,ITOA(btnListview),’,${name};a’”
 // Command the listview to load the data from the controller
 SEND_COMMAND dvTP,”’^LVR-’,ITOA(btnListview)”
 }
}

// The custom event that is raised whenever a listview item is selected on the panel
CUSTOM_
```

```
EVENT[dvTP,btnListview,LISTVIEW_ON_ROW_SELECT_EVENT
] {

 SLONG payloadId
 SLONG payloadType
 CHAR fields[2][16]
 CHAR name[DATA_MAX_VALUE_LENGTH] CHAR number[DATA_MAX_VALUE_LENGTH]
 DATA_RECORD record
    // Get the data access ID from the custom event
payloadId = custom.value1
    // Get the data type from the custom event
payloadType = custom.value2

 if (payloadId > 0 && payloadType == DATA_STRUCTURE_DATARECORD)
 {
 // Specify which fields we want to retrieve from the
payload  fields[1] = ‘name’  fields[2] = ‘number’
 // Populate a record with the requested fields from the event
if (DATA_GET_EVENT_RECORD(dvTP, payloadId, fields, record) > 0)
 {

          // All is well so far so retrieve the values that we are
          // interested in from the selection that the user made on // the panel.
   name = record.content[1].value
   number = record.content[2].value
          // Put the name and number that was selected on a popup and
           // show the popup
           SEND_COMMAND dvTP,”’^TXT-50,0,’,name”
           SEND_COMMAND dvTP,”’^TXT-51,0,’,number”
           SEND_COMMAND dvTP,”’^PPN-Calling’”
 }
 }
}
(**********************************************************)
(*            THE ACTUAL PROGRAM GOES BELOW               *)
(**********************************************************)
DEFINE_PROGRAM
(**********************************************************)
(*                   END OF PROGRAM                       *)
(*         DO NOT PUT ANY CODE BELOW THIS COMMENT        *)
(**********************************************************)
```

3) Configure the Response to a User Selection

Follow the CUSTOM_EVENT example at the end of the *NetLinx Usage Example - ASCII* (above) to retrieve the phone number that was selected by the user.

Appendix C: Text Formatting

Text Formatting Codes for Bargraphs

Text formatting codes for bargraphs provide a mechanism to allow a portion of a bargraphs text to be dynamically provided information about the current status of the level (multistate and traditional). These codes are entered into the text field along with any other text.

The following is a code list used for bargraphs:

Bargraph Text Code Inputs		
Code	**Bargraph**	**Multi-State Bargraph**
$P	Display the current percentage of the bargraph (derived from the Adjusted Level Value as it falls between the Range Values)	Display the current percentage of the bargraph (derived from the Adjusted Level Value as it falls between the Range Values)
$V	Raw Level Value	Raw Level Value
$L	Range Low Value	Range Low Value
$H	Range High Value	Range High Value
$S	N/A	Current State
$A	Adjusted Level Value (Range Low Value subtracted from the Raw Level Value)	Adjusted Level Value (Range Low Value subtracted from the Raw Level Value)
$R	Low Range subtracted from the High Range	Low Range subtracted from the High Range
$$	Dollar sign	Dollar sign

By changing the text on a button (via a VT command), you can modify the codes on a button. When one of the Text Formatting Codes is encountered by the firmware, it is replaced with the correct value. These values are derived from the following operations:

Formatting Code Operations	
Code	**Operation**
$P	(Current Value - Range Low Value / Range High Value - Range Low Value) x 100
$V	Current Level Value
$L	Range Low Value
$H	Range High Value
$S	Current State (if regular bargraph then resolves to nothing)
$A	Current Value - Range Low Value
$R	Range High Value - Range Low Value
$$	Dollar sign

Given a current raw level value of 532, a range low value of 500, and a high range value of 600, the following text formatting codes would yield the following strings as shown in the table below:

Example	
Format	**Display**
$P%	32%
$A out of $R	32 out of 100
$A of 0 - $R	32 of 0 - 100
$V of $L - $H	532 of 500 - 600

Text Area Input Masking

Text Area Input Masking may be used to limit the allowed/correct characters that are entered into a text area. For example, in working with a zip code, a user could limit the entry to a max length of only 5 characters; with input masking, this limit could be changed to 5 mandatory numerical digits and 4 optional numerical digits. A possible use for this feature is to enter information into form fields. The purpose of this feature is to:

- Force the use of correct type of characters (i.e. numbers vs. characters)
- Limit the number of characters in a text area

- Suggest proper format with fixed characters
- Right to Left
- Required or Optional
- Change/Force a Case
- Create multiple logical fields
- Specify range of characters/number for each field With this feature, it is not necessary to:
- Limit the user to a choice of selections
- Handle complex input tasks such as names, days of the week, or month by name
- Perform complex validation such as Subnet Mask validation

Input mask character types

These character types define what information is allowed to be entered in any specific instance. The following table lists what characters in an input mask will define what characters are allowed in any given position.

Character Types	
Character	**Masking Rule**
0	Digit (0 to 9, entry required, plus [+] and minus [-] signs not allowed)
9	Digit or space (entry not required, plus and minus signs not allowed)
#	Digit or space (entry not required; plus and minus signs allowed)
L	Letter (A to Z, entry required)
?	Letter (A to Z, entry optional)
A	Letter or digit (entry required)
a	Letter or digit (entry optional)
&	Any character or a space (entry required)
C	Any character or a space (entry optional)

NOTE: *The number of the above characters used determines the length of the input masking box. Example: 0000 requires an entry, requires digits to be used, and allows only 4 characters to be entered/used.*

Refer to the following SEND_COMMANDs for more detailed information:

- ^BIM- Sets the input mask for the specified addresses.
- ^BMF subcommand %MK - sets the input mask of a text area.

Input Mask Ranges

These ranges allow a user to specify the minimum and maximum numeric value for a field. *Only one range is allowed per field. Using a range implies a numeric entry ONLY*.

Input Mask Ranges	
Character	**Meaning**
[	Start range
]	End range
|	Range Separator

An example from the above table:

[0|255] This allows a user to enter a value from 0 to 255.

Input Mask Operations

Input Mask Operators change the behavior of the field in the following way:

Input Mask Operators	
Character	**Meaning**
<	Forces all characters to be converted to lowercase
>	Forces all characters to be converted to uppercase

Input Mask Literals

To define a literal character, enter any character, other than those shown in the above table *(including spaces, and symbols)*. A back-slash ('\') causes the character that follows it to be displayed as the literal character. For example, **\A** is displayed just as the letter **A**. To define one of the following characters as a literal character, precede that character with a back-slash. Text entry operation using Input Masks.

A keyboard entry using normal text entry is straightforward. However, once an input mask is applied, the behavior of the keyboard needs to change to accommodate the input mask's requirement. When working with masks, any literal characters in the mask will be "skipped" by any cursor movement, including cursor, backspace, and delete keys.

When operating with a mask, the mask should be displayed with placeholders. The "-" character should display where you should enter a character. The arrow keys will move between the "-" characters and allow you to replace them. The text entry code operates as if it is in the overwrite mode. If the cursor is positioned on a character already entered and you type in a new (and valid) character, the new character replaces the old character. There is no shifting of characters.

When working with ranges specified by the [] mask, the keyboard allows you to enter a number between the values listed in the ranges. If a user enters a value that is larger than the maximum, the maximum number of right-most characters is used to create a new, acceptable value.

- **Example 1**: If you type "125" into a field accepting 0-100, then the values displayed will be "1", "12", "25".
- **Example 2**: If the max for the field was 20, then the values displayed will be "1", "12", "5".

When data overflows from a numerical field, the overflow value is added to the previous field on the chain if the overflow character was specified. In the above example, if the overflow flag was set, the first example will place the "1" into the previous logical field and the second example will place "12" in the previous logical field. If the overflow field already contains a value, the new value will be inserted to the right of the current characters and the overflow field will be evaluated. Overflow continues to work until a field with no overflow value is set or no more fields remain (i.e. reached first field).

If a character is typed and that character appears in the Next Field list, the keyboard should move the focus to the next field. For example, when entering time, a ":" is used as a next field character. If you enter "1:2", the 1 is entered in the current field (hours) and then the focus is moved to the next field and 2 is entered in that field.

When entering time in a 12-hour format, entry of AM and PM is required. Instead of adding AM/PM to the input mask specification, the AM/PM should be handled within the NetLinx code. This allows a programmer to show/hide and provide discrete feedback for AM and PM.

Input Mask Output Examples

The following are some common input masking examples

Output Examples		
Common Name	**Input Mask**	**Input**
IP Address Quad	[0|255]{.}	Any value from 0 to 255
Hour	[1|12]{:}	Any value from 1 to 12
Minute/Second	[0|59]{:}	Any value from 0 to 59
Frames	[0|29]{:}	Any value from 0 to 29
Phone Numbers	(999) 000-0000	(555) 555-5555
Zip Code	00000-9999	75082-4567

URL Resources

A URL can be broken into several parts. For example, with the URL *http://www.amx.com/company-info-home.asp*, this URL indicates that the protocol in use is **http** (HyperText Transport Protocol) and that the information resides on a host machine named **www.amx.com**. The image on that host machine is given an assignment (by the program) name of **company-info-home.asp** *(Active Server Page)*.

The exact meaning of this name on the host machine is both protocol dependent and host dependent. The information normally resides in a file, but it could be generated dynamically. This component of the URL is called the file component, even though the information is not necessarily in a file.

A URL can optionally specify a port, which is the port number to which the TCP/IP connection is made on the remote host machine. If the port is not specified, the default port for the protocol is used instead. For example, the default port for http is 80. An alternative port could be specified as: *http://www.amx.com:8080/company-info-home.asp*. **NOTE**: *Any legal HTTP syntax can be used.*

Special Escape Sequences

The system has only a limited knowledge of URL formats, as it transparently passes the URL information onto the server for translation. A user can then pass any parameters to the server side programs such as CGI scripts or active server pages. However; the system will parse the URL looking for special escape codes. When it finds an escape code, it replaces that code with a particular piece of panel, button, or state information.

For example, "http://www.amx.com/img.asp?device=$DV" would become *http://www.amx.com/img.asp?device=10001*. Other used escape sequences include:

Escape Sequences	
Sequence	**Panel Information**
$DV	Device Number

$SY	System Number
$IP	IP Address
$HN	Host Name
$MC	Mac Address
$PX	X Resolution of current panel mode/file
$PY	Y Resolution of current panel mode/file
$BX	X Resolution of current button
$BY	Y Resolution of current button
$BN	Name of button
$ST	Current state
$AC	Address Code
$AP	Address Port
$CC	Channel Code
$CP	Channel Port
$LC	Level Code
$LP	Level Port

Appendix D: Bargraph Functions

Overview

For drag operations on Bargraph and Multi-State Bargraph buttons, each movement increments based on the drag increment field. For centering, the bargraph/multistate bargraph will return to the middle - either the 50% mark for bargraphs, or the median state number, once the touch point is released.

Setup Codes

Bargraph Functions - Setup Codes		
Code	**Code**	**Description**
Channel	2	Panel Setup:Brightness Up
Channel	3	Panel Setup: Brightness Down
Channel	6	Panel Setup: Controller Volume Up
Channel	7	Panel Setup: Controller Volume Down
Channel	8	Panel Setup: Controller Volume Mute
Channel	158	Panel Setup: Mic Volume Mute
Channel	171	Panel Setup: Call Volume Up
Channel	172	Panel Setup: Call Volume Down
Channel	1403	Panel Setup: Notification Alarm Volume Mute
Channel	1404	Panel Setup: Notification Volume Up
Channel	1405	Panel Setup: Notification Volume Down
Channel	1407	Panel Setup: Alarm Volume Up
Channel	1408	Panel Setup: Alarm Volume Down
Address	33	Panel Setup: Brightness
Address	35	Panel Setup: Controller Volume
Address	144	Time Display: AM|PM
Address	46	Panel Setup: Call Volume
Address	450	Panel Setup: Notification Volume
Address	451	Panel Setup: Alarm Volume
Level	1	Panel Setup: Brightness
Level	3	Panel Setup: Controller Volume
Level	9	Panel Setup: Call Volume
Level	450	Panel Setup: Notification Volume
Level	451	Panel Setup: Alarm Volume

Appendix E: Video Streaming

Optimizing Motion JPEG Video Presentation and Speed

In some cases, multiple Motion JPEG streams may slow presentation of individual screen popups, or prevent all of the streams from showing at the same time. This may happen even though the Panel Preview in TPDesign 5 may show no issues. To minimize this and assure a smooth and non-sluggish stream, try these options:

- Limit the number of simultaneous Motion JPEG streams to eight or fewer streams at a time.
- Remove any unnecessary buttons associated with the Motion JPEG streams.
- Make sure that the Refresh rate on a Motion JPEG is set to 0.
- Make sure to hide the preview popup before displaying the full image.
- If possible, uncheck the “Scale to Fit” option, as scaling is very resource-intensive.
- Dial down the frame rate of the server. The frame rate of a Motion JPEG is determined by the server.
- When you go from a page with multiple previews to a page with a single full screen video, it is best to do a page flip rather than popup attach, or hide the preview windows first. Otherwise, the preview windows will continue to decode (taxing the system), even though they may be completely or partially obstructed by the popup.
- Verify that the full-screen image is set for acceleration by checking the “Dynamo” box in Resource Manager.”

Motion JPEG Support for VARIA Panels	
Baseline mode:	ISO 10918-1
Encoding:	ISO-10918-5 (JFIF)
Maximum Resolution:	720p
Recommended resolution:	720x480-NTSC or 720x576-PAL (or less). If the video is defined in the Resource Manager as opposed to video fill, consideration must be made for the video being decoded by the panel, which cannot decode 720p.
Maximum Frame Rate:	Up to 30fps
Latency:	From 1-3 seconds, depending on multiple factors including button size, resolution and network performance.

Streaming a Video File Saved on the Panel via Custom URL Scheme

To use a custom URL scheme and File Transfer (in NetLinx Studio) to play a video stored in the G5 touch panel’s internal storage:

1. In NetLinx Studio **4**, select **Tools -> File Transfer** to open the *File Transfer* dialog - *Send* tab (FIG. 164):

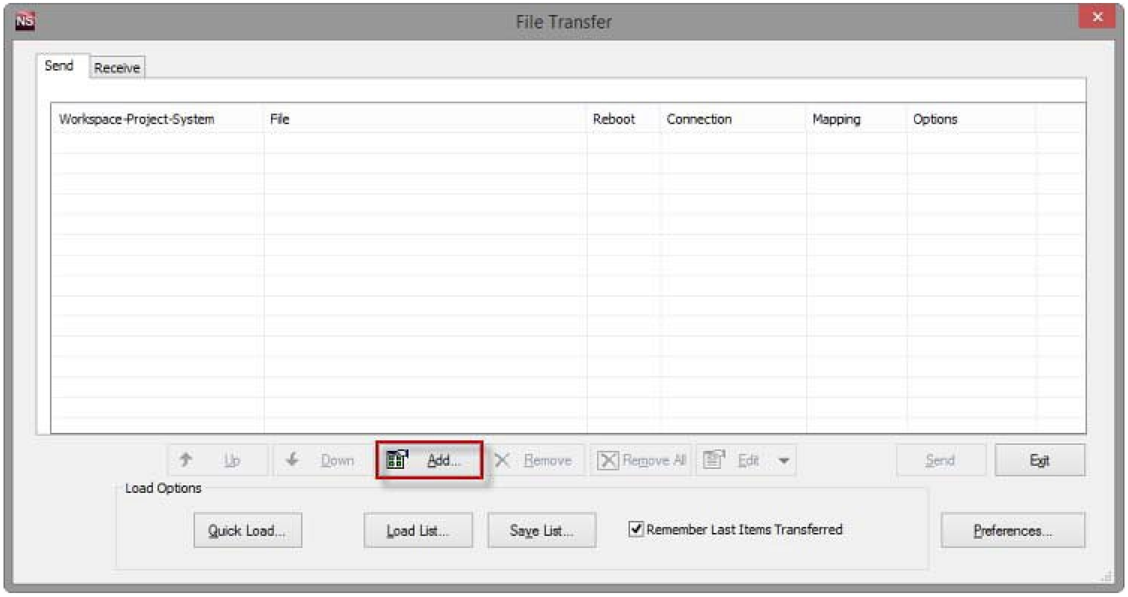
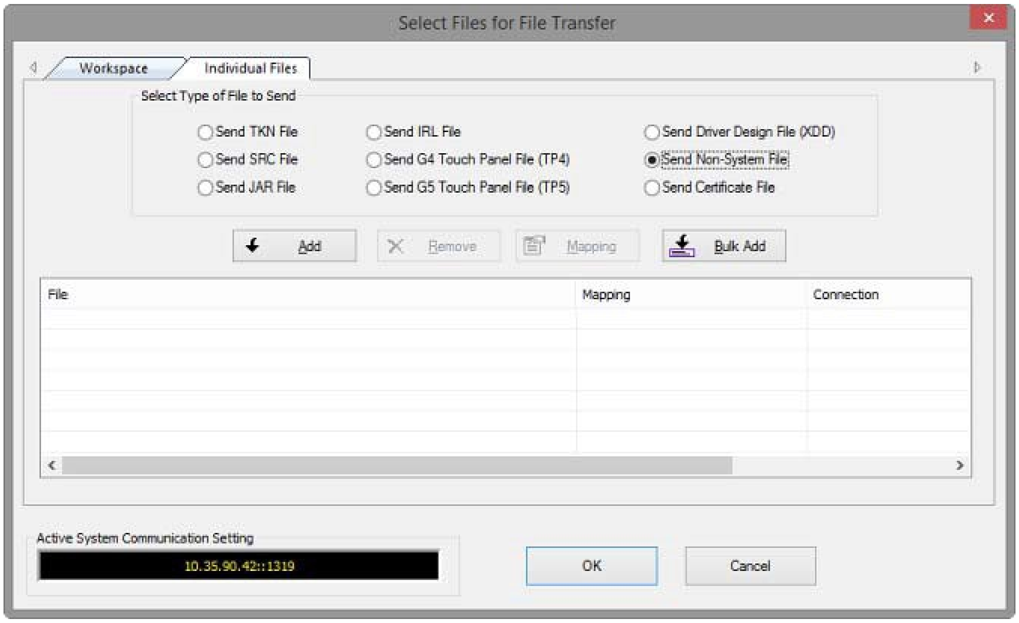


Fig. 164 NetLinx Studio 4 - File Transfer dialog

2. Click **Add** to open the Select *Files for File Transfer* dialog, open the *Individual Files* tab and select **Send Non-System File** (FIG. 165):



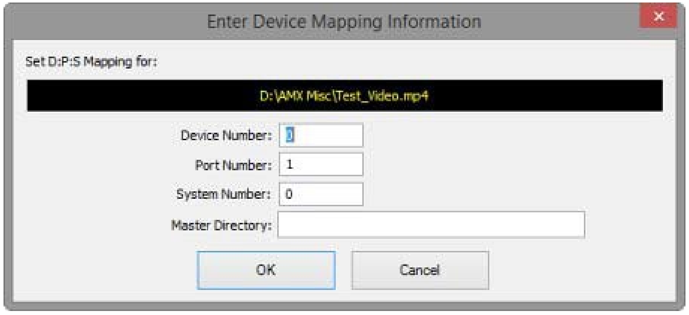
Select Type of File to Send

○ Send TKN File
○ Send SRC File
○ Send JAR File
○ Send IRL File
○ Send G4 Touch Panel File (TP4)
○ Send G5 Touch Panel File (TP5)
○ Send Driver Design File (XDD)
● Send Non-System File
○ Send Certificate File

Active System Communication Setting: 10.35.90.42::1319

Fig. 165 NetLinx Studio 4 - Select Files for File Transfer dialog (Individual Files tab)

3. Click **Add** to select the video file you want to use: Select the video file in the *Open* dialog and click **OK** to invoke the *Enter Device Mapping Information* dialog (FIG. 166):



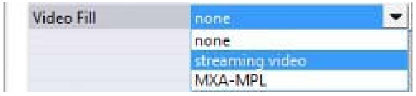
Set D:P:S Mapping for:

D:\AMX Misc\Test_Video.mp4

Device Number: 0
Port Number: 1
System Number: 0
Master Directory:

Fig. 166 NetLinx Studio 4 - Enter Device Mapping Information dialog

4. Enter device mapping information (D:P:S) for the target G5 panel Leave the *Controller Directory* field blank.
5. Click **OK** to save changes and close the *Enter Device Mapping Information* dialog.
6. Click **OK** to close the *Select File For File Transfer* dialog.
7. Click **Send** in the *File Transfer* dialog to transfer the file (this may take time for large video files).
8. In TPDesign5, select the page/button state you want to play the video file.
9. In the desired state tab, set the *Video Fill* property to **streaming video** (FIG. 167). Note that this selection enables the *Streaming Source* property.

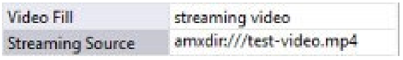


Video Fill: none

none
streaming video
MXA-MPL

Fig. 167 TPDesign5 - Video Fill (State) property

10. For the Streaming Source property, enter the filename of the video file with **amxdir:///** as the prefix. For example, if the video filename is "test-video.mp4" then enter the Streaming Source as "*amxdir:///test-video.mp4*" (FIG. 168):



Video Fill	streaming video
Streaming Source	amxdir:///test-video.mp4

Fig. 168 TPDesign5 - Streaming Source (State) property

NOTE: *There are three slashes after amxdir:, not two as in a standard URL. If there aren't three slashes the video file won't be found.*

11. Load the TP5 file on the panel and the desired state should continually play the video.
If you desire to change the video using the ^SDM command to another that you have transfered, use the same URL scheme as the prefix (**amxdir:///**).
Any file that is transfered to the amxdir:/// directory is not cleared by a panel file transfer or "Remove User Pages". The only way to transfer is to do a Factory Data Reset, or to upload an empty file with the same filename.

To get around this, you can specify a file to be **amxdir:///AMXPanel/images/filename** instead.

To do this using NetLinx Studio File Transfer, set the "Controller Directory" to \AMXPanel\images\ in the device mapping. This will put the file in the panel file images directory. A TP5 file transfer will not remove the file, but a "Remove User Pages" will. The Streaming Source value in the TP5 file would have to correspond to the same path.

NOTE: *See details on the ^SDM Button State Streaming Digital Media command*.

Transcoding Guidelines

For certain H.264 video and audio streaming , you may observe a drift between audio and video the longer the content is streamed. This drift can be more pronounced when streaming from a non- MXA-MPL source such as a Vision 2 steaming server. If the panel detects excessive drift, it will attempt to restart the stream decode. During the restart, the audio will be temporarily interrupted and the video will be frozen on the last frame until the restart is complete (typically a couple of seconds). To reduce the drift issue for Vision 2 H264 steaming, video transcoding tools (such as HandBrake or FFMPEG) are available to convert H.264 video into lower bitrates, reduced resolution and/or lower H.264 profiles. For example you can try the H.264, 2mbps bit rate, 480p resolution, Baseline profile. If this does not work, try transcoding the stream into MPEG2 video, which is less susceptible to A/V drift.

NOTE: *Third-party encoders and digital television devices have not been tested with VARIA touch panels, and are not supported by AMX.*

The table below lists the typical synchronization and latency times for each supported video and audio stream:

Video Performance					
Device	**Typical A/V Sync (offset/ hr)**	**Typical A/V Sync Restart Rate**	**Expected Latency Typical**	**Expected Latency - Max**	**Notes:**
3rd Party Solutions					
H.264	N/A	N/A	N/A	N/A	Third-party encoders and digital television devices have not been tested with Varia touch panels, and are not supported by AMX. Network congestion can cause video glitches. We recommend the panel be installed behind a smart Ethernet switch to filter unintended multicast packets reaching the panel and consuming panel resources. We recommend maintaining aspect ratio of source and following usage guidelines regarding window/button placement.
MPEG2	N/A	N/A	N/A	N/A	Third-party encoders and digital television devices have not been tested with Varia touch panels, and are not supported by AMX. Network congestion can cause video glitches. We recommend the panel be installed behind a smart Ethernet switch to filter unintended multicast packets reaching the panel and consuming panel resources. We recommend maintaining aspect ratio of source and following usage guidelines regarding window/button placement.

Appendix F: Modern Authentication

Steps that are required to configure Modern Authentication (OAuth 2) support for **AMX Book** using Microsoft Graph API.

Pre-requisites:

- An Azure account with an active subscription.
- An Azure AD tenant.
- Please note that some fo the screenshots are extracted from the RMS manual and as such naming of the app registrations may reflect RMS and not AMX Book. The process and result are the same regardless.

I. Self-signed Certificate creation

In order to use certificate as credential for the Application, you need to generate a certificate and upload it to Azure AD.

1. Open **Windows PowerShell**

2. Execute the below given command to create a Self-signed Certificate:

```
New-SelfSignedCertificate -CertStoreLocation "Cert:\LocalMachine\My" -Subject
"CN=<ApplicationName>"
```

NOTE: *Replace the ApplicationName with the name of your application that you create on Azure AD.*

II. Export Certificate

Export the Self-signed certificate to use with the application with **Certificate Manager** as explained below:

1. Execute **mmc** command in the **Command Prompt** / **Windows PowerShell**.
2. Open **File** > **Add/Remove Snap-In**.
3. Add the **Certificates** snap-in from the **Available snap-ins** pane to the **Selected snap-ins** pane.
4. Select the **Computer Account** radio button in the **Certificates snap-in** dialog box and click **Next**
5. Select the **Local Computer** radio button in the **Select Computer** dialog box and then click **Finish**
6. Click **OK**
7. Select **Console Root > Certificates (Local Computer) > Personal > Certificates** from the tree view on the left pane.
8. You will see the newly generated application certificate

a. Export Certificate - Without Private Key (.CER)

We need to export the certificate without Private key and Base-64 Encoded X.509 .CER format to upload in Azure AD.

1. Select the application certificate and right click
2. Select **All tasks > Export** option
3. **Certificate Export Wizard** opens.
4. Click **Next** button
5. Select the **No, do not export the private key** radio button and click **Next**
6. Select the **Base-64 encoded X.509 (.CER)** radio button and click **Next**
7. Click **Browse** button and select a folder to place the exported file
8. Key in the **File name** and click **Save** button
9. Click **Next** button
10. You will see **You have successfully completed the Certificate Export Wizard**
11. Click **Finish** button
12. Click **OK** button

b. Export Certificate - With Private Key (.PFX)

We need to export the certificate with Private key and PKCS #12 (.PFX) format to be loaded in RMS EWS Scheduler Interface application.

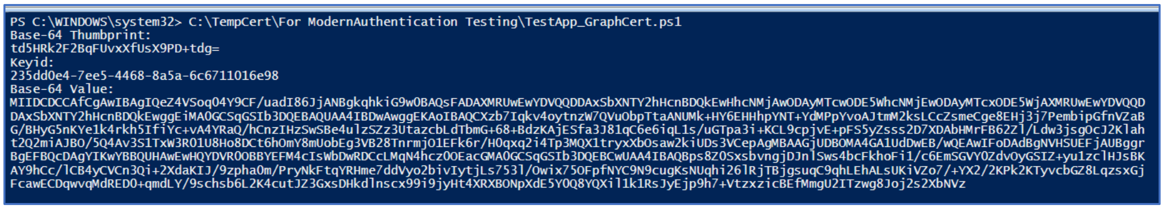
1. Select the application certificate and right click
2. Select **All tasks > Export** option
3. **Certificate Export Wizard** opens.
4. Click **Next** button
5. Select the **Yes, export the private key** radio button and click **Next**
6. Select the **Personal Information Exchange – PKCS #12 (.PFX)** radio button and click **Next**
7. For **Security,** select the **Password** checkbox
8. Key-in the password and repeat the same in **Confirm password**
9. Accept the default option in **Encryption** dropdown and click **Next** button
10. Click **Browse** button and select a folder to place the exported file
11. Key in the **File name** and click **Save** button
12. Click **Next** button
13. You will see **You have successfully completed the Certificate Export Wizard**
14. Click **Finish** button
15. Click **OK** button

III. Generate the keyCredentials value using PowerShell

Run the following **PowerShell** script to generate the keyCredential value based on your **Base-64 encoded** certificate file.

This will get the values that are to be placed in Azure AD App Manifest.

```
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2

$Cert.Import("C:\RmsSchCert\TestApp_GrpCBA.cer")

$BinaryData = $Cert.GetRawCertData()
$Base64Value = [System.Convert]::ToBase64String($BinaryData)

$BinaryData = $Cert.GetCertHash()
$Base64Thumbprint = [System.Convert]::ToBase64String($BinaryData)

$Keyid = [System.Guid]::NewGuid().ToString()

Write-Host "Base-64 Thumbprint:"
$Base64Thumbprint

Write-Host "Keyid:"
$Keyid

Write-Host "Base-64 Value:"
$Base64Value
```



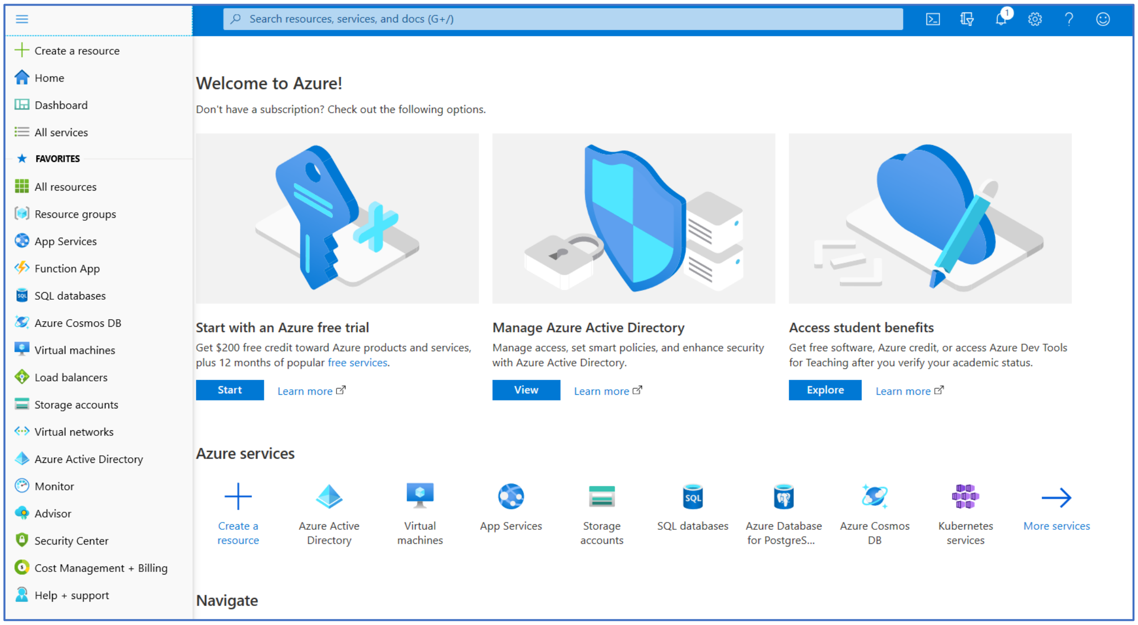
```
PS C:\WINDOWS\system32> C:\TempCert\For ModernAuthentication Testing\TestApp_GraphCert.ps1
Base-64 Thumbprint:
td5HRk2F2BqFUvxXfUsX9PD+tdg=
Keyid:
235dd0e4-7ee5-4468-8a5a-6c6711016e98
Base-64 Value:
MIIDCDCCAfCgAwIBAgIQeZ4VSoq04Y9CF/uadI86JjANBgkqhkiG9w0BAQsFADAXMRUwEwYDVQQDDAxSbXNTY2hHcnBDQkEwHhcNMjAwODAyMTcwODE5WhcNMjEwODAyMTcxODE5WjAXMRUwEwYDVQQD
DAxSbXNTY2hHcnBDQkEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCXzb7Iqkv4oytnzW7QVuObpTtaANUMk+HY6EHHhpYNT+YdMPpYvoAJtmM2ksLCcZsmeCge8EHj3j7PembipGfnVZaB
G/BHyG5nKYe1k4rkh5IfiYc+vA4YRaQ/hCnzIHzSwSBe4ulzSZz3UtazcbLdTbmG+68+BdzKAjESfa3J81qC6e6iqL1s/uGTpa3i+KCL9cpjvE+pFS5yZsss2D7XDAbHMrFB62Zl/Ldw3jsgOcJ2Klah
t2Q2miAJBO/5Q4Av3S1TxW3R01U8Ho8DCt6hOmY8mUobEg3VB28TnrmjO1EFk6r/H0qxq2i4Tp3MQX1tryxXbOsaw2kiUDs3VCepAgMBAAGjUDBOMA4GA1UdDwEB/wQEAwIFoDAdBgNVHSUEFjAUBggr
BgEFBQcDAgYIKwYBBQUHAwEwHQYDVR0OBBYEFM4cIsWbDwRDCcLMqN4hcz0OEacGMA0GCSqGSIb3DQEBCwUAA4IBAQBps8Z0SxsbvngjDJnlSws4bcFkhoFi1/c6EmSGVY0ZdvOyGSIZ+yu1zclHJsBK
AY9hCc/lCB4yCVCn3Qi+2XdaKIJ/9zpha0m/PryNkFtqYRHme7ddVyo2bivIytjLs753l/Owix75OFpfNYC9N9cugKsNUqhi26lRjTBjgsuqC9qhLEhALsUKiVZo7/+YX2/2KPk2KTyvcbGZ8LqzsxGj
FcawECDqwvqMdRED0+qmdLY/9schsb6L2K4cutJZ3GxsDHkdlnscx99i9jyHt4XRXBONpXdE5YOQ8YQXil1k1RsJyEjp9h7+VtzxzicBEfMmgU2ITzwg8Joj2s2XbNVz
```

Copy the value from the PowerShell output terminal in to a Text Editor. We will be using **Base-64 Thumbprint** and **Keyid** values in section *Modify the Manifest of Azure AD Application*

IV. Create your App registration in Azure Active Directory

Access the Azure Portal (https://portal.azure.com) and sign-in with the user ID that has the **Global Administrator** rights

From the Portal Menu, select the **Azure Active Directory** option



Welcome to Azure!

Don't have a subscription? Check out the following options.

Start with an Azure free trial

Get $200 free credit toward Azure products and services, plus 12 months of popular free services.

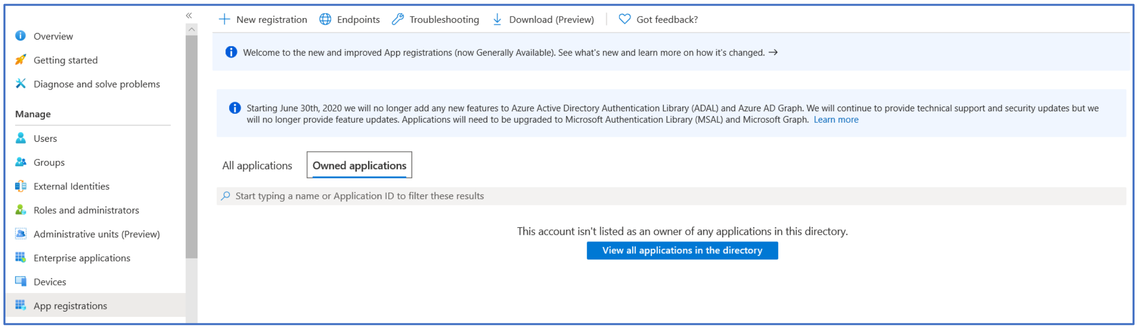
Manage Azure Active Directory

Manage access, set smart policies, and enhance security with Azure Active Directory.

Access student benefits

Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

In the left navigation menu, select **App registrations** from **Manage** section of **Active Directory Menu Blade**



Welcome to the new and improved App registrations (now Generally Available). See what's new and learn more on how it's changed. →

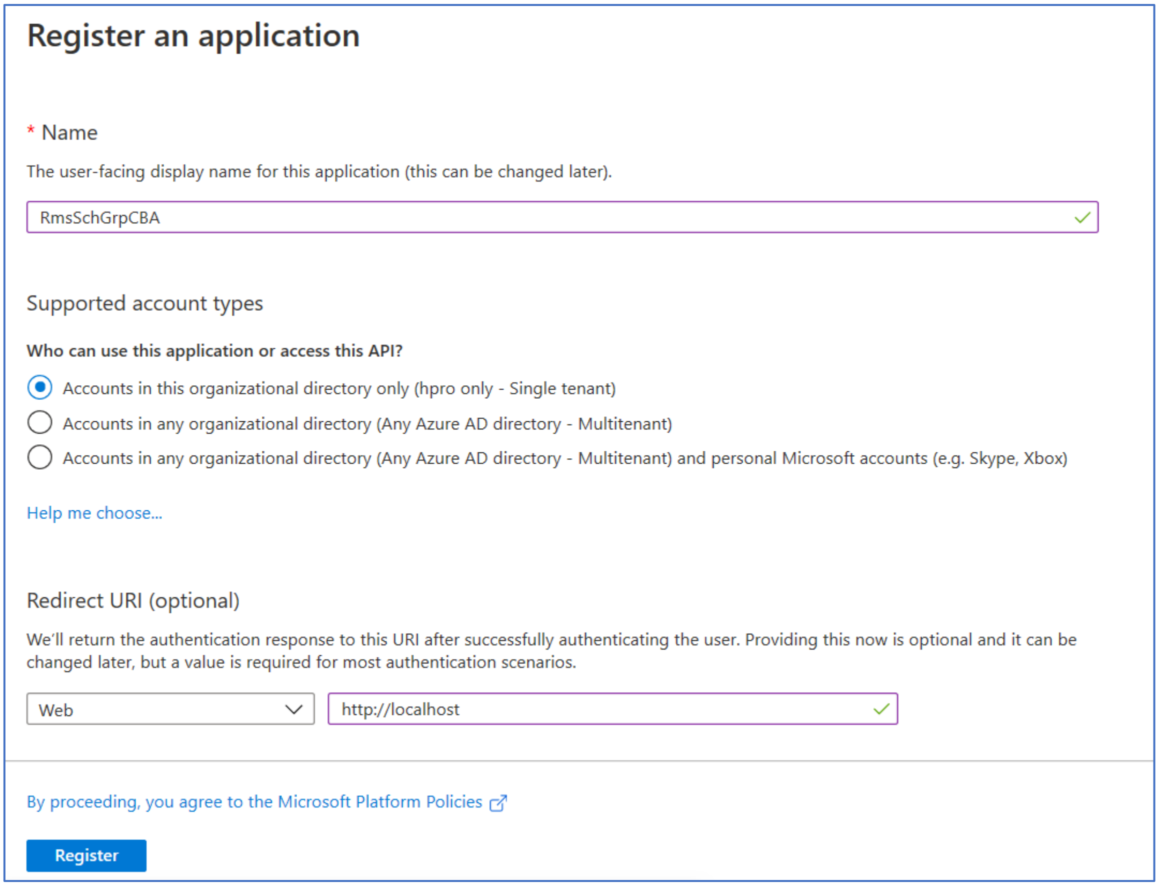
Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

This account isn't listed as an owner of any applications in this directory.

Click on the **New registration** option from the top right section.

In the ‘Register an application’ page:

a. Fill in **Name** for the application.
b. Select the **Supported account types** as “Accounts in this organization directory only”
c. Set the **Redirect URI (optional)** as “Web” and URI as http://localhost



Register an application

* Name

The user-facing display name for this application (this can be changed later).

RmsSchGrpCBA ✓

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (hpro only - Single tenant)
○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)
○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

Help me choose...

Redirect URI (optional)

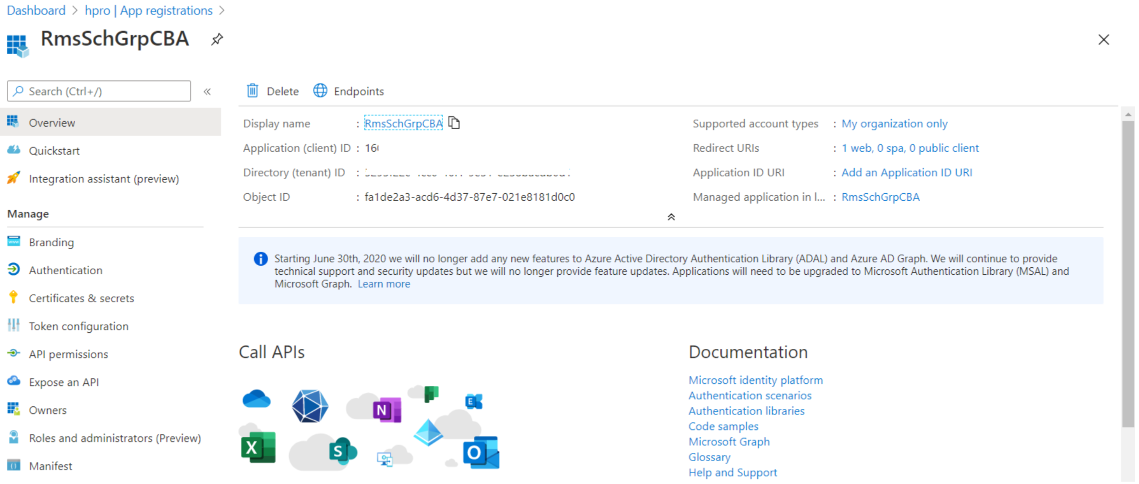
We’ll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web | http://localhost ✓

By proceeding, you agree to the Microsoft Platform Policies

Click **Register** button.

Overview page opens for the newly created application



Display name : RmsSchGrpCBA
Application (client) ID : 16
Directory (tenant) ID : [illegible]
Object ID : fa1de2a3-acd6-4d37-87e7-021e8181d0c0
Supported account types : My organization only
Redirect URIs : 1 web, 0 spa, 0 public client
Application ID URI : Add an Application ID URI
Managed application in l... : RmsSchGrpCBA

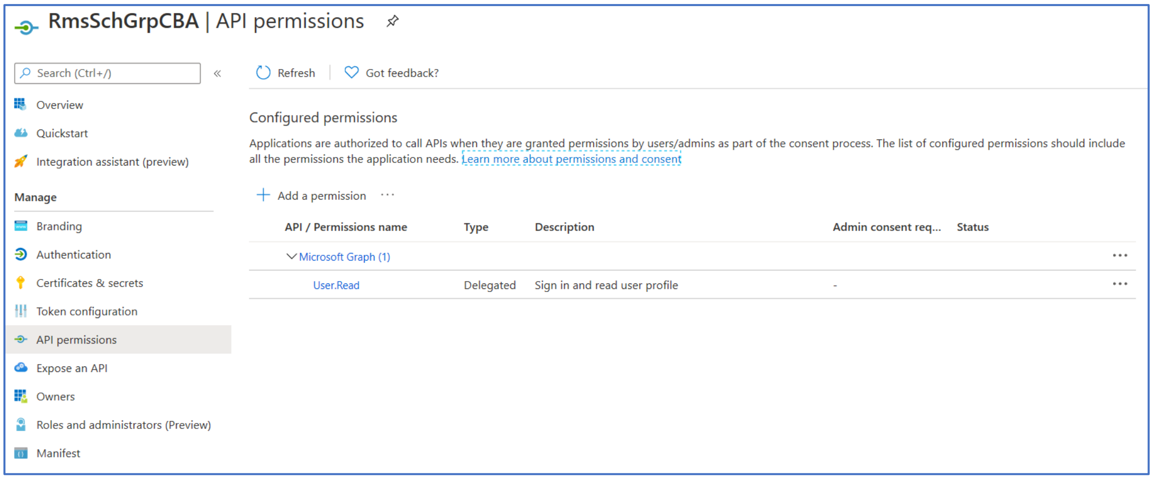
Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

Make note of the **Tenant ID**, **Client ID** values. We will be using them while configuring the **AMX Book Calendar** for Modern Authentication

V. Add Graph API permissions to the app

In the left navigation menu, select **API Permissions** from **Manage** section of **Application Menu Blade**

API Permissions page opens



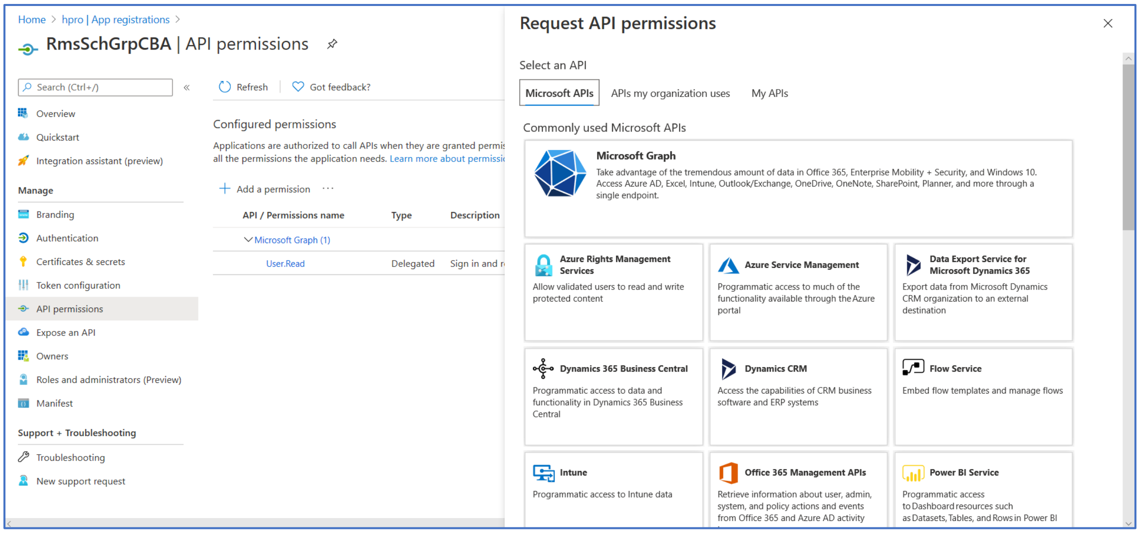
Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	-	

Click on the **Add a permission** option from the **Configured permissions** section.

Request API permissions page opens



Configured permissions

Applications are authorized to call APIs when they are granted permis
all the permissions the application needs. Learn more about permissio

API / Permissions name	Type	Description
Microsoft Graph (1)		
User.Read	Delegated	Sign in and r

Commonly used Microsoft APIs

Microsoft Graph
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

Azure Rights Management Services
Allow validated users to read and write protected content

Azure Service Management
Programmatic access to much of the functionality available through the Azure portal

Data Export Service for Microsoft Dynamics 365
Export data from Microsoft Dynamics CRM organization to an external destination

Dynamics 365 Business Central
Programmatic access to data and functionality in Dynamics 365 Business Central

Dynamics CRM
Access the capabilities of CRM business software and ERP systems

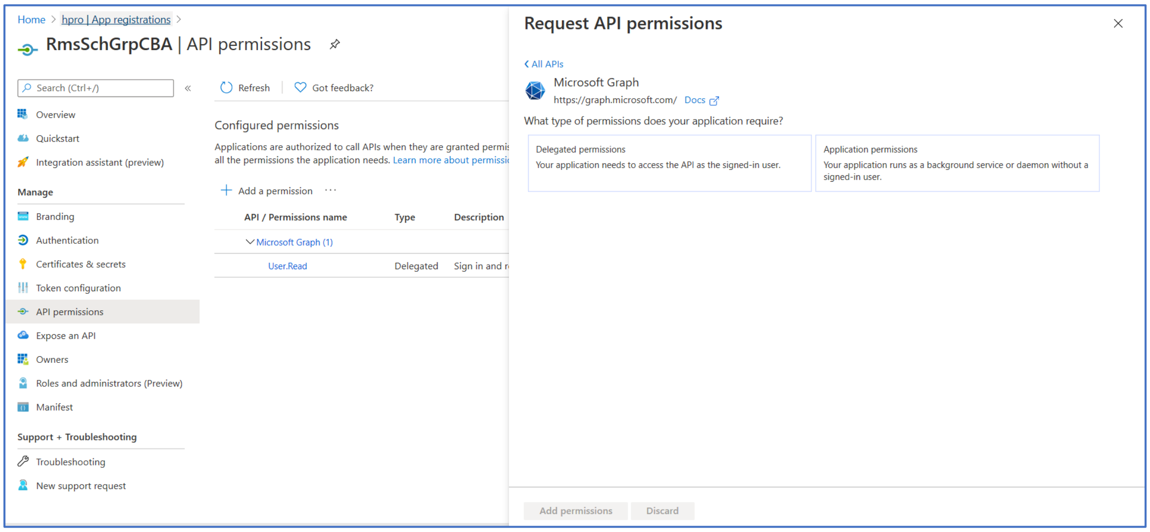
Flow Service
Embed flow templates and manage flows

Intune
Programmatic access to Intune data

Office 365 Management APIs
Retrieve information about user, admin, system, and policy actions and events from Office 365 and Azure AD activity

Power BI Service
Programmatic access to Dashboard resources such as Datasets, Tables, and Rows in Power BI

Click on **Microsoft Graph** option from the **Commonly used Microsoft APIs** section



Configured permissions

Applications are authorized to call APIs when they are granted permis
all the permissions the application needs. Learn more about permissio

API / Permissions name	Type	Description
Microsoft Graph (1)		
User.Read	Delegated	Sign in and r

Microsoft Graph
https://graph.microsoft.com/

What type of permissions does your application require?

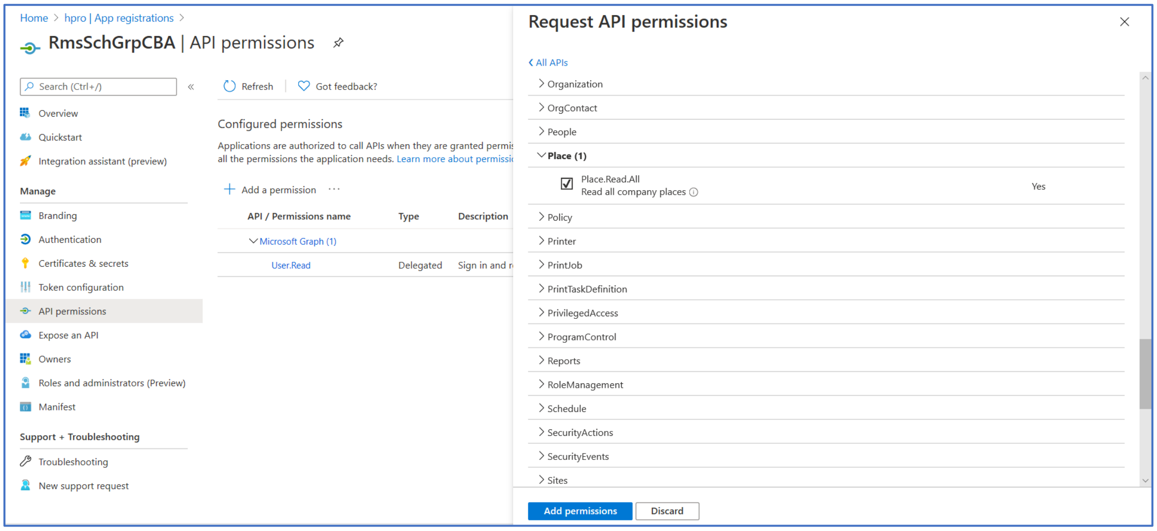
Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select the **Application Permissions** option for ‘What type of permissions does your application require?’

Select the following Permissions:

Calendars > Calendars.ReadWrite
Mail > Mail.ReadWrite
Place > Place.Read.All



Configured permissions

Applications are authorized to call APIs when they are granted permis all the permissions the application needs. Learn more about permissi

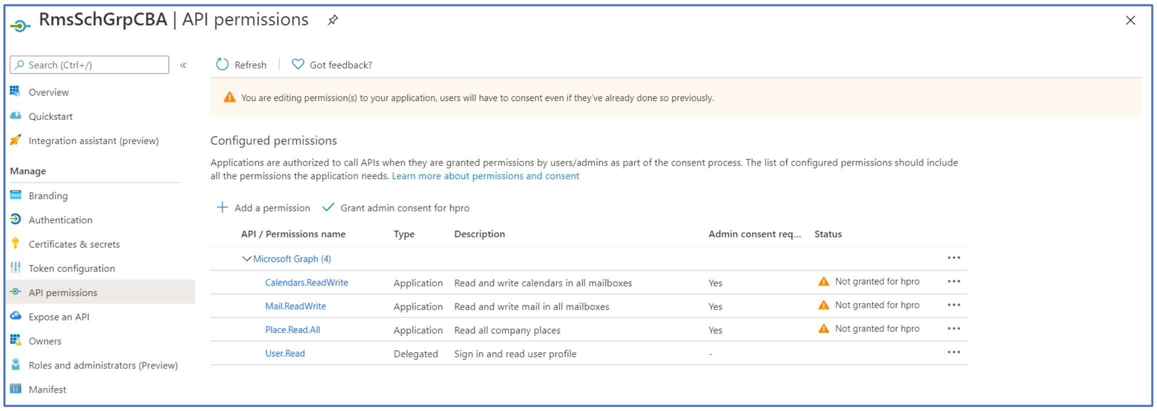
API / Permissions name	Type	Description
Microsoft Graph (1)		
User.Read	Delegated	Sign in and r

Place (1)

☑ Place.Read.All — Read all company places — Yes

Click **Add Permissions** button

The newly added permissions are saved and listed in **Configured permissions**



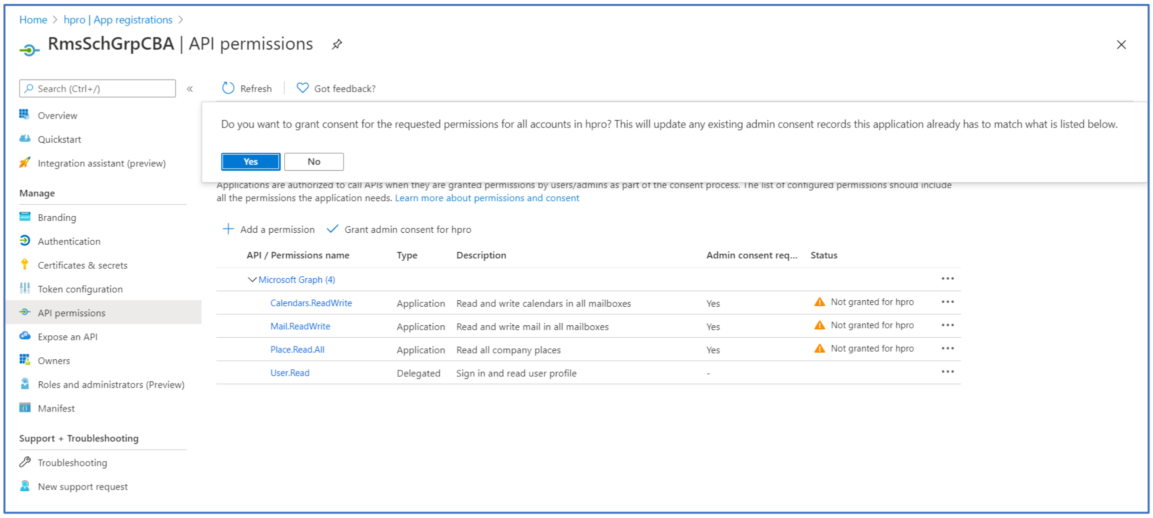
⚠ You are editing permission(s) to your application, users will have to consent even if they've already done so previously.

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (4)				
Calendars.ReadWrite	Application	Read and write calendars in all mailboxes	Yes	⚠ Not granted for hpro
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	⚠ Not granted for hpro
Place.Read.All	Application	Read all company places	Yes	⚠ Not granted for hpro
User.Read	Delegated	Sign in and read user profile	-	

Now, Grant admin consent for the requested permissions by clicking **Grant admin consent for <domain name>** button



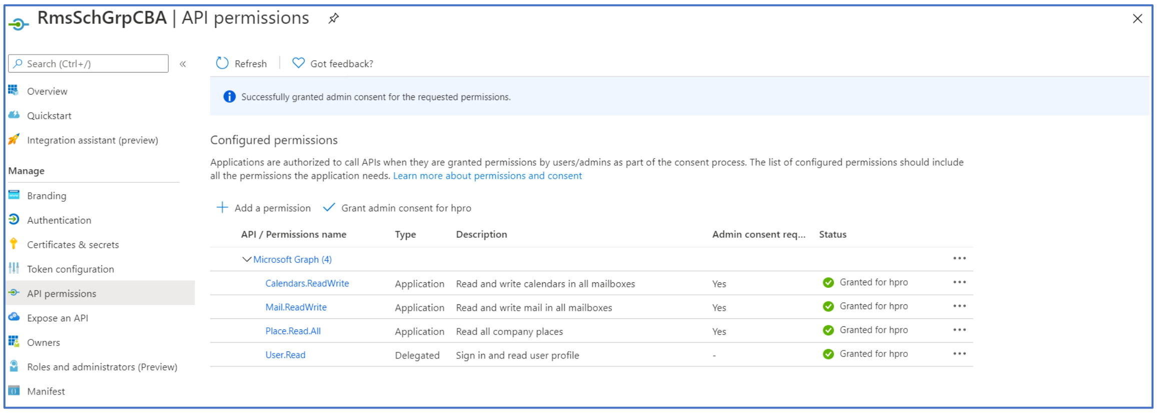
Do you want to grant consent for the requested permissions for all accounts in hpro? This will update any existing admin consent records this application already has to match what is listed below.

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (4)				
Calendars.ReadWrite	Application	Read and write calendars in all mailboxes	Yes	Not granted for hpro
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	Not granted for hpro
Place.Read.All	Application	Read all company places	Yes	Not granted for hpro
User.Read	Delegated	Sign in and read user profile	-	

Click **Yes** button for the confirmation message

Notice that the **Admin consent** granted for the requested permissions



Successfully granted admin consent for the requested permissions.

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (4)				
Calendars.ReadWrite	Application	Read and write calendars in all mailboxes	Yes	Granted for hpro
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	Granted for hpro
Place.Read.All	Application	Read all company places	Yes	Granted for hpro
User.Read	Delegated	Sign in and read user profile	-	Granted for hpro

VI. Application Credentials

Credentials enable applications to identify themselves to the authentication service when receiving tokens.

You can use either **Client Secret** or **Certificate** to achieve this, but not both. Section VII and VIII describes both methods. **Make sure to use any one.**

VII. Adding Credentials to the Application (Client Secret)

Important NOTE:

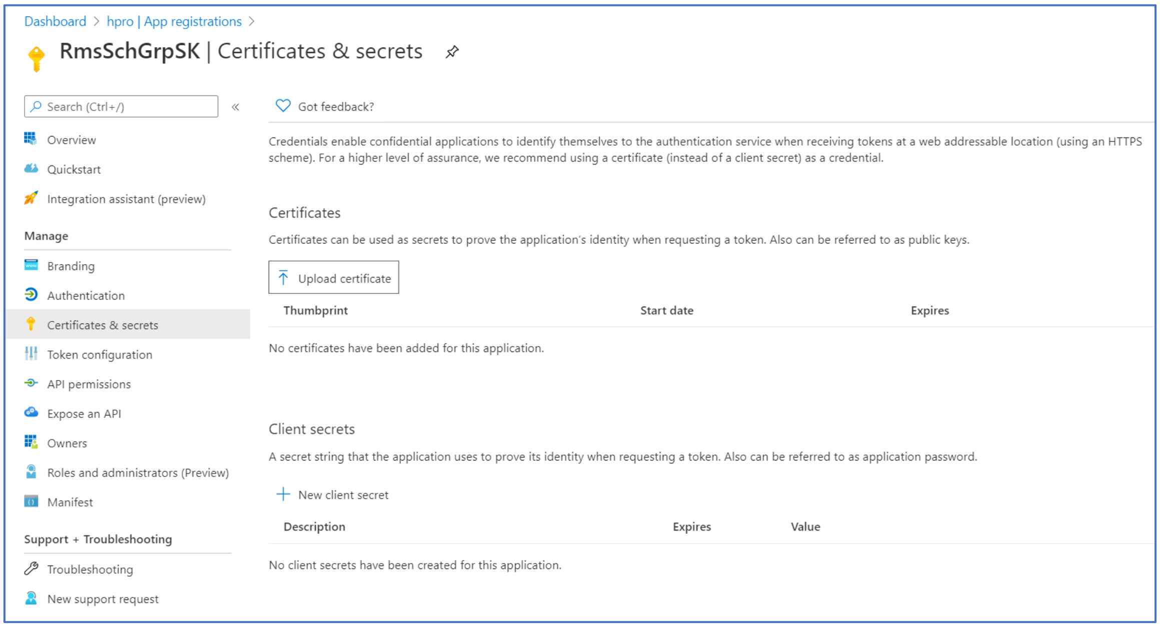
Skip this step and go to Adding Credentials to the Application (Client Certificate) section *below to use Certificate based authentication*

For a higher level of assurance, recommended to use a certificate (instead of a client secret) as a credential.

Make sure to create a new application by following the steps defined in Create your app registration in Azure Active Directory

In the left navigation menu, select **Certificates & Secrets** from **Manage** section of **Application Menu Blade**

Certificates & Secrets page opens



Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Start date	Expires

No certificates have been added for this application.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value

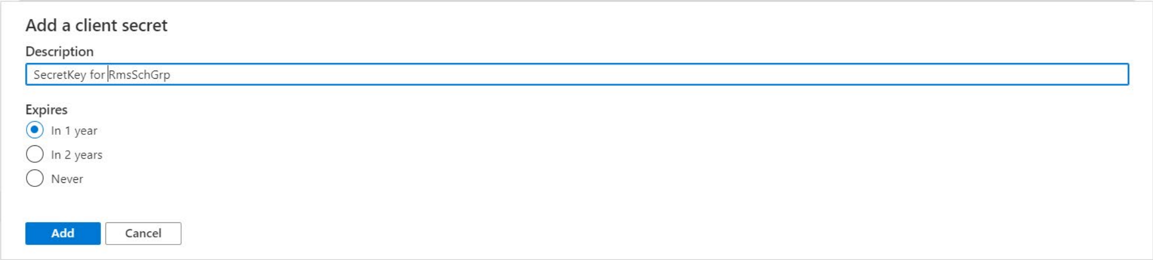
No client secrets have been created for this application.

Click **New Client secret** button

Add a client secret dialog opens

Key-in **Description** for the client secret

Select a value from the **Expires** section as per your need



Add a client secret

Description

SecretKey for RmsSchGrp

Expires

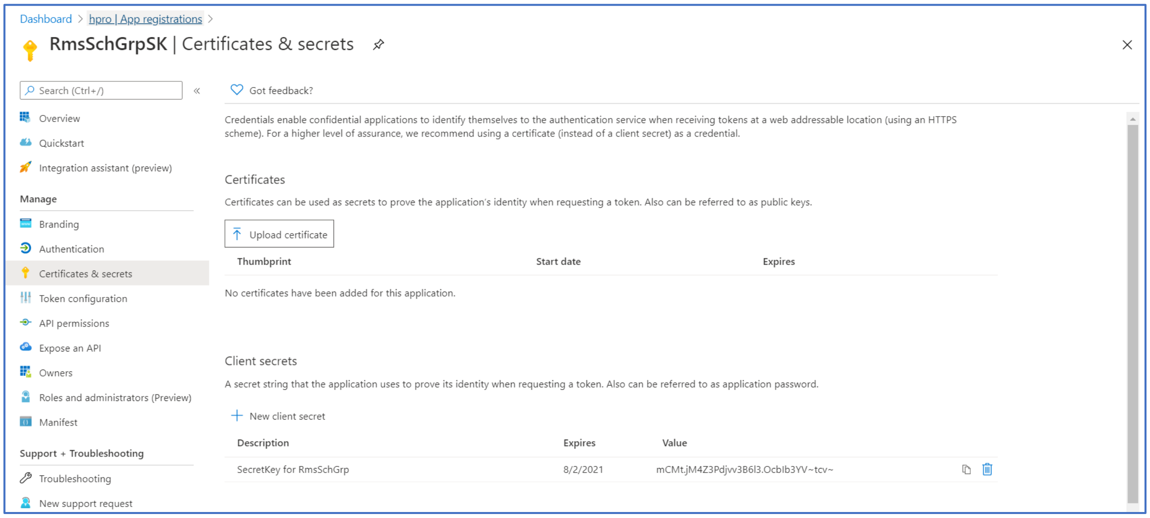
● In 1 year

○ In 2 years

○ Never

Click **Add** button

Once the settings are saved, the key is displayed.



RmsSchGrpSK | Certificates & secrets

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Start date	Expires

No certificates have been added for this application.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value
SecretKey for RmsSchGrp	8/2/2021	mCMt.jM4Z3Pdjvv3B6l3.Ocblb3YV~tcv~

<u>Important NOTE:</u>

Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this screen.

VIII. Adding Credentials to the Application (Client Certificate)

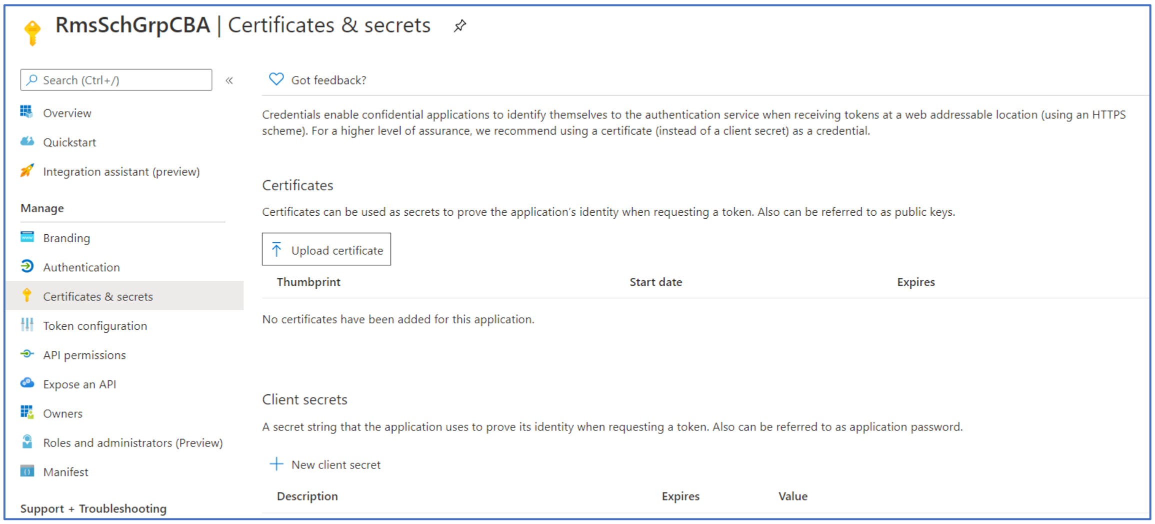
Important NOTE:

If you have already configured [Adding Credentials to the Application (Client Secret)](), Please skip this step.

Make sure to create a new application by following the steps defined in [Create your app registration in Azure Active Directory]()

In the left navigation menu, select **Certificates & Secrets** from **Manage** section of **Application Menu Blade**

Certificates & Secrets page opens



Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Start date	Expires
No certificates have been added for this application.		

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value

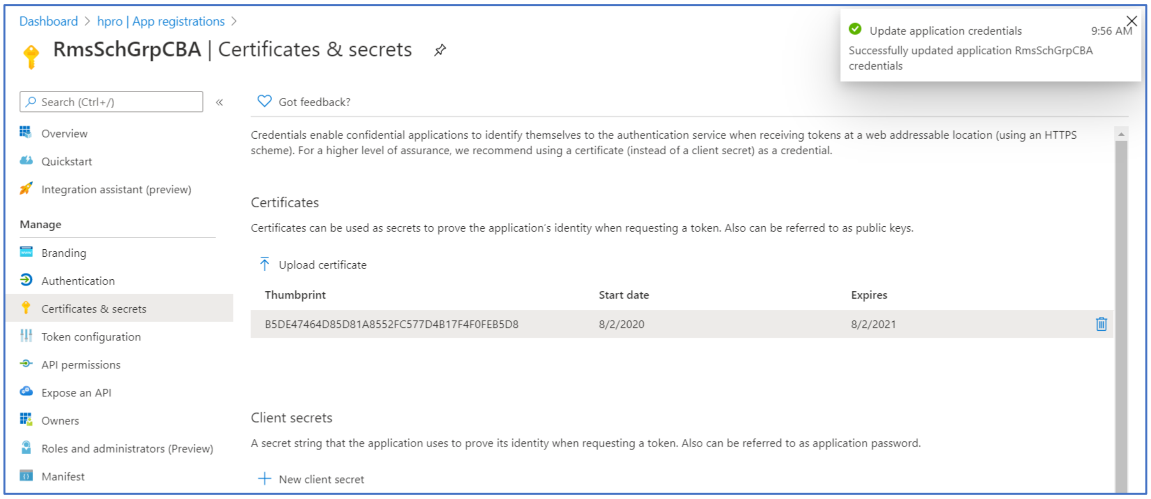
Click **Upload certificate** button

Upload certificate dialog opens. Click on the **Folder** icon to select a certificate file

Select the **.CER** file that exported in section *[Export Certificate - Without Private Key (.CER)]()*

Click **Add** button

Notice that the uploaded certificate listed under **Certificates** section



Update application credentials 9:56 AM
Successfully updated application RmsSchGrpCBA credentials

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

Thumbprint	Start date	Expires
B5DE47464D85D81A8552FC577D4B17F4F0FEB5D8	8/2/2020	8/2/2021

Client secrets

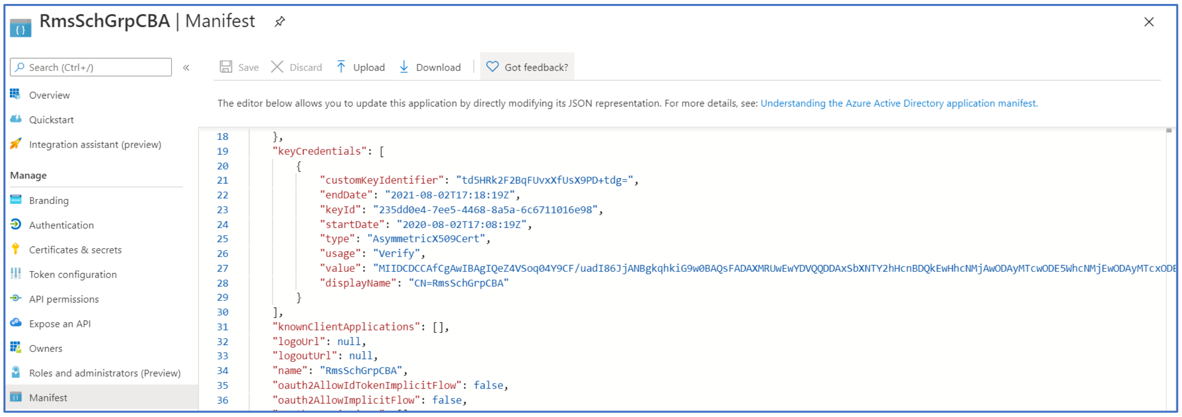
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

New client secret

IX. Modify the Manifest of Azure AD Application

In the left navigation menu, select **Manifest** from **Manage** section of **Application Menu Blade**

The editor shown on right side with JSON elements



The editor below allows you to update this application by directly modifying its JSON representation. For more details, see: Understanding the Azure Active Directory application manifest.

```
    },
    "keyCredentials": [
        {
            "customKeyIdentifier": "td5HRk2F2BqFUvxXfUsX9PD+tdg=",
            "endDate": "2021-08-02T17:18:19Z",
            "keyId": "235dd0e4-7ee5-4468-8a5a-6c6711016e98",
            "startDate": "2020-08-02T17:08:19Z",
            "type": "AsymmetricX509Cert",
            "usage": "Verify",
            "value": "MIIDCDCCAfCgAwIBAgIQeZ4VSoq04Y9CF/uadI86JjANBgkqhkiG9w0BAQsFADAXMRUwEwYDVQQDDAxSbXNTY2hHcnBDQkEwHhcNMjAwODAyMTcwODE5WhcNMjEwODAyMTcxODE
            "displayName": "CN=RmsSchGrpCBA"
        }
    ],
    "knownClientApplications": [],
    "logoUrl": null,
    "logoutUrl": null,
    "name": "RmsSchGrpCBA",
    "oauth2AllowIdTokenImplicitFlow": false,
    "oauth2AllowImplicitFlow": false,
```

The editor shown on right side with JSON elements. In the `"keyCredentials"`**:** section, replace the following with the value that we got from section III

```
"customKeyIdentifier": <Base-64 Thumbprint value got from PowerShell Script>

"keyId": <Keyid value got from PowerShell Script>
```

Click **Save** button to save the changes.

Appendix G: Limiting application permissions to specific Exchange Online mailboxes

https://learn.microsoft.com/en-us/graph/auth-limit-mailbox-access

In this article

Administrators who want to limit app access to specific mailboxes can create an application access policy by using the **New-ApplicationAccessPolicy** PowerShell cmdlet. This article covers the basic steps to configure access control. These steps are specific to Exchange Online resources and do not apply to other Microsoft Graph workloads.

Background

Some apps call Microsoft Graph using their own identity and not on behalf of a user. These are usually background services or daemon apps that run on a server without the presence of a signed-in user. These apps make use of OAuth 2.0 client credentials grant flow to authenticate and are configured with application permissions, which by default enable such apps to access *all* mailboxes in a organization on Exchange Online. For example, the `Mail.Read` application permission allows apps to read mail in all mailboxes without a signed-in user.

Important

By default, apps that have been granted **application permissions** to the following data sets can access all the mailboxes in the organization:

- Calendars
- Contacts
- Mail
- Mailbox settings

Administrators can configure **application access policy** to limit app access to *specific* mailboxes.

There are scenarios where administrators may want to limit an app to only specific mailboxes and *not all* Exchange Online mailboxes in the organization. Administrators can identify the set of mailboxes to permit access by putting them in a mail-enabled security group.

Administrators can then limit third-party app access to only that set of mailboxes by creating an application access policy for access to that group.

As further described in the Supported permissions and additional resources section below, application access policy restricts mailbox access for apps that have been granted any of the Microsoft Graph or Exchange Web Services permission scopes that the policy supports.

Configure ApplicationAccessPolicy

To configure an application access policy and limit the scope of application permissions:

1. Connect to Exchange Online PowerShell. For details, see Connect to Exchange Online PowerShell.
2. Identify the app's client ID and a mail-enabled security group to restrict the app's access to.
   - Identify the app's application (client) ID in the Microsoft Entra admin center > app registrations page.
   - Create a new mail-enabled security group or use an existing one and identify the email address for the group.
3. Create an application access policy.

   Run the following command, replacing the arguments for **AppId**, **PolicyScopeGroupId**, and **Description**.

   PowerShell

   ```
   New-ApplicationAccessPolicy -AppId e7e4dbfc-046f-4074-9b3b-2ae8f144f59b -PolicyScopeGroupId
   EvenUsers@contoso.com -AccessRight RestrictAccess -Description "Restrict this app to members of
   distribution group EvenUsers."
   ```

4. Test the newly created application access policy.

   Run the following command, replacing the arguments for **Identity** and **AppId**.

   PowerShellCopy

   ```
    Test-ApplicationAccessPolicy -Identity user1@contoso.com -AppId e7e4dbfc-046-4074-9b3b-2ae8f144f59b
   ```

   The output of this command will indicate whether the app has access to User1's mailbox.

   > **Note**
   > Changes to application access policies can take longer than 1 hour to take effect in Microsoft Graph REST API calls, even when `Test-ApplicationAccessPolicy` shows positive results.

Supported permissions and additional resources

Administrators can use ApplicationAccessPolicy cmdlets to control mailbox access of an app that has been granted any of the following Microsoft Graph application permissions or Exchange Web Services permissions.

Microsoft Graph application permissions:

- `Mail.Read`
- `Mail.ReadBasic`
- `Mail.ReadBasic.All`
- `Mail.ReadWrite`
- `Mail.Send`
- `MailboxSettings.Read`
- `MailboxSettings.ReadWrite`
- `Calendars.Read`
- `Calendars.ReadWrite`
- `Contacts.Read`
- `Contacts.ReadWrite`

Exchange Web Services permission scope: `full_access_as_app`.

For more information about configuring application access policy, see the PowerShell cmdlet reference for New-ApplicationAccessPolicy.

Handling API errors

You might encounter the following error when an API call is denied access due to a configured application access policy.

JSON

```json
{
    "error": {
        "code": "ErrorAccessDenied",
        "message": "Access to OData is disabled.",
        "innerError": {
            "request-id": "2f038156-cf40-403d-8e46-831fe42a8229",
            "date": "2019-05-24T10:16:21"
        }
    }
}
```

If the Microsoft Graph API calls from your app return this error, work with the Exchange Online administrator for the organization to ensure that your app has permission to access the mailbox resource.

Please see https://learn.microsoft.com for full details.

Appendix H: Google Authentication Configuration

Google Authentication Setup

To authenticate using the Google API, you will need to setup a new app registration in the Google API Console. You can setup a new app registration by following these steps:

Create the App Registration

1. Login to the **Google API Console** (https://console.cloud.google.com/) and tap on the **Select a Project** which is located at the upper left.
2. A popup should show, and then select **New Project**.
3. Give your app registration a name.
4. Select an organization, as well as the location if needed.
5. Click **Create** to finish the app registration.

Create the App Registration

1. Go to **APIs & Services** from the menu of the newly created app.
2. From the side menu, select **OAuth Consent Screen**.
3. Select either an **Internal** or **External** User type. This can be changed later.
4. Fill up the form fields for the **OAuth Consent Screen** section accordingly.
5. Click on **Save and Continue** to proceed.
6. Under the **Scopes** section, select the following:

Non-sensitive scopes

API	Scope	User-facing Description
Google Calendar API	.../auth/calendar.app.created	Make secondary Google calendars, and see, create, change, and delete events on them
Google Calendar API	.../auth/calendar.calendarlist.readonly	See the list of Google calendars you're subscribed to
Google Calendar API	.../auth/calendar.events.freebusy	See the availability on Google calendars you have access to
Google Calendar API	.../auth/calendar.events.public.readonly	See the events on public calendars
Google Calendar API	.../auth/calendar.settings.readonly	View your Calendar settings
Google Calendar API	.../auth/calendar.freebusy	View your availability in your calendars

Sensitive Scopes

API	Scope	User-facing Description
Google Calendar API	.../auth/calendar	See, edit, share, and permanently delete all the calendars you can access using Google Calendar

7. Click on **Save and Continue** to proceed.

8. On the Test User section, add testers accordingly.
9. Click on **Save and Continue** to proceed.
10. On the **Summary Section**, review the details if correct.
11. Tap on **Back to Dashboard** when done.

Creating Client Credentials

1. Select **Credentials** from the side menu.
2. Click on the **Create Credentials** at the top, and select **OAuth 2.0 Client IDs**.
3. Select **Android** as the Application Type.
4. Add Name, Package name, and SHA-1 certificate fingerprint. Instructions for the SHA-1 is on the page.
5. Tap on **Create** to finish.

Getting Client Secret and JSON

1. Select **Credentials** from the side menu.
2. Under the **OAuth 2.0 Client IDs** table, find your client credentials.
3. On the far right of the client credentials, tap on the download icon.
4. On the popup, take note of your **Client ID**. This will be used later. Assume that the real client id is: **<CLIENT_ID>**.apps.googleusercontent.com. For example, on the pop up, the client id shown is: *0000000000-3l6obc54r6ttg9csq8lalvv8sk0aq5ub.apps.googleusercontent.com*, assume that your real client id is: *0000000000-3l6obc54r6ttg9csq8lalvv8sk0aq5ub*.
5. Tap on the **Download JSON** to download json file. Rename the file into **client_secret.json**.

Usage of Web Authentication

1. Open AMX Book.
2. Navigate into the AMX Book settings.
3. Select **Calendar**
4. Tap on Edit
5. Select **Google Calendar**
6. Tap on **Web Authentication**
7. Input client id from the previous section.
8. Tap on **Verify**. If successful, you can login to your account and then select a calendar from your account.

Google Authentication Setup

Create Service Account

1. Login to the **Google API Console** (https://console.cloud.google.com/) and tap on the **Select a Project** which is located at the upper left.
2. Select the project you have created.
3. On the options, select **IAM & Admin**.
4. On the side, select **Service Accounts**.
5. On the top, select **CREATE SERVICE ACCOUNT**.
6. Fill up as needed.
7. When done, take note of the **Email**. This will be used later.

Creating Key for Service Account

1. Select a service account.

2. On the top, select **Keys**.
3. Tap on **ADD KEY**.
4. Tap on **Create new key**.
5. Select on **P12** and tap **Create**.
6. Store and keep note of the P12 file created.

Creating Base64 from the Key

Generate the Base64 for the pfx file using this command:

```
openssl base64 -in <p12_name>.p12
```

You can also copy it directly to your clipboard using this command:

```
openssl base64 -in <p12_name>.p12 | pbcopy
```

NOTE: When getting it directly from the editor, make sure there are no white spaces in between. It should just be one long string with no spaces/line breaks in between.

Usage of Service Account

1. Open AMX Book.
2. Navigate into the AMX Book settings.
3. Select **Calendar**
4. Tap on Edit
5. Select **Google Calendar**
6. Tap on **Service Account**
7. Input the service account email and key base64 string
8. Input email id of user you want to impersonate. This user must be in the organization.
9. Tap on **Verify**. If successful, you can select a calendar delegated to the user you impersonated.

Appendix I: Hardware API

Overview

AMX Varia panels support integration with third-party apps and web apps. Its hardware components can be accessed via the hardware API.

Device Peripherals

Device peripherals like camera, microphone, speakers, NFC, etc. can be accessed in apps & web applications via normal Android configuration methods.

References:
https://source.android.com/docs/setup/about/faqs
https://www.androidauthority.com/aosp-explained-1093505/

Any-Color Side LEDs

AMX Varia touch panels feature programmable *Any-Color Side LEDs* that can be set to one of over a million colors. These can be programmed either by an AMX Netlinx control system or via HTML calls.

Previous generation RoomBook, Acendo Book, and Modero G5 Acendo Book room scheduling panels featured similar side LEDs, but these are preprogrammed and capable of displaying either green or red only. This was based on the room's availability, available or booked, respectively.

See the table below to program the Any-Color side LEDs:

AMX Varia Any-Color Side LED usage	
Web API	The LEDs are controlled via REST communication to an internal server. This server is NOT accessible from external IP. LED information can be queried with a simple GET call to http://localhost:8080/v1/led/side_led will return the RGB value of the LED in a JSON response { "red": 0, "green": 0, "blue": 0 } To SET the LED, POST { "red": 255, "green": 255, "blue": 255 } to http://localhost:8080/v1/led/side_led