



API Command List

by [Touch Panel Control](#)

These commands are additional to AMX API commands. AMX commands can be found in [AMX PI](#).

Overview:

The purpose of the document is to provide an overview of commands specific to TPControl, which are additional to AMX specific API commands. TPControl recognises AMX G4 touch panel commands for applicable supported features, and includes support for legacy G3 touch panel commands.

Implementation:

For control of any TPControl enabled device, commands can be implemented within NetLinx programming, or by parsing the API in the programming *command port : 0 - loopback port* of a TP4 design file. Refer to 'TPDesign4 > Button Properties > Programming'.

Launch TPControl from other applications:

TPControl supports being launched from other applications installed on your device, using a common URI/URL call for each supported operating system; Android, iOS, Windows 8/10/Windows RT:

tpcontrol://

Parsing parameters in URI calls:

TPControl supports parsing API parameters in the URI call. Using the TPControl API commands listed in this document, you can configure and launch TPControl from a single URI call.

For example:

tpcontrol://?tpccmd-1;LocalHost,10.100.1.11;DeviceID,11001;ScreenResize,Scale;ApplyProfile;

The '?' is used to denote parameter parsing follows the URI.

Parameter parsing for the TPControl URI is supported in:

- TPControl v2.5.0.0+ for Android
- TPControl v2.5.0.0+ for iOS
- TPControl v1.4.0.0+ for Windows 8/10/RT

NOTE: To launch other applications from TPControl, please refer to the 'TPCURL' API command, detailed later in this document.

Testing URI calls on your device:

To test URI's on a device, do the following:

Android: Install/Run '[URI Launcher](#)' from Google Play, enter the URI and press *LAUNCH*

iOS: Launch *Safari* browser, enter the URI and press *Go*

Windows: Press the *Windows-key + R*, enter the URI and press *OK*

Demonstration files:

The following resources are provided to assist in reviewing and understanding some of the unique features available within TPControl. Simply download the resources and review at your convenience.

These examples and more are available in the '**Other Resources**' section of our [Downloads](#) page.

- **API command example**

A TP4 file demonstrating use of Touch Panel Control API commands.
Download the example here: [TPC API command examples.TP4](#)

- **Dynamic Profile manipulation**

Have a need to dynamically assign connection profiles, independent of code? One profile can be used dynamically to connect to any number of unique control system configurations.
Check out how to do this in this TP4 file: [TPC API Profile examples.TP4](#)

- **VoiceControl**

Want to talk to your device to do...well anything you like? It can even talk back, and in a variety of languages!
An example workspace for VoiceControl (Speech-to-Text, Text-to-Speech), which includes code and a TP4 file is available for TPControl here: [VoiceControl for TPControl.AXW](#)

- **List Box elements**

A powerful feature of many AMX G4 touch panels, is List Box elements.
Supported in *TPControl for iOS v2.3.2.0*, you can implement the standard AMX vertical List Box elements, or using our *TPCLBO* API command, switch to horizontal List Box elements. And yes, TPControl supports inertial scrolling within List Boxes!
List Boxes are great for source selection options, channel selects, large tables of data just to mention a few ideas. All data can be dynamically updated via the standard AMX API.
An example workspace for List Box implementation, including code and a TP4 file for use with TPControl is available here: [TPC ListBoxes.AXW](#)

- **Push Messaging Module:**

Send personalised messages to TPControl devices, even if TPControl is not running or connected to the control system!
Just like SMS for mobile, you can send messages to validated/registered TPControl devices anywhere in the world, provided WiFi or MobileData connection is available. Messages are routed via Apple, Google, and Microsofts secure messaging services for respective devices.
For the latest version of the Push Messaging NetLinX module, check out the [Downloads](#) page.

- **Dynamic AMX Device ID allocation from code**

Running a BYOD Licensed system or simply need to manage AMX Device IDs from code?
This example code demonstrates a simple method for managing devices connecting using a common AMX Device ID. The example could be expanded to permit devices with full, restricted or no access, depending on your requirements.
Download the example here: [TPControl Dynamic DeviceID assignments from code.AXW](#)

- **H.264 Video/Audio example**

Download an example TP4 file including reference H.264, MJPEG and other resources.
Specifically, H.264 support is available for:
- TPControl v2.5.0.0+ for Android
- TPControl v2.5.0.0+ for iOS
- TPControl v1.4.0.0+ for Windows 8/10/RT
Download the example here: [Video Streaming \(H264, MJPEG, etc\).TP4](#)

Supported commands:

SIP Video & Audio conferencing ^(C)

TPCSIP: Command List	
Show Presents the built-in SIP UI call interface, in full screen mode.	Syntax: TPCSIP-Show
Hide Hides the built-in SIP UI call interface.	Syntax: TPCSIP-Hide
SendVideo Provides the ability to enable or disable the video stream for video calls. Command can be instantiated before or during calls.	Syntax: TPCSIP-SendVideo,<1 0>; Example: TPCSIP-SendVideo,1; TPCSIP-SendVideo,0;
VideoCall Place a video call to an end-point registered SIP device. If video is not supported by the end-point, and audio call will be placed accordingly.	Syntax: TPCSIP-VideoCall,<extension extension name uri>; Example: TPCSIP-VideoCall,101; TPCSIP-VideoCall,TPC; TPCSIP-VideoCall,TPC@mydomain.com;
AudioCall Place an audio call to an end-point registered SIP device.	Syntax: TPCSIP-AudioCall,<extension extension name uri>; Example: TPCSIP-AudioCall,101; TPCSIP-AudioCall,TPC; TPCSIP-AudioCall,TPC@mydomain.com;

NOTE: TPCControl also supports AMX SIP API commands. AMX commands can be found in [AMX PI](#). SIP commands not supported: Hold, Transfer, Privacy, Mailbox.

QR Code Scanner ^(D)

TPCQRC: Command List	
Scan Initiates the QR Code scanner process within the current TPCControl UI.	Syntax: TPCQRC-Scan
Cancel End Ends the QR Code scanner process, removing the view from the TPCControl UI .	Syntax: TPCQRC-<Cancel End>; Example: TPCQRC-Cancel; TPCQRC-End;

TPCURL and TPCCMD commands (2):

<p>TPCURL</p> <p>Device specific functionality such as;</p> <ul style="list-style-type: none"> - launching URLs - launching applications <p><i>NOTE:</i> To determine if an application can be launched using a URI, contact that applications developer.</p>	<p>Examples:</p> <p>TPCURL-http://www.touchpanelcontrol.com TPCURL-itms-apps://itunes.com/apps/TPControl TPCURL-tel:+1234567890 TPCURL-sms:+1234567890</p> <p>TPCURL-mailto:support@touchpanelcontrol.com?subject=Test%20subject&body=Great%20work!</p> <p>For other schemes that may be supported by your device OS, please refer to: http://en.wikipedia.org/wiki/URI_scheme</p>
<p>TPCCMD</p> <p>TPControl specific configuration commands are prefixed with TPCCMD.</p>	<p>Syntax: TPCCMD- [<profileID>;]<cmd>[,<value>;<cmd>,<value>]...;</p> <p>Variables: <profileID> = 0^(A) 1 2 3 4 5 (<i>optional</i>) If <profileID> is not declared, the command(s) will be applied to ProfileID "1". A <profileID> value of "0" will apply the command(s) to the current active profile.</p> <p><cmd> Refer to the Command List.</p> <p><value> Refer to the Command List for specific values for each command.</p>
<p>TPCCMD: Command List</p>	
<p><i>Verify</i> (*)</p> <p>Verify/refresh the TPControl device license based on information stored on Touch Panel Control servers.</p>	<p>Syntax and example: TPCCMD-Verify;</p> <p>A "Device successfully verified" message will be presented upon successful completion of the verification process.</p>
<p><i>Update</i> (*)</p> <p>This option provides the ability to update the TPControl UI file(s) and/or Settings via the Internet.</p> <p>Management services for device Settings and TP4 files is provided via TPcloud at tpcloud.touchpanelcontrol.com</p>	<p>Syntax: TPCCMD-Update,<TPcloud TP4 Settings All BYOD>;</p> <p>Examples: TPCCMD-Update; TPCCMD-Update,TPcloud;</p> <p>Updates the device based on the 'Enable/Disable TPcloud update' options configured for the Token within TPcloud.</p> <p>TPCCMD-Update,TP4; TPCCMD-Update,Settings; TPCCMD-Update,All;</p> <p>Forces update of the specified resource(s) independent of the 'Enable/Disable TPcloud update' options configured for the Token within TPcloud.</p>

	<p>TPCCMD-Update, BYOD; ^(E)</p> <p>When TPControl is connected to a BYOD licensed AMX system, this API command can be invoked to force retrieval of the BYOD UI file which is optionally hosted in TPCloud and/or on the AMX System.</p> <p>How BYOD file retrieval works: TPControl identifies BYOD licensed control systems when connecting to the AMX System, and if TPControl has Internet access, it queries TPCloud to see if a BYOD UI file has been configured via the "Assign BYOD UI file" option for the BYOD Token for the related AMX system. If so, the BYOD UI file will be retrieved from TPCloud. Otherwise, if TPControl does not have access to TPCloud (or there is no assigned BYOD UI file), and a BYOD UI file has been stored on the AMX system, TPControl will retrieve the BYOD UI file from the AMX system.</p> <p>NOTE: Upon connection to a BYOD licensed AMX System TPControl automatically downloads the BYOD UI file from TPCloud (if available) or the AMX system, but only if the file does not match a previously retrieved file for that AMX system.</p> <p><i>The 'TPCCMD-Update, BYOD;' command can be used to overwrite (within TPControl) a previously retrieved BYOD UI file for the related AMX system.</i></p>
<p>TPCloudPIN ^{(*) (5) (6)}</p> <p>PIN codes can be generated specific to each Token that is stored within TPCloud at tpcloud.touchpanelcontrol.com</p> <p>PIN codes are used to complete device registrations and deauthorizations as configured within TPCloud.</p> <p>PIN entries will usually be entered via the device UI, however this command provides an method of PIN entry independently.</p> <p>Please refer to TPCloud for further information.</p>	<p>Syntax: TPCCMD-TPCloudPIN, <value>;</p> <p>Variables: <value> = 5-character alpha-numeric code</p> <p>Example: TPCCMD-TPCloudPIN, A1234;</p>
<p>TPCCMD: Profile related options</p>	
<p>LocalHost</p> <p>Set the NetLinx Master Connection IP/URL</p>	<p>Syntax: TPCCMD- [<profileID>;]LocalHost[:PortNumber], <IPAddress>;</p> <p>Examples: TPCCMD-LocalHost, 192.168.10.11; TPCCMD-1;LocalHost, 192.168.10.11:1319; TPCCMD-2;LocalHost, 192.168.1.101;</p>

<p><i>LocalPort</i></p> <p>Set the ICSP port number value (Default Port value is: 1319)</p>	<p>Syntax: TPCCMD- [<profileID>;]LocalPort,<PortNumber>;</p> <p>Example: TPCCMD-LocalPort,1319;</p>
<p><i>eICSPu</i></p> <p>Set the Encrypted ICSP Username</p>	<p>Syntax: TPCCMD- [<profileID>;]eICSPu,<username>;</p> <p>Example: TPCCMD-1;eICSPu,User 1;</p>
<p><i>eICSPp</i></p> <p>Set the Encrypted ICSP Password</p>	<p>Syntax: TPCCMD- [<profileID>;]eICSPp,<password>;</p> <p>Example: TPCCMD-1;eICSPp,Pass1;</p>
<p><i>eICSP</i></p> <p>Enable or Disable Encrypted ICSP connection method</p>	<p>Syntax: TPCCMD- [<profileID>;]eICSP,<>true false>;</p> <p>Example: TPCCMD-2;eICSP,true;</p>
<p><i>UseMobileData</i> ⁽⁴⁾</p> <p>Options available determine whether to use WiFi and/or MobileData to establish a connection with the NetLinx Master.</p> <p><i>(UseMobileData replaces the legacy Use3G API command)</i></p>	<p>Syntax: TPCCMD- [<profileID>;]UseMobileData,<>false withWiFi withoutWiFi>;</p> <p>Variables: false Device will only use the WiFi adapter to establish a connection</p> <p>withWiFi Device will use the WiFi adapter to establish a connection if a connection is present. If no WiFi is available, MobileData will be used to establish a connection</p> <p>withoutWiFi Device will use only MobileData to establish a connection NOTE: MobileData may not be supported while an active WiFi connection is present.</p> <p>Example: TPCCMD-1;UseMobileData,false; TPCCMD-2;UseMobileData,withWiFi; TPCCMD-2;UseMobileData,withoutWiFi;</p>
<p><i>DeviceID</i></p> <p>Set the Device ID number used upon connection to the NetLinx master.</p>	<p>Syntax: TPCCMD- [<profileID>;]DeviceID,<value>;</p> <p>Example: TPCCMD-1;DeviceID,11001;</p>
<p><i>DeviceName</i></p> <p>Set the Device Name</p>	<p>Syntax: TPCCMD- [<profileID>;]DeviceName,<value>;</p> <p>Example: TPCCMD-1;DeviceName,Cinema;</p>

<p>TP4FileSlot ⁽⁸⁾</p> <p>The TP4FileSlot parameter can be used to assign the TP4 File Slot to be used with any existing profile. TPControl supports multiple TP4 files, with 2 x TP4 File Slots available as a default.</p> <p>NOTE: If more than 2 x TP4 File Slots are required, TPControl 'Full license' Tokens can be licensed for as many TP4 File Slots as required, using TP4 File Slot KEYS which can be applied via TPCloud.</p>	<p>Syntax: TPCCMD- [<profileID>;]TP4FileSlot, <value>;</p> <p>Variables: <value> = 0^(B) n Where 0 = BYOD system_license.tp4 Where n = File Slot number</p> <p>Example: TPCCMD-1;TP4FileSlot, 0; TPCCMD-1;TP4FileSlot, 2;</p> <p>NOTE: BYOD files are unique to each BYOD system even if profiles settings are common between different BYOD systems. TPControl will recall the relevant file based on the BYOD system it is connected to.</p>
<p>ProfileName ⁽⁴⁾</p> <p>Provides the ability to name profiles</p>	<p>Syntax: TPCCMD- [<profileID>;]ProfileName, <value>;</p> <p>Variables: <profileID> = 0^(A) 1 2 3 4 5 (<i>optional</i>) If <profileID> is not declared, the command(s) will be applied to ProfileID "1". A <profileID> value of "0" will apply the command(s) to the current active profile.</p> <p>Example: TPCCMD-ProfileName, Home; TPCCMD-1;ProfileName, Home; TPCCMD-2;ProfileName, Office; TPCCMD-3;ProfileName, Home while in Office;</p>
<p>ApplyProfile ⁽⁴⁾</p> <p>Provides the ability to recall the Settings stored within the declared profile. If the profile is different from the current active profile, TPControl will disconnect the active connection, and attempt to connect using the new profile settings.</p>	<p>Syntax: TPCCMD- [<profileID>;]ApplyProfile;</p> <p>Variables: <profileID> = 0^(A) 1 2 3 4 5 (<i>optional</i>) If <profileID> is not declared, the command(s) will be applied to ProfileID "1". A <profileID> value of "0" will apply the command(s) to the current active profile.</p> <p>Example: TPCCMD-ApplyProfile; TPCCMD-1;ApplyProfile; TPCCMD-3;ApplyProfile;</p>
<p>DefaultProfile ⁽⁴⁾</p> <p>Provides the ability to set the default start-up profile for TPControl, which will be retained between sessions. Default is profileID "1".</p> <p>NOTE: Returning to TPControl from Home-screen or multi-tasking mode,</p>	<p>Syntax: TPCCMD- [<profileID>;]DefaultProfile;</p> <p>Variables: <profileID> = 1 2 3 4 5 (<i>optional</i>) If <profileID> is not declared, the command(s) will be applied to ProfileID "1".</p> <p>Example: TPCCMD-DefaultProfile;</p>

<p>does not represent restarting TPControl. As such, TPControl retains the previous active profile when returning.</p>	<pre>TPCCMD-1;DefaultProfile; TPCCMD-2;DefaultProfile;</pre>
<p>QueryProfile ⁽⁴⁾</p> <p>Returns a STRING including related profile data. The current default profile and active profile information is included in the returned data.</p>	<p>Syntax: TPCCMD- [<profileID>;]QueryProfile;</p> <p>Variables: <profileID> = 0^(A) 1 2 3 4 5 (<i>optional</i>) If <profileID> is not declared, the command(s) will be applied to ProfileID "1". A <profileID> value of "0" will apply the command(s) to the current active profile.</p> <p>Example: TPCCMD-QueryProfile; TPCCMD-1;QueryProfile; TPCCMD-2;QueryProfile; TPCCMD-0;QueryProfile;</p> <p>Response format: ProfileInfo-<profileID>; LocalHost,<host_ip>:<local_port>; DeviceID,<device_id>; eICSP,<0 1>; DeviceName,<device_name>; UseMobileData,<false withWiFi withoutWiFi>; ProfileName,<profile_name>; DefaultProfile,<default_profile_number>; ActiveProfile,<current_active_profile_number>; TP4FileSlot,<tp4_file_slot_number>;</p> <p>Response Example: ProfileInfo-3; LocalHost,192.168.10.12:1319; DeviceID,11001; eICSP,1; DeviceName,Galaxy Nexus; UseMobileData,withoutWiFi; ProfileName,Home while in Office; DefaultProfile,1; ActiveProfile,1; TP4FileSlot,2;</p>
<p>QueryDeviceInfo ⁽⁵⁾</p> <p>Returns a STRING including related device identification information.</p>	<p>Syntax: TPCCMD-QueryDeviceInfo;</p> <p>Response format: ProfileInfo-TPCDeviceID,<TPCDeviceID>; udid,<udid>; push_token,<push_messaging_token>;</p> <p>Response Example: DeviceInfo-TPCDeviceID,DEVID-0003-NKCORA; udid,b560e49d0d07feda01535dac88229e76eecca6b3; push_token,4f08bf3c4a1b30cfb9186ff97f6ae42d7818b7b402f07ac978a6f39fdf659a32;</p>

<p>Example concatenated commands</p> <p>Commands and their associated values i.e. <cmd>, <value> can be concatenated, so that multiple commands can be applied by sending one command expression. Concatenate commands must be separated by semi-colons ';' e.g. <cmd>, <value>; <cmd>, <value></p>	<p>Examples: TPCCMD-1;LocalHost,192.168.10.2;DeviceID,11006; TPCCMD-1;LocalHost,192.168.10.2:1319;DeviceID,11006; TPCCMD-2;LocalHost,192.168.5.51:1319;DeviceID,11004; TPCCMD-2;LocalHost,192.168.5.51;eICSP,true;eICSPu,User1;eICSPp,Pass;</p> <p>NOTE: If you want to activate/apply the profile immediately, then simply append or issue the 'ApplyProfile' command: Examples: TPCCMD-1;LocalHost,10.0.0.2;DeviceID,11007;ApplyProfile; or TPCCMD-3;ApplyProfile;</p>
<p>TPCCMD: Settings related options</p>	
<p><i>KeepWiFiActive</i></p> <p>When Enabled, TPControl will continue to keep a connection live with the NI Master when the device goes to sleep or another application takes device focus e.g. the device Home screen.</p>	<p>Syntax: TPCCMD-KeepWiFiActive, <true false>;</p> <p>Example: TPCCMD-KeepWiFiActive, true;</p>
<p><i>Gestures</i></p> <p>Enable or disable standard AMX gesture recognition.</p> <p><i>NOTE: Recommend disabling gestures when device Accessibility mode is enabled, due to gesture specific operation of Accessibility functions.</i></p>	<p>Syntax: TPCCMD-Gestures, <true false>;</p> <p>Example: TPCCMD-Gestures, true;</p>
<p><i>AutoLock</i></p> <p>When enabled, <true>, this will allow the Device to run the OS screen lock feature as set within the device settings. If AutoLock is set to <false> the screen will stay active until the Power button is pressed.</p>	<p>Syntax: TPCCMD-AutoLock, <true false>;</p> <p>Example: TPCCMD-AutoLock, true;</p>
<p><i>LockRotation</i></p> <p>Enable or disable screen rotation.</p>	<p>Syntax: TPCCMD-LockRotation, <true false>;</p> <p>Example: TPCCMD-LockRotation, false;</p>
<p><i>DarkScreen</i> ⁽⁹⁾</p> <p>When enabled, provides a dark background loading image for TPControl as opposed to the default white</p>	<p>Syntax: TPCCMD-DarkScreen, <true false>;</p> <p>Example: TPCCMD-DarkScreen, true;</p>

<p>background loading image. Avoid the potential for a bright white loading screen when in dark rooms by enabling this option.</p> <p><i>Deprecated: DarkScreen is a permanent/fixd method in relevant versions of TPControl.</i></p>	
<p>DisableMultitouch</p> <p>Multitouch functionality (if supported) can be disabled/enabled with this option.</p>	<p>Syntax: TPCCMD-DisableMultitouch,<true false>;</p> <p>Example: TPCCMD-DisableMultitouch,false;</p>
<p>AccessibilityIncludeNoTextBtns</p> <p>When enabled, buttons with no text will be included when navigating during Accessibility VoiceOver mode.</p>	<p>Syntax: TPCCMD-AccessibilityIncludeNoTextBtns,<true false>;</p> <p>Example: TPCCMD-AccessibilityIncludeNoTextBtns,true;</p>
<p>AccessibilityLevelIncrement</p> <p>Defines the level increment/decrement percentage when a bargraph/level is selected during Accessibility VoiceOver mode. Gesture Up/Down defines increment/decrement action for the selected level.</p>	<p>Syntax: TPCCMD-AccessibilityLevelIncrement,<0-100>;</p> <p>Example: TPCCMD-AccessibilityLevelIncrement,10;</p>
<p>InactivityTimeout</p> <p>TPControl will flip to the Inactivity page that has been defined within the properties of the TP4 file, based on the timeout defined.</p>	<p>Syntax: TPCCMD-InactivityTimeout,<Never 0 1 2 5 10 15 30 60 120 180 240>;</p> <p>Example: TPCCMD-InactivityTimeout,Never;</p>
<p>ScreenResize</p> <p>Options allow the TP4 project file to be presented in various formats: No-scaling (None), Scale-to-fit (Scale), and Stretch-to-fit (Stretch).</p>	<p>Syntax: TPCCMD-ScreenResize,<None Scale Stretch>;</p> <p>Variables: None TP4 project will be presented in the original TP4 resolution</p> <p>Scale Upsizes/downsizes to maintain original aspect ratio of the TP4 file to fully extend to fill at least one dimension of the device display. Scale-to-fit.</p> <p>Stretch Upsizes/downsizes to fill the usable native screen area of the device display. Stretch-to-fit.</p> <p>Example: TPCCMD-ScreenResize,Scale;</p>

<p>ProfilePrompt</p> <p>When enabled, TPControl will present a dialog requesting confirmation of the connection profile to use whenever returning from the Home-screen or multi-tasking mode. This is independent of whether an active profile connection exists or not.</p> <p><i>NOTE:</i> Within supported versions of TPControl, the "Not Connected" dialog presents any <i>named</i> profiles for selection.</p> <p>The following named or unnamed profiles will always be shown:</p> <ul style="list-style-type: none"> ● the current active profile ● the current default profile ● any named profile 	<p>Syntax: TPCCMD-ProfilePrompt, <true false>;</p> <p>Example: TPCCMD-ProfilePrompt, false;</p>
<p>ButtonHit</p> <p>When enabled, Button Hit produces a "Beep" sound when a valid button area is pressed within the touch panel design file.</p>	<p>Syntax: TPCCMD-ButtonHit, <true false>;</p> <p>Example: TPCCMD-ButtonHit, true;</p>
<p>ButtonMiss</p> <p>When enabled, Button Miss produces a "Double Beep" sound when any area outside of a valid button area is pressed within the touch panel design file.</p>	<p>Syntax: TPCCMD-ButtonMiss, <true false>;</p> <p>Example: TPCCMD-ButtonMiss, true;</p>
<p>BeepLevel</p> <p>Sets the level at which the volume for the Beep will be announced.</p>	<p>Syntax: TPCCMD-BeepLevel, 0-100;</p> <p>Example: TPCCMD-BeepLevel, 30;</p>
<p>DeveloperMode</p> <p>When enabled, this will allow TPControl to communicate with TPTransfer.</p>	<p>Syntax: TPCCMD-DeveloperMode, <true false>;</p> <p>Example: TPCCMD-DeveloperMode, true;</p>
<p>TransferPort</p> <p>The port used by TPControl to communicate with TPTransfer. (Default port value is: 10700).</p>	<p>Syntax: TPCCMD-TransferPort, <integer>;</p> <p>Example: TPCCMD-TransferPort, 10700;</p>
<p>IntercomCallNotify</p> <p>When enabled, if TPControl is running but does not currently have application</p>	<p>Syntax: TPCCMD-IntercomCallNotify, <true false>;</p> <p>Example: TPCCMD-IntercomCallNotify, true;</p>

<p>focus, an alert notification will be presented on the device.</p>	<p><i>NOTE: If the Keep WiFi Active option within device settings has been disabled, notifications will not be presented in multi-tasking mode, as TPControl disconnects when it loses application focus. Enable Keep WiFi Active to sustain connection when operating multi-tasking mode.</i></p> <p><i>NOTE: iOS5 introduced an enforced 2.5 minute application timeout, which overrides the Multitasking timeout feature. So, if the device is running iOS5+, TPControl will be forced to disconnect after ~2.5 minutes when running in multi-tasking mode.</i></p>
<p>STTAutoHide ⁽⁴⁾ (Speech-to-Text, AutoHide)</p> <p>When disabled, TPControl will hide the Speech-to-Text recording dialog only after a successful result is processed.</p> <p>When enabled, TPControl will hide the Speech-to-Text recording dialog once processing of recorded audio commences, irrespective of the result.</p>	<p>Syntax: TPCCMD-STTAutoHide,<true false>;</p> <p>Example: TPCCMD-STTAutoHide,true;</p>
<p>STTDisplayResult ⁽⁴⁾ (Speech-to-Text, DisplayResult)</p> <p>The result of Speech-to-Text analysis can be displayed via a brief notification on-screen.</p> <p><i>NOTE: Errors in processing will still solicit a notification.</i></p>	<p>Syntax: TPCCMD-STTDisplayResult,<0 1 2 3>;</p> <p>Variables: 0: Do Not Display 1: 1 result will be displayed (default) 2: Up to 2 results will be displayed 3: Up to 3 results will be displayed</p> <p>Example: TPCCMD-STTDisplayResult,1;</p>
<p>TTSOfflineMode ⁽⁴⁾ (Text-to-Speech, OfflineMode) (Android only)</p> <p>When enabled, TPControl will utilise device defined language packs to determine text-to-speech translation.</p> <p>When disabled, TPControl will utilise online resources to determine text-to-speech translation.</p> <p><i>NOTE: Results are cached with each completed translation when operating in online mode.</i></p>	<p>Syntax: TPCCMD-TTSOfflineMode,<true false>;</p> <p>Example: TPCCMD-TTSOfflineMode,true;</p>
<p>TTSClearCache ⁽⁴⁾</p> <p>Clears the text-to-speech cache which is created during online operation.</p>	<p>Syntax and example: TPCCMD-TTSClearCache;</p>
<p>RestoreAllSettings</p>	<p>Syntax and example:</p>

<p>This will restore all settings within the device Settings page back to defaults</p>	<p>TPCCMD-RestoreAllSettings;</p>
<p><i>ClearUserPages</i></p> <p>The design file will be removed and the original Demo Pages will be loaded back onto the device.</p>	<p>Syntax and example: TPCCMD-ClearUserPages;</p>
<p><i>DownloadDemo</i></p> <p>Downloads the most recent TP4 demonstration file from Touch Panel Control.</p>	<p>Syntax and example: TPCCMD-DownloadDemo;</p> <p>NOTE: The device must have unrestricted access to the Internet in order to complete the command successfully.</p>
<p><i>ReprocessTP4</i></p> <p>Clears any caching, and reprocesses the installed TP4 file. This is the same process that runs whenever a file is transferred to the device.</p>	<p>Syntax and example: TPCCMD-ReprocessTP4;</p>

Speech-to-Text and Text-to-Speech commands ⁽⁴⁾:

<p>Speech-To-Text</p> <p><i>LISTEN</i></p> <p>When <i>Speech-to-Text</i> is activated, a recording window will appear on the device prompting you to speak.</p> <p><i>An option for selecting supported languages is provided.</i></p> <p>At the first discernible pause during speech, recording will automatically end, and processing will commence.</p> <p>The result of the speech analysis will be parsed to the NetLinx master in string format. The string is based on the language definition and will include up to three (3) closely matched results.</p> <p>NOTE: Results can be displayed on the device display. By default <result1> will be displayed, however control over displayed results is also provided via <i>STTDisplayResult</i>.</p>	<p>Syntax: TPCSTT TPCSTT-LISTEN</p> <p>Example: TPCSTT</p> <p>Response format: TPCSTT-<result1>;<result2>;<result3>; [<UNI>];</p> <p>Response Example: TPCSTT-lights on;light song;like song;</p> <p>If the response contains a UTF8 symbol, unicode format will be parsed for each result, with an appended <UNI> delimiter. e.g. "télé;tlf;tsr;" would be parsed as: TPCSTT- 007400E9006C00E9;0074006C0066;007400730072;<UNI>;</p> <p>NOTE: The device must have unrestricted access to the Internet in order to complete the command successfully.</p> <p>Errors that may arise during Speech-to-Text operations invoke an onscreen notification message. Similarly, an error message is parsed to the NetLinx master in the following STRING format: TPCSTTError-<errorID>,<errorText></p>
--	---

	<p>Where <errorID> and related <errorText> are:</p> <ol style="list-style-type: none"> 1 - Unable to recognise speech due to connection errors. 2 - Unable to recognise speech due to connection errors. 3 - Unable to recognise speech due to an audio recording error. 4 - Unable to recognise speech due to server errors. 5 - Unable to recognise speech due to unknown errors. 6 - Unable to recognise speech.\n Please, try again. 7 - Unable to recognise speech.\n Please, try again. 8 - Recogniser is busy at the moment.\n Please, try again later. 9 - Unable to recognise speech, application does not have appropriate permission.
<p>END</p> <p>Ends recording, and processes the recorded result when applicable. The recording window will be removed from the UI, provided no errors are detected during processing.</p>	<p>Syntax and example: TPCSTT-END</p>
<p>CANCEL</p> <p>Cancels recording, and the recording window will be removed from the UI.</p>	<p>Syntax and example: TPCSTT-CANCEL</p>
<p>Text-To-Speech</p> <p>Android: Text to Speech will operate in one of two modes; offline or online, as determined by <i>TTSOfflineMode</i>.</p> <p>iOS: Text to Speech will operate in online mode only.</p> <p>Where applicable, the device will present the audio, and the result of processing will be parsed to the NetLinx master in string format.</p>	<p>Syntax: TPCTTS-<text>;<locale></p> <p>Variables: <text> The text to convert to speech</p> <p><locale> The target language for the device to speak. Refer to appendix for suggested <locale> list.</p> <p>Example: TPCTTS-The lights are at 65%;en</p> <p>Response format: TPCTTS-<OK Unsupported Language/Locale Error_tts Error_player>;</p> <p>NOTE: When operating in online mode, the device must have unrestricted access to the Internet in order to complete the command successfully.</p> <p>NOTE: When operating in offline mode, a supported language pack (locale) must be installed on the device in order to function as intended.</p>

Listbox implementation ⁽⁶⁾:

Listbox elements are used for the display of table data. The table data can be dynamically created and updated via associated AMX API commands.

Refer to [AMX PI](#) (Mio Modero R-4) for further information.

<p>TPCLBO</p> <p>Listbox elements and their associated table data can be dynamically updated via related AMX API commands. The <i>TPCLBO</i> command extends on this, providing the ability to switch a Listbox element between vertical (default) and horizontal display modes. Inertial scrolling is supported for Listbox elements, fully independent of other Listbox elements.</p>	<p>Syntax: TPCLBO-<address code>,<Horizontal Vertical H V></p> <p>Variables: address code: programming address code for Listbox horizontal h: associate Listbox with Horizontal scrolling vertical v: associate Listbox with Vertical scrolling (default)</p> <p>Examples: TPCLBO-1,Horizontal TPCLBO-2,Vertical TPCLBO-3,H TPCLBO-4,V</p>
--	---

Device orientation ⁽⁷⁾:

<p>TPCACC</p> <p>Used to return device orientation data, parsed to the controller in string format on Port 1. Unparsed parameters will assume disabling the function.</p>	<p>Syntax: TPCACC-<Enable Disable Query></p> <p>Variables: Enable: orientation data will be actively returned upon change of device orientation Disable: orientation data will stop being actively issued Query: returns the current device orientation</p> <p>Examples: TPCACC-ENABLE TPCACC-DISABLE TPCACC-QUERY</p> <p>Response format: TPCACC-<orientation>;</p> <p>Examples: TPCACC-DeviceOrientationPortrait; TPCACC-DeviceOrientationPortraitUpsideDown; TPCACC-DeviceOrientationLandscapeLeft; TPCACC-DeviceOrientationLandscapeRight; TPCACC-DeviceOrientationFaceUp; TPCACC-DeviceOrientationFaceDown;</p>
--	---

Notifications ⁽¹⁾:

<p>TPCNotify</p> <p>Notification messages can be presented when the device is running in Background multi-tasking mode.</p> <p><i>Apple device:</i> Presents a notification window when the device is running in background mode, titled "TPControl". If a <code><text message></code> is included, the text will be presented within the dialog notification window. The notification window will provide 2 options; <i>Close</i> and <i>TPControl</i>.</p> <ul style="list-style-type: none">● Pressing <i>Close</i> will close the notification.● Pressing <i>TPControl</i> will return <code>TPCNotify-Accept</code> to the NI Master and the device will relaunch TPControl. <p><i>Android device:</i> Presents a notification alert when the device is running in background mode, with title "TPControl". If a <code><text message></code> is included, the text will be provided within the notification list for review.</p> <ul style="list-style-type: none">● Pressing the notification will return to the NI Master <code>TPCNotify-Accept</code> and the device will relaunch TPControl.	<p>Syntax: <code>TPCNotify-<text message></code></p> <p>Examples: <code>TPCNotify</code> <code>TPCNotify-A visitor is at the Front Door</code> <code>TPCNotify-There is an incoming call</code></p> <p>NOTE: If the <i>Keep WiFi Active</i> option within device settings has been disabled, notifications will not be presented in multi-tasking mode, as TPControl disconnects when it loses application focus. Enable Keep WiFi Active to sustain connection when operating multi-tasking mode.</p> <p>NOTE: iOS5 introduced an enforced 2.5 minute application timeout, which overrides the Multitasking timeout feature. So, if the device is running iOS5+, TPControl will be forced to disconnect after ~2.5 minutes when running in multi-tasking mode.</p>
<p>TPCWarn_Intercom</p> <p>All incoming or outgoing intercom functionality is indicated via a <i>Intercom Warning</i> icon relating to the current mode of Intercom operation.</p> <p>Modes indicated include: TX (Outgoing audio only) i.e. Device is being <i>Monitored</i></p> <p>RX (Incoming audio only) i.e. Receiving a <i>Page-All</i></p> <p>TX/RX (Audio incoming and outgoing) i.e. A two-way call is in progress</p>	<p>Syntax: <code>TPCWarn_Intercom-<value></code></p> <p>Variables: <i>Show (default)</i> All Intercom warnings will be shown</p> <p><i>Hide</i> No Intercom warnings will be shown</p> <p><i>Posn, <x>, <y></i> Updates the top-left draw position for the TPCWarn icons. All Intercom warnings will appear from the x,y location.</p> <p>Examples: <code>TPCWarn_Intercom-Show</code> <code>TPCWarn_Intercom-Hide</code> <code>TPCWarn_Intercom-Posn,100,100</code></p>

TPCLockOut commands supported in TPControl for Android ⁽³⁾:

TPCLockOut functionality may vary between different Android devices. Factors that can affect the intended operation can relate to the operating system version installed, and manufacturer or service-provider software operating on the device.

<p><i>LockOut</i></p> <p>Enabling <i>LockOut</i> will restrict usage of the "Home", "Back", "Menu/Setting", and "Search" soft-or-external button functions on the Android device. TPControl retains application focus, and if TPControl detects that application focus has been lost, will attempt to automatically regain focus.</p> <p>Restricted access to TPControl <i>Settings</i> is provided through a pin-code (see <i>LockOutPin</i>) when <i>LockOut</i> is enabled.</p>	<p>Syntax: TPCCMD-LockOut,<true false>;</p> <p>Example: TPCCMD-LockOut,true;</p>
<p><i>LockOutPin</i></p> <p>When the <i>LockOut</i> functionality is enabled, access to TPControl Settings is restricted by entry of a pin-code. The <i>LockOutPin</i> pin-code can be updated using this command. There is no restriction on pin-code length.</p>	<p>Syntax: TPCCMD-LockOutPin,<ascii-numeric>;</p> <p>Example: TPCCMD-LockOutPin,1234567890; TPCCMD-LockOutPin;</p> <p>NOTE: Non-numeric characters parsed will invalidate the command.</p>
<p><i>AutoLaunch</i></p> <p>Option to automatically launch TPControl when the device boots up.</p>	<p>Syntax: TPCCMD-AutoLaunch,<true false>;</p> <p>Example: TPCCMD-AutoLaunch,true;</p>

^(*) **NOTE:** The device must have unrestricted access to the Internet in order to complete the command successfully.

⁽¹⁾ Added in: v2.0.0.0

⁽²⁾ Added in: v2.2.0.0

⁽³⁾ Added in: v2.2.0.11 Android (TPCLockOut pre-release)

⁽⁴⁾ Added in: v2.3.0.0

⁽⁵⁾ Added in: v2.3.1.0 Android

⁽⁶⁾ Added in: v2.3.2.0 iOS

⁽⁷⁾ Added in: v2.3.3.0 iOS

⁽⁸⁾ Added in: v2.4.0.0 iOS

⁽⁹⁾ Deprecated in: v2.4.0.0 iOS

^(A) Added in: v2.4.2.0 iOS, v1.1.0.0 Windows

^(B) Added in: v2.5.0.0 iOS/Android, v1.4.0.0 Windows

^(C) Added in: v2.6.0.0 iOS/Android

^(D) Added in: v2.5.1.0 iOS/Android

^(E) Added in: v2.6.3.0 iOS/Android, v1.5.3.0 Windows

APPENDIX

Text-to-Speech language and Locale information

Language	Locale
Catalan	ca
Chinese (Simp.)	zh-CN
Chinese (Trad.)	zh-TW
Croatian	hr
Danish	da
Dutch	nl
English	en
Esperanto	es
Finnish	fi
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Norwegian	no
Polish	pl
Portuguese	pt
Russian	ru
Spanish	es
Swedish	sv

Problem Reporting

We aim to make your integration experience of TPControl within your AMX environment, as seamless as possible.

If you encounter any difficulties using the product or any of its features, please let us know and we will be happy to assist.

The helpdesk on our website at support.touchpanelcontrol.com operates from:

- Monday to Friday; 09:00 to 17:00 (GMT)
- Monday to Friday; 07:00 to 15:00 (AEST)

We appreciate your support,
Touch Panel Control Team.

AMX is a trademark of AMX, LLC registered in the US and other countries.
Android is a trademark of Google Corporation registered in the US and other countries.
Apple is a trademark of Apple Inc. registered in the US and other countries.
Windows is a trademark of Microsoft Corporation registered in the US and other countries.
All other trademarks and copyrights are the property of their respective owners.