TPDESIGN5

TOUCH PANEL DESIGN/PROGRAMMING FOR MODERO X G5 TOUCH PANELS

AV FOR AN IT WORLD®

Table of Contents

TPDesign5 (v1.3 or higher)

Overview

The TPDesign5 Touch Panel Design program ("TPD5") is designed to assist you in creating a state-of-the-art touch panel interface for AMX's X Series G5 touch panels. Use TPD5 to create *.TP5 project files containing all of the information required to define a user-interface to be utilized on a G5 touch panel. This includes Pages, Popup Pages and associated navigation/page-flip information, as well as buttons (including function code information), and all image and sound-related files that are used in the design (including dynamic images and video feeds).

Software and Hardware Requirements

To install TPDesign5, you must have the following software installed on your computer:

- Microsoft Windows 7 or Windows 8 (32- or 64-bit)
- Operating System Languages: English, Spanish, French, German

Minimum hardware requirements

- 60 MB of free disk space (minimum requirement); 80 MB recommended
- Minimum (VGA) screen resolution of 800x600
- Windows-compatible mouse (or other pointing device)

Getting Started

Before starting a new TPDesign5 project, you should have a clear idea of how your touch panel needs to function and how it should look. This includes identifying all image and sound files that will be used in the project, as well as a clear understanding of the control system to which the target touch panel will connect. A basic outline of a typical workflow for creating a new project in TPDesign5 would be:

1. Create a new TPD5 project with the New Project Wizard:
   - Setting Project Properties - see page 23

2. Import all image and sound files into the Resource Manager to make them available to this project:
   - Importing Image Files Into the Project - see page 32
   - Adding Dynamic Images to the Project - see page 38
   - Importing Sound Files Into the Project - see page 43

3. Create additional Pages if necessary (depending on your design), and set the properties for each:
   - Adding Pages to the Project - see page 33
   - Setting Page Properties - see page 34

4. Create Popup Pages as necessary, and set the properties for each:
   - Adding Popup Pages to the Project - see page 38
   - Setting Popup Page Properties - see page 40
   - Creating Popup Page Groups - see page 42

5. Create Sub-Pages as necessary, and set the properties for each. To use Sub-Pages, it necessary to create Sub-Page View Buttons and Scrolling Regions:
   - Adding Sub-Pages to the Project - see page 46
   - Setting Sub-Page Properties - see page 48
   - Creating Sub-Page Sets - see page 49
   - Sub-Page View Buttons - see page 52
   - Assigning a Sub-Page Set to the Sub-Page View Button - see page 53
   - Creating a Scrolling Region - Overview - see page 53

6. Add Application windows as necessary:
   - Adding Applications - see page 60
   - Setting Application windows Properties - see page 61
   - Editing Application Parameters - see page 62

7. Draw Buttons on Pages Popup Pages and Sub-Pages, and set Button properties for each to specify button type and functionality as well as button text and other visual properties:
   - Creating New Buttons - see page 87
   - Setting Button Properties - see page 87
   - Text Effects - see page 90
   - Animation Effects - see page 90

8. Assign Bitmaps, Sounds, Borders, Fills, Text and Video Fills to the TP5 Elements in your project:
   - Assigning Bitmaps to TPD5 Elements - see page 33
   - Assigning Sounds to Buttons - see page 43

- Assigning Borders to TPD5 Elements - see page 103
- Assigning Fills (Type and Color) to TPD5 Elements - see page 104
- Assigning Text to TPD5 Elements - see page 104
- Assigning Video Fills to TPD5 Elements - see page 106

9. Assign Events including Page Flips, Launch Actions (which open Applications on the panel), and Command or String Actions to Pages and Buttons:
   - Action Types - see page 271
   - Adding a Command (Action) to a Page or Button - see page 272
   - Adding a String (Action) to a Page or Button - see page 273

10. Assign Function Codes according your system design requirements. See *Function Codes* on page 276 for details.

11. Transfer the TP5 project file to the NetLinx Master to which the G5 panel is connected. See *File Transfer Operations* on page 285 for details.

The TPDesign5 Work Area

The TPD5 work area consists of several main components (FIG. 1):
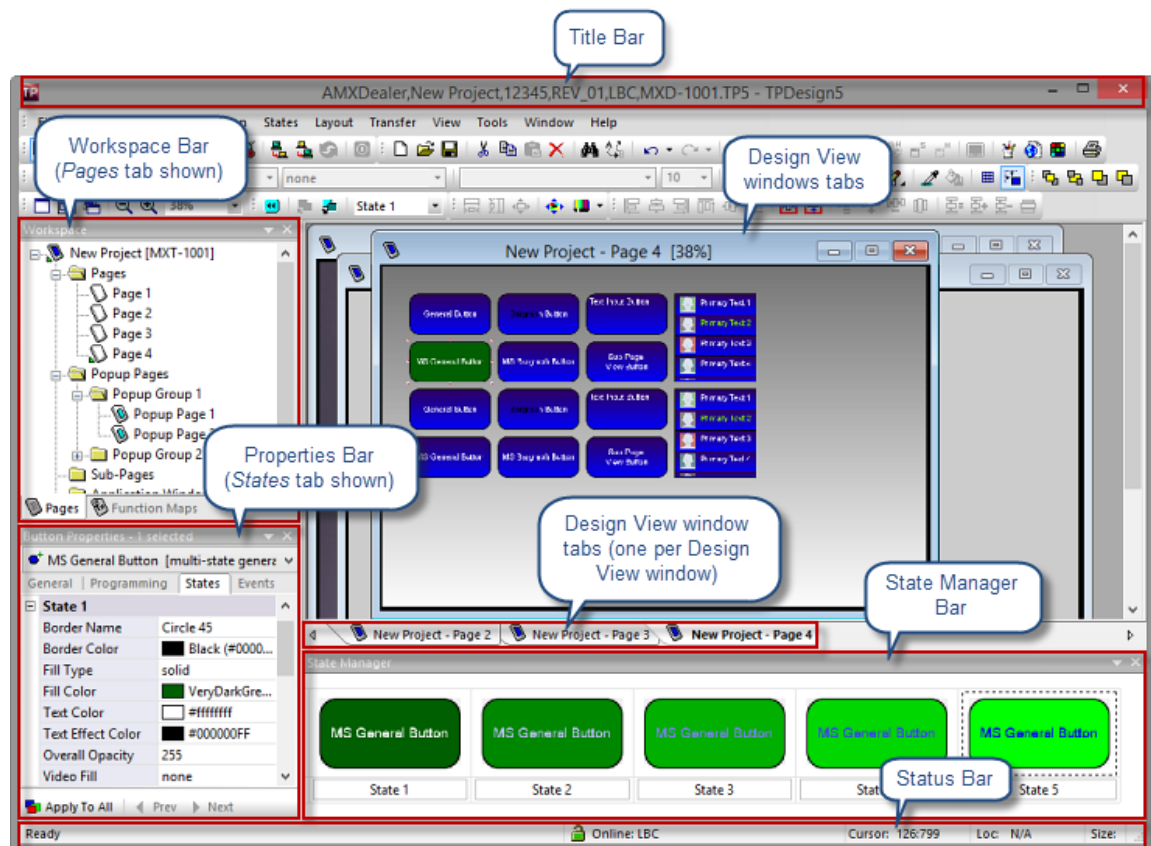
FIG. 1 The TPDesign5 Work Area

In its default configuration, from top to bottom, left to right, the main components are:

- **Title Bar** - Lists the name of the active project file.
- **Menu Bar** - Contains the main menu options (*File*, *Edit*, *Panel*, *Page*, *Button*, *States*, *Layout*, *Transfer*, *View*, *Tools*, *Window*, and *Help*).
- **Toolbars** - You can choose which toolbars to show in the workspace via the **View > Toolbars** sub-menu. Click View, then click in the checkboxes to select/de-select the listed toolbars (including custom toolbars). If a toolbar has a check next to it in the sub-menu, then it is displayed. These settings are saved, so they'll apply the next time the application is launched.
- **Workspace Bar** - Contains two tabs: The *Pages* tab contains a tree structure representing all open projects, and their pages and popup pages. Use the Pages tabs to open/edit the pages / popup pages in each project. The *Function Maps* tab allows you to view/edit the function codes associated with each page.
- **Properties Bar** - Contains four tabs:
  - the *General* tab allows you to view/edit general (non-state oriented) button properties (see *General Properties* on page 235)
  - the *Programming* tab allows you view/edit programming-oriented properties (see *Programming Properties* on page 247)
  - the *States* tab allows you to view/edit button state information (see *State Properties* on page 251)

- the *Events* tab allows you to assign Gestures and Page Flips (see *Event Properties* on page 257)
- **State Manager** - Displays each state of the selected button as a thumbnail image in this window. The State Manager window allows you to view/edit the various states of a selected button. See *States* on page 261.
- **Design View windows** - Each tabbed Design View window represents a Touch Panel page, Popup Page or Application window. Note that a MDI tab is displayed for each opened page. Use the **View > MDI Page Tabs** option to toggle the page tabs. When this option is enabled, a tab is displayed for each open page. These tabs normally appear along the bottom edge of the Design View window area, but can be moved, toggled or modified via the MDI tabs context menu (right-click inside any of the tabs to open).
- **Status Bar** - By default, the Status Bar shows the current XY cursor position, and (for the button the mouse is over): Channel code, Address code, Level code, button size, panel revision, Feedback and Initial page file target information. The status bar also indicates the status of your connection to the NetLinx Master.

Supported G5 Panels and Screen Resolutions

Supported G5 Panels and Screen Resolutions	
MXT-2001-PAN	• 20.3" Modero X® Series G5 Panoramic Tabletop Touch Panel • Resolution: 1920x800 (landscape)
MXD-2001-PAN	• 20.3" Modero X® Series G5 Panoramic Wall Mount Touch Panel • Resolutions: 1920x800 (landscape) / 800x1920 (portrait)
MXT-1901-PAN	• 19.4" Modero X® Series G5 Panoramic Tabletop Touch Panel • Resolution: 1920x530 (landscape)
MXD-1901-PAN	• 19.4" Modero X® Series G5 Panoramic Wall/Flush Mount Touch Panel • Resolutions: 1920x530 (landscape) / 530x1920 (portrait)
MXT-1001	• 10.1" Modero X® Series G5 Tabletop Touch Panel • Resolution: 1280x800 (landscape)
MXD-1001	• 10.1" Modero X® Series G5 Wall Mount Touch Panel • Resolutions: 1280x800 (landscape) / 800x1280 (portrait)
MXT-701	• 7" Modero X® Series G5 Tabletop Touch Panel • Resolution: 1024x600 (landscape)
MXD-701	• 7" Modero X® Series G5 Wall Mount Touch Panel • Resolutions: 1024x600 (landscape) / 600x1024 (portrait)

NOTE: *TPDesign5 is specifically intended for use with AMX G5 Touch Panels. To work with G4 Touch Panels, use TPDesign4, or use the G4Utility (included with TPD5) to convert a G4 project to a G5 project for use in TPDesign5.*

Related AMX Software

The related AMX software applications described below are available to download from www.amx.com:

G4Utility (TPD4-to-TPD5 Conversion)

TPDesign5 is not backward-compatible with TPDesign4 - TPD4 project files must be converted in order to be compatible with TPDesign5 and G5 touch panels. The conversion of TPD4 projects to TPD5 projects is accomplished via the *G4Utility* software, available in the TPDesign5 Tools menu. Refer to the *G4Utility (TPD4-to-TPD5 Conversion)* section on page 301 for details.

NetLinx Studio

NetLinx Studio is a full featured Integrated Development Environment (IDE) for NetLinx and Axcess Control Systems. NetLinx Studio is available for free download from www.amx.com. Refer to the NetLinx Studio online help and Instruction Manual for instructions.

Web Update

The *Web Update* program is a stand-alone application that communicates with the AMX website that allows a user to select from a list of available AMX Software programs to choose for updating. Web Update determines the latest version of the selected applications, returns a listing of available updates, allows a user to download the selected installation files, and upon request, launches the installation of those downloads.

- Select **Help > Web Update** to launch the Web Update application. If not found, TPD5 will prompt you to download the application from www.amx.com.
- Refer to the WebUpdate on-line help for details and instructions.

VisualArchitect

VisualArchitect is an intuitive drag-and-drop interface for programming meeting rooms, classrooms, home theaters, and other single NetLinx Control system environments. The comprehensive system design platform reduces training requirements and increases the productivity and efficiency of an organization by simplifying the generation of programming, touch panel files, IR codes and system documentation for thousands of AMX and third-party devices. Even entry-level programmers can build their own system by starting with one of the System Design Library (SDL) modules and customizing as needed with simple point and click commands.

- VisualArchitect is available for purchase from www.amx.com.
- Refer to the VisualArchitect online help and Instruction Manual for instructions.

TPDesign5 Project Files

Overview

TPDesign5 project files (*.TP5) contain all of the information required to define the user-interface that is displayed on the touch panel, including Pages, Popup Pages, Sub-Pages, Application windows and Buttons, as well as all image and sound files that are used in the design and function code information.

NOTE: *All image and sound files that will be used in a project must be imported in to the project via the Resource Manager. See Resource Manager on page 31 for details.*

- Use the *New Project Wizard* to create new TP5 Projects - see *Creating a New TPD5 Project* (below) for details.
- There are several ways to open an existing TP5 project:
  - Select **File > Open** and select a TP5 file via the *Open* dialog.
  - Select a project from the listing of Recently Opened Files in the File menu
  - Drag and drop a *.TP5 file from Windows Explorer onto the TPDesign5 workspace.

NOTE: *Projects created in previous versions of TPD5 will be updated to the current project format when opened. After this migration, the project cannot be opened by prior versions of TPD5. In this case, the application will prompt you to verify this action before proceeding with the migration process.*

Creating a New TPD5 Project

The *New Project Wizard* steps you through the process of creating a new project file (also known as a Panel file), complete with a start page. Use the New Project Wizard as a shortcut to starting new TPD5 projects.

1. Select **File > New** To open the *New Project Wizard - Step 1* dialog (FIG. 2):
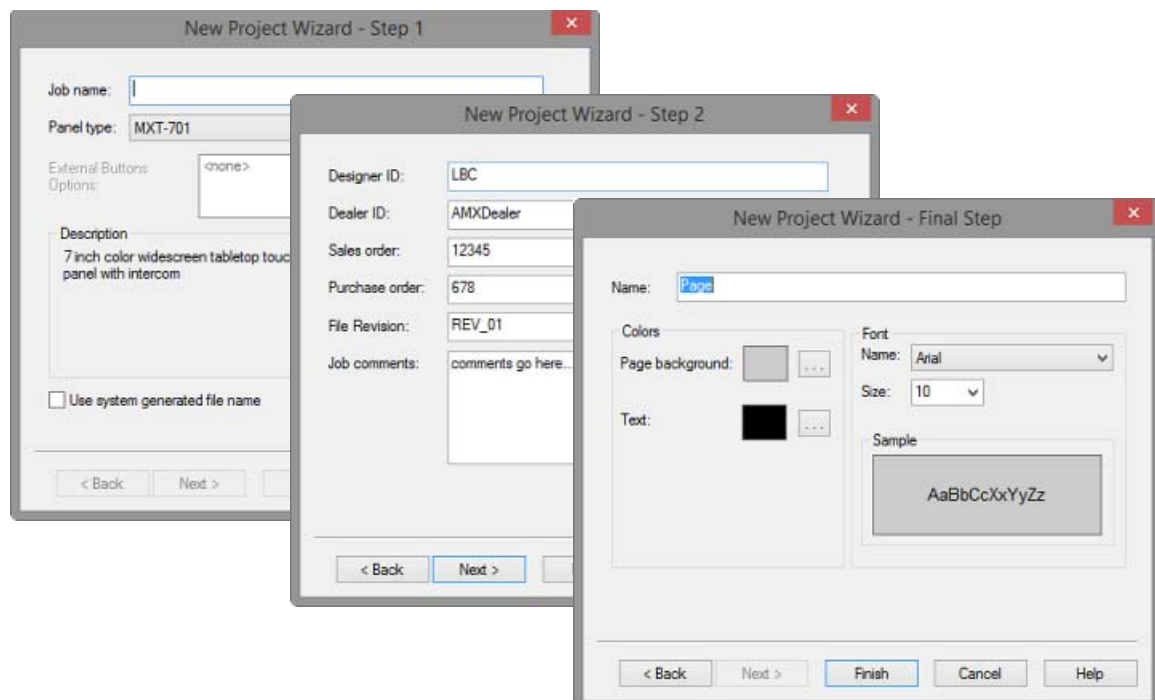
FIG. 2 New Project Wizard dialogs

2. Fill in the fields in this dialog:
   - **Job Name**: Enter a name for the new project. Avoid using inappropriate file name characters:

|	Vertical Bar	**/**	Forward Slash
?	Question Mark	****	Back Slash
*	Asterisk	**"**	Double Quotes
<	Less Than Sign	**:**	Colon
>	Greater Than Sign	**.**	Period

   - All name collision checks in the program are case insensitive.
   - TPD5 has a limit of 59 characters for the project name/filename. If you try to save with a longer name, TPD5 will automatically truncate the name to 59 characters.
   - **Panel Type**: Click the down-arrow to view a list of supported panel types.

- **External Button Options**: This option is disabled at this time.
- **Resolution**: If the Panel Type selected supports multiple screen resolutions, click the down-arrow to select a target resolution for the project. This is usually determined by the option of a touch panel having landscape and portrait models.
- **Use System Generated File Name**: Click this checkbox to use a System Generated File Name for this project.

   NOTE: *The Panel Type and Resolution selections cannot be modified once the panel project has been created. To change either of these, use the Save As Different Panel Type option (File Menu). See Converting a Project to a Different Panel Type on page 27 for details.*

3. Select **Next** to proceed to the *New Project Wizard - Step 2* dialog (see FIG. 2 on page 21). This dialog collects the information that is used for System-Generated File Names.

   If the *Use System Generated File Names* option (in the Step 1 dialog) is not selected, this dialog is skipped since the information entered here would not apply. When you utilize System-Generated File Names, the resulting filename for this project file consists of each of these entries, separated by commas.

   - These fields are all optional.
   - If you leave any of the fields blank, they are simply omitted from the file name.

4. Fill in the fields in this dialog:
   - **Designer**: Enter the name of the project designer in this field.
   - **Dealer ID**: Enter the Dealer ID in this field.
   - **Sales Order**: Enter the Sales Order number in this field.
   - **Purchase Order**: Enter the Purchase Order number in this field.
   - **File Revision**: Enter the File Revision identifier in this field.
   - **Job Comments**: Enter any project-related comments in this field.

5. Select **Next** to proceed to the *New Project Wizard - Final Step* dialog (see FIG. 2 on page 21). The options in this dialog allow you to set up the initial touch panel page in the project.

6. Fill in the fields in this dialog:
   - **Name**: Enter a name for the initial touch panel page in the project.
   - **Colors** (*Page background* and *Text*): Click the browse button (...) to open the *Colors* dialog, where you can select the colors for the page background and text.

   NOTE: *TPDesign5 supports custom palettes. See Working With Colors and Palettes on page 290 for details.*

   - **Font**: Click the down arrow to select the Page Font, from a list of all available fonts on your PC.
   - **Font Size**: Click the down arrow to select the Page Font size.

7. Click **Finish** to close the *New Project Wizard*. The new project is opened in the Workspace window (*Pages* tab), and the initial touch panel page is opened in a Design View window (FIG. 3):
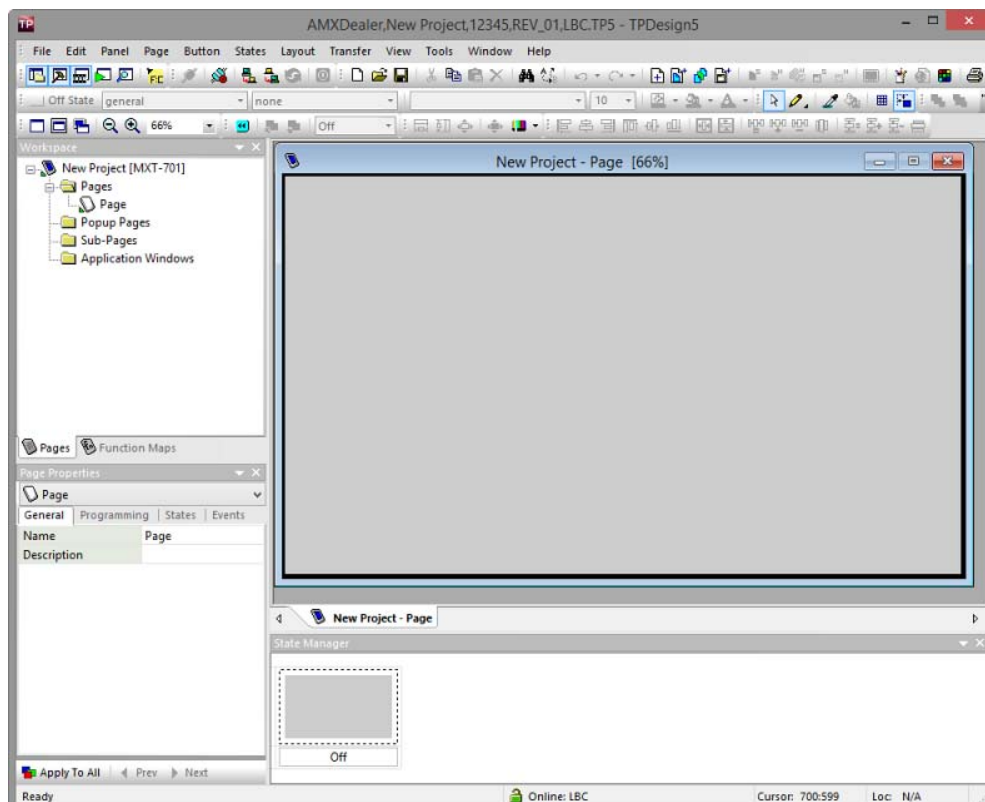
FIG. 3 New Project created

Setting Project Properties

Select **File > Project Properties** to open the *Project Properties* dialog. This multi-tab dialog provides options to view and edit properties for the active project. Depending upon the panel selected, each panel will support all or part of the following properties at the panel level:

Project Properties dialog - Project Information tab

Includes basic project information as well as options for applying password protection to your project file and the *Use system generated filenames* option (FIG. 4):

FIG. 4 Project Properties dialog - Project Information tab

Project Properties dialog - Project Information tab options	
Job Name, Designer ID, Dealer ID, Sales Order, Purchase Order	Use these fields to view/edit basic project information. Click and type to edit these fields as desired.
Created, Modified,	These read-only fields indicate when the project file was \created and last modified.
File Revision, Revision Date, File Name, Job comments	Use these fields to view/edit file revision information for the project file. Click and type to edit these fields as desired.
Protection	Select a level of protection for this project file from the drop-down menu: • *none (default setting)*: with *none* selected, this file is unprotected - anyone can open and edit this project file. • *read-only*: the project file can be opened, but requires a valid password to edit. • *locked*: a valid password is required to open the project file. Refer to the *Applying Password Protection to a Project File* section on page 26 for details.
Password, Confirm	Enter and confirm the password required for protected files. Note that these fields are enabled only if *read-only* or *locked* is selected.
Use system generated file name for this project	With this option selected, the filename for this project file will consist the Job Name, Designer ID, Dealer ID, Sales Order and Purchase Order entries, separated by commas.

Project Properties dialog - Panel Setup Information tab

Includes touch panel setup options including Refresh Frequency, Panel Strings, and Power-up and Inactivity Settings (FIG. 5):

FIG. 5 Project Properties dialog - Panel Setup Information tab

Project Properties dialog - Panel Setup Information tab options	
Panel Type, Screen size	The read-only fields indicate the panel type and screen size that this project is currently designed for. For information on changing the panel type for this project, see the *Converting a Project to a Different Panel Type* section on page 27.
Refresh Frequency	For touch panels that support multiple refresh frequencies, select a refresh frequencies from the drop-down list of supported frequencies.
Panel Strings	Use these fields to specify text strings to appear on the panel, during each condition (*Startup*, *Wakeup*, *Sleep*, *Stand-By* and *Shutdown*). For example, if you entered "Hello!" for the Wakeup string, the panel will send this string to the NetLinx Master on wakeup. *Note: In order to receive strings from a device, the programmer typically creates a buffer or adds a DATA_EVENT/ STRING: handler for the device. This will cause an RXON command to be sent to the device, then the master will pass the strings from the device.*
Power up/Inactivity Settings	The use of dynamic list tables with channel codes may require the designer to specify the ports and/or channels to be allocated on the panel, if the ports/channels utilized in the dynamic table(s) will exceed those declared in the project. These options include: • **Power up page**: Select the initial startup page for the panel from the drop-down list of all pages currently saved in this project. Refer to the *Setting a Power Up Page* section on page 54 for details. • **Power up popups**: Select the popup pages that will appear over the Power up page from the drop-down list of all popup pages currently saved in this project. Refer to the *Setting Power Up Popup Pages* section on page 63 for details. • **Inactivity page**: Click the down arrow to select which page to flip to after the specified period of inactivity (as set on the touch panel). Refer to the *Setting an Inactivity Page Flip* section on page 54 for details. • **Feedback blink rate (10th of second)**: Set the blink rate for feedback on the panel in .10-second increments (default = 5 or one-half second).

Project Properties dialog - Panel Setup Information tab options (Cont.)	
Device Port/Channel Allocation	The use of dynamic list tables with channel codes may require the designer to specify the ports and/or channels to be allocated on the panel, if the ports/channels utilized in the dynamic table(s) will exceed those declared in the project. These options include: • **Automatically calculate port allocation**: Select to have TPDesign5 calculate the maximum ports required (default setting). • **Specify port allocation**: Select to supply a specific port number to be allocated to dynamic list tables. When this option is selected, a text field is provided for you to manually enter the port number to be allocated. • **Automatically calculate channel allocation**: Select to have TPDesign5 calculate the maximum channels required (default setting). • **Specify channel allocation**: Select to supply a specific channel number to be allocated to dynamic list tables. When this option is selected, a text field is provided for you to manually enter the channel number.

Project Properties dialog - Sensors tab

Includes Channel/Level and Port settings for Light Sensors, Motion Sensors, Battery Levels, Voice Over IP, and Cradle Sensors (FIG. 6):

FIG. 6 Project Properties dialog - Sensors tab

Project Properties dialog - Sensors tab options	
Light Sensor	Use these fields to specify the Level and Channel port/code assignments for the on-board light sensor.
Voice over IP	Use this field to specify the command port to be used for VoIP (default = 1).
Motion Sensor	Use these fields to specify the Channel port/code assignments for the on-board motion sensor.
Battery Levels	Use these fields to specify the Level port/code assignments for an on-board battery charger, if the panel uses a battery charger. If the touch panel does not, these fields will be disabled.
Cradle Sensor	Use these fields to set the Channel port/code assignments for the Cradle Sensor on touch panels that use a charging cradle. The channel will be turned on when the panel is docked, either in a tabletop docking station or in a wall cradle. If the touch panel does not use a charging cradle of some sort, these fields will be disabled.

Working With Multiple Projects

TPD5 supports working on multiple projects simultaneously:

Select **File > Open** to open as many Projects as desired. You can copy/paste pages, popup pages and buttons across projects. Each Project is indicated in the Workspace Bar (Pages tab), and edit focus is indicated with a small green icon next to the Project that currently has edit focus (FIG. 7):
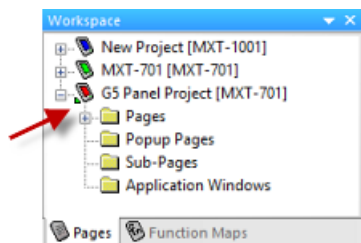
FIG. 7 Workspace Bar - Multiple Projects/Project Edit Focus

Applying Password Protection to a Project File

TPD5 supports two levels of password protection for project (.TP5) files:

- **Read-only** - the next time this project file is opened, the *Enter Access Password* dialog appears, prompting the user to enter the correct password in order to gain write access to the file. The password is not required to open and view the file. Note that if a read-only file is opened without using the password, it cannot be saved under a new file name.
- **Locked** - the next time the panel file is selected to open; the *Enter Access Password* dialog appears, prompting the user to enter the correct password to open the file. If the password is not entered correctly, the file will not be opened at all.

NOTE: *These protection options are not windows file attributes, and are only relevant within the context of the TPD5 application.*

1. Select **File > Project Properties** to open the *Project Properties* dialog - *Project Information* tab.
2. Click the down-arrow next to **Protection** to select a level of password protection from the drop-down list (either read-only, locked or none).
3. Enter the password in the *Password* text field. Passwords can be from 1 to 259 characters in length.
4. Re-type the password in the *Confirm* field. If the passwords don't match, a "*Passwords do not match*" message box is displayed, in which case you'll need to re-enter the password, or re-confirm the password, or both.
5. Click **Apply** to save the changes and apply the specified password to the project file.

Generating the Programmer's Report

Select **Panel > Generate Programmers Report** to generate a report that summarizes various aspects of the active project. This selection invokes the *Generate Programmer's Report* sub-menu, where you can select a file format for the generated file (FIG. 8):
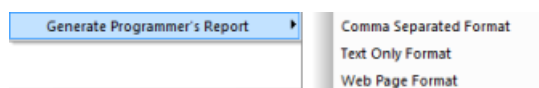
FIG. 8 Panel > Generate Programmer's Report sub-menu

- **Comma Separated Format** - This option produces a .CSV file (compatible with most spreadsheet applications, including Microsoft Excel).
- **Text Only Format** - This option produces a .TXT file (compatible with most text editor applications, including   Microsoft Notepad).
- **Web Page Format** - This option produces a .HTML file (compatible with most web browser applications, including Internet Explorer).

When you select a format option, TPD5 generates the Programmer's Report and opens it in the default application according to the file type. An example (in Web Page format) is shown below (FIG. 9):
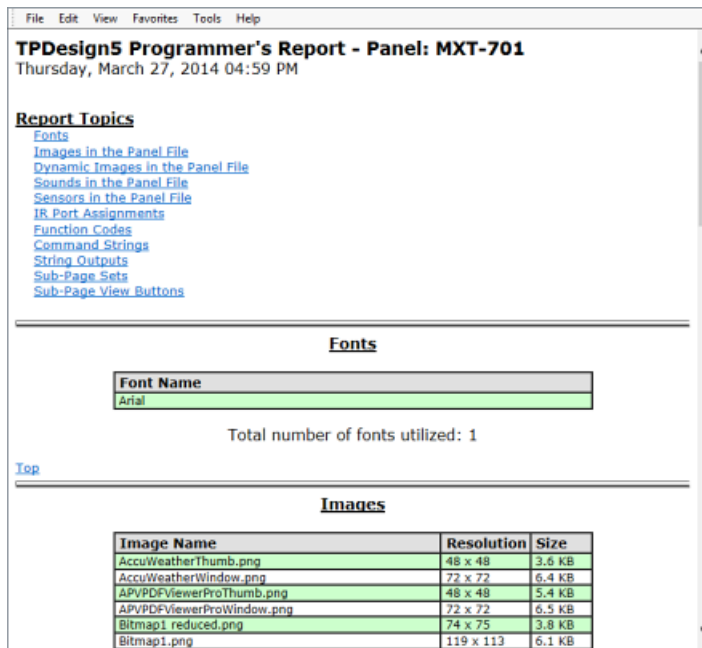
FIG. 9 Example Programmer's Report (Web Page format)

Converting a Project to a Different Panel Type

TPD5 allows you to save the active panel project to be compatible with a different panel type than was specified when the file was created.

NOTE: *TPDesign5 is not backward-compatible with TPDesign4 - TPD4 project files must be converted in order to be compatible with TPDesign5 and G5 touch panels. The conversion of TPD4 projects to TPD5 projects is accomplished via the G4Utility, available in the TPDesign5 Tools menu. See the G4Utility (TPD4-to-TPD5 Conversion) section on page 301 for details.*

1. Open the project (.TP5) file that you want to convert (**File > Open**). The panel that this project was designed for is indicated in the *Panel Setup Information* tab of the *Project Properties* dialog.

2. Select **File > Save As Different Panel Type** to open the *Save As Different Panel Type* dialog (FIG. 10):
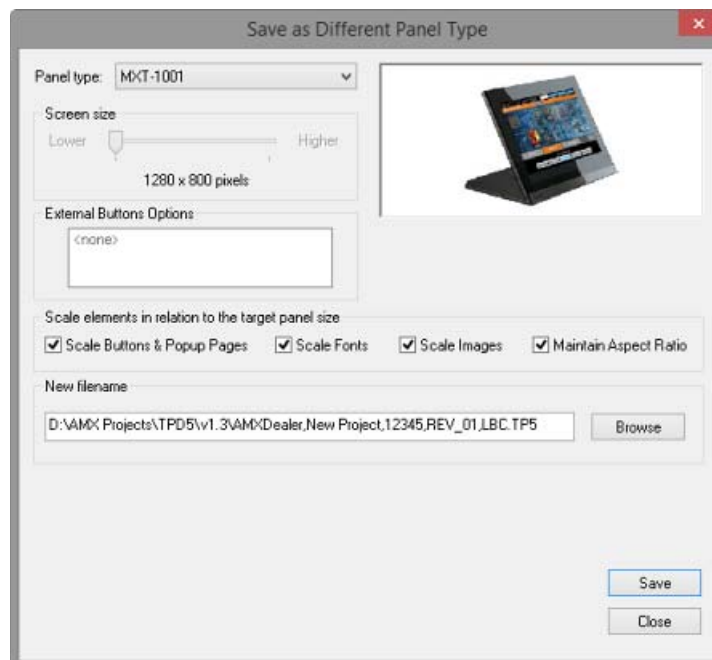
FIG. 10 Save As Different Panel Type dialog

3. Click the down arrow next to *Panel Type* to access the drop-down list of supported panel types. Note that this field initially shows the panel type currently associated with this project.

4. Select a the desired panel type from the list.

NOTE: *If the selected panel type has a single resolution, the Screen Size field is read-only. If the selected panel supports multiple resolution settings, move the Screen Size slider to select the desired resolution.*

5. Use the *Scale Elements in relation to the target panel size* set of options to specify how to scale various elements of the project to fit the target panel (Scale Buttons & Popup Pages, Scale Fonts and/or Scale Images). These options may be enabled/disabled independently of each other.

   NOTE: *By default, the program scales all elements of the panel file, which should produce the best overall effect. Note that scaling an image to a greater size produces more artifacts in the final image than reducing the image.*

6. Use the *New Filename* field to rename the project file, if desired. Use the *Browse* button to navigate to a different target directory if you need to.

Cut, Copy and Paste

TPD5 allows you to cut, copy and paste Pages, Popup Pages, Sub-Pages, Application windows and Buttons (including some or all of their attributes) within a project, or across Projects. This can obviously be a major time saver, and there are a few key points to keep in mind in doing so.

● Use the *Paste Controls* dialog to specify whether to retain Function Codes (Address, Channel and Level codes), Page Flip Options, and Images and Sounds when the Page, Popup Page, Sub-Page, Application window or Button(s) are pasted into a separate project. This a powerful tool. Give some thought to which elements of the copied TPD5 element you want to retain in the target project.

● When you copy/paste across projects, if the selected TPD5 elements contain bitmaps and/or sounds, they are copied into the target project along with the Page, Popup Page, Sub-Page, Application window or Button(s) (assuming that Retain image references and Retain sound references are all selected in the Paste Controls dialog).

● When a TPD5 element are pasted into a target project, the bitmaps and sounds that came over with the pasted element are available in the Resource Manager for the target project.

● If you copy/paste a Page, Popup Page or Sub-Page containing buttons, the buttons are copied as well.

NOTE: *Cut and Copy works on the TPD5 Element that currently has the Edit Focus. For Pages Popup Pages, Sub-Pages and Application windows, edit focus is indicated in the Workspace Navigator (Pages tab) by a small green arrow at the lower-left corner of the icon for the selected element. For buttons, edit focus is indicated in the Design View window with small red squares on the edges of the selected button. See Edit Focus on page 29 for more details.*

Cut, Copy and Paste Controls

Cut, Copy and Paste controls
in the Standard Toolbar

Cut, Copy and Paste controls
in the Edit Menu
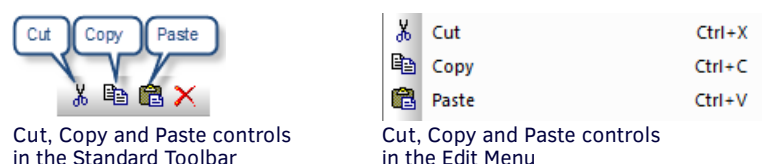
FIG. 11 Cut, Copy and Paste Controls

Note that the shortcut keys (for all functions) are indicated in the menus.

● **Cutting** - To Cut a selected TPD5 Element to clipboard memory, select the element in the Workspace window and select Cut. The program will prompt you to verify any Cut action before the selected element is removed from the project. Shift+click to select multiple Buttons in a Design View to cut.

● **Copying** - To Copy a selected TPD5 Element to clipboard memory, select the element in the Workspace window and select Copy. Shift+click to select multiple Buttons in a Design View to copy.

● **Pasting** - To paste Pages, Popup Pages, Sub-Pages and Application windows from clipboard memory, select the target project in the Workspace window (Pages tab) and select Paste. The element will automatically be pasted into the appropriate project folder in the Workspace window. Note that if the element is pasted into a folder that already has an element with the same name, the existing element is not overwritten; rather, the name of the pasted element is modified to indicate that it is a copy of another element (i.e. "Page 2 - Copy"):
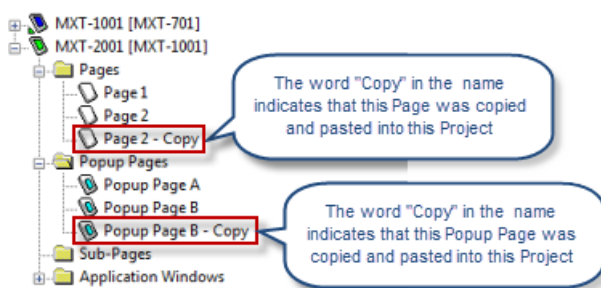
FIG. 12 Cut, Copy and Paste Controls

● To paste Buttons from clipboard memory, select the target Page, Popup Page or Sub-Page and select Paste. The Button(s) are pasted into the target element, in their original positions.

- If a set of copied (or cut) buttons is pasted into a smaller area (for example a Popup Page), the buttons will automatically be resized and re-positioned to fit.
- Note that for Buttons, the attributes of the copied (or cut) buttons are retained, according to the selections made in the *Paste Controls* dialog (FIG. 13).
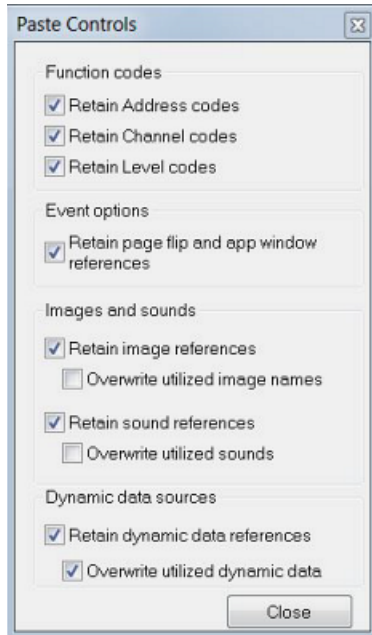
FIG. 13 Paste Controls dialog

- See *Cut, Copy and Paste - Pages* on page 37.
- See *Cut, Copy and Paste - Popup Pages* on page 44
- See *Cut, Copy and Paste - Sub-Pages* on page 47
- See *Cut, Copy and Paste - Application Windows* on page 64

Edit Focus

The element that is the target for change actions is said to have the *Edit Focus*.

For Pages Popup Pages, Sub-Pages and Application windows, edit focus is indicated in the Workspace Navigator (Pages tab) by a small green arrow at the lower-left corner of the icon for the selected element (see "*Page 1*" in FIG. 14):
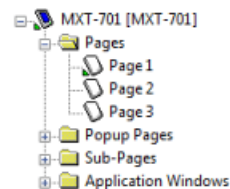
FIG. 14 Edit Focus - Workspace window

For buttons, edit focus is indicated in the Design View window with small red squares on the edges of the selected button (FIG. 15):
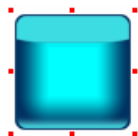
FIG. 15 Edit Focus - Design View windows

Note that the last thing selected (not necessarily a button displayed in the Design View windows) has the edit focus.

Drag and Drop Support

- **Project Files**: Drag and drop TPDesign5 project files (*.TP5) from a windows Explorer window onto the TPDesign5 workspace to open the project.

NOTE: *You can also drag and drop TPDesign4 (*.TPD) project files into the workspace, to open the file via the TPD Conversion Wizard.*

- **Popup Pages**: Popup Pages can be dragged from the Pages tab and dropped on a Design View as an alternate method of displaying the Popup Page on the full sized Page.

NOTE: *Dragging a Popup Page onto another Popup Page is not supported.*

- **States**: Copies of states can be added within the same button via drag-and-drop. See Adding States via Drag-and-Drop for details.
- **Properties**: Drag and drop properties from the Properties window onto elements in the Design View to apply the selected property.

Undo/Redo Support

TPD5 supports full Undo / Redo functionality at the panel level. This means that each panel file that is open in the program maintains a separate undo/redo stack and manages this for the user. You may either click the Undo or Redo toolbar icons, or click **Edit > Undo** or **Edit > Redo**.

Click the down-arrows to view a history list that lists up to 25 of the most recent actions (FIG. 16):
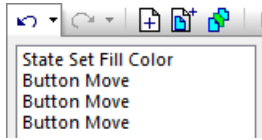
FIG. 16 Undo List

Click to select one or more actions to be undone/redone.

- The commands alter their text dynamically to indicate which action is next in the undo/redo list. Selecting either option will immediately undo/redo the last action.
- These actions are independent of any file saves, therefore you may undo past a save if you so desire.
- Once a panel file is closed, its undo/redo stack is flushed and is no longer available.

The following actions may be undone/redone:

- Button Property Changes
- Page / Popup Page Renames
- Button Deletions / Cuts
- Button Creations / Pastes
- Page / Popup Page Deletions / Cuts
- Page / Popup Page Creations / Pastes
- State Deletions / Cuts
- State Creations / Pastes
- Copying/Cutting/Pasting images and sounds

NOTE: *Use the options in the Undo/Redo tab of the Preferences dialog to disable/enable the Undo/Redo system.*

Resource Manager

Overview

All image and sound files as well as dynamic data sources in the TPD5 project are managed via the Resource Manager (**Panel > Resource Manager**). The *Resource Manager* dialog provides the ability to import image and sound files so that they can be applied to various panel elements in the active project.

Once a image file, dynamic image source, sound file or dynamic data source has been added to the Resource Manager, it is available to be applied to any panel element, and is available to share among any number of elements.

NOTE: *All Image, Dynamic Image and Sound files as well as Dynamic Data Sources must be imported into the Resource Manager before they can be used in the Project.*

The Resource Manager dialog consists of four tabs:

- Resource Manager dialog - *Images* tab (see below)
- Resource Manager dialog - *Dynamic Images* tab (see page 38)
- Resource Manager dialog - *Sounds* tab (see page 43)
- Resource Manager dialog - *Dynamic Data Sources* tab (see page 43)

Images

The *Images* tab of the Resource Manager dialog provides a convenient way to import and preview all image files to be used in your project. Use this tab as a "library" of every image file that will be used in your project - for pages, popup pages, and buttons (FIG. 17):
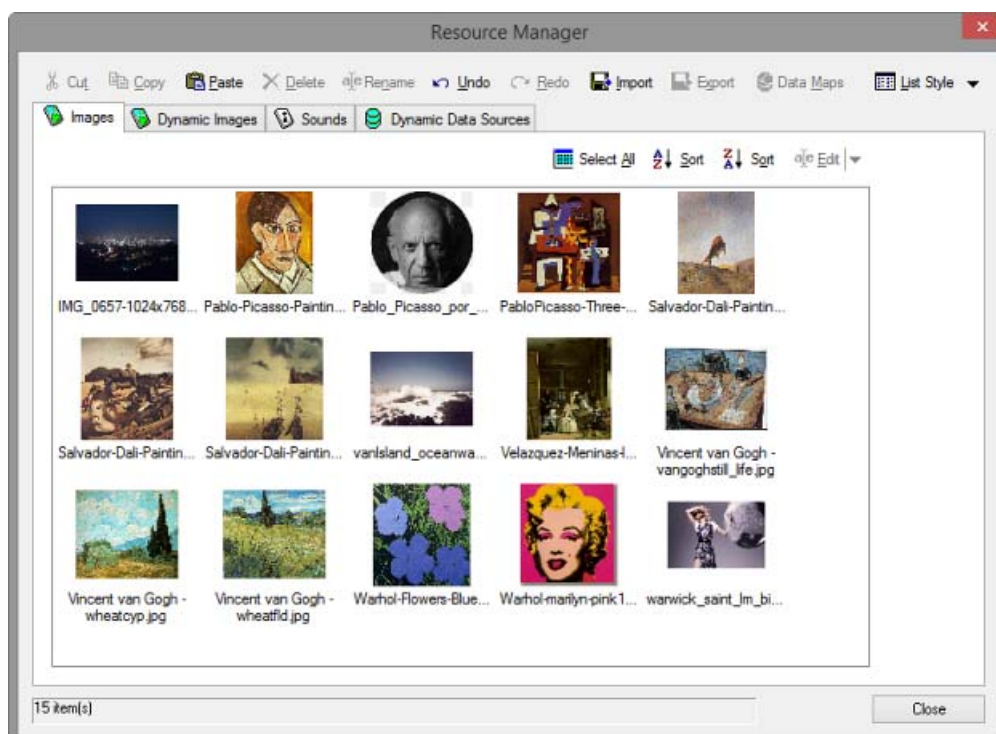
FIG. 17 Resource Manager dialog - Images tab

TPD5 uses one concept for image files: *Bitmaps*.

The term "Bitmap" is a generic term that describes any pixel-based image file. In TPD5, a Bitmap can be any supported image type (not limited to BMP files). Bitmaps and Dynamic Images are applied to Pages, Popup Pages, Sub-Pages and Buttons via the *Bitmap* (State) Property (see page 251).

Multiple Bitmaps can be applied to Pages, Popup Pages, Sub-Pages and Buttons. When multiple bitmaps are assigned to an element, they are displayed in a "stack" in the order specified in the *Bitmaps* dialog. See *Assigning Bitmaps to TPD5 Elements* on page 33

By default, bitmaps are drawn beneath Text.

To use images in your project, they must first be imported into the project via the Resource Manager. See *Importing Image Files Into the Project* on page 32

- *Static Images* (bitmaps) that have been imported into the project are listed in the *Images* tab of the Resource Manager.
- *Dynamic Images* that have been imported into the project are listed in the *Dynamic Images* tab of the Resource Manager.

Supported Image File Types

TPD5 supports the following image file types:

- BMP
- IFF
- JPG
- PCT
- PNG
- TGA
- WMF

NOTE: *Of all the image file formats supported by TPD5, only one accommodates transparency as a color: PNGs. For the transparent color to be interpreted correctly by TPD5, the PNG file must be saved with RGB colors. Unless you are actually using transparency, JPGs are recommended over PNGs, since they are usually slightly smaller in size.*

Importing Image Files Into the Project

1. Select **Panel > Resource Manager** (or click the toolbar button) to open the *Resource Manager* dialog.
2. Open the *Images* tab.
3. Click the **Import** button to invoke the *Open* dialog.

   Click the *Overwrite resources with duplicate names* checkbox (at the bottom of the dialog) to automatically overwrite any existing files that have the same filename. If this option is selected, anytime a file with a duplicate name is imported, it will replace the original file on every button or page that referenced that filename in the project.
4. Locate and select the file(s) to import.
5. Click **Open** to import the files to the *Images* tab.

Notes on Importing Image Files

- When images of any supported file type (except PNG) are imported into a project, the files are automatically converted to JPGs. For this reason, if you import (or paste) a file that has the same name as a previously imported file (even if it has a different extension), TPD5 automatically adds the "copy of" prefix to the file.

    PNGs are not converted because they are already compressed.
- The largest image size supported on the panels is 1280x1024. Any image files that are imported to the project that are larger are automatically scaled down to fit.

Exporting Image Files From the Project

To export image files from Resource Manager dialog to a specified directory:

1. Select **Panel > Resource Manager** (or click the toolbar button) to open the *Resource Manager* dialog.
2. Open the *Images* tab.
3. Select one or more files to export (Ctrl + click to select multiple files individually, or Shift + click to select a range of files).
4. Click the **Export** button. This opens the *Choose Directory* dialog.
5. Use the Browse button to locate a target directory for the files.
6. Click **OK**.

Renaming Image Files

1. Select **Panel > Resource Manager** (or click the toolbar button) to open the *Resource Manager* dialog.
2. Open the *Images* tab.
3. Select an image to rename.
4. Click the **Rename** button to invoke the *Rename* dialog.
5. Enter the new file name in the *New Name* text field.
6. Click **OK** to close the *Rename* dialog.

Deleting Image Files From the Project

1. Select **Panel > Resource Manager** (or click the toolbar button) to open the *Resource Manager* dialog to the *Images* tab.
2. Select one or more files to delete (Ctrl + click to select multiple files individually, or Shift + click to select a range of files).
3. Click the **Delete** button.
   - The files are not deleted from the hard drive, just from this project.
   - If any of the files selected for deletion are used by the active project, the *Resource(s) In Use* dialog is displayed, with a listing of all files targeted for deletion, as well as the Page(s) on which each file is used (FIG. 18).
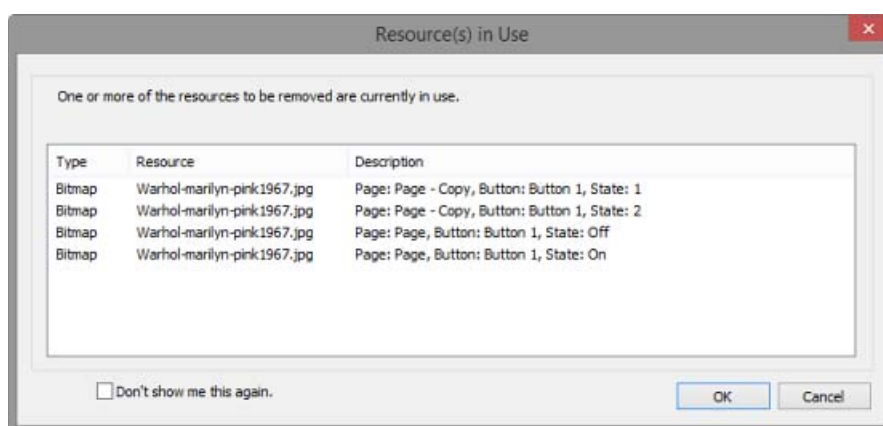
FIG. 18 Resource(s) In Use dialog

By default, the *Resource(s) In Use* dialog is set to display any time a used resource is deleted. However, you can toggle the display of this dialog either by selecting the *Don't show me again* checkbox, or by selecting When deleting resources in use in the Application tab of the *Preference* dialog.

Assigning Bitmaps to TPD5 Elements

TPD5 uses one concept for image files: Bitmaps. Bitmaps can be applied to Pages, Standard Popup Pages, Sub-Pages and Buttons.

NOTE: *The term Bitmap is a generic term that describes any pixel-based image file. In TPD5, a Bitmap can be any supported image type (not limited to . BMP files). Bitmaps and Dynamic Images are applied to Pages, Popup Pages and Buttons via the Bitmap (State) Property.*

Unlike TPDesign4, TPD5 supports assigning multiple (up to five) bitmaps to each state of an element.

When multiple bitmaps are assigned, they are displayed in a "stack" according to the order specified in the Bitmaps dialog. This concept replaces the concept of Icons that were used in TPDesign4 (icons were necessary in TPD4 to control the Draw Order so that an icon could be displayed on top of a bitmap graphic).

As an example, the figure below indicates three simple bitmaps (FIG. 19):

Bitmap 3 image   Bitmap 2 image   Bitmap 1 image
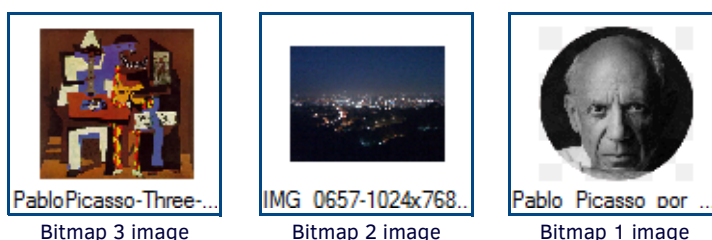
FIG. 19 Example - 3 bitmaps

Using the *Bitmaps* dialog, these three bitmaps can be assigned to a single state of any TPD5 element. As an example, these three bitmaps can be assigned to a Page and arranged to display like this (FIG. 20):
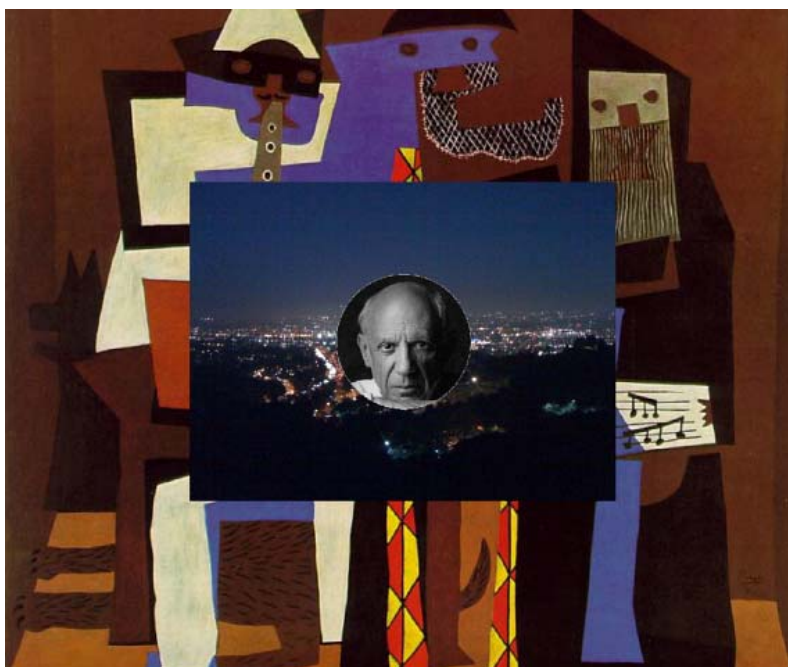
FIG. 20 Example - 3 bitmaps displayed on a Page

Each of the bitmaps has it's own justification settings. In this example, all three bitmaps are set to "Center-middle (see the *Adjusting Bitmap Position* section on page 36 for details).

- Text is drawn on top of bitmap images.
- Static Bitmaps (Images) that have been imported into the project are listed in the *Images* tab of the Resource Manager.
- Dynamic Images that have been imported into the project are listed in the *Dynamic Images* tab of the Resource Manager.

In buttons where the bitmap state is set to "Absolute", the presentation may be adjusted either through the Bitmaps dialog, or through the I*mage/Text Positioning* dialog. See*Adjusting Bitmap Position* on page 36 for details.

Assigning Bitmaps to a Page, Popup Page or Button

NOTE: *In order to apply Bitmaps to Pages, Popup Pages or Buttons, the image files must be pre-loaded in the Resource Manager. See the Importing Image Files Into the Project section on page 32 for details.*

1.   In a Design View window, select the Page, Popup Page or Button to which you want to add or change the image.

2.   In the *States* tab of the Properties window, click on the **Bitmap** property to enable the browse (...) button.

3.   Click the browse button to open the *Bitmaps* dialog (FIG. 21):
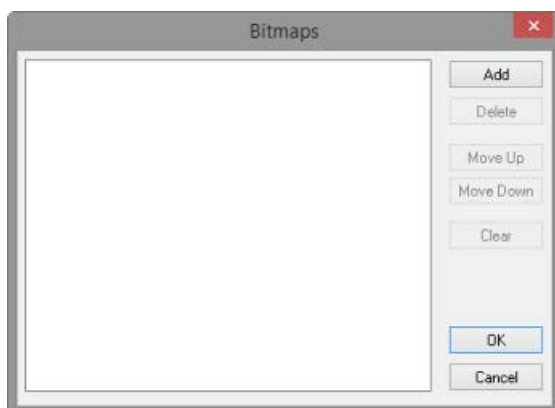
FIG. 21 Bitmaps dialog

4.   Click **Add** to open the *Select Resource* dialog where you can select an image (or Dynamic Image) to apply to the selected page (FIG. 22).
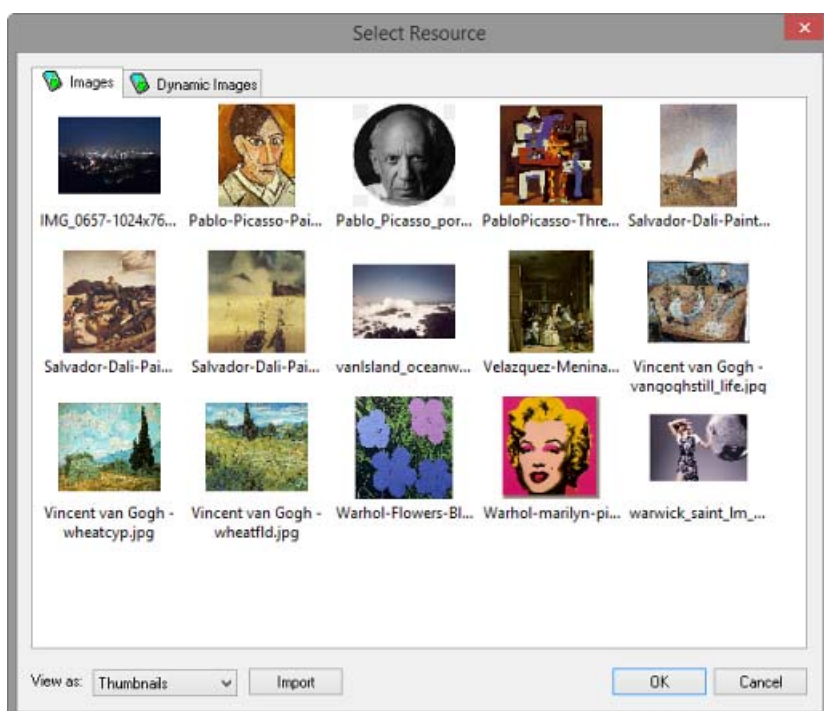
FIG. 22 Select Resource dialog - Images tab

Note that only images that have been imported into the project are listed in the *Select Resource* dialog (see *Importing Image Files Into the Project* section on page 32 for details).

5. Select an image and click **OK** to close the *Select Resource* dialog and return to the *Bitmaps* dialog. The selected image is indicated as **Bitmap 1** in the *Bitmaps* dialog (FIG. 23):
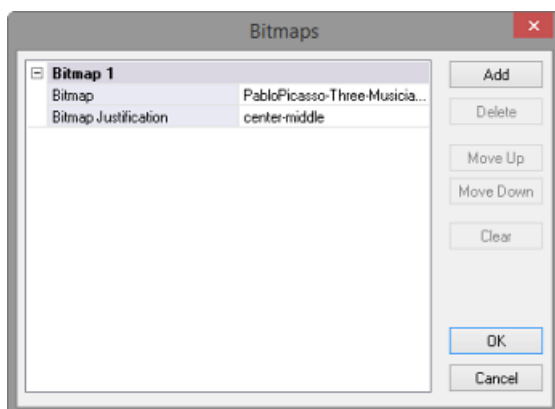
FIG. 23 Bitmaps dialog - Bitmap 1 configured

6. Set the **Bitmap Justification** setting as desired (default = *center-middle*).

To add additional bitmaps to the selected Page, click **Add** again and select another bitmap and set it's justification. Repeat this process to add as many bitmaps to the Page as you need.

When multiple bitmaps are assigned, they are displayed in a "stack" according to the order specified in the Bitmaps dialog: "Bitmap 1" is drawn first, then "Bitmap 2", "Bitmap 3", etc...

For example, the *Bitmaps* dialog shown in FIG. 24 has three bitmaps assigned
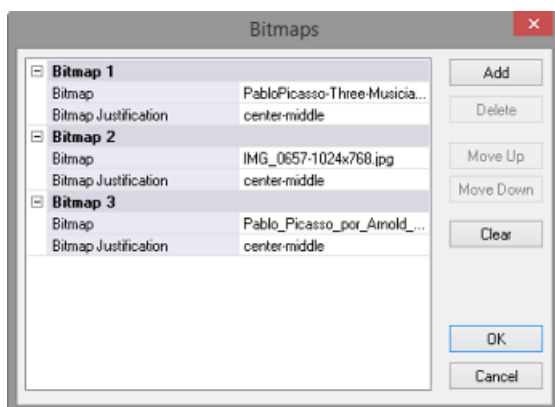
FIG. 24 Bitmaps dialog - three bitmaps configured

In the *Add Bitmap* dialog, the bitmaps are ordered as follows: "Bitmap 1" is drawn first, followed by "Bitmap 2" (which appears on top of Bitmap 1), and "Bitmap" 3 is drawn last, so it appears on top of the others (FIG. 25):
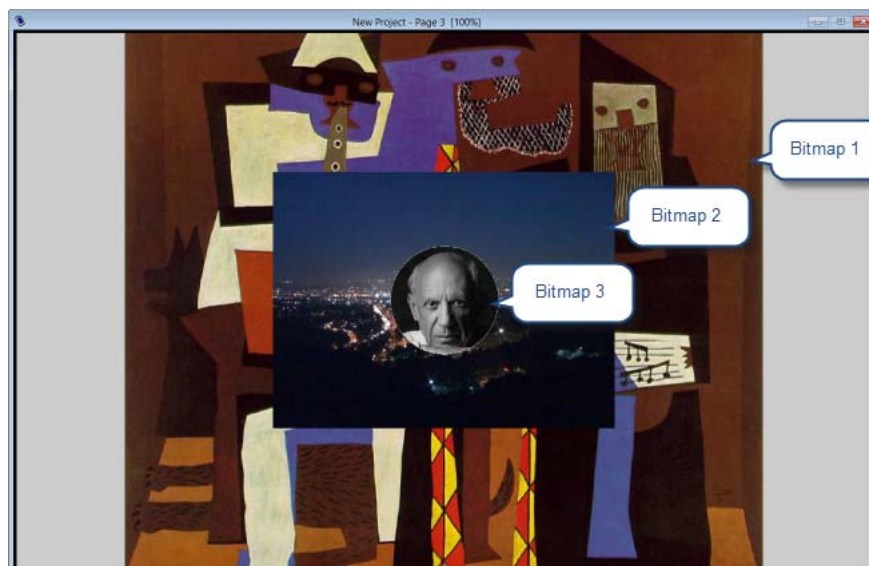
FIG. 25 Example of a Page with three bitmaps

Use this ordering to control the "Z-Order" (or draw order) for multiple bitmaps that are layered on the selected TPD5 element.

7.   Click **OK** to close the *Bitmaps* dialog.

Adjusting Bitmap Position

In Pages, Popup pages, Sub-Pages or Buttons that have images with a Justification setting of "**Absolute**" (as specified in the *Bitmaps* dialog), the positioning of the Bitmap(s) can be adjusted either via the *Bitmaps* dialog or the *Image/Text Positioning* dialog.

Via the Bitmaps dialog

1.   With a Page, Popup page, Sub-Page or Button state selected, click the browse button (...) in the Bitmaps (State) property to open the *Bitmaps* dialog (FIG. 26).

2.   Use the *Bitmap X Offset* and *Bitmap Y Offset* fields to adjust the position of the selected bitmap with pixel values.

    The default values for Absolute Justification are X=0, Y=0. These values will position the selected bitmap in the upper-left corner of the Page, Popup page or Button (the upper left corner of the bitmap is placed at pixel position 0,0 of the element - see FIG. 26):
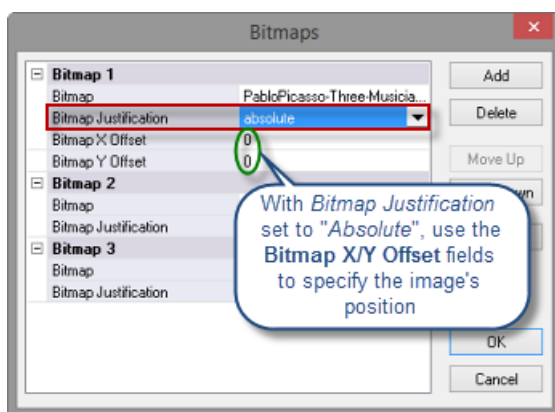
FIG. 26 Bitmaps dialog

Via the Image/Text Positioning dialog

1. Right-click on a Page, Popup page or Button state in the *State Manager* window and select **Image/Text Positioning...** to open the *Image/Text Positioning* dialog (FIG. 27):
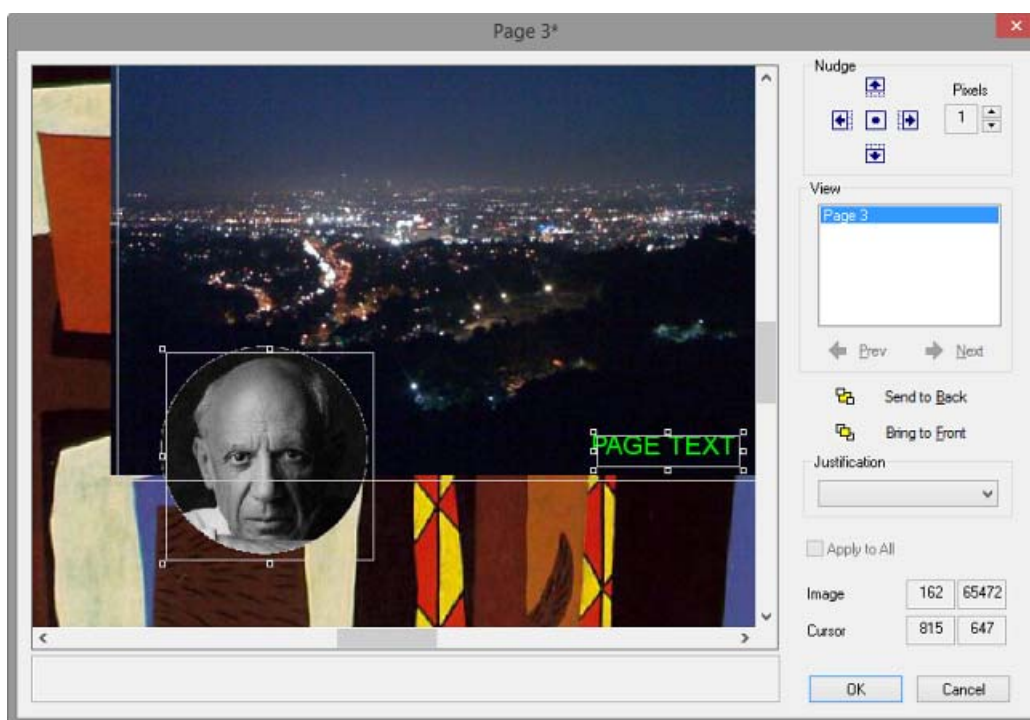
FIG. 27 Image/Text Positioning dialog

2. Click on a bitmap to enable the *Nudge* and *Justification* controls. You can also simply drag and drop the selected bitmap to it's new position. The controls in this dialog also allow you to adjust the draw order of bitmaps (if there are multiple bitmaps) and set justification via drop-down.

3. Note that if there are multiple bitmaps assigned to the selected element, each one can be adjusted individually.

4. When you finish making your changes, click **OK** to save them and close the dialog.

Editing Image Files

In order to edit image and sound files via the TPDesign5 UI, default image and sound editor applications must be defined in the *Editor Selection* tab of the Preferences dialog. See *Adding an External Image Editing Program* on page 45.

1. Open the Images tab of the *Resource Manager* dialog.

2. Select the image file that you want to edit.

3. Click the **Edit** button to launch the external program specified as the default editor for image files.

4. When the edit session begins, the image or sound file is opened in the default editing program, and control is returned to TPD5. Any saved changes to the resource made in the external editor will be immediately reflected in TPD5.

   NOTE: *Any action taken in TPDesign5 that would change the state of the image or sound file being edited (e.g., delete, rename, undo/redo, etc.) will cause the link between TPDesign5 and the external application to be broken. Any subsequent changes made in the external editor will not be reflected nor applied.*

Dynamic Images

The *Dynamic Images* tab of the Resource Manager dialog provides a convenient way to import and preview all dynamic image files to be used in your project. Dynamic Images are images that exist on an HTTP server, external to the panel. This feature requires you to specify a URL in place of image file.

Dynamic images can be refreshed at specified regular intervals or via the **Panel > Refresh Dynamic Images** option.
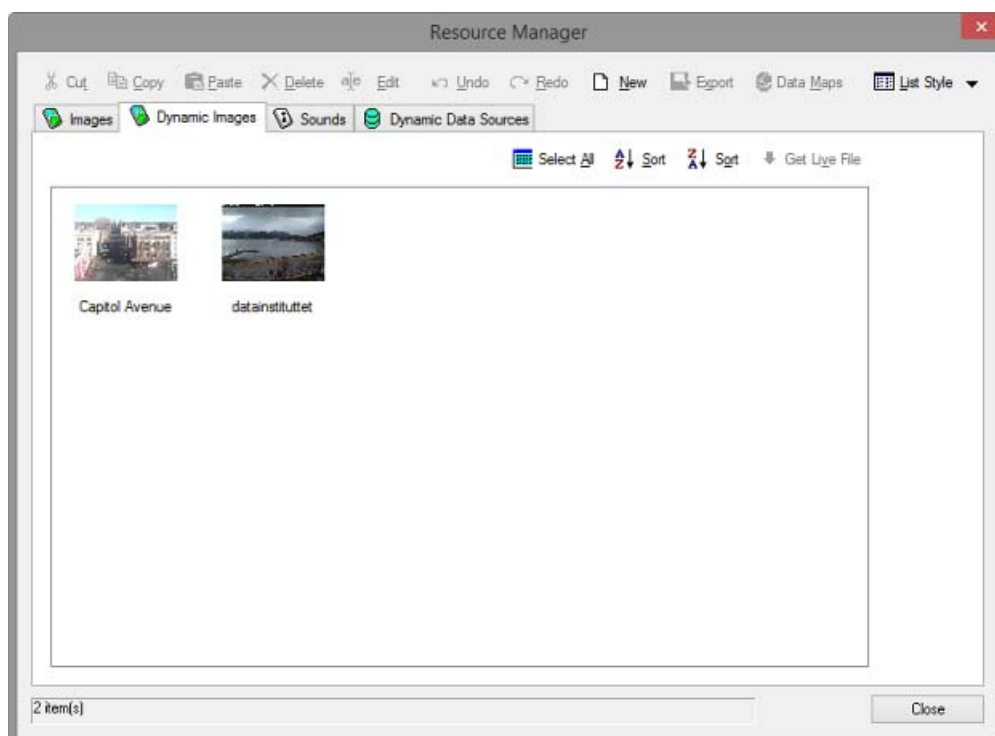
FIG. 28 Resource Manager dialog - Dynamic Images tab

- To use dynamic images in your project, specify a URL rather than a directory path to a static image file.
- Dynamic images can be applied to Pages, Popup Pages, Sub-Pages and all Button types except Text Input buttons.
- The *Dynamic Images* tab of the Resource Manager dialog provides a convenient way to import and preview all dynamic image files to be used in your project.
- Dynamic images can be refreshed at specified regular intervals or via the **Panel > Refresh Dynamic Images** option.
- Use the *Scale Bitmap To Fit* state property (Properties window - States tab) to automatically scale (down only) the dynamic image to fit the button on which it will be displayed.

Adding Dynamic Images to the Project

The following example steps you through the process of adding a Dynamic Image to the Resource Manager.

NOTE: *This example starts with selecting a dynamic image on the Internet, which may not be necessary if you have a specific URL to use:*

1. In your browser, locate the Dynamic Image that you want to use.

2. Copy the Dynamic Image's URL to the clipboard. In this example the image URL is:

    http://kamera.harpefossen.no/mjpg/video.mjpg

    The specific method will depend on your browser. For example Google Chrome provides the option to right-click on the image and select **Copy Image URL**, while IE provides the option to right-click on the image and select Properties to view the image properties including the URL in the *Properties* dialog (FIG. 29):
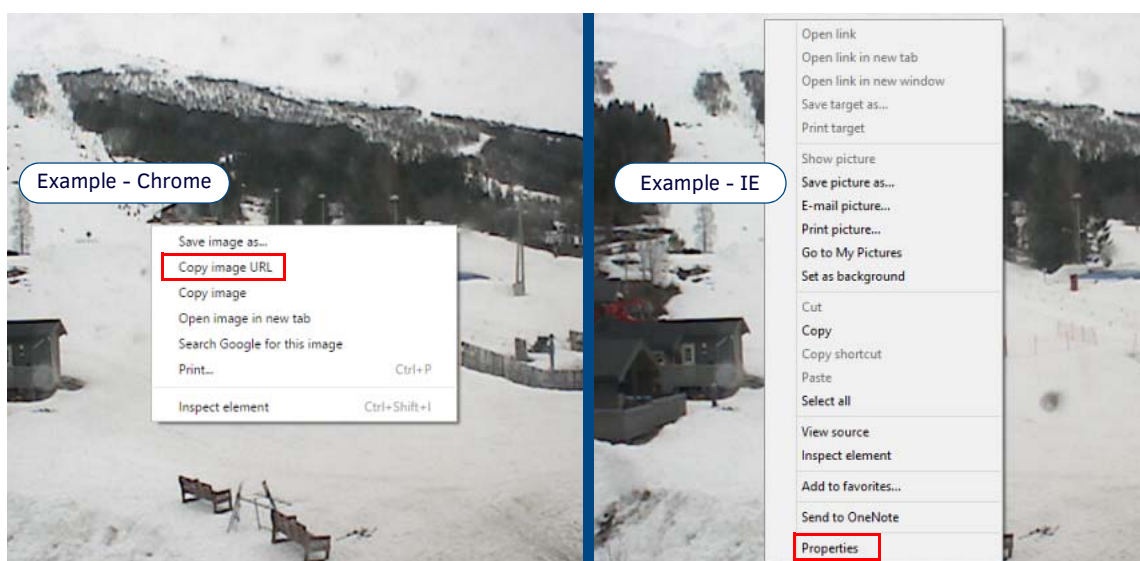
FIG. 29 Example - Dynamic Image (Copy image URL / Properties)

3.   In the *Dynamic Images* tab of the Resource Manager, click **New** to open the *Create Dynamic Image* dialog (FIG. 30):
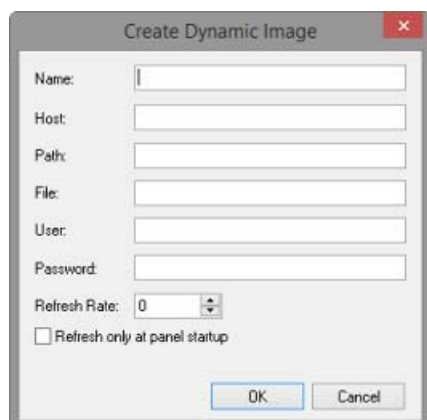
FIG. 30 Create Dynamic Image dialog

4.   In the *Name* field, enter a descriptive name for this Dynamic Image (in this example - "Harpefossen Ski Center").

5.   Paste the copied URL into the *Host* field. Note that the pasted URL consists of not only the Host address information, but also protocol and path information as well, so it necessary to separate it into the different elements represented in the *Create Dynamic Image* dialog.

Here is the full string provided by the image source for this image (FIG. 31):

FIG. 31 Example Dynamic Image URL (full path)

a.   Enter a name for the dynamic image in the

b.   Cut and paste the host portion of the URL into the *Host* field (delete "HTTP://" from the pasted string).

c.   Copy and paste the path portion of the URL into the *Path* field (excluding the backslash). Note that not all dynamic image URLs require or use a path.

d.   Copy and paste the file portion of the URL into the *File* field (excluding the backslash).

e.   The *User* and *Password* fields are left blank, since no credentials are required by the host to access the webcam in this example.

f.   The *Refresh Rate* should be left to zero (default setting), to allow the image to update automatically based on the source's refresh rate. When the Refresh Rate is set to zero, the *Refresh Only at Panel Startup* option is available. This option directs the panel to refresh this image only when the panel is rebooted.

At this point, the *Host* field should include the host information only, and the other fields should be filled in respectively (FIG. 32):
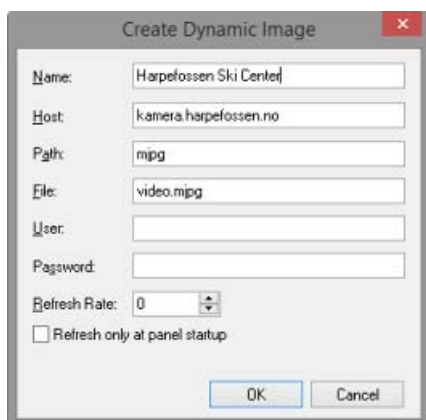
FIG. 32 Create Dynamic Image dialog with example data

6. Click **OK** to close the *Create Dynamic Image* dialog and add the new image to the Dynamic Images tab (FIG. 33):

Dynamic Image Icon   Live File image Icon   Dynamic Image Preview Window
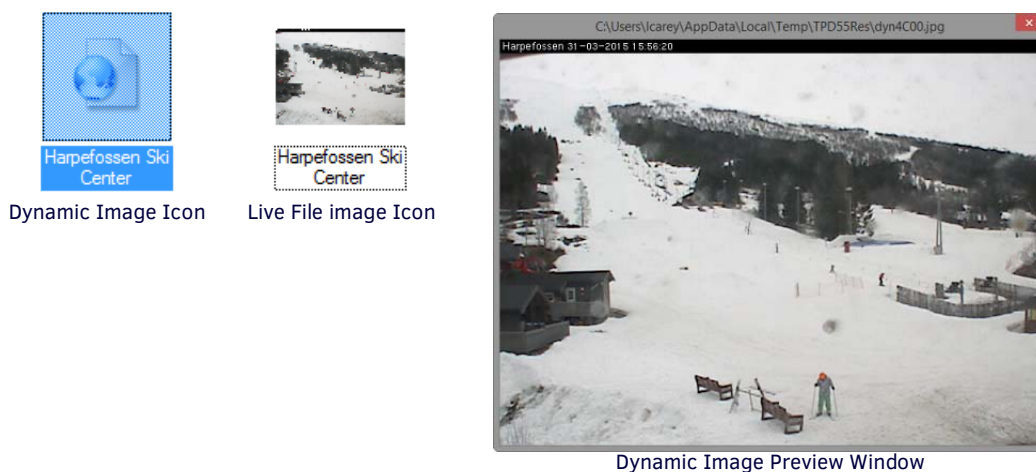
FIG. 33 Resource Manager (Dynamic Images tab) - Example Dynamic Image (icons and preview window)

● Click **Get Live File** to update the dynamic image icon with a preview image (if *List Style* is set to *Thumbnail*).
● Double-click on the dynamic image icon to show the cam feed in a preview window

Now that the dynamic resource has been added to the Resource Manager, it is available to be applied to TPD5 elements (as a Bitmap). See *Assigning Dynamic Images to TPD5 Elements* on page 41 for details.

Editing Dynamic Images

1. Select **Panel > Resource Manager** (or click the toolbar button) to open the *Resource Manager* dialog.
2. Open the *Dynamic Images* tab.
3. Click the **Edit** button to open the *Edit Dynamic Image* dialog (FIG. 36):
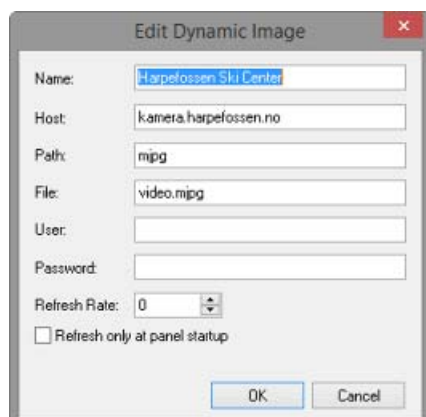
FIG. 34 Edit Dynamic Image dialog with example data

4. Edit the image information as desired.
5. Click **OK** to save changes and close the dialog.

Deleting Dynamic Image Files From the Project

1. Select **Panel > Resource Manager** (or click the toolbar button) to open the *Resource Manager* dialog.
2. Open the *Dynamic Images* tab.
3. Select one or more files to delete (Ctrl + click to select multiple files individually, or Shift + click to select a range of files).
4. Click the **Delete** button.

NOTE: *Deleting a Dynamic Image from the Project cannot be undone.*

Assigning Dynamic Images to TPD5 Elements

The following example illustrates assigning a Dynamic Image to a TPD5 Element (

1. Select a Page, Popup Page, Sub-Page or Button (all types except Text Input).
2. In the Properties window (States tab) - *Bitmaps* property, click the Browse (...) button to open the *Bitmaps* dialog.
3. Click **Add** to access the *Select Resource* dialog, and open the *Dynamic Images* tab (FIG. 35):
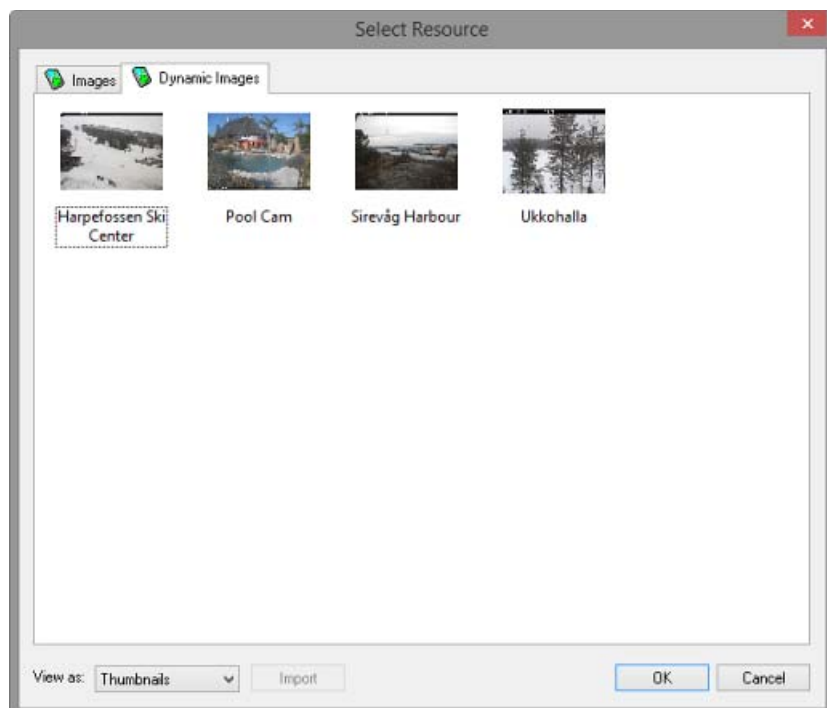
FIG. 35 Select Resource dialog (Dynamic Images tab) indicating four dynamic images in the project

4. Select a dynamic image and click **OK** to close the *Select Resource* dialog and return to the *Bitmaps* dialog, where you can adjust the Bitmap Justification as desired, and add other bitmaps to the selected UI element if desired (FIG. 36):
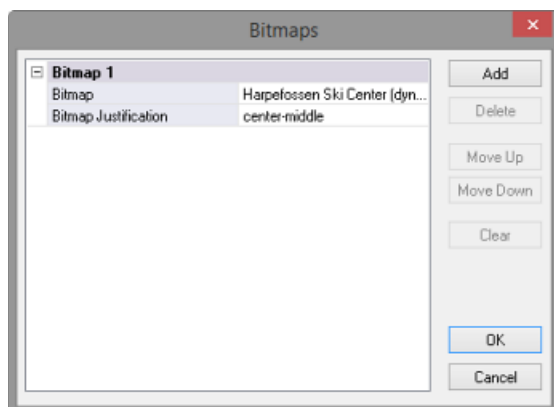
FIG. 36 Bitmaps dialog indicating a dynamic image to be applied to the selected UI element

5. Click **OK** to close the *Bitmaps* dialog - the dynamic image will be represented in the selected UI element. As an example, FIG. 37 shows a button with a dynamic image applied:
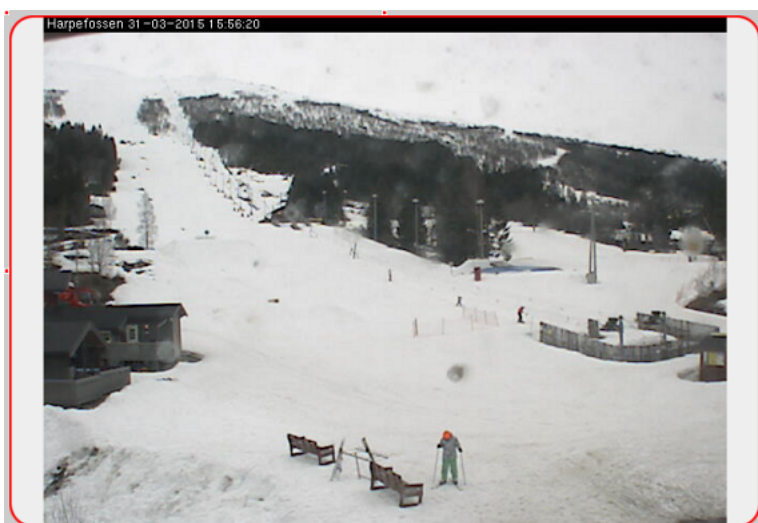
FIG. 37 Example button with a Dynamic Image applied

Dynamic Image Settings - Camera Examples

Like with any other type of equipment AMX controls, manufacturer's documentation and customer support are the most reliable ways of obtaining information on the device's communication protocol/syntax. This can also help you fully utilize optional features available on that specific device. However, at times it can be difficult to get the needed information with respect to the protocol/syntax of a particular camera/server.

One way to work around this is connecting to your networked camera or video server using an Internet browser that captures the location or path to the stream. An example of such browser is Mozilla FireFox. Using the browser you can go to your network device's IP address, left click on the streaming image and select Copy Image Location.

Dynamic Image Settings - Example 1: Axis

- Manufacturer: Axis
- Model: 2100 (camera)
- Path: axis-cgi/mjpg/video.cgi?camera=&resolution=320x240

Axis equipment supports a number of resolutions, and therefore requires that the target resolution be indicated. Each camera can also have a camera ID number but that is optional, just as a number of other features that can be indicated in the path.

Dynamic Image Settings - Example 2: Panasonic

- Manufacturer: Panasonic
- Model: BL-C10A (camera)
- Path: nphMotionJpeg?resolution=320x240&Quality=Standard

Dynamic Image Settings - Example 3: Vivotek

- Manufacturer: Vivotek
- Model: 2111 (camera)
- Path: cgi-bin/ video.jpg?cam=1&quality=3&size=2

Working With Trendnet IP Cameras

To get a streaming image from the TV-IP301 Trendnet IP camera on to a dynamic image window of a touch panel:

1. Configure the camera for JPEG and 15fps. The default settings of "640x480" and "Highest Quality" should be OK.
2. In TPD5, set the dynamic image properties to:
   - Protocol: HTTP (default)
   - Host: (the default IP of this camera is 192.168.1.30)
   - Path: goform
   - File: video2
   - user & password are blank
   - Refresh Rate: 1 (or more)

Sounds

The *Sounds* tab of the Resource Manager dialog provides a convenient way to import and preview all sound (.WAV and .MP3) files to be used in your project. Use this tab as a "library" of every sound file that will be used in your project - for pages, popup pages, and buttons (FIG. 38).
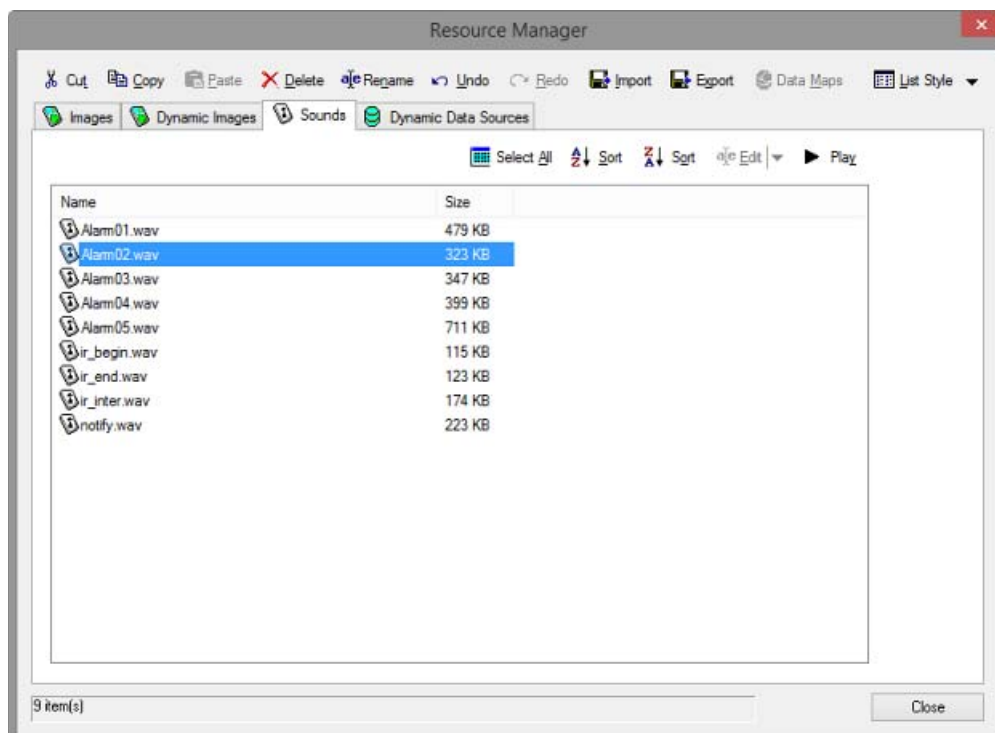

FIG. 38 Resource Manager dialog - Sounds tab

Importing Sound Files Into the Project

1. Select **Panel > Resource Manager** (or click the toolbar button) to open the *Resource Manager* dialog.

2. Open the *Sounds* tab.

3. Click the **Import** button to invoke the *Open* dialog.

    Click the *Overwrite resources with duplicate names* checkbox (at the bottom of the dialog) to automatically overwrite any existing sound files that have the same filename. If this option is selected, anytime a file with a duplicate name is imported, it will replace the original file on every button or page that referenced that filename in the project.

4. Locate and select the file(s) to import.

    NOTE: *TPD5 supports WAV and MP3 sound file formats.*

5. Click **Open** to import the files to the Sounds tab.

NOTE: *If you import an sound file that has already been imported to the Sounds tab, the filename is changed to include the prefix "Copy of...". This is true even if the second version of the file you have imported is of a different file type with a different extension.*

Previewing Sound Files

1. Select **Panel > Resource Manager** (or click the toolbar button) to open the *Resource Manager*.

2. Open the *Sounds* tab.

3. Select a Sound file.

4. Click the **Play** button.

Assigning Sounds to Buttons

Sounds (WAV or MP3 sound files) can be assigned to General, Multi-State General and Multi-State Bargraph buttons, via the Sound State Property.

NOTE: *In order to apply a Sound, the sound file must be pre-loaded in the Resource Manager. See page 43 for details.*

1. In a Design View window, select the Button to which you want to add or change the Sound file.

2. In the States tab of the Properties window, click on the *Sound* field. Click the browse (...) button to open the *Select Resource* dialog (FIG. 39):
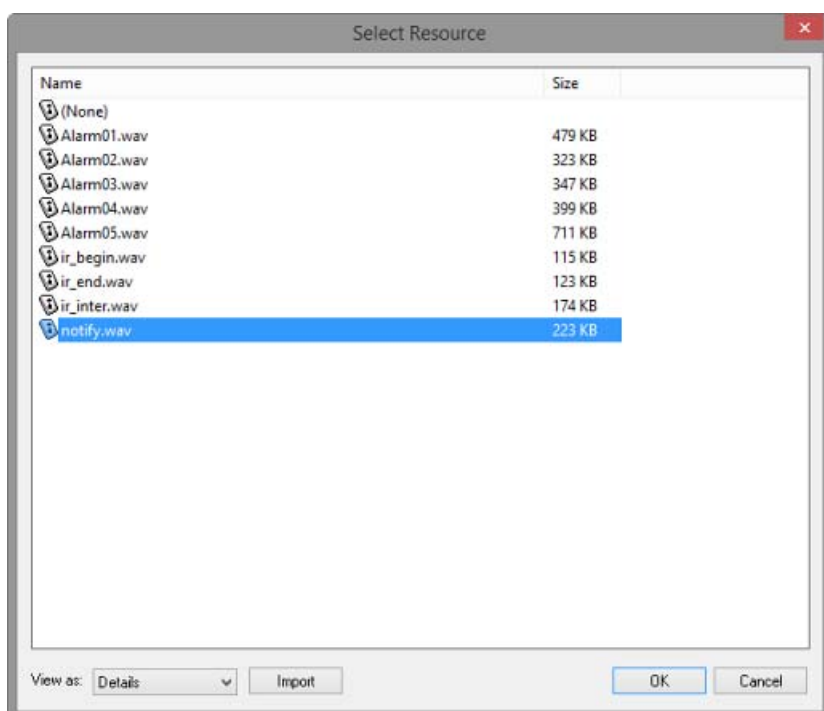
FIG. 39 Select Resource dialog (Sounds)

3. Select a sound file and click **OK** to save changes and close the *Select Resource* dialog.

Editing Sound Files

In order to edit image and sound files via the TPDesign5 UI, default image and sound editor applications must be defined in the Editor Selection tab of the Preferences dialog. See *Adding an External Sound Editing Program* on page 45 for details.

1. Open the *Sounds* tab of the Resource Manager dialog.
2. Select the sound file that you want to edit.
3. Click the **Edit** button to launch the external program specified as the default editor for sound files.
4. When the edit session begins, the image or sound file is opened in the default editing program. Any saved changes to the resource made in the external editor will be immediately reflected in TPDesign5.

NOTE: *Any action taken in TPDesign5 that would change the state of the image or sound file being edited (e.g., delete, rename, undo/ redo, etc.) will cause the link between TPDesign5 and the external application to be broken. Any subsequent changes made in the external editor will not be reflected nor applied.*

Custom Ringtones & Ringback Tones

G5 supports custom ringtones and ringback tones:

Customizing the Ringtone

The default incoming call ringtone can be overridden by including a wave file named "**ringtone.wav**" in the TP5 project file.

If a "*ringtone.wav*" file is found in the TP5 file, it is used instead of the default ringtone. If no such file exists, then the default ringtone is used.

Customizing the Ringtone For Particular Caller Numbers

A custom ringtone can be configured for calls from a specific extension by including a wave file named "**ringer_xxxx.wav**" file in the TP5 project file. For example, a file named "*ringer_6001.wav*" would be used as a custom ringtone for incoming calls from extension *6001*.

This will override the custom *ringtone.wav* for any extension that has a custom ringer_xxxx.wav sound defined.

Customize Ringback Tone

Te default ringback tone (for an outgoing call) can be overridden by placing a "*ringback.wav*" file in the TP5 project file.

Editing Image and Sound Files Using External Programs

Adding an External Image Editing Program

Use the options in the *Editor Selection* tab of the Preferences dialog to associate one or more image editing programs with image files in TPD5 projects. Note that you can associate multiple editor programs with image files, but one is specified as the default image editor:

1. Select **Edit > Preferences** to open the *Preferences* dialog, and open the *Editor Selection* tab. Note that *Image Editors* is already selected in the *Editor Type* drop-down menu.
2. Click the **Add** button (**+**) to access the *Choose Editor* dialog.
3. Click the Browse button (...) to locate and select the desired program's executable (.EXE) file, in the *Open* dialog.

NOTE: *The first program added to the Editors list is automatically designated as the default image editor. If you add additional programs to the list, you have the option (in the Choose Editor dialog) to set the default image editor.*

4. Click **OK** in the *Choose Editor* dialog to add the selected program to the Editors list (FIG. 40)
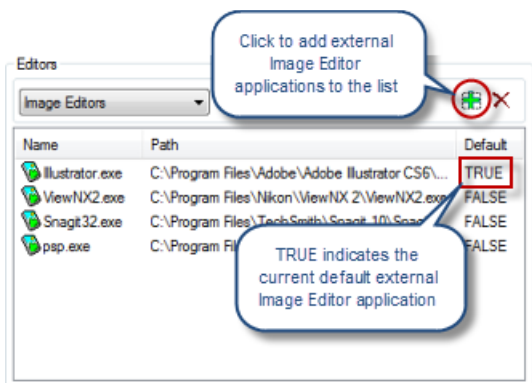
FIG. 40 Preferences dialog - Editor Selection tab

Changing the Default External Image Editor Program

1. In the Preferences dialog (Editor Selection tab), double-click the Image Editor application that you want to set as the new default program. This opens the *Choose Editor* dialog (FIG. 41).
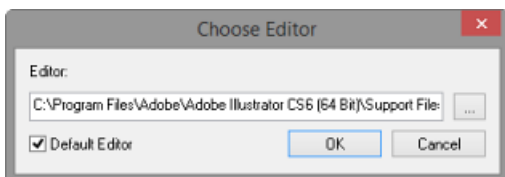
FIG. 41 Choose Editor dialog with Default Editor option selected

2. Click in the *Default Editor* checkbox and click **OK** to save changes and close the dialog.
3. The application now indicates **TRUE** in the *Default* column in the *Editor Selection* tab.

Adding an External Sound Editing Program

Use the options in the Editor Selection tab of the Preferences dialog to associate one or more sound editing programs with sound files in TPD5 projects. Note that you can associate multiple editor programs with sound files, but one is specified as the default image editor:

1. Select **Edit > Preferences** to open the *Preferences* dialog, and open the *Editor Selection* tab.
2. Click the down arrow and select **Sound Editors** from the *Editor Type* drop-down menu.
3. Click the Add (+) button to access the *Choose Editor* dialog.
4. Click the Browse button to locate and select the desired program's executable (.EXE) file.

NOTE: *The first program added to the Editors list is automatically designated as the default sound editor. If you add additional programs to the list, you have the option (in the Choose Editor dialog) to set the default sound editor.*

5. Click **OK** in the *Choose Editor* dialog to add the selected program to the Editors list.

Changing the Default External Sound Editor Program

1. In the Preferences dialog (Editor Selection tab), double-click the Sound Editor application that you want to set as the new default program. This opens the *Choose Editor* dialog.
2. Click in the *Default Editor* checkbox and click **OK** to save changes and close the dialog.
3. The application now indicates **TRUE** in the *Default* column in the *Editor Selection* tab.

Dynamic Data Sources

The *Dynamic Data Sources* tab of the Resource Manager dialog provides a convenient way to configure dynamic data sources for use with Listview buttons in your TPD5 project (FIG. 42).
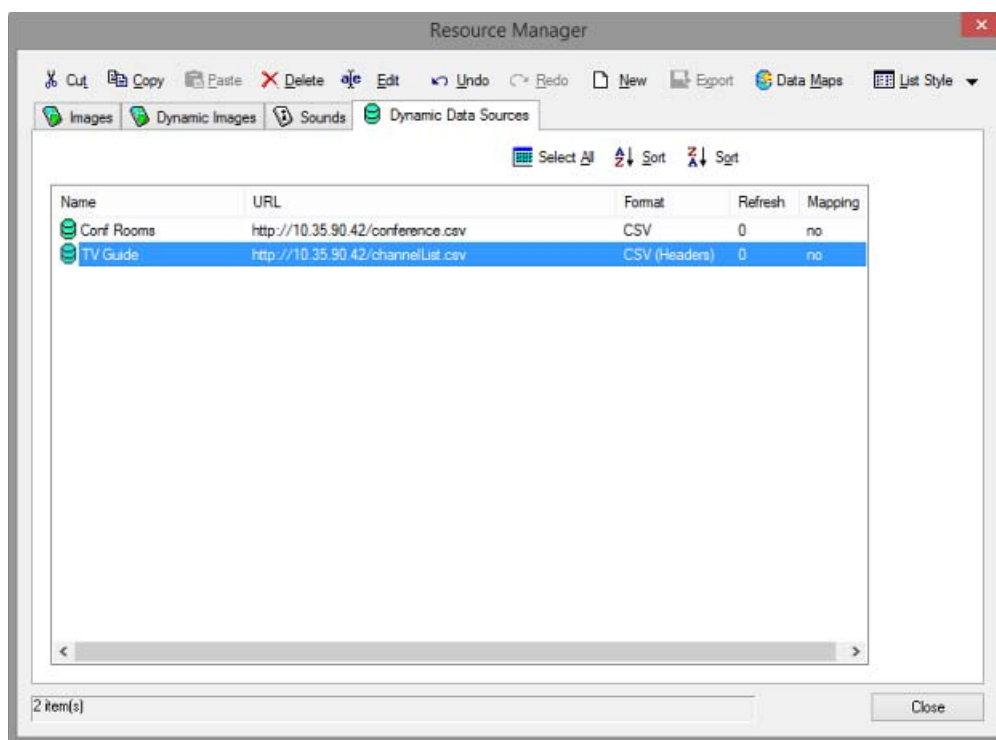
FIG. 42 Resource Manager dialog - Dynamic Data Sources tab

Modero X Series G5 touch panels and TPDesign5 (v1.2, build 65 or greater) support *Listview* buttons. Listview buttons provide the ability to display a listing of items from a dynamic data source on a G5 touch panel.

NOTE: *See the Listview Buttons & Dynamic Data section on page 104 for information on Listview buttons.*

Dynamic data defines data files/feeds URL where the data can be loaded by the touch panel at runtime via HTTP (GET) or HTTPS (GET) transport protocols.

Dynamic Data Sources are data sources that exist on the NetLinx Master (or an HTTP server) external to the panel. Data source files can be CSV files either with or without headers, XPort-generated XML, or NetLinx code.

The creator of the data can specify how many fields comprise a record and the format of those fields. As many records as necessary can be specified. This data can be used to populate a Listview button displayed on a G5 touch panel, where the end user can scroll or search through the list and make a selection.

Once a selection has been made, a CUSTOM_EVENT is raised in the NetLinx Master to retrieve the data fields comprising the selected record and potentially trigger events.

Refer to the *Listview Buttons & Dynamic Data* section for working demos of creating Listview buttons using four types of source data:

1) CSV file with headers (page 114)
2) CSV file without headers (page 128)
3) XPort-generated XML file (page 141)
4) NetLinx Data Source (page 154)

Adding Dynamic Data Sources to the Project

The following example steps you through the process of adding a Dynamic Data Source file to the Resource Manager (for use with Listview buttons).

1.   In the *Dynamic Data Sources* tab of the Resource Manager, click **New** to open the *Create Dynamic Data Source* dialog (FIG. 43):
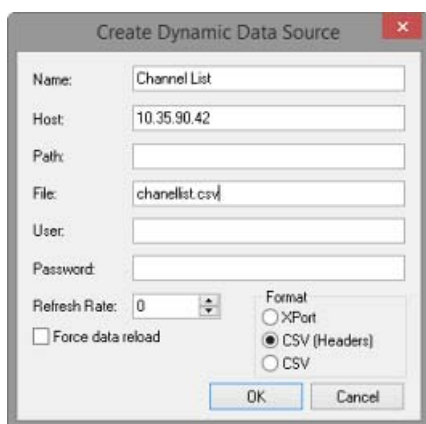
FIG. 43 Create Dynamic Data Source dialog (with sample data)

2. In the **Name** field, enter a unique name for the Data Source. The name entered here will be used to identify this file in the *Resource Manager - Dynamic Data Sources* tab and the *Select Resources* dialog.

3. In the **Host** field, enter the host name, which must be a fully qualified DNS or IP address.

4. In the **Path** field, enter the path to the source file. The path must be a valid HTTP URL minus the protocol and host. The only exception to this is the inclusion of special escape sequences and regular expressions.

5. In the **File** field, enter a file name that indicates the full path to the location of the source file.

6. In the **User** field, enter the user name required by the NetLinx Master or server for authentication (if required).

7. In the **Password** field, enter the password required by the NetLinx Master or server for authentication (if required).

8. In the **Refresh Rate** field, use the up/down arrows to adjust the number of seconds between refreshes in which the resource is downloaded again. Refreshing resources will cause the button displaying that resource to refresh as well. The default value is 0, which means that the resource is only downloaded once.

9. Set the **Refresh only at panel startup** option. This option is only available if *Refresh Rate* is set to zero, and causes the dynamic data to refresh only upon restart of the panel, as opposed to upon each visit to the page (as is the default).

10. Specify the format of the source file:
    ● **XPort** - Select if the Data Source file is XPort-generated XML (default selection).
    ● **CSV (Headers)** - Select if the Data Source is a CSV file with headers.
    ● **CSV** - Select if the Data Source is a CSV file that does not have headers.

11. Click **OK** to save changes and close this dialog. The new data source is indicated in the *Dynamic Data Sources* tab (FIG. 44):
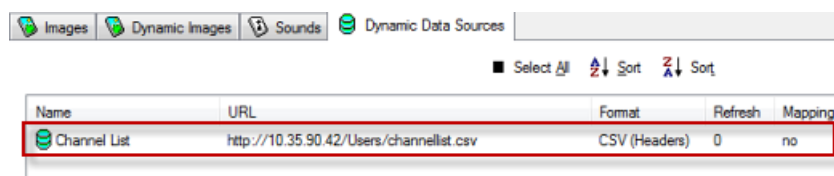
FIG. 44 Resource Manager - Dynamic Data Sources tab indicating an example Data Source File

Editing Dynamic Data Sources

1. In the *Dynamic Data Sources* tab of the Resource Manager, select a data source.

2. Click the **Edit** button to open the *Edit Dynamic Data Source* dialog (FIG. 45).
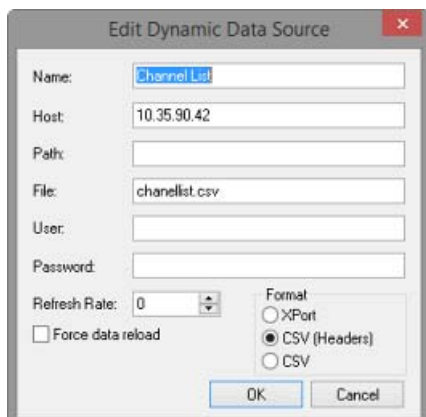
FIG. 45 Edit Dynamic Data Source dialog

3. Edit the Data Source information as desired.

4. Click **OK** to save changes and close the dialog.

Deleting Dynamic Data Sources From the Project

1. In the *Dynamic Data Sources* tab of the Resource Manager, select one or more files to delete (Ctrl + click to select multiple files individually, or Shift + click to select a range of files).

2. Click the **Delete** button (FIG. 46):
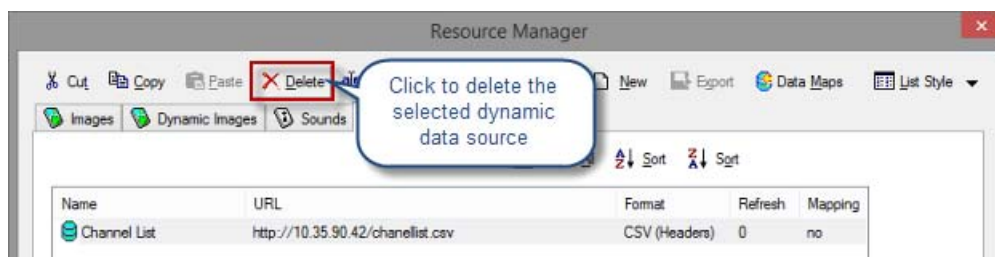
FIG. 46 Deleting a data source from the project

Pages

Overview

Pages are not only containers for other TPD5 Elements (Popup Pages, Sub-Pages, Application windows and Buttons), but can also have up to one address port / address code and up to one channel port / channel code combination.

When a new project is created, it contains a single page - "Page 1" (see *Creating a New TPD5 Project* on page 21 for details).

- Each page must be uniquely named within its project (name collision checks are case-insensitive).
- The maximum number of pages in a panel file is *500*.
- Each page in the project is represented in its own Design View window
- Use the Properties window to set the available properties for Pages. These include name/description, background color, text, bitmap, icon and sound as well as address port / address codes and channel port / channel code. If you change any of these properties (except the name) your changes will be saved and subsequent page creations will use those settings.
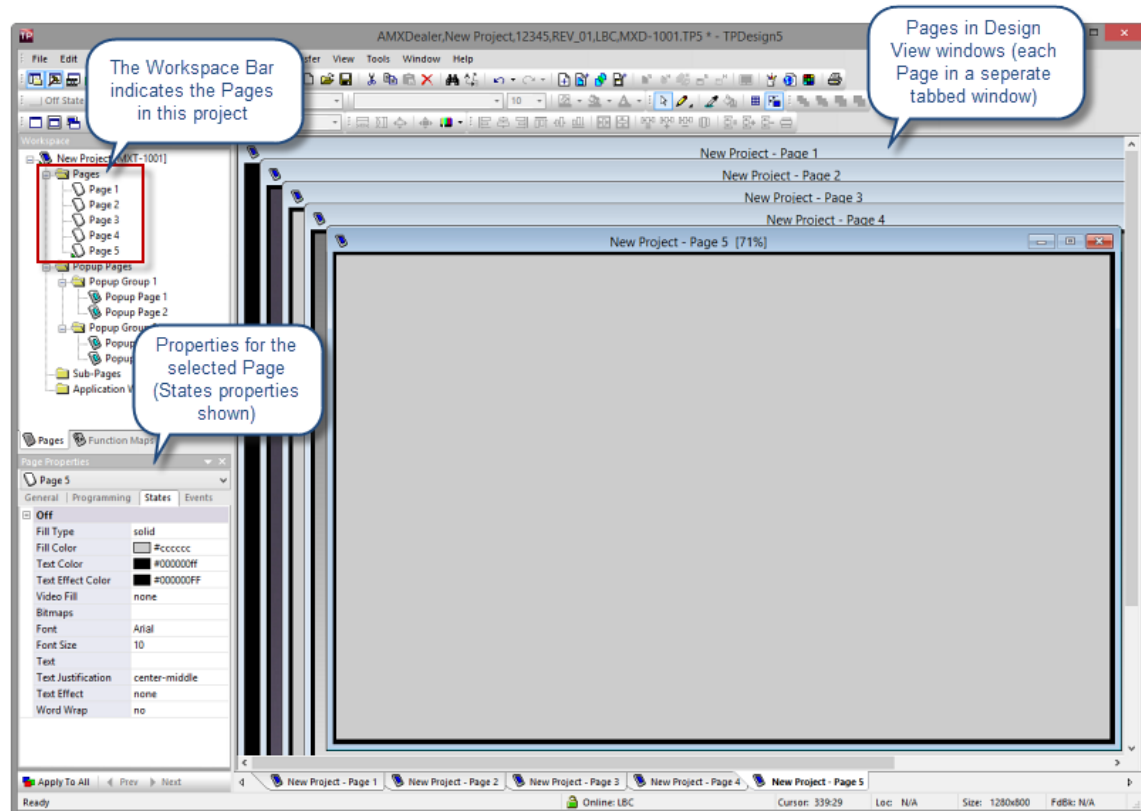
FIG. 47 Pages

NOTE: *Page Flips (which provide the ability to "flip" from one page to another via a button press) are a button-oriented function - see Adding a Page Flip to a Button on page 270 for details.*

Adding Pages to the Project

Use the options in the *Add Page* dialog to define a new Page to add to the active project (FIG. 48):
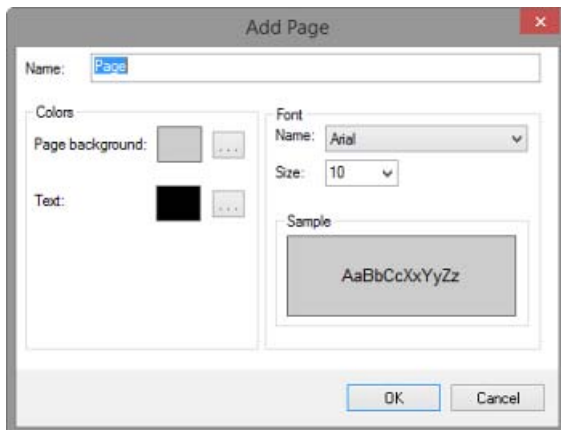
FIG. 48 Add Page dialog

There are several ways to open the *Add Page* dialog:

- Select **Panel > Add Page**
- Click the **Add Page** toolbar button
- Right-click on the Pages folder (or on any individual Page) in the Workspace window and select **Add Page**.

To add a new Page to your project:

1. Open the *Add Page* dialog.
2. Fill in the information in this dialog to specify the basic properties for the new Page.
   - **Name**: Enter a name for the new Page here. By default, the pages are named Page (2), Page (3), etc... You can edit the page name at any time directly via the Workspace window (Pages tab).Note the name assigned to the Page appears in the Workspace, but is not displayed in the Page itself. Any text that is to be displayed on the Page is entered via the Text property in the Workspace window (Pages tab).
   - **Colors**: Set the *Page background* and *Text* colors for the new Page. Click on the palette buttons (...) to open the *Colors* dialog, where you can select the desired colors.
   - **Font**: Select a font (*Name*) and *Size* for any text that will be displayed on this Page. Note that the actual text is entered via the Text property in the Workspace window (Pages tab). This is also where you can specify the Text Justification as well as apply a Text Effect and Text Effect Color as desired.
3. Click **OK** to add the new Page to the active project.

   The new Page is added to the Workspace window (Pages tab), under the project to which the Page was added as the active page (see "*Page (2)*" below):
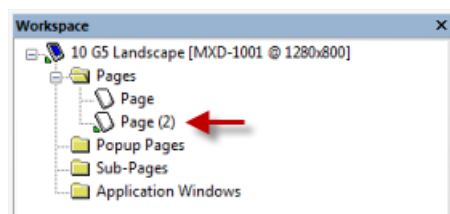
FIG. 49 Workspace window - "New Page (2)" added

Note that once the page has been created, all Page Properties can be viewed/edited via the Properties window.

Copying and Pasting Pages

1. Cut or copy a Page to clipboard memory:
   - To cut a Page to the clipboard, select a Page in the Workspace Window (Pages tab) and select Cut. The program will prompt you to verify this action before the Page is removed from the project.
   - To copy a Page to the clipboard, select a Page in the Workspace Window (Pages tab) and select Copy.
2. Select the target Project for the Page in the Workspace Window (Pages tab). Pages can be pasted into the current Project, or into any other Project that is open in the Workspace Window. Note that all Buttons present on the original Page are also copied, and when pasted they retain the attributes of the original buttons, according to the selections made in the *Paste Controls* dialog (see page 66).
3. Select Paste to paste a copy of the Page into the selected Project. If a Page with the same name already exists in the target Project, the Page's name will be modified to indicate that it is a copy of another page. This prevents existing Pages from being overwritten by a Paste operation.
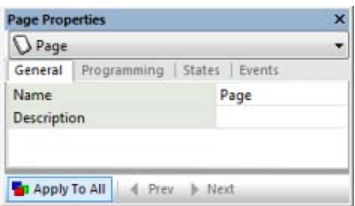
Setting Page Properties

Pages have a set of Properties that can be configured via the fields in the Properties window. To set Page-level properties, click on the Page area in the Design View. With the Page selected, the Properties window displays the properties available for the Page.

NOTE: *To edit any of the listed properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.*
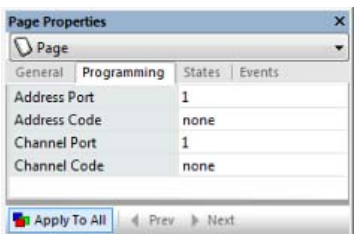
Pages - General Properties

With a Page selected, use the options in the *General* tab of the Properties window to set General properties. The following *General* properties are supported at the Page level:

Pages - General Properties		
	• Name	see page 242
	• Description	see page 236
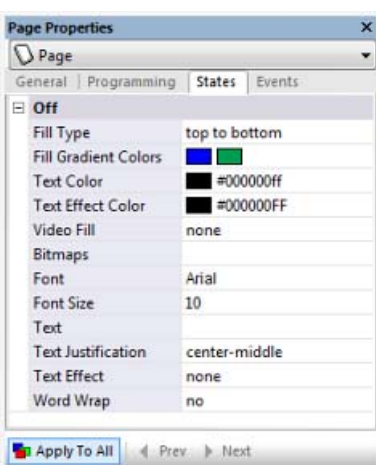
Pages - Programming Properties

With a Page selected, use the options in the *Programming* tab of the Properties window to set Programming properties. The following Programming properties are supported at the Page level:

Pages - Programming Properties		
	• Address Port	see page 247
	• Address Code	see page 247
	• Channel Port	see page 249
	• Channel Code	see page 249

Pages - States Properties

With a Page selected, use the options in the *States* tab of the Properties window to set State properties. The following States properties are supported at the Page level:
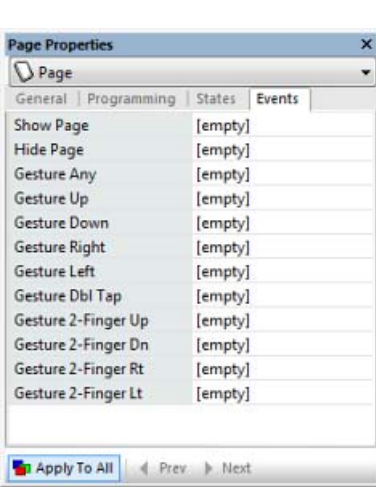
Pages - States Properties		
	• Fill Type	see page 252
	• Fill Color	see page 252
	• Fill Gradient Colors	see page 252
	• Gradient Radius	see page 253
	• Gradient Center X%	see page 253
	• Gradient Center Y%	see page 253
	• Text Color	see page 254
	• Text Effect Color	see page 254
	• Video Fill	see page 254
	• Streaming Source	see page 253
	• Bitmaps	see page 251
	• Font	see page 252
	• Font Size	see page 253
	• Text	see page 254
	• Text Justification	see page 254
	• Text X Offset	see page 254
	• Text Y Offset	see page 254
	• Text Effect	see page 254
	• Word Wrap	see page 254

Pages - Events Properties

Modero X Series panels support *Gestures* for on-screen navigation. Gestures can be used (in addition to Buttons) for navigating the panel UI. For example, a "Swipe" gesture can invoke a page flip when the user swipes a finger across the screen. Gestures are presented in TPDesign5 under the **Events** tab of the Properties window.

With a Page selected, use the options in the *Events* tab of the Properties window to set *Event* properties.

To choose a particular event property for a page, highlight the row and click the Browse button (...) to open the *Edit Event Actions* dialog. Use this dialog to add actions instigated by the gesture selected. Click the **Add Action** button to select between a Send Command or a command string, and enter the command or string in the field. When finished, click **OK**. The following Events are supported at the Page level:

Pages - Events Properties		
	• Show Page	see page 257
	• Hide Page	see page 257
	• Gesture Any	see page 257
	• Gesture Up	see page 257
	• Gesture Down	see page 257
	• Gesture Right	see page 258
	• Gesture Left	see page 258
	• Gesture Dbl Tap	see page 258
	• Gesture 2-Finger Up	see page 258
	• Gesture 2-Finger Dn	see page 258
	• Gesture 2-Finger Rt	see page 258
	• Gesture 2-Finger Lt	see page 258

Renaming a Page

To rename any Page in the active project, simply click on a Page entry in the Workspace and type directly into the text field (FIG. 50):

FIG. 50 Renaming a Page

Opening Pages via the Workspace window

You can open a Page in the active project by double-clicking on the Page in the Workspace window (Pages tab). This action opens the selected Page in its own Design View window.

Deleting Pages From a Project

To delete a Page from the active project, select a Page to delete in the Workspace window (Pages tab), and press the Delete key. Alternatively, select **Edit > Delete** (or click the **Delete** toolbar button).

The program will prompt you to verify this action before the selected Page(s) are deleted (FIG. 51):
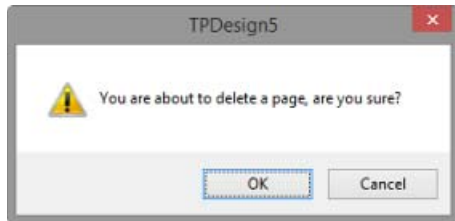
FIG. 51 Prompt - Verify Deleting a Page

NOTE: *All panels must include at least one page, therefore, deleting the last remaining page is not allowed.*

Exporting Pages as Image Files

Use the options in the *Export Page Images* dialog to export one or more Pages (and/or Popup Pages) in the current project as image files:

1. Select **Panel > Export Page Images** to open the *Export Page Images* dialog (FIG. 52):

FIG. 52 Export Page Images dialog

2. Use the checkboxes in the *Pages* and *Popup Pages* lists to select which Pages (and Popup Pages) to export.

3. Specify a target directory for the exported image files in the *Export Directory* field. Use the browse button (...) to navigate to a target directory via the *Browse For Folder* dialog.

4. Specify a template for the resulting filenames in the *Filename Template* field. By default, the template is set to include the Panel file name and the Page name into the generated image's file name.

    Use **$P** to insert the Panel name and **$p** to insert the Page name into the exported file's name. The file extension is automatically, based on the *File Format* selection.

5. Use the *Scale% up* and *down* arrows indicate a percentage to down-scale the exported images. The default setting is 100% (no scaling).

6. Select a image file type (JPG, PNG or BMP) for the resulting image files from the *File Format* drop-down. If JPG is selected, you can select the desired level of image compression

7. Select **Export current display state for buttons** to capture the Pages as they are currently displayed in the Design View, with the button's current display state. If this option is not selected, it will export the buttons in the (default) Off state. By default, this option is enabled.

8. Select **Export popup pages shown on pages** to include any Popup Pages that are being displayed on the Design View at the time of selection in the exported Page image file. If this option is not selected, only the Page underneath any currently viewed Popup Pages will be exported. By default, this option is enabled.

9.   Select **Export information overlay** to include any function code information this is currently being displayed on the Design View at the time of selection in the exported Page image file. If this option is not selected, only the Page underneath any currently displayed function codes will be exported. By default, this option is disabled.

10. Click **Export**.

Cut, Copy and Paste - Pages

1.   Cut or Copy a Page to clipboard memory:
     ● To cut a Page to the clipboard, select a Page in the Workspace window (Pages tab) and select **Cut**. The program will prompt you to verify this action before the Page is removed from the project.
     ● To copy a Page to the clipboard, select a Page in the Workspace window (Pages tab) and select **Copy**.

2.   Select the target project for the Page in the Workspace window (Pages tab). Pages can be pasted into the current project, or into any other project that is open in the Workspace window.

     Note that all Buttons present on the original Page are also copied, and when pasted they retain the attributes of the original buttons, according to the selections made in the *Paste Controls* dialog.

3.   Select **Paste** to paste a copy of the Page into the selected project. If a Page with the same name already exists in the target project, the Page's name will be modified to indicate that it is a copy of another page. This prevents existing Pages from being overwritten by a Paste operation.

Setting a Power Up Page

Use the *Power-up page* option in the *Panel Setup Information* tab of the *Project Properties* dialog to specify a particular page in your project to be displayed when the panel is turned on:

1.   Select **File > Project Properties** to open the *Project Properties* dialog.

2.   In the *Panel Setup Information* tab, click the down arrow next to the **Power up page** field to display a list of all of the pages in this project (FIG. 53):
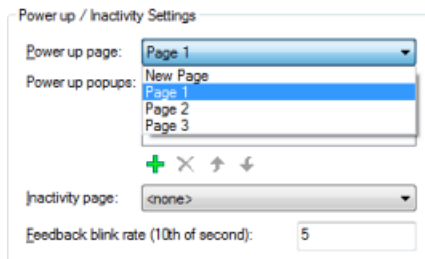
FIG. 53 Project Properties dialog (Panel Setup Information tab) - Power Up Page

3.   Select the page that you want to be used as the Power-up page from this list.

Setting an Inactivity Page Flip

Use the *Inactive Page Flip* option in the *Panel Setup Information* tab of the *Project Properties* dialog to specify a particular page in your project to be displayed when the panel is inactive for a specified period of time:

1.   Select **File > Project Properties** to open the *Project Properties* dialog.

2.   In the *Panel Setup Information* tab, click the down arrow next to the **Inactivity page flip** field to display a list of all of the pages in this project (FIG. 54):
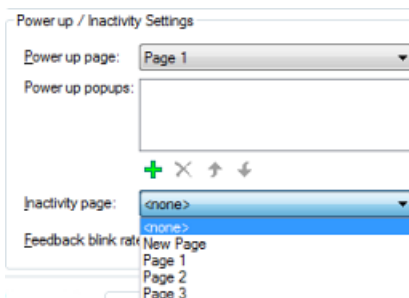
FIG. 54 Project Properties dialog (Panel Setup Information tab) - Power Up Page

3.   Select the page that you want the panel to flip to when the panel is inactive for the amount of time specified on the panel.

Printing Pages

1. Select **File > Print Preview** to display a preview of the pages and popup pages in the active project, as they will appear when printed in the *Print Preview* window (FIG. 55):
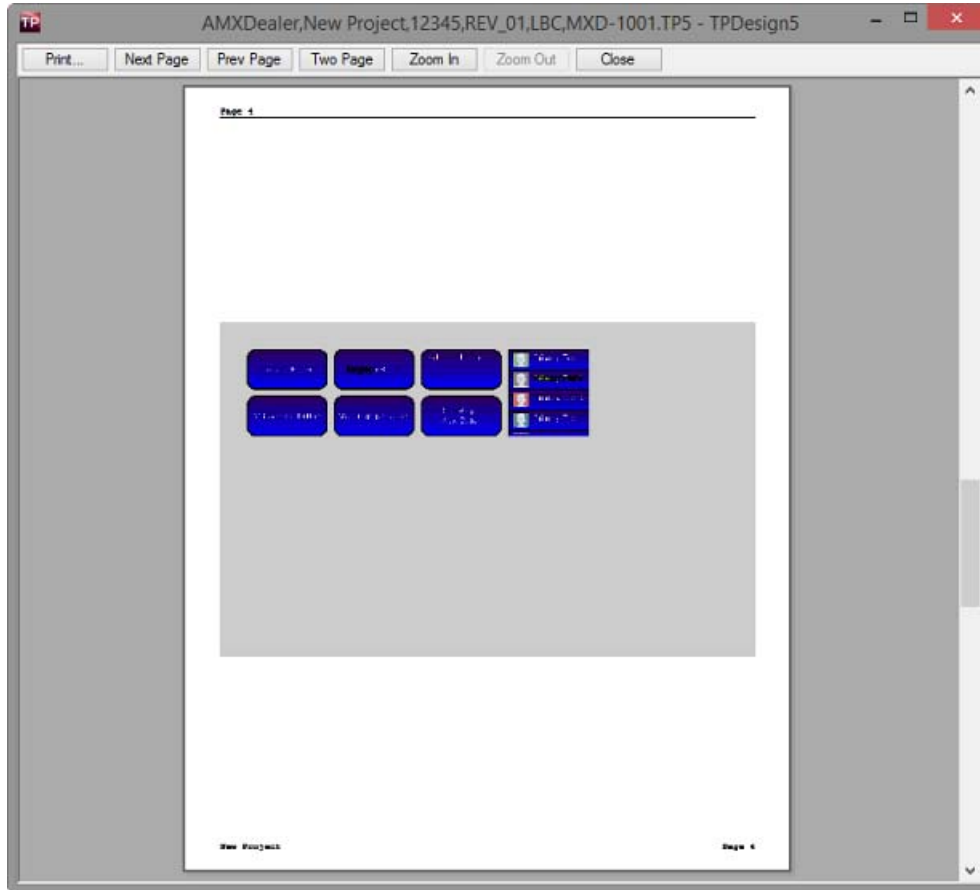
FIG. 55 Print Preview window

2. Select **Print**.

Each page and popup page in the project is printed one per page.

Popup Pages

Overview

Popup Pages provide a mode of sub-navigation for the panel's UI by presenting a set of options that sit on top of a main Page. Popup Pages are similar to Pages in some respects as they are a container for Buttons, they may have up to one address and channel port code assignments, and contain only one state. Otherwise, Popup Pages have many typically Button specific properties, as well as some special properties that apply only to Popup Pages. For example, Popup pages (unlike Pages) can be assigned a border style and border color, as well as left, top, width or height values.

- Each Popup Page must be uniquely named within its project (name collision checks are case-insensitive).
- The minimum size for Popup Pages is 15 pixels in height or width.
- Each Popup Page in the project is represented in its own Design View window.

Use the Properties window to set the available properties for Popup Pages. These include name/description, background color, text, bitmap, icon and sound as well as address port / address codes and channel port / channel code. If you change any of these properties (except the name) your changes will be saved and subsequent page creations will use those settings.
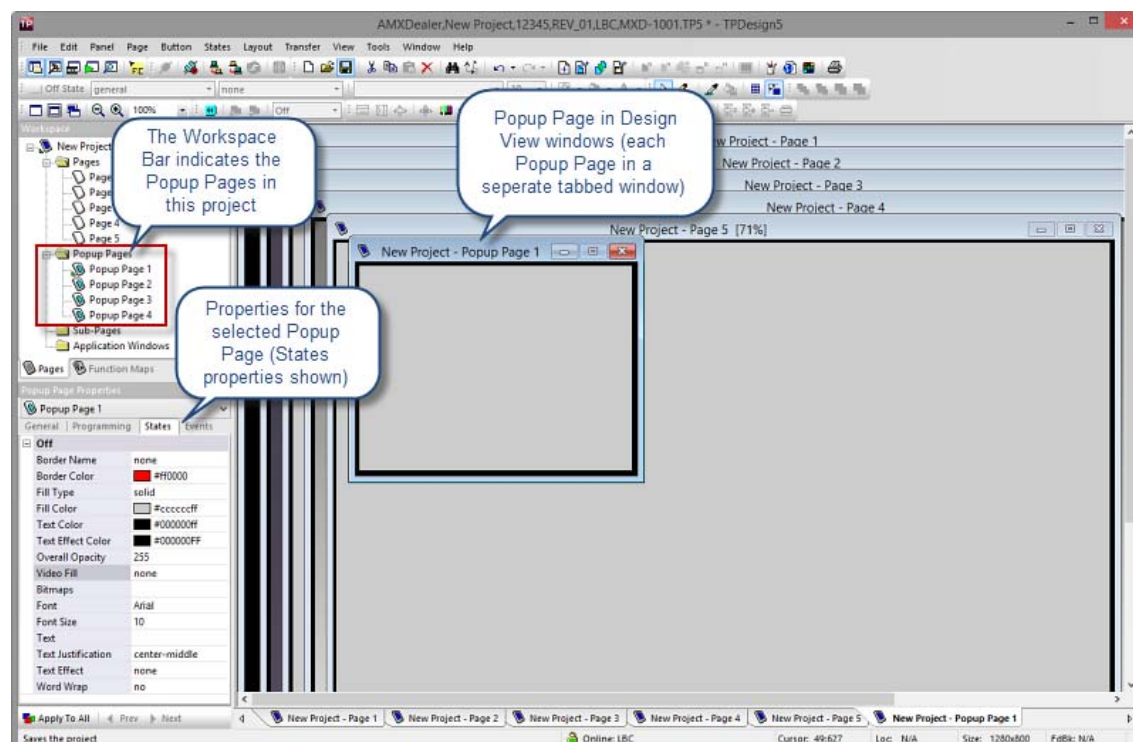
FIG. 56 Popup Pages

Popup Pages can be organized in to *Popup Page Groups*. Popup Page Groups provide a mechanism to group Popup Pages into mutually exclusive groups for display purposes (see *Creating Popup Page Groups* on page 60).

Adding Popup Pages to the Project

There are two ways to add a new Popup Page to your project:

- Via the *Add Popup Page* dialog (see page 57)
- Via the Popup Draw tool (see page 57)

Adding a Popup Page via the Add Popup Page dialog

1. Select **Panel > Add Popup Page** to open the *Add Popup Page* dialog (FIG. 57):
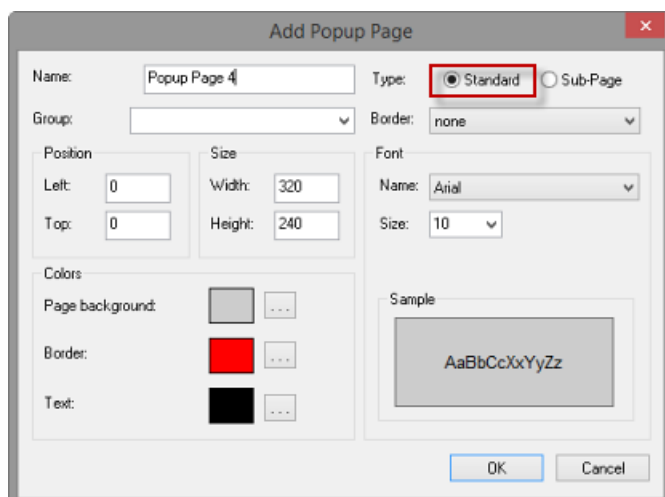
FIG. 57 Add Popup Page dialog

By default, the **Type** property is set to *Standard*. This indicates that the new Popup Page will be a Standard Popup Page (as opposed to a *Sub-Page*).

NOTE: *"Sub-Page" refers to a specific type of a popup page that is only used with the Sub-Page View button type. Sub-Pages and Sub-Page View buttons are required for Scrolling Regions. See the Scrolling Regions (Sub-Pages & Sub-Page View Buttons) section on page 89 for details.*

2. Fill in the information in this dialog to specify the basic properties for the new Popup Page. Note that the settings made here can be adjusted at any time via the Properties window:

Popup Page Properties (add Popup Page dialog)			
Name	see page 242	Height	see page 237
Group	see page 237	Page Background (Fill Color)	see page 252
Left	see page 238	Border (Border Color)	see page 251
Top	see page 246	Font (Name)	see page 252
Width	see page 247	Font Size	see page 253

3. Click **OK** to close the *Add Popup Page* dialog and add the new Popup Page to the active project. The new popup page will be appear in the Workspace window (Pages folder), under the project to which the page was added (as the active page).

4. Edit and set additional Popup Page Properties as desired in the Properties window.

Showing Popup Pages on a Page in the Design View

Note that when a Popup Page is created via the *Add Popup Page* dialog, it is represented in the Design view in it's own window, and not on a Page. To show a Popup Page on a Page (to verify its placement and other display attributes):

1. In the Design view, select a Page.

2. In the Workspace window, right-click on a Popup Page and select **Show Popup Page**.

Adding a Popup Page via the Popup Draw tool

1. Select **Edit > Popup Draw Tool** (or click the toolbar button) to activate the Popup Draw tool.

    To access the Popup Draw toolbar button, click and hold the Button Draw tool (in the Selection/Drawing Tools toolbar) for one second to open the drop-down menu containing the Popup Draw tool (FIG. 58):

FIG. 58 Selection/Drawing Tools toolbar - Popup Draw tool

2. Left-click inside the desired page (in the active Design View window), and while holding the left mouse button down, drag to draw the popup page to the desired size and shape (FIG. 59):
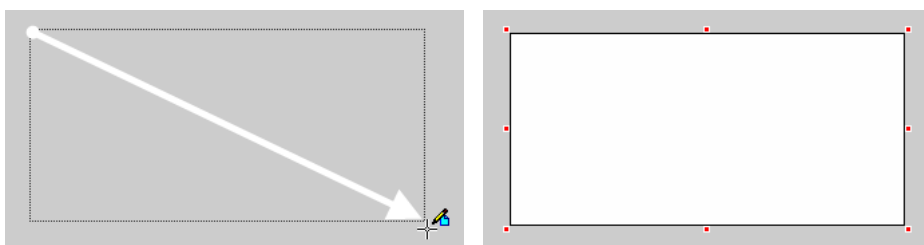
FIG. 59 Popup Draw tool - Drawing a Popup Page

- When using the Button Draw or Popup Draw tools, hold down the SHIFT key while drawing to constrain the item to a square.
- When using the Selection tool, hold down the ALT key while clicking and dragging in a Design View window to move the current selections without selecting anything new on the mouse press (useful for moving popup pages whose entire area is filled with buttons).
- When using the Selection tool, hold down the CTRL key while clicking and dragging to force a "lasso" selection to occur (even if the mouse was clicked over a button or Popup Page shown on the Page). Lasso selection forces you to draw a square around the outside of the item to select it, as opposed to clicking on the item.
- With an item selected in a Design View window, hold down the CTRL key while moving the item with the keyboard arrow keys to move by the grid size instead of a single pixel (regardless of the grid visibility or snap to grid setting). Holding the CTRL key while resizing the selected items with the keyboard will resize by the grid size.

3. Set the other properties as desired - see *Setting Popup Page Properties* on page 58 for details.
4. Select **File > Save** to save your changes.

Hiding Popup Pages on a Page in the Design View

Note that when a Popup Page is created via the Popup Draw tools, it is represented in the Design view on a Page. To hide a Popup Page on a Page:

1. In the Design view, select a Page displayed with Popup Pages.
2. In the Workspace window, right-click on a Popup Page that you want to hide and select **Hide Popup Page**.
   To hide all popup pages, right-click on the page in the Design View and select **Hide All Popup Pages**.

Setting Popup Page Properties

Popup Pages have *General*, *Programming* and *States* Properties that can be configured via the fields in the Properties window.

NOTE: *Note that Popup Pages do not support Events.*

To set Popup Page properties, click on the Popup Page in the Design View, or select a Popup Page in the Workspace window (Pages tab). With the Popup Page selected, the Properties window displays the properties available for the Popup, separated into three tabs (*General*, *Programming* and *States*).

NOTE: *To edit any of the listed properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.*

Popup Pages - General Properties

With a Popup Page selected, use the options in the *General* tab of the Properties window to set General properties.

The following General properties are supported at the Popup-Page level:

Popup Pages - General Properties		
	• Popup Type	see page 242
	• Name	see page 242
	• Description	see page 236
	• Left	see page 238
	• Top	see page 246
	• Width	see page 247
	• Height	see page 237
	• Reset Pos On Show	see page 243
	• Group	see page 237
	• Timeout	see page 246
	• Collapse Direction	see page 236
	• Collapse Offset	see page 236
	• Show Open	see page 236

Popup Pages - Programming Properties

With a Popup Page selected, use the options in the *Programming* tab of the Properties window to view/edit Address and Channel Port/Channel Code assignments. The following Programming properties are supported at the Popup-Page level:

Popup Pages - Programming Properties		
	• Address Port	see page 247
	• Address Code	see page 249
	• Channel Port	see page 249
	• Channel Code	see page 249

Popup Pages - States Properties

With a Popup Page selected, use the options in the *States* tab of the Properties window to set state-related properties.

Note that Popup Pages have only one State - **Off**.

The following State properties are supported at the Popup-Page level:

Popup Pages - States Properties		
	• Border Name	see page 252
	• Border Color	see page 251
	• Fill Type	see page 252
	• Fill Color	see page 252
	• Fill Gradient Colors	see page 252
	• Gradient Radius	see page 253
	• Gradient Center X%	see page 253
	• Gradient Center Y%	see page 253
	• Text Color	see page 254
	• Text Effect Color	see page 254
	• Overall Opacity	see page 253
	• Video Fill	see page 254
	• Streaming Source	see page 253
	• Bitmaps	see page 251
	• Font	see page 252
	• Font Size	see page 253
	• Text	see page 254
	• Text Justification	see page 254
	• Text X Offset	see page 254
	• Text Y Offset	see page 254
	• Text Effect	see page 254
	• Word Wrap	see page 254

NOTE: *Popup Pages do not support Events properties.*

Naming Popup Pages

If the popup page name starts with an underscore (ex: " _sources"), it will be always be displayed on top.

- If you rename the popup page without the underscore, it will act normally.
- If you call two popups with the underscore, the last one called will be on top.

Renaming Popup Pages

To rename any Popup Page in the active Project, simply click on a Popup Page entry in the Workspace and type directly into the text field (FIG. 60):
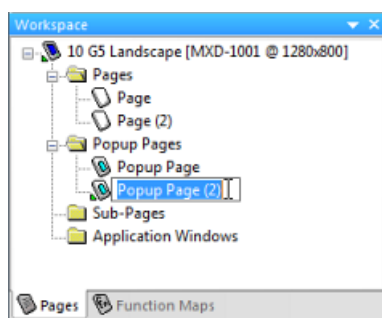
FIG. 60 Renaming a Page

Popup Page Groups

Standard Popup Pages can be organized into *Popup Page Groups*. Popup Page Groups provide a mechanism to group popup pages into mutually exclusive groups for display purposes.

NOTE: *Groups apply only to Standard Popup Pages (not to Sub-Pages).*

Popup Page Groups are represented in the Workspace Navigator (Pages tab) in Popup Page Group folders, nested within the Popup Pages folder (see *Popup Group 1* and *Popup Group 2* in FIG. 61):
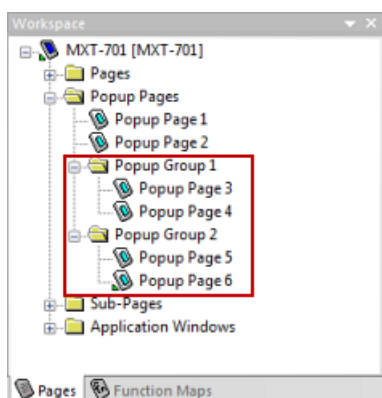
FIG. 61 Popup Page Groups

Creating Popup Page Groups

There are two ways to create a new Popup Page Group:

Via the Add Popup Page dialog (as part of creating a new Popup Page)

1.   Select **Panel > Add Popup Page** to open the *Add Popup Page* dialog.

2.   Enter a name for the new Popup Page Group in the *Group* text field (FIG. 62):
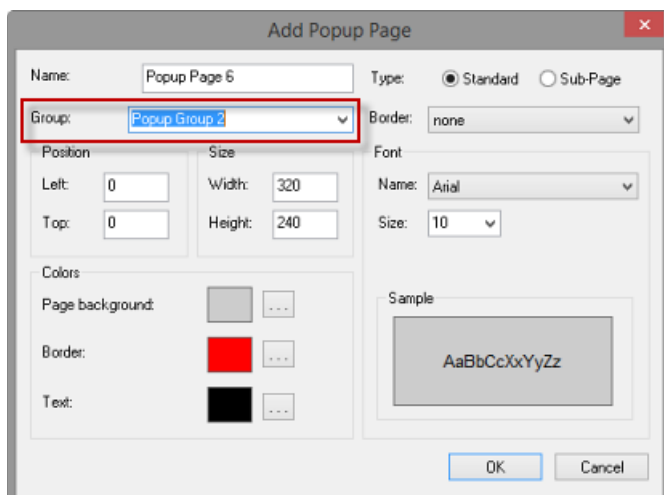
FIG. 62 Add Popup Page dialog - Group field

Note that if any Groups have been defined previously, they are available for selection via the *Group* drop-down menu.

3. Fill in the remaining information in this dialog to specify the basic properties for the new Popup Page (see *Adding Popup Pages to the Project* on page 56 for details).

4. Click **OK** to add the new Popup Page and Group to the active project. The Popup Page and Group are indicated in the Workspace tab (FIG. 63):
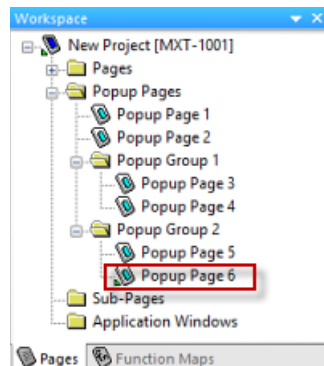
FIG. 63 Workspace window (Pages tab) - Indicating Popup Groups

Via the Properties window (for an Existing Popup Page)

1. Select a Popup Page in the Workspace window (Pages tab) to open the popup page and populate the Properties window.

2. In the **Group** (General) property, type a name for the new Popup Page Group (Example - "*Popup Group 1*"):

FIG. 64 Group (General) Property

3. Press Enter to add the new Group to the active project.

The new Popup Page Group will appear in the Workspace Navigator (Pages tab) within the *Popup Pages* folder:
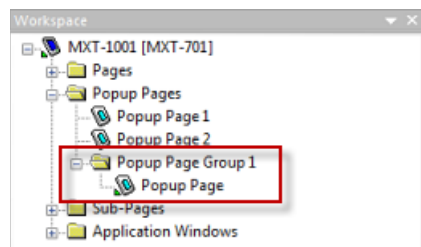
FIG. 65 Popup Page Group containing one Popup Page

Adding Popup Pages To a Popup Page Group

There are two ways to add Popup Pages to an existing Popup Page Group:

1. In the Workspace window, drag-and-drop Popups from the *Popup Pages* folder into the desired Popup Group folder.

2. In the Properties window, use the **Group** (General) property to select a Group for the selected Popup Page:
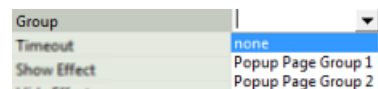
FIG. 66 Popup Page Group containing one Popup Page

Removing Popup Pages From a Popup Page Group

There are two ways to remove Standard Popup Pages from a Popup Page Group:

1. Drag-and-drop popup page(s) from the Popup Page Group folder into the Popup Pages folder:

2. Set the Group (General) property to *None* (in the Properties window):

Renaming Popup Page Groups

To rename a Popup Page Group, simply click on the group folder in the Workspace window and type directly into the text field.
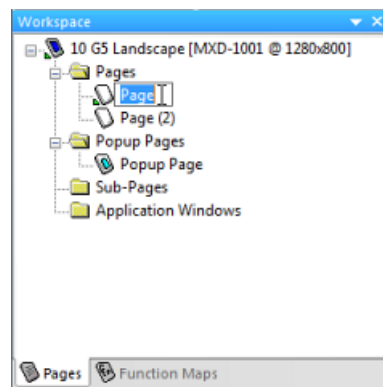
Opening Popup Pages via the Workspace Window

You can open a Popup Page in the active project by double-clicking on the Popup Page in the Workspace window (Pages tab). This action opens the selected Popup in its own Design View window.

NOTE: *Use the Show/Hide Popup Pages functions to display the selected Popup on the active Design View window (for a Page).*

Show/Hide Popup Pages

The *Show Popup Page* and *Hide Popup Page* options can be selected via the Page menu, the Workspace Navigator context menu or the toolbar buttons (contained in the Main toolbar).

These options deal with showing a selected Popup Page on a page that is currently open in the Design View

- **To show a Popup Page**: Select a Popup Page in the Workspace window (Popup Pages folder), and select Show Popup Page (or click the toolbar button).
- **To hide a selected Popup Page**: Select a Popup Page in the Workspace window (Popup Pages folder), and select Hide Popup Page (or click the toolbar button).
- **To hide all Popup Pages**: Select Hide All Popup Pages in the Page menu, Design View context menu, or click the Hide All Popup Pages toolbar button.

NOTE: *Additionally, you can drag and drop a Popup Page from the Workspace window (Pages tab) onto a Design View window to show the selected Popup Page.*

Deleting Popup Pages From a Project

To delete a popup page (Standard or Sub-Page) from the active project, select the Popup Page to delete in the Workspace Navigator (Pages tab), and press the Delete key (or toolbar button). Alternatively, select **Edit > Delete** (or select **Delete** from the Workspace window (Pages Tab Context Menu). The program will prompt you to verify this action before the selected Popup Page(s) are deleted (FIG. 67):
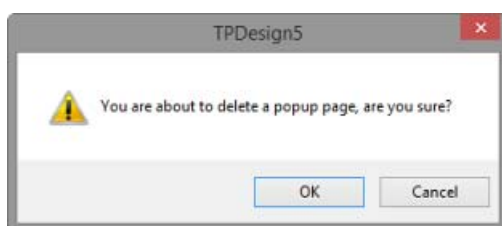
FIG. 67 Prompt - Verify Deleting a Page

Exporting Popup Pages as Image Files

Use the options in the *Export Page Images* dialog to export one or more Popup Pages (and/or Pages) as image files:

1. Select **Panel > Export Page Images** to open the *Export Page Images* dialog (FIG. 68):
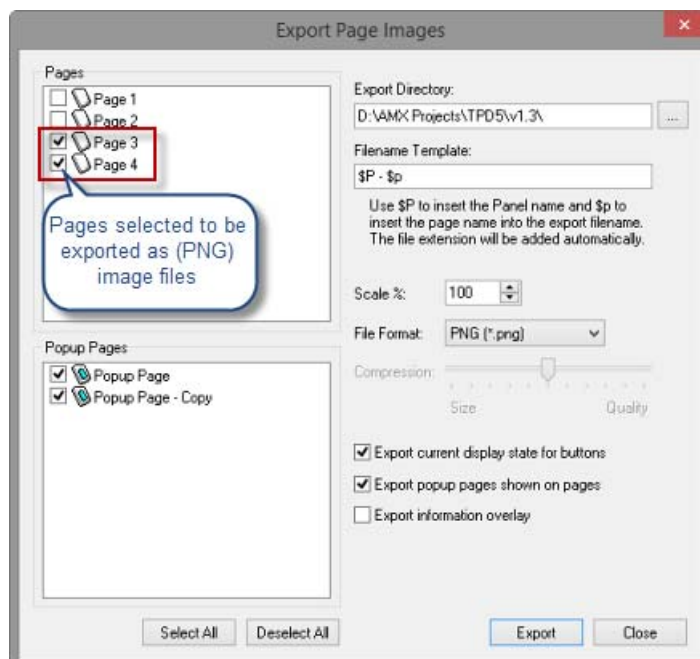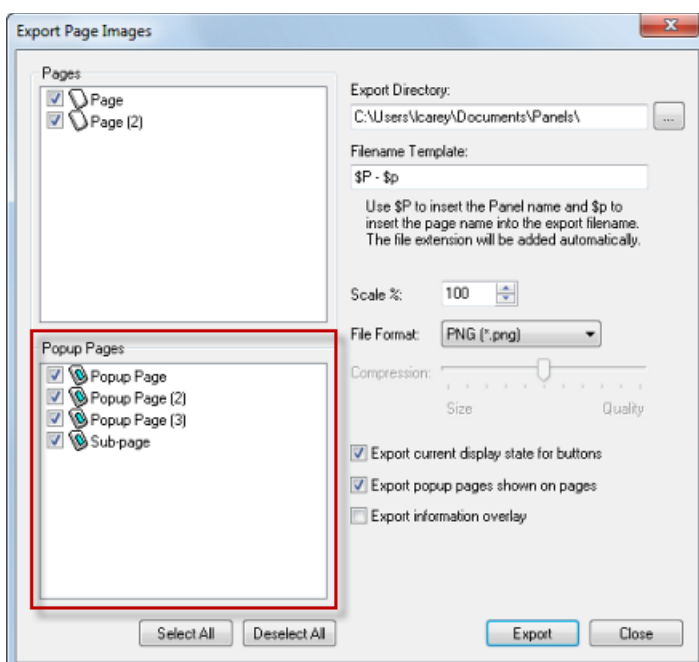

FIG. 68 Export Page Images dialog

2. Use the checkboxes in the *Pages* and *Popup Pages* lists to select which Popup Pages (and Sub-Pages) to export.

3. Specify a target directory for the exported image files in the *Export Directory* field. Use the browse button (…) to navigate to a target directory via the *Browse For Folder* dialog.

4. Specify a template for the resulting filenames in the *Filename Template* field. By default, the template is set to include the Panel file name and the Page name into the generated image's file name. As described on the dialog, use **$P** to insert the Panel name and **$p** to insert the Page name into the exported file's name. The file extension is automatically, based on the *File Format* selection.

5. Use the *Scale% up* and *down* arrows indicate a percentage to down-scale the exported images. The default setting is 100% (no scaling).

6. Select a image file type (JPG, PNG or BMP) for the resulting image files from the File Format drop-down. If JPG is selected, you can select the desired level of image compression

7. Select **Export current display state for buttons** to capture the Pages as they are currently displayed in the Design View, with the button's current display state. If this option is not selected, it will export the buttons in the (default) Off state. By default, this option is enabled.

8. Select **Export popup pages shown on pages** to include any Popup Pages that are being displayed on the Design View at the time of selection in the exported Page image file. If this option is not selected, only the Page underneath any currently viewed Popup Pages will be exported. By default, this option is enabled.

9. Select **Export information overlay** to include any function code information this is currently being displayed on the Design View at the time of selection in the exported Page image file. If this option is not selected, only the Page underneath any currently displayed function codes will be exported. By default, this option is disabled.

10. Click **Export**.

Cut, Copy and Paste - Popup Pages

1. Cut or Copy a Popup Page to clipboard memory:
   - To cut a Popup Page to the clipboard, select a Page in the Workspace window (Pages tab) and select **Cut**. The program will prompt you to verify this action before the Popup Page is removed from the project.
   - To copy a Popup Page to the clipboard, select a Page in the Workspace window (Pages tab) and select **Copy**.

2. Select the target project for the Page in the Workspace window (Pages tab). Popup Pages can be pasted into the current project, or into any other project that is open in the Workspace window.

   Note that all Buttons present on the original Popup Page are also copied, and when pasted they retain the attributes of the original buttons, according to the selections made in the *Paste Controls* dialog.

3. Select **Paste** to paste a copy of the Popup Page into the selected project. If a Popup Page with the same name already exists in the target project, the Popup Page's name will be modified to indicate that it is a copy of another popup. This prevents existing Popup Pages from being overwritten by a Paste operation.

Setting Power Up Popup Pages

Use the *Power-up Popups* option in the *Panel Setup Information* tab of the *Project Properties* dialog to specify one or more popup pages in your project to be displayed over the Power up page when the panel is turned on:

1. Select **File > Project Properties** to open the *Project Properties* dialog.

2. In the *Panel Setup Information* tab, click the green Plus sign icon below the *Power up Popups* field to invoke the *Add Power Up Popup* dialog, presenting a list of all popup pages in the active project (FIG. 69):
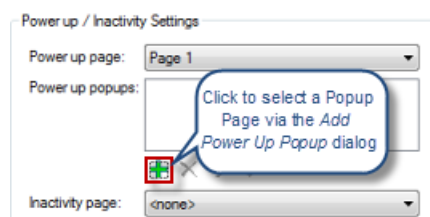
FIG. 69 Project Properties dialog (Panel Setup Information tab) - Add Power Up Popup

3. Select a popup page that you want to be displayed over the Power-up page from this list and click **OK**.

4. The selected popup page is indicated in the Power up Popups field.
   - To display multiple Popups over the Power up page, repeat steps 2 and 3 to select another popup. If you select more than one popup page, you can specify the display order via the up/down arrow buttons below the Power up Popups field. The popup at the top of the list is the first to be displayed.
   - To remove a popup from the Power up Popups list, click the *Remove Popup* **(X)** button below the Power up Popups field.
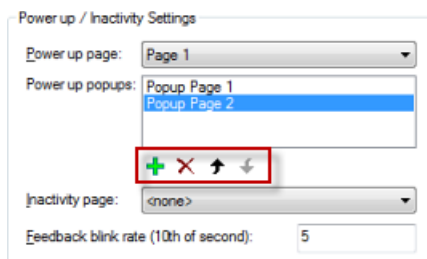
FIG. 70 Project Properties dialog (Panel Setup Information tab) - Power Up Popup options

5.   Click **Apply** to apply this change.

Buttons

Overview

Buttons can be placed on Pages, Popup Pages and Sub-Pages, and can be used to trigger events, to provide level controls, to provide text input, to display images and video fills and more (FIG. 71):
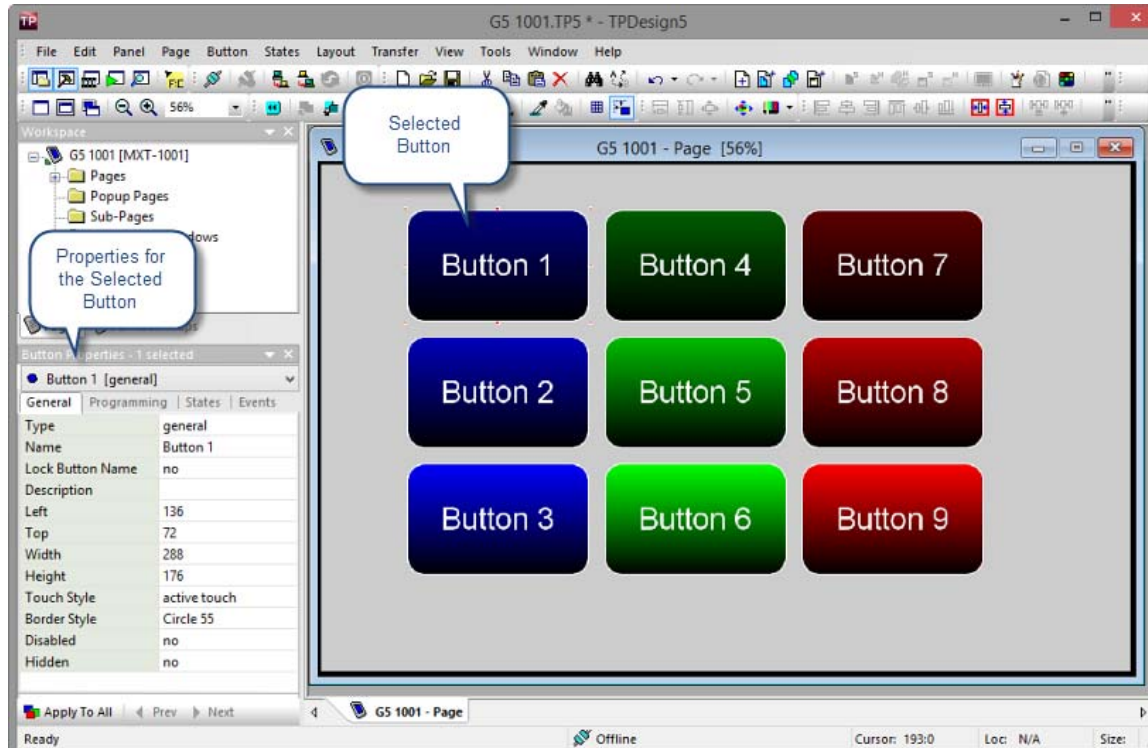
FIG. 71 Buttons

Like other TPD5 Elements, Button properties are managed via the Properties window. TPDesign5 provides various Button Types to accommodate different types of functions:

- **General** buttons - Basic dual-state buttons that can be used for most touch panel functions.
- **Multi-State General** buttons - Basic multi-state buttons that can be used for most touch panel functions, that support up to 256 states. Use multi-state buttons when you want to utilize animation effects.
- **Bargraph** buttons - Level monitors and adjustable level controls that can be configured to monitor or adjust audio outputs and lighting levels.
- **Multi-State Bargraph** buttons - Level monitors and adjustable level controls that can be configured to monitor or adjust audio outputs and lighting levels, that support up to 256 states.
- **Text Input** buttons - Provide a method for the user to enter text on the panel.
- **Sub-Page View** buttons - Serve as "containers" for Sub-Page Sets, and define the area of a Scrolling Region on the panel page. See the *Scrolling Regions (Sub-Pages & Sub-Page View Buttons)* section on page 89 for details on using Sub-Page View buttons and Sub-Pages to create Scrolling Regions on the panel.
- **Listview** buttons - Listview buttons provide the ability to display a listing of items from a dynamic data source on a G5 touch panel. Dynamic data can be created either using an XPort server, NetLinx code or a generic CSV file. The creator of the data can specify how many fields comprise a record and the format of those fields. As many records as necessary can be specified. See the *Listview Buttons & Dynamic Data* section on page 104 for detailed information on implementing Listview buttons.

Note that all button types support General, Programming and State properties, but only General, Multi-State General and Listview buttons support Event properties.

Creating New Buttons

In the Workspace window (Pages tab), double-click the Page or Popup Page to which the button will be added. This opens the target Page or Popup Page in a Design View window for editing. There are two ways to create new buttons:

Drawing a Button

Draw a button using the Button Draw Tool - click and drag on the active Design View window (FIG. 72).
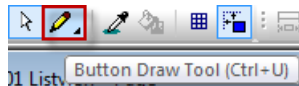
FIG. 72 Button Draw Tool

The Button Draw Tool is used to create all button types (FIG. 73):

Selection handles indicate that this button is selected for editing
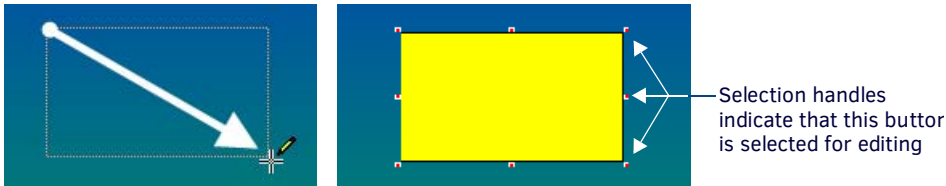
FIG. 73 Drawing a Button

- Upon the creation of a button, note the selection handles displayed on the outsides of the button - these indicate that the button is currently selected for editing.
- The minimum button size is 4 pixels in height or width.

NOTE: *When the Button Draw tool is selected, the Drawing toolbar becomes activated. By default, the options in the Drawing toolbar retain their most recent settings. That is, once you set these options (button type, border, font type/size, border/fill/text colors), the same settings will automatically apply to all consequent new buttons, until they are explicitly changed.*

Copying and Pasting Buttons

1. Cut or Copy one or more Buttons to clipboard memory:
   - Shift+click to select multiple Buttons in a Design View window.
   - To cut a Button to the clipboard, select a Button in the Workspace Window (Pages tab) and select **Cut**.
   - To copy a Button to the clipboard, select a Button in the Workspace Window (Pages tab) and select **Copy**.

2. With one or more Buttons either cut or copied to clipboard memory, select a target Page, Popup Page or Sub-Page in the Design View.

   NOTE: *To copy buttons across Projects, open the Page, Popup Page or Sub-Page in the target Project in a Design View Window.*

3. Select **Paste** to paste the Button(s) into the selected Project/Page, Popup Page or Sub-Page.
   - If a set of copied (or cut) buttons is pasted into a smaller area (for example a Popup Page), the buttons will automatically be resized and re-positioned to fit.
   - The attributes of the copied (or cut) buttons are retained, according to the selections made in the *Paste Controls* dialog (FIG. 74 on page 67).
   - If buttons are copied from one project into another project that is using a different palette, the pasted button will use the palette in the target project. Depending on the differences between the palettes, this can cause color shifting on the button.

Paste Controls dialog

Select **Button > Paste Controls** to open the *Paste Controls* dialog (FIG. 74). Use the options in this dialog to specify exactly which elements of copied buttons you want to retain when they are pasted into a project. You can also select to overwrite image function codes, event options, images/sounds and (in the case of Listview buttons) dynamic data sources, in case you are pasting over existing buttons that have their own image and sound associations. By default, all "overwrite" options are disabled.
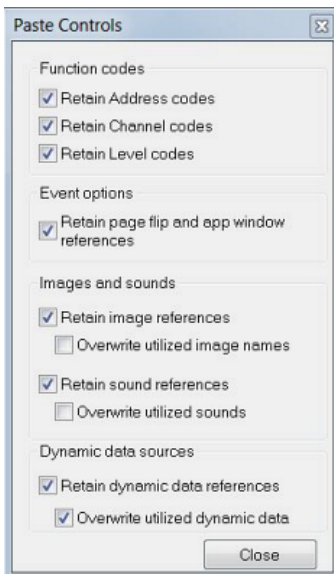
FIG. 74 Paste Controls dialog

Paste Controls dialog options	
Function Codes: Select which aspects of function codes (Address Codes, Channel Codes, and Level Codes) will be retained when the button is pasted from clipboard memory. By default all function codes are retained when pasting the button:	
• Retain Address Codes	Select to retain all Address codes when pasting the button (default = *enabled*)
• Retain Channel Codes	Select to retain all Channel codes when pasting the button (default = *enabled*)
• Retain Level Codes	Select to retain all Level codes when pasting the button (default = *enabled*)
Event options: Select which aspects of Event options (Page Flips, and Application window references) will be retained when the button is pasted from clipboard memory (default = *enabled*).	
• Retain Page Flip and App Window References	Select to retain Page Flip and Application Window references when pasting the button (default = *enabled*).
Images and sounds: Select which collateral information (Image references and Sound references) will be retained when the button is pasted. By default, all collateral information is retained, but the "overwrite" options are disabled:	
• Retain Image References	Select to retain all image file references when pasting the button (default = *enabled*)
• Overwrite Utilized Image Names	Select to overwrite image names, in case you are pasting over existing buttons that have their own image associations (default = *disabled*)
• Retain Sound References	Select to retain all sound file references when pasting the button (default = *enabled*)
• Overwrite Utilized Sound Names	Select to overwrite sound names, in case you are pasting over existing buttons that have their own sound associations (default = *disabled*)
Dynamic Data Sources: Select which collateral information (dynamic data source references) will be retained when the Listview button is pasted. By default, all collateral information is retained, but the "overwrite" options are disabled:	
• Retain Dynamic Data References	Select to retain the dynamic data source assigned to the selected Listview button when pasting (default = *disabled*).
• Overwrite Utilized Dynamic Data	Select to overwrite dynamic data source assignments, in case you are pasting over existing Listview buttons that have their own sound associations (default = *disabled*).

Generated Button Names

When new buttons are created, by default the buttons are automatically given a sequential name composed of the button number (relative to the number of buttons already created in the project). However, TPD5 goes further in automatically generating a descriptive name for the button, based on the text and/or bitmap applied to the button:

- If you apply text to the button, the button text is substituted for the button number. For example if you create a new button (which is automatically named "Button 9"), then add the text "Welcome", note that the button is automatically renamed to "Welcome". Note that this name change occurs on the fly, and does not require a Save operation.
- Text on a button always overrides the presence of a bitmap. If there is text associated with a button, and you add a bitmap, the button name will not change. However, if the button has no text, when you apply a bitmap to the button, the button is automatically renamed to reflect the bitmap file name (i.e. "Background").
- Note that if you change the text, the button is again renamed to reflect the updated text. Furthermore, any time you change the bitmap or the text on the button, the button name is automatically updated with either the latest text or bitmap assignment.

This is where the *Lock Button Name* option (General tab of the Properties window) comes into play. By default, the Lock Button Name option is set to *Off*. At any point in the design process, click *Lock Button Name* and select **Yes** to prevent the button from being automatically renamed by the program when you edit the text or bitmap assignment(s). However, Lock Button Name does not prevent you from manually renaming the button, via the *Name* field.

Setting Default Properties for New Buttons

Use the *Drawing Tools* toolbar to set the default properties for new Buttons (FIG. 75):

FIG. 75 Drawing Tools toolbar

The settings that are specified in this toolbar become the default settings for all new buttons, until the settings are changed. This way, you can quickly create sets of buttons that are visually consistent.

NOTE: *The options in this toolbar are enabled only when a Drawing tool is selected.*

This toolbar is normally at the top of your screen and provides you with a mechanism for controlling various elements of future button creation. This toolbar can either be free floating or docked, but cannot be docked in a vertical fashion, as the drop down lists contained on the toolbar do not support this type of docking.

On the extreme left side of the toolbar, you will see a button that will allow you to toggle the default choices for both the Off and the On state of a button. You can control the button type, border family, font, font size, button fill color, border color and text color.

To set default parameters for new buttons:

1. Select the Button Draw tool from the Selection/Drawing Tools Toolbar. When the Button Draw tool is selected, the Drawing Tools toolbar becomes activated.

2. Buttons are always drawn in the *Off* State. The Drawing toolbar allows you to control certain new button creation properties for both the Off and the On state of a button.

3. Click the down-arrow on the first drop-down menu to the right of the State button to select the type of button you want to create (default = *General*). This selection becomes the default for all subsequent buttons, until the Button Type selection is changed.

4. Click the down-arrow on the second drop-down menu (to the right of the **Button Type** drop-down) to select a Border Style for the button. This selection becomes the default for all subsequent buttons, until the Border Style selection is changed.

5. Use the next two drop-down menus to specify a **Font** and **Font Size** as the default text attributes for new buttons. These selections become the default text settings for all subsequent buttons, until the Font and Font Size specifications are changed.

6. Click the down-arrow on the first palette button to set the **Border Color** for the button. Again, this selection becomes the default for all subsequent buttons, until the Border Color setting is changed.

7. Click the down-arrow on the second palette button to set the **Fill Color** for the button. This selection becomes the default for all subsequent buttons, until the Fill Color setting is changed.

8. Click the down-arrow on the third palette button to set the **Text Color** for the button. Of course, this selection becomes the default for all subsequent buttons, until the Text Color setting is changed.

Drawing Assist Tools

With Buttons, Popup Pages, Sub-Pages and/or Application windows displayed on a Page in the Design View, you can utilize several of the Drawing Assist tools to edit their size and position:

- Order Assist Toolbar
- Position Assist Toolbar
- Size Assist Toolbar

Order Assist Toolbar

The *Order Assist Tool* toolbar (FIG. 76) contains shortcuts to control options for controlling the layering order of selected Buttons, Popup Pages and Sub-Pages.

FIG. 76 Order Assist Toolbar

Note that the Order Assist Tools do not apply to Application windows, since they always appear above other popups and buttons on a page.

Order Assist Toolbar options	
Send to Front:	Click to bring the selected object(s) to the front (or top) layer, so that they appear to be in "front" of the other objects on the page.
Send to Back:	Click to send the selected object(s) to the back (or bottom) layer, so that they appear to be "behind" the other objects on the page.
Shift Up:	Click to shift the selected object(s) one layer up on the page. Depending on the amount of layering, this may or may not place the object(s) on the "Front" or top level of the page.
Shift Down:	Click to shift the selected object(s) one layer down on the page. Depending on the amount of layering, this may or may not place the object(s) on the "Back" or bottom level of the page.

- All toolbars in TPD5 are dockable, so they can be arranged within the application window.
- To undock any toolbar, double-click anywhere inside the toolbar (but not on a toolbar button). Double-click again to dock the toolbar.
- To move the toolbar, click and drag using the handle (the single vertical bar) at the far-left side of the toolbar.
- Select **View > Toolbars > Order Assist Toolbar** to show/hide this toolbar:

Position Assist Toolbar

The Position Assist toolbar (FIG. 77) contains shortcuts to many layout control options for controlling various placement elements of existing buttons.

FIG. 77 Position Assist Toolbar

Position Assist Toolbar options	
Align Left:	With two or more objects selected, click to align the left edges of the selected objects.
Align Horizontal Center:	With two or more objects selected, click to distribute the center points of the selected objects in a horizontal line.
Align Right:	With two or more objects selected, click to align the right edges of the selected objects.
Align Top:	With two or more objects selected, click to align the top edges of the selected objects.
Align Vertical Center:	With two or more objects selected, click to distribute the center points of the selected objects in a vertical line.
Align Bottom:	With two or more objects selected, click to align the bottom edges of the selected objects.
Center Horizontal:	With one or more objects selected, click to center the object horizontally, relative to the page. With multiple objects selected, click to center the objects horizontally as a group.
Center Vertical:	With one or more objects selected, click to center the object vertically, relative to the page. With multiple objects selected, click to center the objects vertically as a group.
Equal Horizontal Spacing:	With three or more objects selected, use this option to automatically distribute the objects with an equal amount of horizontal space between them.
Increase Horizontal Spacing:	With three or more objects selected, use this option to increase the amount of horizontal space between them. Use this option in conjunction with the Equal Horizontal Spacing option to quickly align and distribute objects horizontally on the page.
Decrease Horizontal Spacing:	With three or more objects selected, use this option to decrease the amount of horizontal space between them. Use this option in conjunction with the Equal Horizontal Spacing option to quickly align and distribute objects horizontally on the page.
Remove Horizontal Spacing:	With three or more objects selected, use this option to remove all horizontal space between them.
Equal Vertical Spacing:	With three or more objects selected, use this option to automatically distribute the objects with an equal amount of vertical space between them.
Increase Vertical Spacing:	With three or more objects selected, use this option to increase the amount of vertical space between them. Use this option in conjunction with the Equal Vertical Spacing option to quickly align and distribute objects horizontally on the page.
Decrease Vertical Spacing:	With three or more objects selected, use this option to decrease the amount of vertical space between them. Use this option in conjunction with the Equal Vertical Spacing option to quickly align and distribute objects horizontally on the page.
Remove Vertical Spacing:	With three or more objects selected, use this option to remove all vertical space between them.

- All toolbars in TPD5 are dockable, so they can be arranged within the application window.
- To undock any toolbar, double-click anywhere inside the toolbar (but not on a toolbar button). Double-click again to dock the toolbar.
- To move the toolbar, click and drag using the handle (the single vertical bar) at the far-left side of the toolbar.
- Select **View > Toolbars > Position Assist Toolbar** to show/hide this toolbar:

Size Assist Toolbar

The Size Assist Tool toolbar (FIG. 78) contains shortcuts to many layout control options for controlling various size elements of existing buttons:

FIG. 78 Size Assist Toolbar

Size Assist Toolbar options	
Make Same Width:	With two or more objects selected, click to adjust the objects to match the width of the first object selected. Select the object with the desired width first, then select the object(s) that you want to resize to match.
Make Same Height:	With two or more objects selected, click to adjust the objects to match the height of the first object selected. Select the object with the desired width first, then select the object(s) that you want to resize to match.
Make Same Size:	With two or more objects selected, click to adjust the objects to match the size (height/width) of the first object selected. Select the object with the desired size first, then select the object(s) that you want to resize to match.
Size To Fit Image:	This option automatically resizes the selected button(s) to accommodate the largest bitmap contained within the selected button(s).
Size for Video:	When using a video fill button, use this option to specify an aspect ratio to apply to the selected button.

- All toolbars in TPD5 are dockable, so they can be arranged within the application window.
- To undock any toolbar, double-click anywhere inside the toolbar (but not on a toolbar button). Double-click again to dock the toolbar.
- To move the toolbar, click and drag using the handle (the single vertical bar) at the far-left side of the toolbar.
- Select **View > Toolbars > Size Assist Toolbar** to show/hide this toolbar.

Setting Button Properties

Buttons have a set of Properties that can be configured via the fields in the Properties window.

To set button-level properties, click on a button (on a page or popup page) in the Design View. With the button selected, the Properties window displays the properties available for the selected button type.

Each button type has its own requirements in terms of properties, so the properties listed depend on the type of button selected.

Editing Button Properties

Edit button properties via the Properties window: select the button property that you wish to modify, and enter (type or select) a new value in the corresponding field.

Once a property has been modified, either press the Enter key, the *Prev* or *Next* button, or left click the mouse in any box other than the current box to apply the change.

NOTE: *If the value is inappropriate for the selected object, you will be notified via a message, and the previous value will be replaced in the box.*

Using the Selection Tool

To activate the Selection tool, select **Edit > Selection Tool**, or click the toolbar button in the Selection/Button Draw Toolbar (FIG. 79):
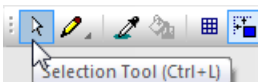
FIG. 79 Selection/Button Draw Toolbar

Use the Selection tool to pick/select objects in the Design View window(s). You can also select multiple buttons, using any of the following techniques:

1. Left click on the remaining desired buttons while simultaneously depressing the Shift key on your keyboard.
2. You may perform a marquis selection by holding down the left mouse button outside the boundaries of the desired buttons and drawing a selection box around the desired buttons.
3. In the Properties window, you may turn the Apply To All toggle button (located at the bottom of the Properties window) On, then individually select each desired button.
4. To select all available buttons on a page or popup page you may either use the Ctrl+A hotkey, or select Edit > Select All from the main menu.

NOTE: *Although you may select and retain the selection of buttons on multiple pages, you may only act on the selected buttons on one page / popup page at a time.*

- When using the Selection Tool, hold down the ALT key while clicking and dragging in a Design View window to move the current selections without selecting anything new on the mouse press (useful for moving popup pages whose entire area is filled with buttons).

- When using the Selection Tool, hold down the CTRL key while clicking and dragging to force a "lasso" selection to occur (even if the mouse was clicked over a button or Popup Page shown on the Page). Lasso selection forces you to draw a square around the outside of the item to select it, as opposed to clicking on the item.
- With an item selected in a Design View window, hold down the CTRL key while moving the item with the keyboard arrow keys to move by the grid size instead of a single pixel (regardless of the grid visibility or snap to grid setting). Holding the CTRL key while resizing the selected items with the keyboard will resize by the grid size.
- To de-select a button, either select another button, or left-click on the background of the page or popup page.

Editing Multiple Selections

Buttons may be acted upon individually by clicking the Selection Tool from the toolbar or selecting **Edit > Selection Tool** from the main menu, then clicking on the desired button.

You can also select multiple buttons, using any of the following techniques:

- Hold the Shift key and left-click.
- Click and drag outside the boundaries of the desired buttons to perform a marquis selection.
- To select all available buttons on a page or popup page you may either use the Ctrl+A hotkey, or select Edit > Select All from the main menu. Although you may select and retain the selection of buttons on multiple pages, you may only act on the selected buttons on one page / popup page at a time.

NOTE: *You can de-select a button by either selecting another button, or by left-clicking on the background of the Page or Popup Page.*

Previewing Buttons

The *Button Preview* window allows you to preview a button so you can check size, border, text and color settings on a Push and Release of the button. To preview a button:

1. Select the button that you want to preview.
2. Select **View > Button Preview** to open the Button Preview window (FIG. 80):
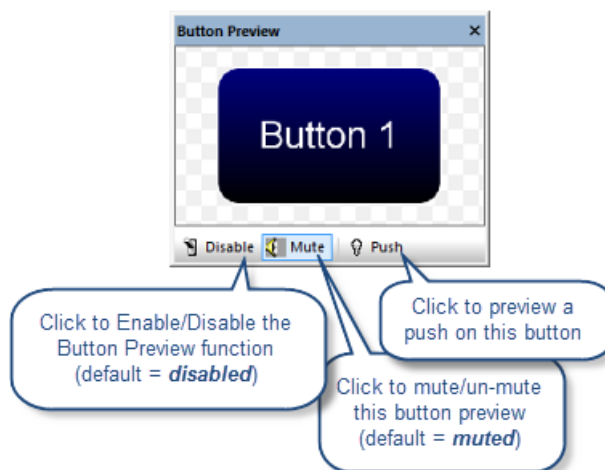
FIG. 80 Button Preview window

- Like the other windows in TPD5, the Button Preview window is fully dockable, but initially it opens undocked. Double-click inside the title bar to dock the window.
- The Button Preview window displays a preview of the selected button. Initially the button is shown in its Off state.

3. Click on the **Push** button to preview the selected button's behavior when pressed.

Deleting Buttons

To delete a button from the active Page, select a button to delete in the Design View window, and press the Delete key.

Alternatively, select **Edit > Delete** (or click the Delete toolbar button).

General Buttons

General buttons are basic dual-state buttons that can be used for most touch panel functions, and are not associated with a specific functionality (as opposed to other button types, which have specific functionalities).

Creating General Buttons

General buttons are basic dual-state buttons that can be used for most touch panel functions, and are not associated with a specific functionality (as opposed to other button types, which have specific functionalities).

To create a new General button:

1. Open the Page, Popup Page or Sub-Page that the button will be added to (double-click on the page in the Workspace window (Pages tab) to open the page and bring it into edit focus).
2. In the Design View, select the Page, Popup Page or Sub-Page to which you want to add the button.

3. Use the Button Draw tool to create a new button. See *Creating New Buttons* on page 66 for details.

4. In the *General* tab of the Properties window, set the *Type* property to **General**.

   This selection updates the Properties window to represent properties specific to this button type.

5. Set the button properties as desired by editing the *General*, *Programming*, *States* and *Events* properties in the Properties window.

6. Select **File > Save** to save your changes.

General Buttons - General Properties

Once you have created a General button, you can use the *General* tab of the Properties window to set/edit general (non-state oriented) button properties. To edit any of the properties in the table, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The following General button properties are supported for *General* buttons:

General Buttons - General Properties		
	• Type	see page 247
	• Name	see page 242
	• Lock Button Name	see page 242
	• Description	see page 236
	• Left	see page 238
	• Top	see page 246
	• Width	see page 247
	• Height	see page 237
	• Z-Order	see page 247
	• Drag/Drop Type	see page 236
	• Drop Group	see page 236
	• Touch Style	see page 247
	• Border Style	see page 236
	• Disabled	see page 236
	• Hidden	see page 237

General Buttons - Programming Properties

Once you have created a General button, you can use the *Programming* tab of the Properties window to set/edit programming-oriented button properties. To edit any of the properties, click in the right-hand table cell to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The following Programming properties are supported for *General* buttons:

General Buttons - Programming Properties		
	• Feedback	see page 247
	• Address Port	see page 247
	• Address Code	see page 249
	• Channel Port	see page 249
	• Channel Code	see page 249
	• Level Control Type	see page 250

NOTE: *Maximum command, string and text length = 4096 characters.*

General Buttons - States Properties

Once you have created a General button, you can use the *States* tab of the Properties window to set/edit state-oriented button properties. To edit any of the listed button properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The State tab of the Properties window works in conjunction with the State Manager window. Note that if the State Manager is not displayed, or if no state(s) are selected in the State Manager, the States tab shows a list of all states associated with the selected button.

Each state represented in the States tab is a collapsed folder containing the state properties for that particular state. Click the + symbol to expand each folder. If you select a state (or multiple states) in the State Manager, then the States tab only represents the selected state(s).

Use the **All States** option to apply any changes you make to all states on the selected button. Note that if you have multiple buttons selected (Shift+click to select multiple buttons a page), the All States option only affects states for the button that has Edit Focus. The button with edit focus would be the last one selected, and is indicated by having red-colored square handles (as opposed to the black squares that indicate that a button is selected, but does not currently have edit focus).

The following State properties are supported for General buttons (for each state). Note that depending on the Panel associated with your project some of these options may not be available.

General Buttons - States Properties

	Property	Reference
•	Border Name	see page 252
•	Border Color	see page 251
•	Fill Type	see page 252
•	Fill Color	see page 252
•	Fill Gradient Colors	see page 252
•	Gradient Radius	see page 253
•	Gradient Center X%	see page 253
•	Gradient Center Y%	see page 253
•	Text Color	see page 254
•	Text Effect Color	see page 254
•	Overall Opacity	see page 253
•	Video Fill	see page 254
•	Streaming Source	see page 253
•	Bitmaps	see page 251
•	Font	see page 252
•	Font Size	see page 253
•	Text	see page 254
•	Text Justification	see page 254
•	Text X Offset	see page 254
•	Text Y Offset	see page 254
•	Text Effect	see page 254
•	Word Wrap	see page 254
•	Sound	see page 253

General Buttons - Events Properties

Modero X Series panels support Gestures for on-screen navigation. Gestures can be used (in addition to Buttons) for navigating the panel UI. For example, a "Swipe" gesture can invoke a page flip when the user swipes a finger across the screen. Gestures are presented in TPDesign5 under the *Events* tab of the Properties window.

To choose a particular property for a button , highlight the row and click the Browse button (…) to open the *Edit Event Actions* dialog. Use this dialog to add actions instigated by the gesture selected. Click the **Add Action** button to select between a Send Command or a command string, and enter the command or string in the field. When finished, click **OK**.

The following Events are supported for *General* buttons:

General Buttons - Events Properties

	Event	Reference
•	Button Press	see page 257
•	Button Release	see page 257
•	Gesture Any	see page 257
•	Gesture Up	see page 257
•	Gesture Down	see page 257
•	Gesture Right	see page 258
•	Gesture Left	see page 258
•	Gesture Dbl Tap	see page 258
•	Gesture 2-Finger Up	see page 258
•	Gesture 2-Finger Dn	see page 258
•	Gesture 2-Finger Rt	see page 258
•	Gesture 2-Finger Lt	see page 258

Multi-State General Buttons

Multi-State General buttons are basic multi-state buttons that can be used for most touch panel functions, and that support up to 256 states. Use multi-state buttons when you want to utilize animation effects.

NOTE: *Since multi-state buttons are pre-rendered, meaning that touch panel memory is allocated in advance for each state, be aware of the potential for excessive memory usage by multi-state buttons. Always take into account that large multi-state buttons, depending upon their size and their functionality, may use more memory than is available to the touch panel.*

Creating Multi-State General Buttons

Multi-State General buttons are basic multi-state buttons that can be used for most touch panel functions, and that support up to 256 states. Use multi-state buttons when you want to utilize animation effects. To create a new Multi-State General button:

1. Open the Page, Popup Page or Sub-Page that the button will be added to (double-click on the page in the Workspace window (Pages tab) to open the page and bring it into edit focus).
2. Select the Button Draw tool from the Button Selection/Draw toolbar.
3. Click on a Page, Popup Page or Sub-Page (in the active Design View window), and while holding the mouse button down, drag to draw the button to the desired size and shape. Upon the creation of a button on a page (or popup page) you will see selection handles appear on the outsides of the button that are small yellow squares with red interiors. This is a visual indication that the newly created button has the Edit Focus.
4. In the General tab of the Properties window, set the *Type* property to **Multi-State General**.
   This selection updates the Properties window to represent properties specific to this button type.
5. Set the button properties as desired by editing the *General*, *Programming*, *States* and *Events* properties in the Properties window.
6. Select **File > Save** to save your changes.

Multi-State General Buttons - General Properties

Once you have created a Multi-State General button, you can use the *General* tab of the Properties window to set/edit general (non-state oriented) button properties. To edit any of the properties in the table, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both. The following General button properties are supported for *Multi-State General* buttons:

Multi-State General Buttons - General Properties	
• Type	see page 247
• Name	see page 242
• Lock Button Name	see page 242
• Description	see page 236
• Left	see page 238
• Top	see page 246
• Width	see page 247
• Height	see page 237
• Z-Order	see page 247
• Drag/Drop Type	see page 236
• Drop Group	see page 236
• Touch Style	see page 247
• Border Style	see page 236
• State Count	see page 246
• Animate Time Up	see page 235
• Animate Time Down	see page 235
• Auto-Repeat	see page 236
• Disabled	see page 236
• Hidden	see page 237

Multi-State General Buttons - Programming Properties

Once you have created a Multi-State General button, you can use the Programming tab of the Properties window to set/edit programming-oriented button properties. To edit any of the properties, click in the right-hand table cell to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The following Programming properties are supported for *Multi-State General* buttons:

Multi-State General Buttons - Programming Properties		
	• Feedback	see page 247
	• Address Port	see page 247
	• Address Code	see page 249
	• Channel Port	see page 249
	• Channel Code	see page 249
	• Level Control Type	see page 250

NOTE: *Maximum command, string and text length = 4096 characters.*

Multi-State General Buttons - States Properties

FIG. 81 Multi-State General Buttons - States Properties

Once you have created a Multi-State General button, you can use the *States* tab of the Properties window to set/edit state-oriented button properties. To edit any of the listed button properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The State tab of the Properties window works in conjunction with the State Manager window. Note that if the State Manager is not displayed, or if no state(s) are selected in the State Manager, the States tab shows a list of all states associated with the selected button. Each state represented in the States tab is a collapsed folder containing the state properties for that particular state. Click the + symbol to expand each folder. If you select a state (or multiple states) in the State Manager, then the States tab only represents the selected state(s).

Use the **All States** option to apply any changes you make to all states on the selected button. Note that if you have multiple buttons selected (Shift+click to select multiple buttons a page), the All States option only affects states for the button that has Edit Focus. The button with edit focus would be the last one selected, and is indicated by having red-colored square handles (as opposed to the black squares that indicate that a button is selected, but does not currently have edit focus).

NOTE: *The maximum number of states for Multi-State type buttons = 256.*

The following State properties are supported for *Multi-State General* buttons (for each state). Note that depending on the Panel associated with your project some of these options may not be available.

Multi-State General Buttons - States Properties

•	Border Name	see page 252
•	Border Color	see page 251
•	Fill Type	see page 252
•	Fill Color	see page 252
•	Fill Gradient Colors	see page 252
•	Gradient Radius	see page 253
•	Gradient Center X%	see page 253
•	Gradient Center Y%	see page 253
•	Text Color	see page 254
•	Text Effect Color	see page 254
•	Overall Opacity	see page 253
•	Video Fill	see page 254
•	Streaming Source	see page 253
•	Bitmaps	see page 251
•	Font	see page 252
•	Font Size	see page 253
•	Text	see page 254
•	Text Justification	see page 254
•	Text X Offset	see page 254
•	Text Y Offset	see page 254
•	Text Effect	see page 254
•	Word Wrap	see page 254
•	Sound	see page 253

NOTE: *Since multi-state buttons are pre-rendered, meaning that touch panel memory is allocated in advance for each state, be aware of the potential for excessive memory usage by multi-state buttons. Always take into account that large multi-state buttons, depending upon their size and their functionality, may use more memory than is available to the touch panel.*

Multi-State General Buttons - Events Properties

Modero X Series panels support Gestures for on-screen navigation. Gestures can be used (in addition to Buttons) for navigating the panel UI. For example, a "Swipe" gesture can invoke a page flip when the user swipes a finger across the screen. Gestures are presented in TPDesign5 under the *Events* tab of the Properties window.

To choose a particular property for a button, highlight the row and click the Browse button (...) to open the *Edit Event Actions* dialog. Use this dialog to add actions instigated by the gesture selected. Click the **Add Action** button to select between a Send Command or a command string, and enter the command or string in the field. When finished, click **OK**.

The following Events are supported at the button level:

Multi-State General Buttons - Events Properties

•	Button Press	see page 257
•	Button Release	see page 257
•	Gesture Any	see page 257
•	Gesture Up	see page 257
•	Gesture Down	see page 257
•	Gesture Right	see page 258
•	Gesture Left	see page 258
•	Gesture Dbl Tap	see page 258
•	Gesture 2-Finger Up	see page 258
•	Gesture 2-Finger Dn	see page 258
•	Gesture 2-Finger Rt	see page 258
•	Gesture 2-Finger Lt	see page 258

Bargraph Buttons

Bargraph Buttons are level monitors and adjustable level controls that can be configured to monitor or adjust audio outputs and lighting levels.

Creating Bargraph Buttons

1. Open the Page, Popup Page or Sub-Page that the button will be added to (double-click on the page in the Workspace window (Pages tab) to open the page and bring it into edit focus).
2. Select the Button Draw tool from the Button Selection/Draw toolbar.
3. Click on a Page, Popup Page or Sub-Page (in the active Design View window), and while holding the mouse button down, drag to draw the button to the desired size and shape. Upon the creation of a button, you will see selection handles appear on the outsides of the button that are small yellow squares with red interiors. This is a visual indication that the newly created button has the Edit Focus.
4. In the *General* tab of the Properties window, set the *Type* property to **Bargraph**. This selection updates the Properties window to represent properties specific to this button type.
5. Set the button properties as desired by editing the *General*, *Programming* and *States* properties in the Properties window.
6. Select **File > Save** to save your changes.

Bargraph Buttons - General Properties

Once you have created a Bargraph button, you can use the *General* tab of the Properties window to set/edit general (non-state oriented) button properties. To edit any of the properties in the table, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The following General button properties are supported for *Bargraph* buttons:

Bargraph Buttons - General Properties		
	• Type	see page 247
	• Name	see page 242
	• Lock Button Name	see page 242
	• Description	see page 236
	• Left	see page 238
	• Top	see page 246
	• Width	see page 247
	• Height	see page 237
	• Border Style	see page 236
	• Disabled	see page 236
	• Hidden	see page 237
	• Value Direction	see page 247
	• Slider Name	see page 246
	• Slider Color	see page 246

Bargraph Buttons - Programming Properties

Once you have created a Bargraph button, you can use the Programming tab of the Properties window to set/edit programming-oriented button properties. To edit any of the properties, click in the right-hand table cell to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both. The following Programming properties are supported for *Bargraph* buttons:

Bargraph Buttons - Programming Properties		
	• Address Port	see page 247
	• Address Code	see page 249
	• Channel Port	see page 249
	• Channel Code	see page 249
	• Level Port	see page 250
	• Level Code	see page 250
	• Level Function	see page 251
	• Range Low	see page 251
	• Range High	see page 251
	• Range Inverted	see page 251
	• Range Time Up	see page 251
	• Range Time Down	see page 251

NOTE: *Maximum command, string and text length = 4096 characters.*
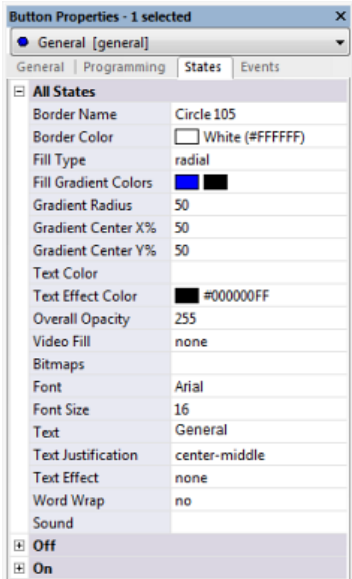
Bargraph Buttons - States Properties

Once you have created a Bargraph button, you can use the *States* tab of the Properties window to set/edit state-oriented button properties. To edit any of the listed button properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The State tab of the Properties window works in conjunction with the State Manager window. Note that if the State Manager is not displayed, or if no state(s) are selected in the State Manager, the States tab shows a list of all states associated with the selected button. Each state represented in the States tab is a collapsed folder containing the state properties for that particular state. Click the + symbol to expand each folder. If you select a state (or multiple states) in the State Manager, then the States tab only represents the selected state(s).

The following State properties are supported for *Bargraph* buttons (for each state). Note that depending on the Panel associated with your project some o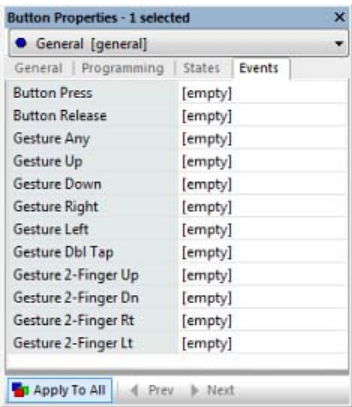f these options may not be available. For example if you have specified a non-video capable panel in your project, the *Video Fill* option(s) will not appear:

Bargraph Buttons - States Properties		
	• Border Name	see page 252
	• Border Color	see page 251
	• Fill Type	see page 252
	• Fill Color	see page 252
	• Fill Gradient Colors	see page 252
	• Gradient Radius	see page 253
	• Gradient Center X%	see page 253
	• Gradient Center Y%	see page 253
	• Text Color	see page 254
	• Text Effect Color	see page 254
	• Overall Opacity	see page 253
	• Video Fill	see page 254
	• Streaming Source	see page 253
	• Bitmaps	see page 251
	• Font	see page 252
	• Font Size	see page 253
	• Text	see page 254
	• Text Justification	see page 254
	• Text X Offset	see page 254
	• Text Y Offset	see page 254
	• Text Effect	see page 254
	• Word Wrap	see page 254

NOTE: *Bargraph buttons do not support Events.*

Multi-State Bargraph Buttons

Multi-State Bargraph Buttons are level monitors and adjustable level controls that can be configured to monitor or adjust audio outputs and lighting levels, and that (like Multi-State General buttons) support up to 256 states. Use multi-state buttons when you want to utilize animation effects. Multi-state Bargraph buttons also allow you to create a custom Bargraph slider (using an image icon).

NOTE: *Since multi-state buttons are pre-rendered, meaning that touch panel memory is allocated in advance for each state, be aware of the potential for excessive memory usage by multi-state buttons. Always take into account that large multi-state buttons, depending upon their size and their functionality, may use more memory than is available to the touch panel.*
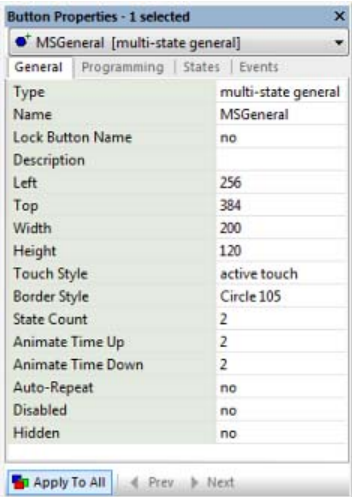
Creating Multi-State Bargraph Buttons

1. Open the Page, Popup Page or Sub-Page that the button will be added to (double-click on the page in the Workspace window (Pages tab) to open the page and bring it into edit focus).

2. Select the Button Draw tool from the Button Selection/Draw toolbar.

3. Click on a Page, Popup Page or Sub-Page (in the active Design View window), and while holding the mouse button down, drag to draw the button to the desired size and shape.

   Upon the creation of a button, you will see selection handles appear on the outsides of the button that are small yellow squares with red interiors. This is a visual indication that the newly created button has the Edit Focus.

4. In the *General* tab of the Properties window, set the *Type* property to **Multi-State Bargraph**.

   This selection updates the Properties window to represent properties specific to this button type.

5. Set the button properties as desired by editing the *General*, *Programming* and *States* properties in the Properties window.

6. Select **File > Save** to save your changes.

Multi-State Bargraph Buttons - General Properties

Once you have created a Multi-State Bargraph button, you can use the General tab of the Properties window to set/edit general (non-state oriented) button properties. To edit any of the properties in the table, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The following General button properties are supported for *Multi-State Bargraph* buttons:

Multi-State Bargraph Buttons - General Properties		
	• Type	see page 247
	• Name	see page 242
	• Lock Button Name	see page 242
	• Description	see page 236
	• Left	see page 238
	• Top	see page 246
	• Width	see page 247
	• Height	see page 237
	• Border Style	see page 236
	• State Count	see page 246
	• Disabled	see page 236
	• Hidden	see page 237
	• Value Direction	see page 247
	• Touch Map	see page 246

Multi-State Bargraph Buttons - Programming Properties

Once you have created a Multi-State Bargraph button, you can use the Programming tab of the Properties window to set/edit programming-oriented button properties. To edit any of the properties, click in the right-hand table cell to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The following Programming properties are supported for *Multi-State Bargraph* buttons:

Multi-State Bargraph Buttons - Programming Properties		
	• Address Port	see page 247
	• Address Code	see page 249
	• Channel Port	see page 249
	• Channel Code	see page 249
	• Level Port	see page 250
	• Level Code	see page 250
	• Level Function	see page 251
	• Range Low	see page 251
	• Range High	see page 251
	• Range Inverted	see page 251
	• Range Time Up	see page 251
	• Range Time Down	see page 251

NOTE: *Maximum command, string and text length = 4096 characters.*

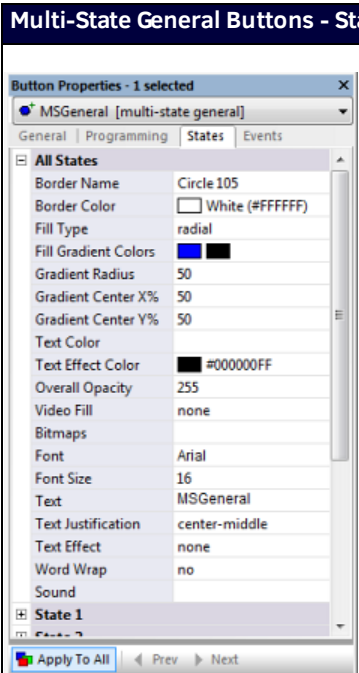## Multi-State Bargraph Buttons - States Properties

Once you have created a Multi-State Bargraph button, you can use the *States* tab of the Properties window to set/edit state-oriented button properties. To edit any of the listed button properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The State tab of the Properties window works in conjunction with the State Manager window. Note that if the State Manager is not displayed, or if no state(s) are selected in the State Manager, the States tab shows a list of all states associated with the selected button. Each state represented in the States tab is a collapsed folder containing the state properties for that particular state. Click the + symbol to expand each folder. If you select a state (or multiple states) in the State Manager, then the States tab only represents the selected state(s).

Use the **All States** option to apply any changes you make to all states on the selected button. Note that if you have multiple buttons selected (Shift+click to select multiple buttons a page), the All States option only affects states for the button that has Edit Focus. The button with edit focus would be the last one selected, and is indicated by having red-colored square handles (as opposed to the black squares that indicate that a button is selected, but does not currently have edit focus).

FIG. 82 Multi-State Bargraph Buttons - States Properties

NOTE: *The maximum number of states for Multi-State type buttons = 256.*

The following State properties are supported for *Multi-State Bargraph* buttons (for each state). Note that depending on the Panel associated with your project some of these options may not be available.

Multi-State Bargraph Buttons - States Properties		
	• Border Name	see page 252
	• Border Color	see page 251
	• Fill Type	see page 252
	• Fill Color	see page 252
	• Fill Gradient Colors	see page 252
	• Gradient Radius	see page 253
	• Gradient Center X%	see page 253
	• Gradient Center Y%	see page 253
	• Text Color	see page 254
	• Text Effect Color	see page 254
	• Overall Opacity	see page 253
	• Video Fill	see page 254
	• Streaming Source	see page 253
	• Bitmaps	see page 251
	• Font	see page 252
	• Font Size	see page 253
	• Text	see page 254
	• Text Justification	see page 254
	• Text X Offset	see page 254
	• Text Y Offset	see page 254
	• Text Effect	see page 254
	• Word Wrap	see page 254
	• Sound	see page 253

NOTE: *Multi-State Bargraph buttons do not support Events.*

Creating a Custom Slider

Multi-State Bargraph buttons work differently than regular Bargraph buttons. Note that when you draw a Multi-State Bargraph button, that there is no slider indicated on the button (in the Design View window).

Also note that unlike regular Bargraph buttons, there are no slider-oriented settings to make in the Properties window. This is because multi-state Bargraph buttons, like multi-state General buttons, use up to 255 states to animate the button action. In the case of Bargraph buttons, you'll be animating a change in levels as opposed to a push/release, as on Multi-State General buttons. Rather than assign a prepared slider, like you would for a normal Bargraph button, you can animate an icon across the states to serve as a custom slider.

Custom sliders on Multi-state Bargraph Buttons work on the panel basically the same as regular sliders. You adjust a level, you touch the Bargraph button and move the slider up and down (or side to side on a horizontal Bargraph button).

To create a custom slider:

1. Create a Multi-State Bargraph button.

2. Select **State 1** in the Properties window (States tab), or in the State Manager window.

3. Apply a bitmap assignment to State 1.

4. Set the *Bitmap Justification* to **Absolute**.

5. In the State Manager, right-click on State 1 and select **Image/Text Positioning** to open the *Image and Text Positioning* dialog.

6. In the *Image and Text Positioning* dialog, move the bitmap into the position that you want to be the "start position" for the Bargraph slider. The start position for Multi-State Bargraph buttons is always the bottom of the Bargraph (representing the minimum level setting).

   NOTE: *One key difference between regular and Multi-State Bargraph buttons is the way they work on the panel. The button action is the same (press the Bargraph slider and drag to adjust the level), but while regular Bargraph buttons can be set as either horizontal or vertical, Multi-State Bargraph buttons are always oriented vertically. The user will always press and drag the slider up and down to adjust the level. Keep this in mind when setting up a custom slider. Always begin the icon animation starting (at State 1) at the bottom position, and ending at the top.*

7. Add States to the button (up to 255 total), duplicating State 1 (containing the bitmap). Generally, you'll want to delete the "extra" state at the end of the sequence, which does not contain the bitmap (the original State 2 setting).

   NOTE: *Keep in mind that the more states used to animate the movement of the icon, the smoother and more accurate the slider will be. The range of motion (i.e. the size of the button that the icon will travel across) needed for the slider should be taken in to consideration as well. A "short" Bargraph button would require less states to create a smooth motion than a "long" one (that for example spans the entire touch panel page).*

8. Select the last State in the Properties window (*States* tab), or in the State Manager window.

9. In the *Image and Text Positioning* dialog, move the bitmap into the position that you want to be the "end position" for the Bargraph slider. The end position for Multi-State Bargraph buttons is always the top of the Bargraph (representing the maximum level setting).

10. In the State Manager window, Ctrl+click to select just the first and last states.

11. Right-click on either of the selected states and select **Slot Position** from the *Tweeners* sub-menu. The results of the tweening are displayed in the State Manager window.

To preview the custom slider in action, open the Button Preview window, and click and drag the cursor up and down within the button to move the slider up and down.

Remember, you can also utilize the other tweeners in conjunction with the custom slider to create complex color transition effects.

Working With Touch Maps

TPD5 supports Touch Maps for Multi-State Bargraph buttons. Touch Map images allow you to use irregular shapes for active Bargraph buttons. If you select *Touch Map* as the Value Direction (state property), an additional state property is enabled (displayed directly below *Value Direction* in the Properties window) called **Touch Map**.

Click the browse button (...) next to **Touch Map** to select an image to use as a Touch Map (via the Resource Manager). The alpha values in the selected image represent the areas where touch will be registered, and the red-channel values represent the overall value to change the control to.

Formatting Codes

Formatting codes can be used in the Text for Bargraph and Multi-State Bargraph buttons. The following formatting codes will be replaced with the identified values:

- $P: level percentage
- $V: raw level value
- $L: range low
- $H: range high
- $A: adjusted level value (raw level value - range low)
- $R: range (range high - range low)
- $$: $ character

NOTE: *Bargraph and Multi-State Bargraph buttons do not support Events.*

Text Input Buttons
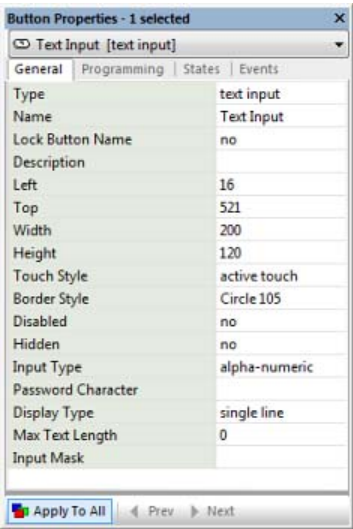
Text Input Buttons provide a method for the user to enter text on the panel.

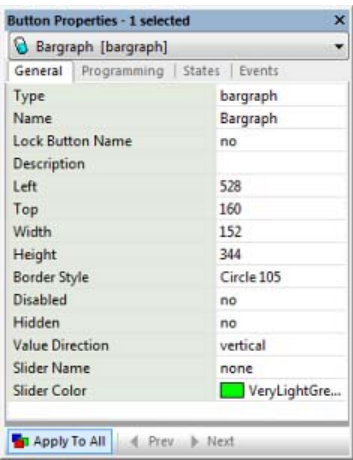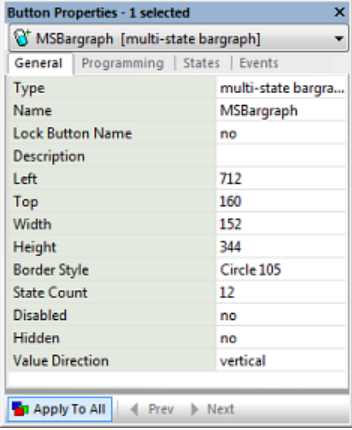### Creating Text Input Buttons

1. Open the Page, Popup Page or Sub-Page that the button will be added to (double-click on the page in the Workspace window (Pages tab) to open the page and bring it into edit focus).

2. Select the Button Draw tool from the Button Selection/Draw toolbar.

3. Click on a Page, Popup Page or Sub-Page (in the active Design View window), and while holding the mouse button down, drag to draw the button to the desired size and shape.

   Upon the creation of a button, you will see selection handles appear on the outsides of the button that are small yellow squares with red interiors. This is a visual indication that the newly created button has the Edit Focus.

4. In the *General* tab of the Properties window, set the *Type* property to **Text Input**.

   This selection updates the Properties window to represent properties specific to this button type.

5. Set the button properties as desired by editing the *General*, *Programming* and *States* properties in the Properties window.

6. Select **File > Save** to save your changes.

Text Input Buttons - General Properties

Once you have created a Text Input button, you can use the General tab of the Properties window to set/edit general (non-state oriented) button properties. To edit any of the properties in the table, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both. The following General button properties are supported for *Text Input* buttons:

Text Input Buttons - General Properties		
	• Type	see page 247
	• Name	see page 242
	• Lock Button Name	see page 242
	• Description	see page 236
	• Left	see page 238
	• Top	see page 246
	• Width	see page 247
	• Height	see page 237
	• Touch Style	see page 247
	• Border Style	see page 236
	• Disabled	see page 236
	• Hidden	see page 237
	• Input Type	see page 238
	• Password Character	see page 242
	• Display Type	see page 236
	• Max Text Length	see page 242
	• Input Mask	see page 238

Text Input Buttons - Programming Properties

Once you have created a Text Input button, you can use the Programming tab of the Properties window to set/edit programming-oriented button properties. To edit any of the properties, click in the right-hand table cell to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both. The following Programming properties are supported for *Text Input* buttons:

Text Input Buttons - Programming Properties		
	• Address Port	see page 247
	• Address Code	see page 249
	• Channel Port	see page 249
	• Channel Code	see page 249

Text Input Buttons - States Properties

Once you have created a Text Input button, you can use the *States* tab of the Properties window to set/edit state-oriented button properties. To edit any of the listed button properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.
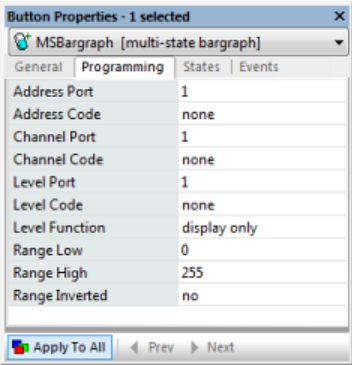
The State tab of the Properties window works in conjunction with the State Manager window. Note that if the State Manager is not displayed, or if no state(s) are selected in the State Manager, the States tab shows a list of all states associated with the selected button. Each state represented in the States tab is a collapsed folder containing the state properties for that particular state. Click the + symbol to expand each folder. If you select a state (or multiple states) in the State Manager, then the States tab only represents the selected state(s).

The following button state properties are supported for Text Input buttons (for each state):

Text Input Buttons - States Properties

	• Border Name	see page 252
	• Border Color	see page 251
	• Fill Type	see page 252
	• Fill Color	see page 252
	• Fill Gradient Colors	see page 252
	• Gradient Radius	see page 253
	• Gradient Center X%	see page 253
	• Gradient Center Y%	see page 253
	• Text Color	see page 254
	• Text Effect Color	see page 254
	• Overall Opacity	see page 253
	• Bitmaps	see page 251
	• Font	see page 252
	• Font Size	see page 253
	• Text	see page 254
	• Text Effect	see page 254

NOTE: *Text Input buttons do not support Events.*

Sub-Page View Buttons

The Sub-Page View button type serves as a "container" for a Sub-Page Set, and defines the area of the Scrolling Region on the panel page.
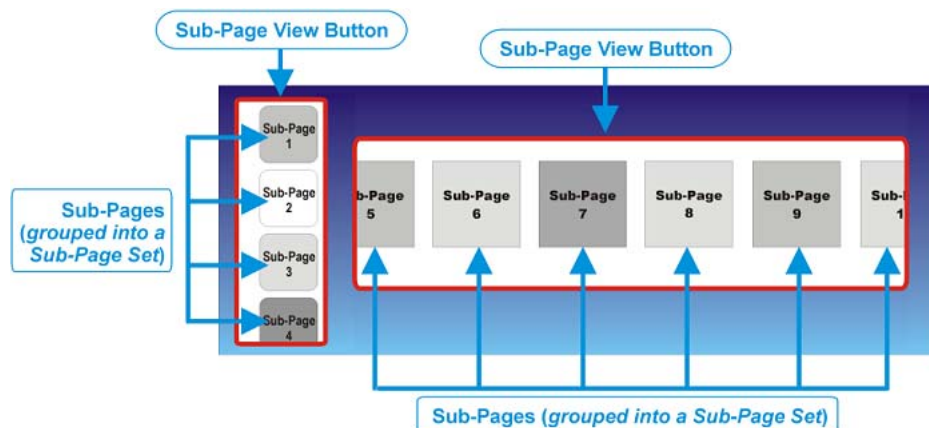
FIG. 83 Scrolling Region elements

See the *Scrolling Regions (Sub-Pages & Sub-Page View Buttons)* section on page 89 for details.

Creating Sub-Page View Buttons

1. Open the Page, Popup Page or Sub-Page that the button will be added to - double-click on the page in the Workspace Navigator (Pages tab) to open the page and bring it into edit focus.
2. Select the Button Draw tool from the Selection/Drawing Tools Toolbar.
3. With the Button Draw tool still selected, left-click inside the desired page (in the active Design View window), and while holding the left mouse button down, drag to draw the button to the desired size and shape.
4. In the *General* tab of the Properties window - Type property, set **Sub-Page View** as the button type.

This selection updates the Properties window, and the Design View window to represent the properties specific to this button type.
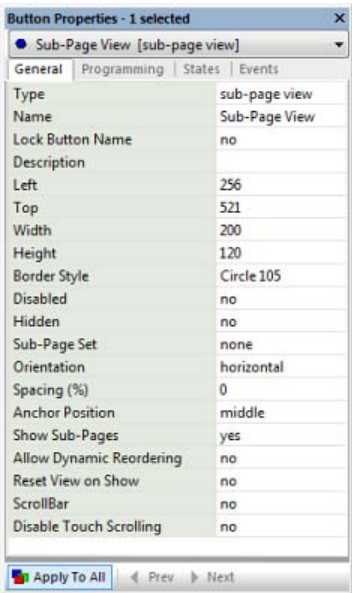
5. Set the button properties as desired by editing the *General*, *Programming* and *States* properties in the Properties window.

    NOTE: *The Address Port and Address Code assignments for Sub-Page View buttons are provided only for use in SEND-COMMANDS (not to trigger actions). Sub-Pages do not utilize Channel Port/Code addresses.*

6. Select **File > Save** to save your changes.

Sub-Page View Buttons - General Properties

Once you have created a Sub-Page View button, you can use the General tab of the Properties window to set/edit general (non-state oriented) button properties. To edit any of the properties in the table, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both. The following General button properties are supported for *Sub-Page View* buttons:

Sub-Page View Buttons - General Properties		
	• Type	see page 247
	• Name	see page 242
	• Lock Button Name	see page 242
	• Description	see page 236
	• Left	see page 238
	• Top	see page 246
	• Width	see page 247
	• Height	see page 237
	• Touch Style	see page 247
	• Border Style	see page 236
	• Disabled	see page 236
	• Hidden	see page 237
	• Sub-Page Set	see page 246
	• Orientation	see page 242
	• Spacing (%)	see page 246
	• Anchor Position	see page 235
	• Show Sub-Pages	see page 246
	• Reset View on Show	see page 243
	• ScrollBar	see page 243
	• ScrollBar Offset	see page 244
	• Disable Touch Scrolling	see page 236

Sub-Page View Buttons - Programming Properties

Once you have created a Sub-Page View button, you can use the Programming tab of the Properties window to set/edit programming-oriented button properties. To edit any of the listed properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The following Programming properties are supported for *Sub-Page View* buttons:

Sub-Page View Buttons - Programming Properties		
	• Address Port	see page 247
	• Address Code	see page 249

NOTE: *The Address Port and Code assignments for Sub-Page View buttons are provided only for use in SEND_COMMANDs (not to trigger actions).*

Sub-Page View Buttons - States Properties

Once you have created a Sub-Page View button, you can use the States tab of the Properties window to set/edit state-oriented button properties. To edit any of the listed button properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

Sub-Page View buttons use only one State - *Off*.

The following button state properties are supported for Sub-Page View buttons for each state. Note that depending on the Panel associated with your project some of these options may not be available. For example, if you have specified a non-video capable panel in your project, the Video Fill option(s) will not appear:

Sub-Page View Buttons - States Properties		
	• Border Name	see page 252
	• Border Color	see page 251
	• Fill Type	see page 252
	• Fill Color	see page 252
	• Fill Gradient Colors	see page 252
	• Gradient Radius	see page 253
	• Gradient Center X%	see page 253
	• Gradient Center Y%	see page 253
	• Text Color	see page 254
	• Text Effect Color	see page 254
	• Sub-Page Layout Color	see page 252
	• Overall Opacity	see page 253
	• Video Fill	see page 254
	• Streaming Source	see page 253
	• Bitmaps	see page 251
	• Font	see page 252
	• Font Size	see page 253
	• Text	see page 254
	• Text Justification	see page 254
	• Text X Offset	see page 254
	• Text Y Offset	see page 254
	• Text Effect	see page 254
	• Word Wrap	see page 254

NOTE: *Sub-Page View buttons do not support Events.*

Listview Buttons

Modero X Series G5 touch panels and TPDesign5 (v1.2.0, build 47 or greater) support a new button type called *Listview* buttons. Listview buttons provide the ability to display a listing of items from a dynamic data source on a G5 touch panel. Dynamic data can be created either using an XPort server, NetLinx code or a generic CSV file. The creator of the data can specify how many fields comprise a record and the format of those fields. As many records as necessary can be specified.

NOTE: *Dynamic data defines data files/feeds URL where the data can be loaded by the touch panel at runtime via HTTP (GET) or HTTPS (GET) transport protocols.*

This data can be used to populate a Listview button displayed on a G5 touch panel, where the end user can scroll or search through the list and make a selection. Once a selection has been made, a CUSTOM_EVENT is raised in the NetLinx Master to retrieve the data fields comprising the selected record.

NOTE: *Listview buttons will not work with NetLinx Masters that are in DoD Security Mode. Refer to the NX-Series Controllers, Enova DGX, Enova DVX, Massio WebConsole & Programming Guide for information on security mode settings on Central Controllers.*

Refer to the *Listview Buttons & Dynamic Data* **section on page 104** for working demos of creating Listview buttons using three types of data source files:

1) CSV with headers (page 114),

2) CSV without headers (page 128) and

3) XPort-generated XML (page 141).
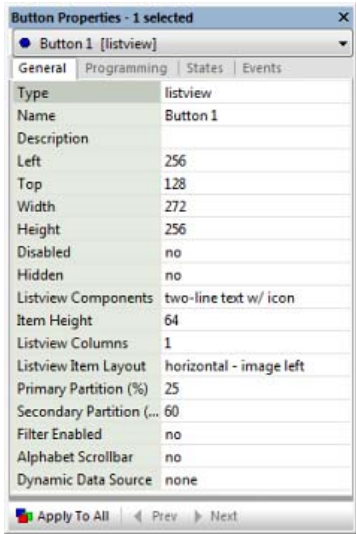
Creating Listview Buttons

To create a new Listview button:

1. Open the Page, Popup Page or Sub-Page that the button will be added to (double-click on the page in the Workspace window (Pages tab) to open the page and bring it into edit focus).

2. In the Design View, select the Page, Popup Page or Sub-Page to which you want to add the button.

3. Use the Button Draw tool to create a new button. See *Creating New Buttons* on page 66 for details.

4. In the *General* tab of the Properties window, set the *Type* property to **Listview**.
   This selection updates the Properties window to represent properties specific to this button type.

5. Set the button properties as desired by editing the *General*, *Programming*, *States* and *Events* properties in the Properties window.
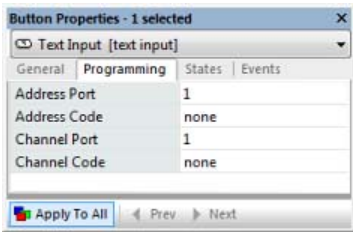
6. Select **File > Save** to save your changes.

Listview Buttons - General Properties
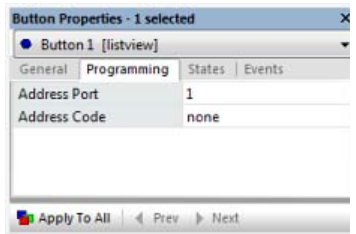
Once you have created a Listview button, you can use the *General* tab of the Properties window to set/edit general (non-state oriented) button properties. To edit any of the properties in the table, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both. The following General button properties are supported for *Listview* buttons:

Listview Buttons - General Properties		
	• Type	see page 247
	• Name	see page 242
	• Description	see page 236
	• Left	see page 238
	• Top	see page 246
	• Width	see page 247
	• Height	see page 237
	• Disabled	see page 236
	• Hidden	see page 237
	• Listview Components	see page 239
	• Item Height	see page 238
	• Listview Columns	see page 239
	• Listview Item Layout	see page 241
	• Primary Partition (%)	see page 242
	• Secondary Partition (%)	see page 245
	• Filter Enabled	see page 236
	• Filter Height	see page 237
	• Alphabet Scrollbar	see page 235
	• Dynamic Data Source	see page 236

Listview Buttons - Programming Properties

Once you have created a Listview button, you can use the *Programming* tab of the Properties window to set/edit programming-oriented button properties. To edit any of the properties, click in the right-hand table cell to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

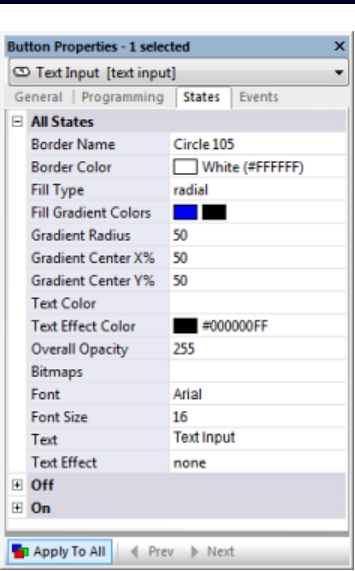The following Programming properties are supported for *Listview* buttons:

Listview Buttons - Programming Properties		
	• Address Port	see page 247
	• Address Code	see page 249

NOTE: *Maximum command, string and text length = 4096 characters.*

Listview Buttons - States Properties

Once you have created a Listview button, you can use the *States* tab of the Properties window to set/edit state-oriented button properties. To edit any of the listed button properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

The State tab of the Properties window works in conjunction with the State Manager window. Note that if the State Manager is not displayed, or if no state(s) are selected in the State Manager, the States tab shows a list of all states associated with the selected button.

Each state represented in the States tab is a collapsed folder containing the state properties for that particular state. Click the + symbol to expand each folder. If you select a state (or multiple states) in the State Manager, then the States tab only represents the selected state(s).

Listview buttons use three states:

- **All States** - Use the *All States* option to apply any changes you make to all states on the selected button. Note that if you have multiple buttons selected (Shift+click to select multiple buttons a page), the All States option only affects states for the button that has Edit Focus. The button with edit focus would be the last one selected, and is indicated by having red-colored square handles (as opposed to the black squares that indicate that a button is selected, but does not currently have edit focus).
- **Default** - This represents the non-selected state of the list items in the Listview button.
- **Selected** - This represents the selected state of the list items in the Listview button.

The following State properties are supported for Listview buttons (for each state).
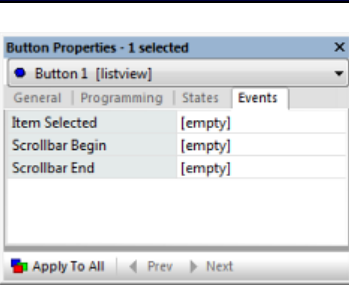
Listview Buttons - States Properties		
	• Border Color	see page 251
	• Fill Type	see page 252
	• Fill Color	see page 252
	• Fill Gradient Colors	see page 252
	• Gradient Radius	see page 253
	• Gradient Center X%	see page 253
	• Gradient Center Y%	see page 253
	• Text Color	see page 254
	• Text Effect Color	see page 254
	• Overall Opacity	see page 253
	• Font	see page 252
	• Font Size	see page 253
	• Text Effect	see page 254
	• Secondary Font	see page 253
	• Secondary Font Size	see page 253

Listview Buttons - Events Properties

Modero X Series panels support Gestures for on-screen navigation. Gestures can be used for navigating the panel UI. For example, a "Swipe" gesture can invoke a page flip when the user swipes a finger across the screen. Gestures are presented in TPDesign5 under the *Events* tab of the Properties window.

To choose a particular property for a button, highlight the row and click the Browse button (...) to open the *Edit Event Actions* dialog. Use this dialog to add actions instigated by the gesture selected. Click the **Add Action** button to select between a Send Command or a command string, and enter the command or string in the field. When finished, click **OK**.

The following Events are supported for *Listview* buttons:

Listview Buttons - Events Properties		
	• Item Selected	see page 258
	• Scrollbar Begin	see page 258
	• Scrollbar End	see page 258

Scrolling Regions (Sub-Pages & Sub-Page View Buttons)

Scrolling Regions - Overview

Scrolling Regions represent a powerful method of presenting functions on AMX Modero X Series Touch Panels. A "Scrolling Region" is a specific area on a touch panel page that contains a set of elements that scroll as a group.

The illustration below shows a basic example of a touch panel page with two Scrolling Regions - one vertical and one horizontal. The red borders represent the boundaries of the Scrolling Regions, and the items labeled "SubPage <x>" scroll within the boundaries of the Scrolling Region (either vertically or horizontally). End-users use swipes to scroll through the items in the Scrolling Region.

- All of the items within the Scrolling Region move as a group.
- Each item within the Scrolling Region can be pressed like traditional buttons (FIG. 84):
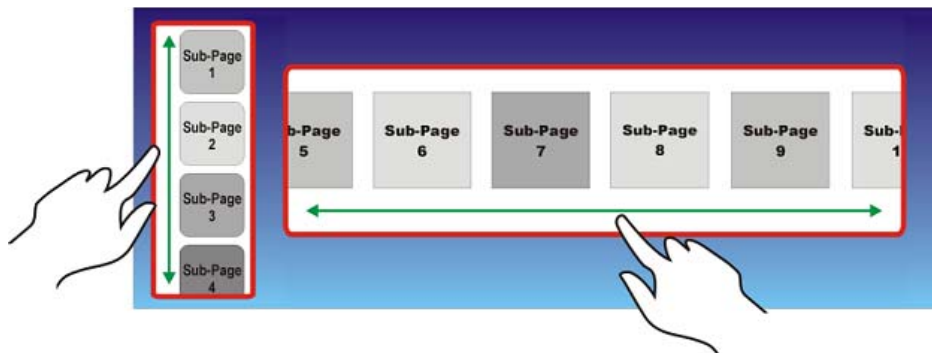
FIG. 84 Scrolling Region

Creating Scrolling Regions in TPDesign5 requires three elements):

- Sub-Pages - these are the items that are displayed within the Scrolling Region (see page 91)
- Sub-Page Sets - a grouped set of Sub-Pages (see page 94)
- Sub-Page View Buttons - a container button that defines the size and shape of the Scrolling Region (see page 97)

Scrolling Regions present a set of Sub-Pages called a Sub-Page Set that scroll as a group within a container button called a Sub-Page View button (FIG. 85):
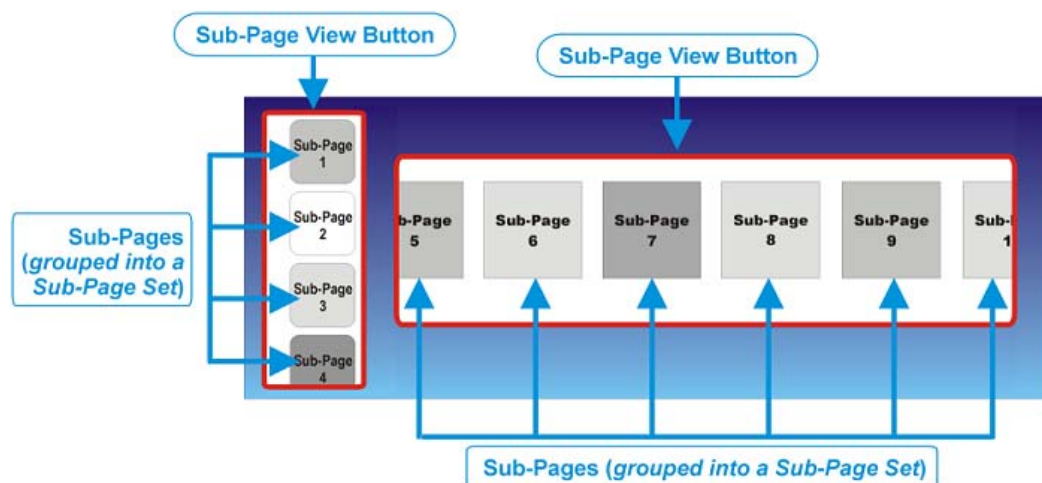
FIG. 85 Scrolling Region elements

Since touch panels that support Scrolling Regions also support Gestures at the page level, it is important to understand that finger gestures used on the page are separate from the swipes used within Scrolling Regions.

For example, if a page uses a swipe gesture to trigger a Page Flip, the page flip will only occur if the swipe gesture is used on the Page, and not within a Scrolling Region (FIG. 86):
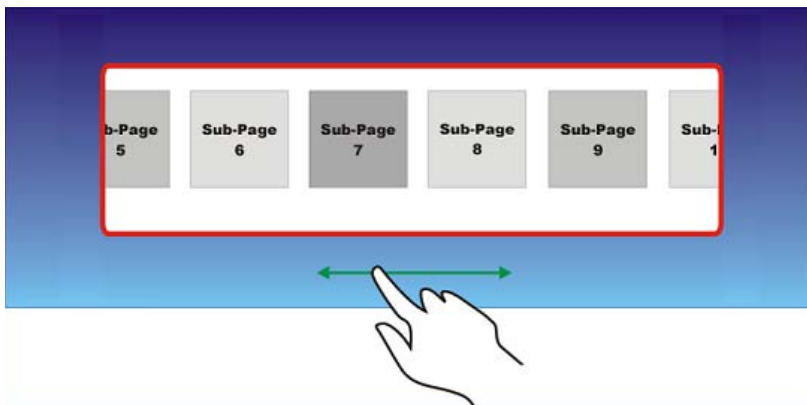
FIG. 86 Gesture example - Page Flip

When the User swipes (or selects) within a Scrolling Region, it only affects the Scrolling Region (FIG. 87):
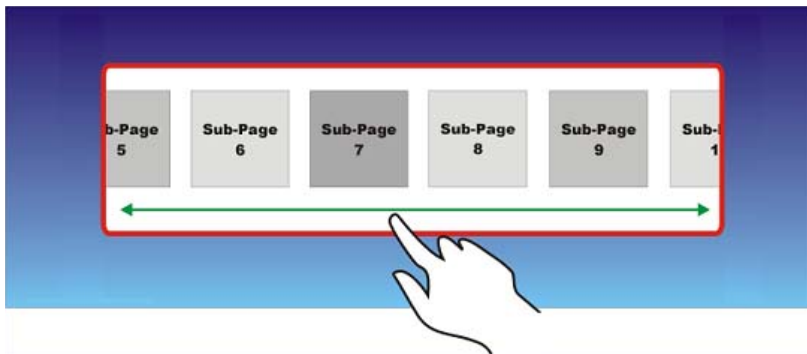
FIG. 87 Gesture example - Scrolling Region

See *Gestures* on page 273 for details on using gestures outside of Scrolling Regions.

Creating a Scrolling Region - Overview

The process of creating a Scrolling Region involves three basic steps (each described in the following sub-sections). Technically, these steps can be performed in any order. However there are fundamental dependencies between the three basic elements that comprise a Scrolling Region (Sub-Page View Button, Sub-Page Set and Sub-Pages) that must be understood before starting:

- A Sub-Page View Button can only be used as a container for a Sub-Page Set. Therefore it makes sense to define a Sub-Page Set before creating the Sub-Page View button, so that there is a Sub-Page Set available to associate to the Sub-Page View Button once it is created.
- In order to create a Sub-Page Set, two or more Sub-Pages must be available in the project to add to the set. Therefore, it is necessary to define the Sub-Pages that will be grouped into a set before a Sub-Page Set can be defined.

If a series of Sub-Pages are created as the first step, they can be grouped into a Sub-Page Set as the second step, and applied to a Sub-Page View button as the final step. The instructions for creating a Scrolling Region are presented in the following order to follow these basic dependencies:

1. Add Sub-Pages to the Project (see page 91)
2. Create a Sub-Page Set (see page 94)
3. Create a Sub-Page View Button (see page 97)

NOTE: *Depending on the context of your work, you may find it advantageous to create a Sub-View Button first to define the visual boundaries of the Scrolling Region, then define the Sub-Pages that will be displayed as a Sub-Page Set within the Sub-Page View button. Again, these steps can be performed in any order.*

NOTE: *The scroll bars on scrolling regions cannot be set a specific position. They can only be on the left or right of a vertical scroll region, or only at the bottom of a horizontal scroll region. With a vertical subpage, if you set an offset greater than half the width of the viewer button, the scrollbar will move to the left.*

Sub-Pages

A *Sub-Page* is a specific type of Popup Page that is used to represent an individual item within a Scrolling Region. Each item displayed with a Scrolling Region is a separate Sub-Page (FIG. 88):
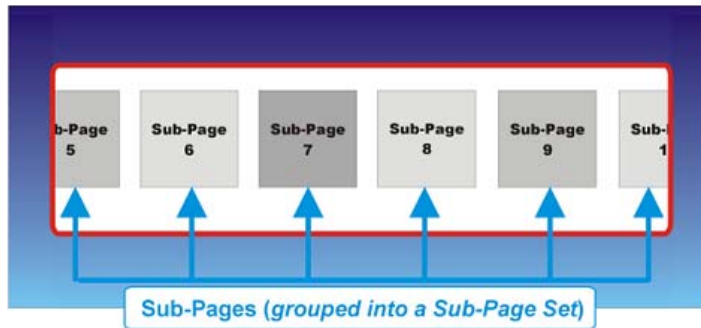
FIG. 88 Scrolling Region with multiple Sub-Pages

NOTE: *Sub-Pages must be grouped into a Sub-Page Set in order to be displayed in a Scrolling Region.*

Sub-Pages are represented in the Workspace window (Pages tab) within the *Sub-Pages* folder (FIG. 89):
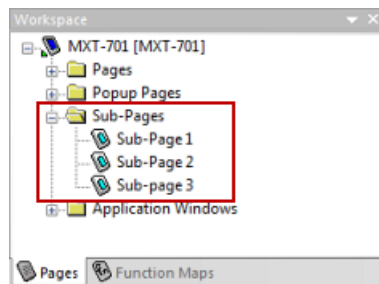
FIG. 89 Workspace window (Pages tab) - Sub-Pages folder

Sub-Pages are created in the same way as Standard Popup Pages (either via the Popup Draw Tool, or via the Add Popup Page dialog) - see *Adding Sub-Pages to the Project* on page 91 for details.

- Sub-Pages are differentiated from Standard Popups via the **Popup Type** (General) property for Popup Pages.
- Sub-Pages can only be displayed within a Sub-Page View button.
- Unlike Popup Pages, Sub-Pages do not appear in Page Flip lists.
- Unlike Popup Pages, Sub-Pages do not have Top, Left, or Group properties, since their position within the Sub-Page View button is determined by button properties set for the Sub-Page View button. See *Setting Sub-Page Properties* on page 93 for details.
- The *Address Port* and *Address Code* assignments for Sub-Pages are provided only for use in SEND COMMANDS (not to trigger actions). Unlike Standard Popup Pages, Sub-Pages do not utilize Channel Port/Code addresses.
- *Sub-Page Sets* are used to define an ordered group of Sub-Pages. Sub-Pages must be added to a Sub-Page Set to be displayed in a Sub-Page View button. Sub-Pages can be shared among multiple Sub-Page Sets.
  See *Sub-Page Sets* on page 94 for details

NOTE: *As is the case with Popup Pages, only one instance of any Sub-Page can be displayed on a touch panel page at a time.*

NOTE: *TPD5 will convert any Sub-Pages to Standard Popups on a Save As Different Panel Type operation from a panel that supports the sub-page view feature to a legacy panel-type that doesn't.*

Adding Sub-Pages to the Project

There are two ways to add a new Sub-Page to your project:
- Adding a Sub-Page via the Add Popup Page dialog
- Adding a Sub-Page via the Popup Draw tool

Adding a Sub-Page via the Add Popup Page dialog

Sub-Pages are a specific type of a popup page that is only used with the Sub-Page View button type. Sub-Pages and Sub-Page View buttons are required for Scrolling Regions. See *Scrolling Regions - Overview* on page 89 for details.

1. Select **Panel > Add Popup Page** (or select **Add Popup Page** from the Workspace Context Menu, or click the toolbar icon) to open the *Add Popup Page* dialog.
2. For the *Type* property, select **Sub-Page** (FIG. 90):
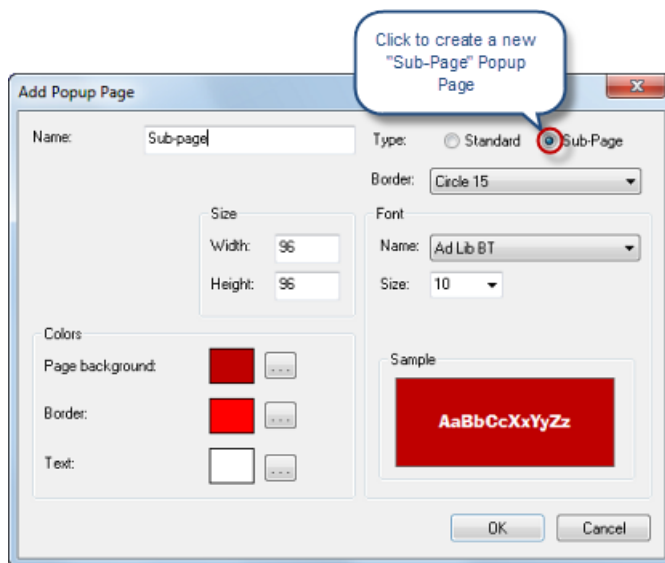
FIG. 90 Add Popup Page dialog

3. Fill in the information in this dialog to specify the basic properties for the new Sub-Page.

4. Click **OK** to add the new Sub-Page to the active project. The new Sub-Page will be appear in the Workspace window (Sub-Pages folder), under the project to which the page was added (as the active page).

5. Set the Sub-Page Properties as desired.

Adding a Sub-Page Popup via the Popup Draw tool

1. Select **Edit > Popup Draw Tool** (or click the toolbar button) to activate the Popup Draw tool.

    To access the Popup Draw toolbar button, click and hold the Button Draw tool (in the Selection/Drawing Tools toolbar) for one second to open the drop-down menu containing the Popup Draw tool:

2. In the Properties window (*General* tab), set the *PopupType* property to **SubPage** (FIG. 91):
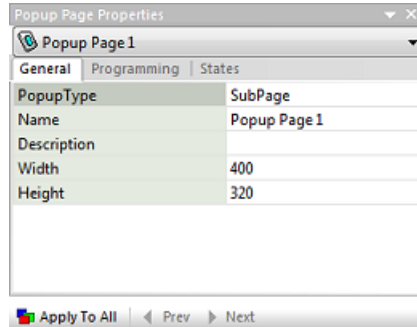
FIG. 91 Popup Type (General Property) - SubPage

3. Left-click inside the desired page (in the active Design View window), and while holding the left mouse button down, drag to draw the popup page to the desired size and shape.

    ● The minimum popup page size is 15 pixels in height or width.

    ● When using the Button Draw or Popup Draw tools, hold down the SHIFT key while drawing to constrain the item to a square.

    ● When using the Selection tool, hold down the ALT key while clicking and dragging in a Design View window to move the current selections without selecting anything new on the mouse press (useful for moving popup pages whose entire area is filled with buttons).

    ● When using the Selection tool, hold down the CTRL key while clicking and dragging to force a "lasso" selection to occur (even if the mouse was clicked over a button or Popup Page shown on the Page). Lasso selection forces you to draw a square around the outside of the item to select it, as opposed to clicking on the item.

    ● With an item selected in a Design View window, hold down the CTRL key while moving the item with the keyboard arrow keys to move by the grid size instead of a single pixel (regardless of the grid visibility or snap to grid setting). Holding the CTRL key while resizing the selected items with the keyboard will resize by the grid size.

4. Set the other properties as desired - see *Setting Sub-Page Properties* on page 93 for details.

5. Select **File > Save** to save your changes.

Naming Sub-Pages

If the Sub-Page name starts with an underscore (ex: " _sources"), it will be always be displayed on top.

- If you rename the Sub-Page without the underscore, it will act normally.
- If you call two Sub-Pages with the underscore, the last one called will be on top.

System Template Pages always use a double underscore prefix.

- A copied template page will have a single underscore prefix.
- Note that the single underscore applies the effect of "always on top".

Cut, Copy and Paste - Sub-Pages

1. Cut or Copy a Sub-Page to clipboard memory:
   - To cut a Sub-Page to the clipboard, select a Page in the Workspace window (Pages tab) and select **Cut**. The program will prompt you to verify this action before the Sub-Page is removed from the project.
   - To copy a Sub-Page to the clipboard, select a Page in the Workspace window (Pages tab) and select **Copy**.

2. Select the target project for the Sub-Page in the Workspace window (Pages tab). Sub-Pages can be pasted into the current project, or into any other project that is open in the Workspace window.

   Note that all Buttons present on the original Sub-Page are also copied, and when pasted they retain the attributes of the original buttons, according to the selections made in the *Paste Controls* dialog.

3. Select **Paste** to paste a copy of the Sub-Page into the selected project. If a Sub-Page with the same name already exists in the target project, the Sub-Page's name will be modified to indicate that it is a copy of another Sub-Page. This prevents existing Sub-Pages from being overwritten by a Paste operation.

Setting Sub-Page Properties

Sub-Pages have General, Programming and States Properties that can be configured via the fields in the Properties window.

To set Page-level properties for Sub-Pages, click on the Sub-Page Popup area in the Design View, or select a Sub-Page in the Workspace window (Pages tab). With the Sub-Page selected, the Properties window displays the properties available for the Sub-Page, separated into three tabs (General, Programming and States). Sub-Pages do not support Events.

NOTE: *To edit any of the listed properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.*

Sub-Pages - General Properties

Use the *General* tab of the Properties window to set/edit general properties for the selected Sub-Page. The following General properties are supported at the Sub-Page level:

Sub-Pages - General Properties		
	• Popup Type	see page 242
	• Name	see page 242
	• Description	see page 236
	• Width	see page 247
	• Height	see page 237

Sub-Pages - Programming Properties

Use the *Programming* tab of the Properties window to view/edit program-related properties (Address assignments) for the active Sub-Page. The following Programming properties are supported at the Sub-Page level:

Sub-Pages - Programming Properties		
	• Address Port	see page 247
	• Address Code	see page 249

NOTE: *The Address Port and Code assignments for Sub-Pages are provided only for use in SEND-COMMANDS (not to trigger actions). Unlike Popup Pages, Sub-Pages do not utilize Channel Port/Code addresses.*

Sub-Pages- States Properties

Use the States tab of the Properties window to set/edit state-related properties for the selected Sub-Page.

Note that Sub-Pages have only one State (Off). The following State properties are supported at the Sub-Page level:

Sub-Pages - States Properties		
	• Border Name	see page 252
	• Border Color	see page 251
	• Fill Type	see page 252
	• Fill Color	see page 252
	• Fill Gradient Colors	see page 252
	• Gradient Radius	see page 253
	• Gradient Center X%	see page 253
	• Gradient Center Y%	see page 253
	• Text Color	see page 254
	• Text Effect Color	see page 254
	• Overall Opacity	see page 253
	• Video Fill	see page 254
	• Streaming Source	see page 253
	• Bitmaps	see page 251
	• Font	see page 252
	• Font Size	see page 253
	• Text	see page 254
	• Text Justification	see page 254
	• Text X Offset	see page 254
	• Text Y Offset	see page 254
	• Text Effect	see page 254
	• Word Wrap	see page 254

Sub-Page Sets

A *Sub-Page Set* represents an ordered list of Sub-Pages that are to be displayed within a Sub-Page View button (FIG. 92):
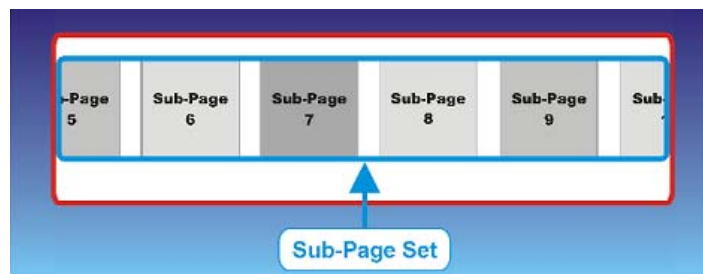
FIG. 92 SubPage Set

Sub-Pages must be assigned to a Sub-Page Set to be used in a scrolling region. Sub-Page Sets are created in the *Create Sub-Page Set* dialog and edited via the *Edit Sub-Page Sets* dialog.

- A panel project must contain at least one Sub-Page Set to implement the scrolling region feature.
- A Sub-Page Set defines the name of the set, the Sub-Pages to be included in the set, and the order in which they will be displayed.
- Note that the size of all Sub-Pages in a Sub-Page Set is determined by the first Sub-Page in the set. See the Sub-Page Sets - Slot Sizes topic for details.

Creating Sub-Page Sets

The options in the *Create Sub-Page Set* dialog allow you to create new Sub-Page Sets.

1.   Select **Panel > Edit Sub-Page Sets** to open the *Edit Sub-Page Sets* dialog (FIG. 93):
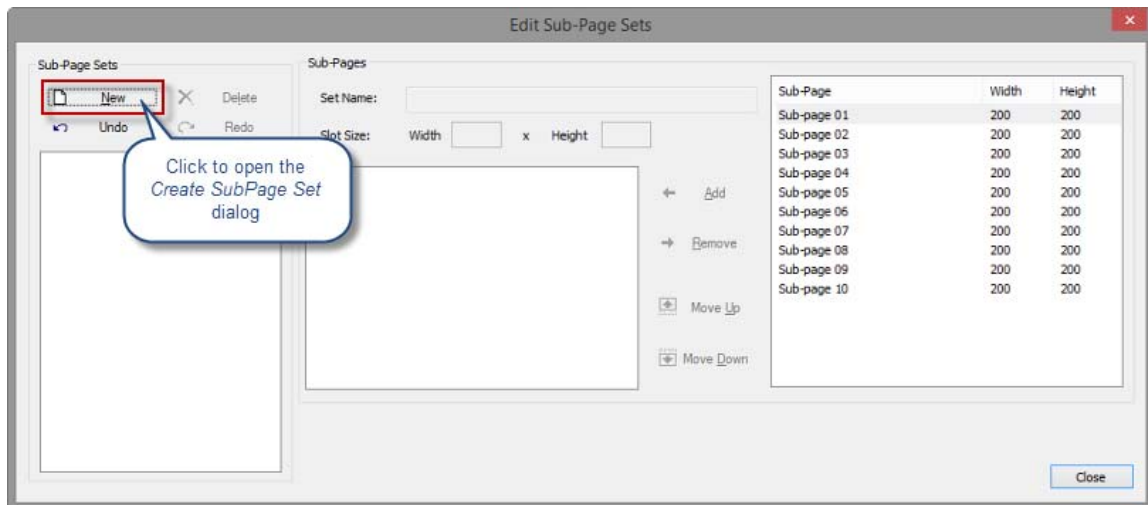
FIG. 93 Edit Sub Page Sets dialog - New

2.   Click **New** to open the *Create Sub-Page Set* dialog.

3.   Enter a unique name for the new Sub-Page Set in the *Set Name* field (FIG. 94):
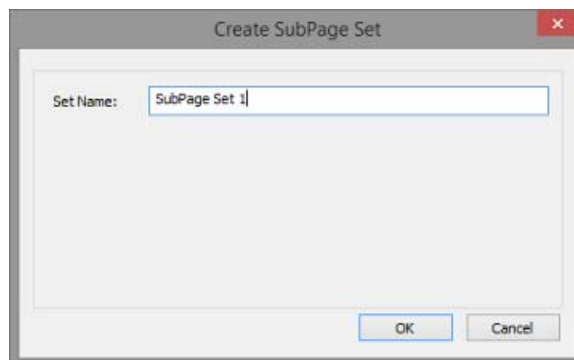
FIG. 94 Create SubPage Sets dialog

4.   Click **OK** to save your changes, close the *Create SubPage Set* dialog, and return to the *Edit Sub-Page Sets* dialog. The new Sub-Page Set is indicated in the *Sub-Page Sets* list on the left-side of the dialog (FIG. 95):
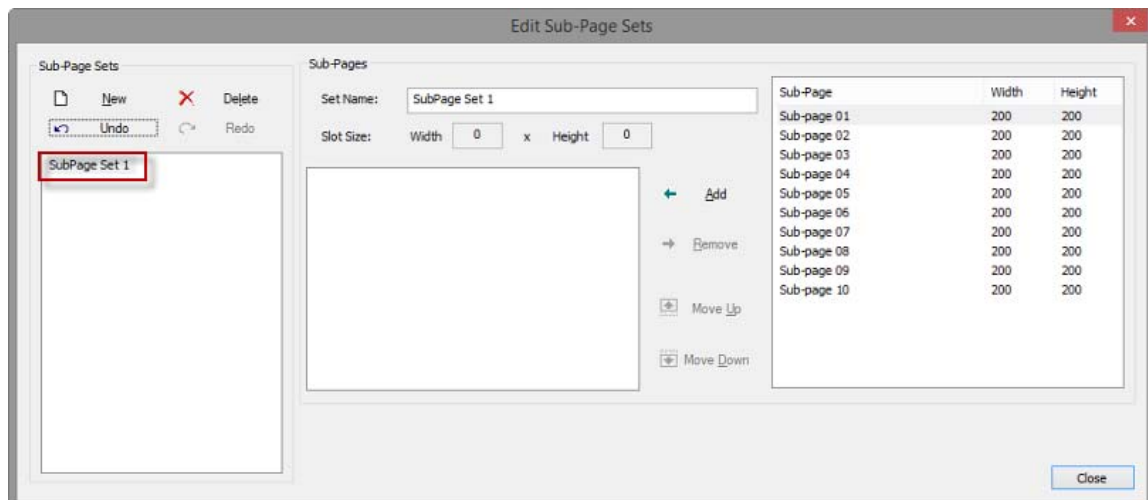
FIG. 95 Edit Sub Page Sets dialog - Sub-Page Sets list

5.   Add Sub-Pages to the new Sub-Page Set:
     a.   Select a Sub-Page from the Sub-Pages list on the right-side of the dialog (FIG. 96):
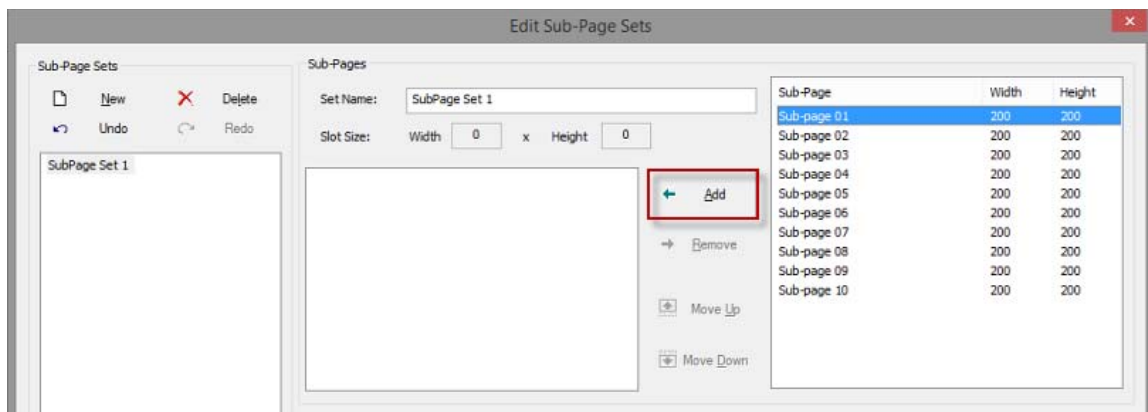
FIG. 96 Edit Sub Page Sets dialog - Select a Sub-Page

  b.   Click **Add** (or double-click the Sub-Page) to add it to the selected Sub-Page Set list, in the center window (FIG. 97):
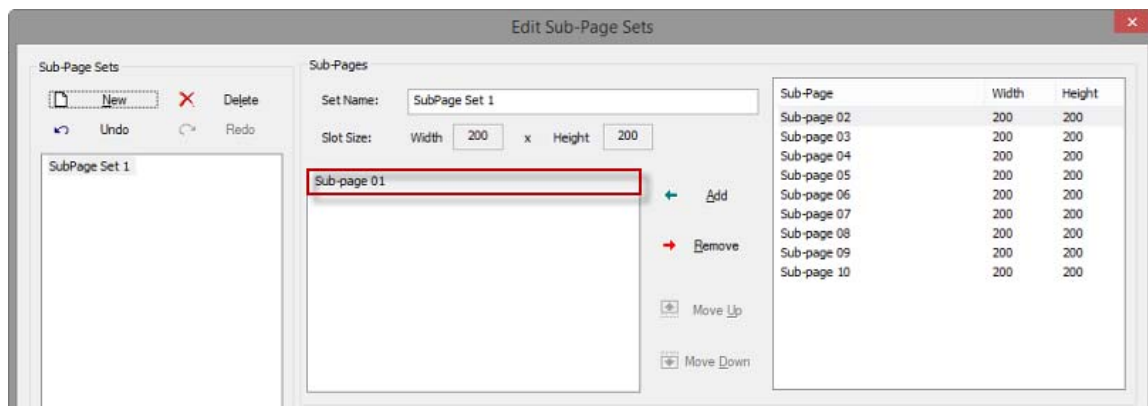
FIG. 97 Edit Sub Page Sets dialog - Adding a Sub-Page to a Sub-Pages list

  c.   Repeat this process to add other Sub-Pages to the selected Sub-Page Set (FIG. 98):
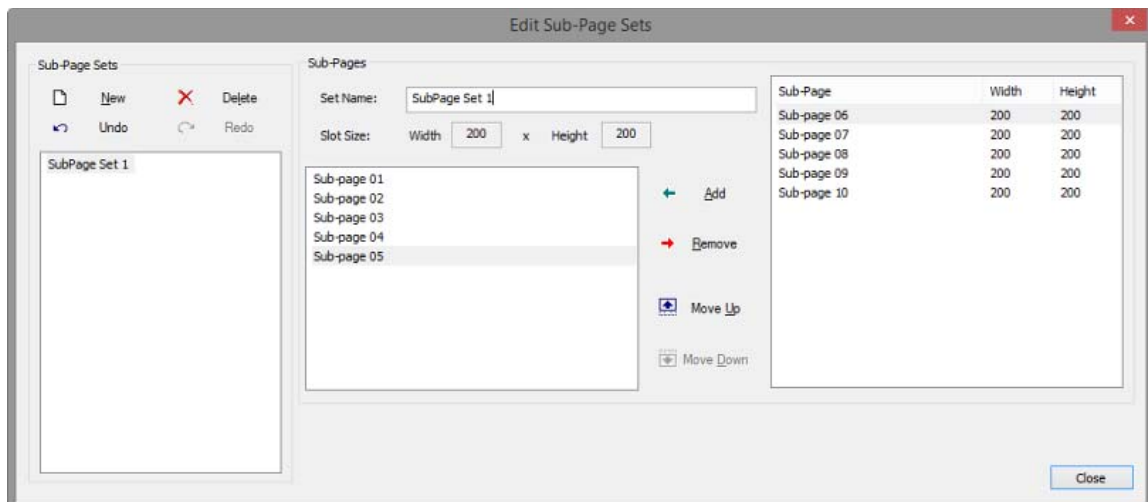
FIG. 98 Edit Sub Page Sets dialog - Adding more Sub-Pages to a Sub-Pages list

> **NOTE:** *Sub-Pages can be added to multiple Sub-Page Sets. However, only one instance of a Sub-Page can be added to any Sub-Page Set. Also, only one instance of any Sub-Page can only be displayed on a touch panel page at a time (as is the case with Standard Popup Pages).*

6.   Specify the order that the Sub-Pages will display via the Move Up and Move Down buttons.

> **NOTE:** *The dimensions of all of the Sub-Pages in the Set are determined by the dimensions of the first Sub-Page in the list.*

7.   Click **Close** to save your changes and close the Edit Sub-Page Sets dialog.

At this point, the new Sub-Page Set has been added to the active project. Once a Sub-Page Set has been saved, it is available for selection via the Sub-Page Set (General) button property for Sub-Page View buttons (FIG. 99):
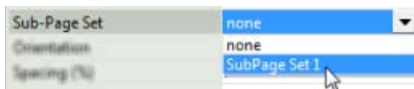
FIG. 99 Sub-Page View Button (General) Property - Sub-Page Set

See *Sub-Pages - General Properties* on page 93 for details.

An entry for the new Sub-Page Set will be added to the Sub-Page Sets list (on the left side of the Edit Sub-Page Sets dialog). This entry is pre-selected, and the field is populated with the appropriate value.

Editing Sub-Page Sets

The options in the *Edit Sub-Page Sets* dialog allow you to edit existing Sub-Page Sets.

1. Select **Panel > Edit Sub-Page Sets** to open the *Edit Sub-Page Sets* dialog.

2. Select a Sub-Page Set in the *Sub-Page Sets* list box (left list box).

3. Edit the *Set Name* value as desired.

4. Use the **Add** and **Remove** buttons to add or remove Sub-Pages from the list.

5. Use the **Move Up** and **Move Down** buttons to reorder the Sub-Pages in the list.

   NOTE: *The dimensions of all of the Sub-Pages in the Set are determined by the dimensions of the first Sub-Page in the list.*

6. Click **Close** to save your changes and close the *Edit Sub-Page Sets* dialog.

Deleting Sub-Page Sets

The options in the *Edit Sub-Page Sets* dialog allow you to delete existing Sub-Page Sets.

1. Select **Panel > Edit Sub-Page Sets** to open the *Edit Sub-Page Sets* dialog.

2. Select a Sub-Page Set in the *Sub-Page Sets* list box (left list box).

3. Click **Delete**.

4. Click **Close** to save your changes and close the *Edit Sub-Page Sets* dialog.

Sub-Page View Buttons

The *Sub-Page View* button type serves as a "container" for a Sub-Page Set, and defines the area of the Scrolling Region on the panel page (FIG. 100):
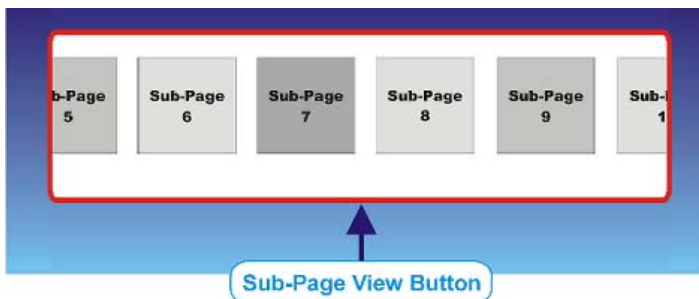
FIG. 100 Sub-Page View Button

The main functions of a Sub-Page View button are to:

1. Establish the display order of the Sub-Pages contained in the Sub-Page Set associated with the Sub-Page View button.

2. Process the scrolling motion of the Sub-Pages displayed.

3. Clip the Sub-Pages as they move out of the boundaries of the Sub-Page View button. Note that Sub-Page 4 in the figure above is shown to be clipped as it leaves the right-side boundary of the Sub-Page View button.

4. Sub-Page View buttons are created like any other button, but utilize button properties that are specific to configuring a scrolling region. Properties are set for Sub-Page View buttons via the Properties window, just like other button types.
   - The orientation of the scrolling region is set via the *Orientation* (General) button property.
   - The spacing between each of the Sub-Pages is defined via the *Spacing (%)* button property.
   - Other properties specific to Sub-Page View buttons include *Anchor Position, Wrap Sub-Pages*, *Allow Dynamic Reordering* and *Reset View on Show*.

Sub-Page View Buttons - Design View

Sub-Page View buttons are displayed in the Design View with placeholders that indicate the relative positions of the Sub-Page Set that it contains, to represent the size, placement, spacing and weighting options currently assigned to the Sub-Page Set. The figure below is an example of a Sub-Page View button in the Design View (FIG. 101):
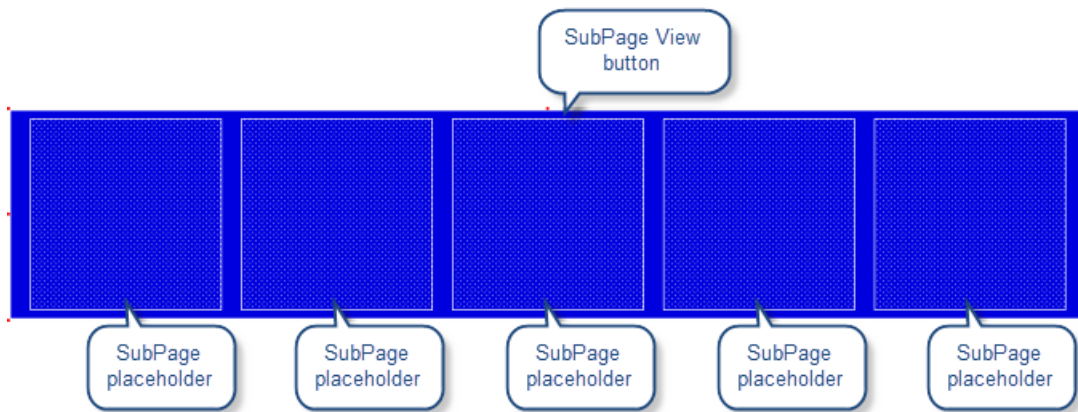
FIG. 101 Sub-Page View button in the Design View

This example shows a Sub-Page View button that has been assigned a Sub-Page Set containing five Sub-Pages. The Sub-Pages have been centered (via the *Anchor Position* property), and have a small amount of spacing applied (via the Spacing property).

These placeholders provide a preview of how the Sub-Pages will appear on the touch panel page. Use them to visualize how changes made to various Sub-Page View button properties will affect the arrangement of Sub-Pages within a Sub-Page View button (FIG. 102):

Anchor Position = Left
Spacing = 0

Anchor Position = Middle
Spacing = 25

Anchor Position = Right
Spacing = 50
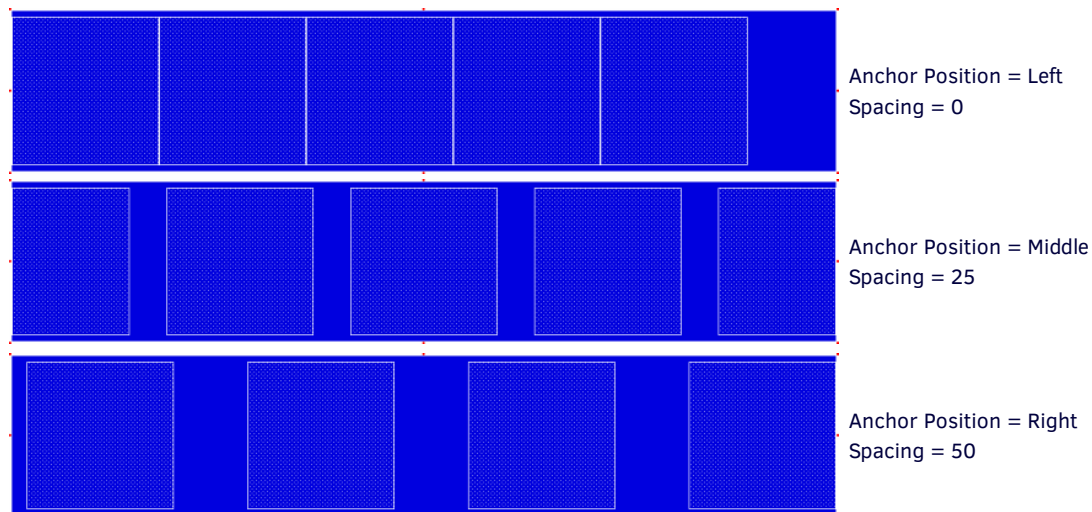
FIG. 102 Sub-Page View button in the Design View - Anchor Positions and Spacing

Sub-Page View Buttons - ScrollBar

Modero-X panel firmware supports a *ScrollBar* for Sub-Page View buttons. The ScrollBar is a position indicator within the Sub-Page Set, and does not provide dragging or scrolling functionality (FIG. 103):
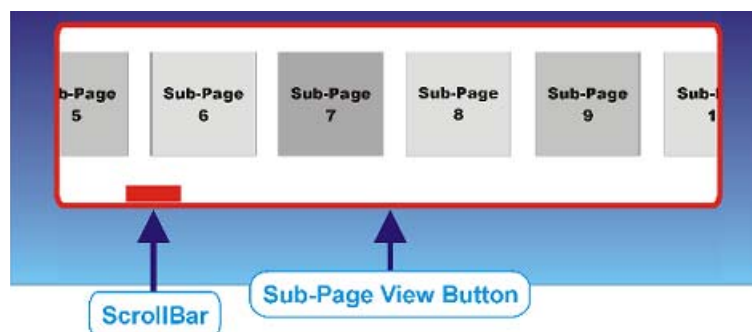
FIG. 103 Sub-Page View button - Scrollbar

Adding a ScrollBar to a Sub-Page View Button

1. Select a Sub-Page View button in the Design View.
2. In the *General* tab of the Properties window, select the *Scrollbar* property.
3. Select **Yes** from the drop-down menu.
4. Select the *Scrollbar Offset* property to specify an offset value for the ScrollBar.

Assigning a Sub-Page Set to the Sub-Page View Button

Once you created a series of Sub-Pages and added them to a Sub-Page Set, the final step is to assign the Sub-Page Set to the Sub-Page View button. This is accomplished via the Sub-Page Set (General) button property for the Sub-Page View button.

1. Select a Sub-Page View button.

2. In the *General* tab of the Properties window, select a Sub-Page Set from the Sub-Page Set drop-down menu. This menu lists all Sub-Page Sets that have been defined for this project (FIG. 104):
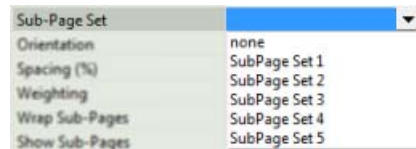
FIG. 104 Sub-Page View button (General) Property - Sub-Page Set

Creating a Scrolling Region - Example

Scrolling Regions represent a way to organize and navigate the functions on the panel by providing a set of Sub-Pages that scroll as a group, within a container button called a Sub-Page View button. The following elements are required to define a Scrolling Region:

- Sub-Pages - see page 91
- Sub-Page Sets - see page 94
- Sub-Page View Button - see page 97

Creating a Scrolling Region involves three basic procedures, as illustrated in the following workflow example:

Step 1 - Create Sub-Pages

The first step in creating a scrolling region is to create the Sub-Pages that will display within the Scrolling Region:

1. Select **Panel > Add Popup Page** to open the *Add Popup Page* dialog.

2. Select **Sub-Page** as the *Type*.

3. Enter a *Name* for the new Sub-Page and set the *Size*, *Colors*, *Border* and *Font* settings, just as you would for a Standard Popup Page. For this example:
   - *Name* = **Sub-Page 01**
   - *Size* = **100x100**
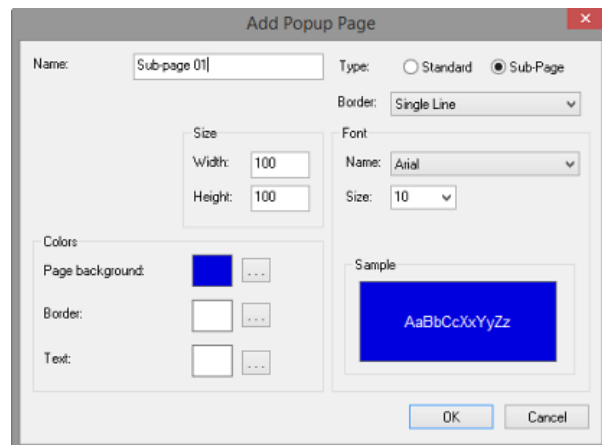
FIG. 105 Add Popup Page dialog - "Sub-Page 01"

4. Click **OK** to save your changes and close the *Add Popup Page* dialog.

5. The newly created Sub-Page is listed in the Workspace window - Pages tab, in the *Sub-Pages* folder. (FIG. 106):
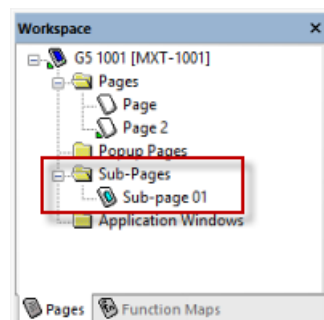
FIG. 106 Workspace window - "Sub-page 01" added to the project

6. Repeat steps 1 - 5 to create additional Sub-Pages, named "*Sub-Page 02*" - "*Sub-Page 10*" (FIG. 107):
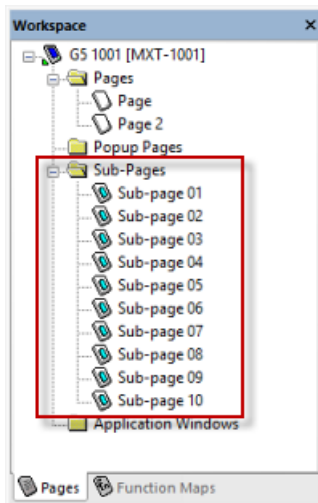
FIG. 107 Workspace window - Sub-pages 01 - 10 added to the project

NOTE: *As with Standard Popup Pages, you can always copy a Sub-Page and paste it into the project. The copy will initially be named "Copy Of <Page Name>". Double-click on the pasted entry in the Workspace window to edit the name as desired. In this example, you could copy "Sub-Page 01", and rename the pasted copies to "Sub-Page 02", then "Sub-Page 03", etc.*

Once you have created a total of ten Sub-Pages, you are ready to proceed to the next step - adding the Sub-Pages to a Sub-Page Set (next step).

NOTE: *Keep in mind that are no hard limitations to the number of Sub-Pages that can be used in a scrolling region, ten is simply an easy number to use for our example.*

Step 2 - Create a Sub-Page Set

The second step in creating a scrolling region is to define a Sub-Page Set to group the Sub-Pages together so that they can be associated as a set to the horizontal scrolling region:

1. Select **Panel > Edit Sub-Page Sets** to open the *Edit Sub-Page Sets* dialog (FIG. 108):
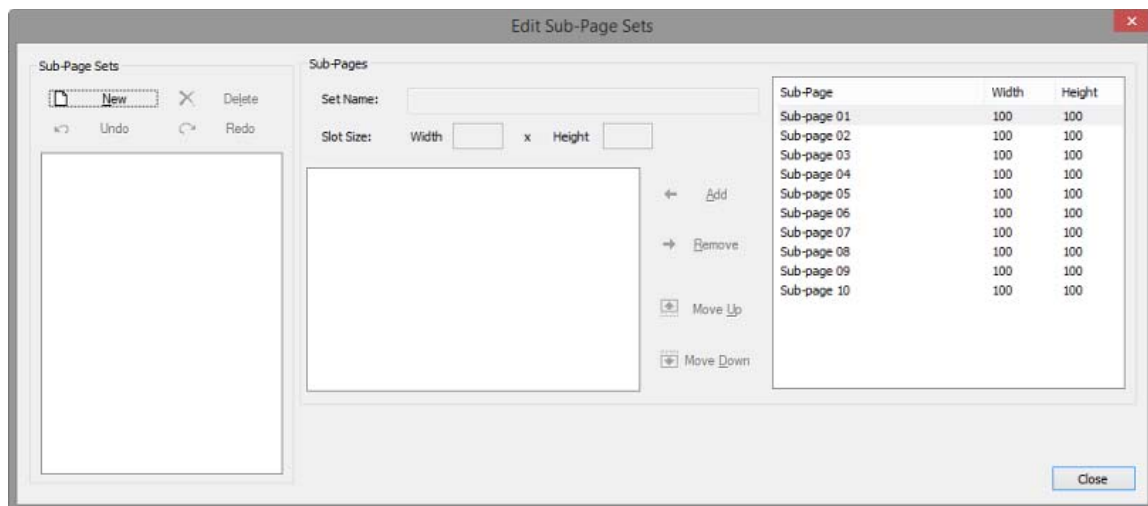
FIG. 108 Edit Sub-Page Sets dialog

Note the Sub-Pages that have been created (*Sub-page 01 - Sub-page 10*) are indicated in the Sub-Pages list on the left-side of the dialog.

2. Under *Sub-Page Sets*, click **New**. This invokes the *Create Sub-Page Set* dialog (FIG. 109):
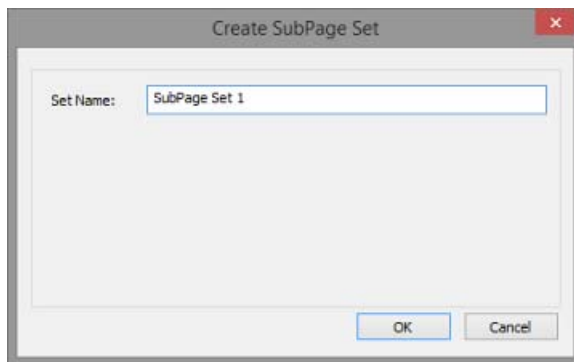
FIG. 109 Create SubPage Set dialog

3. Enter "**Sub-Page Set 1**" as the *Set Name*.

4. Click **OK** to save the new Sub-Page Set and close the *Create Sub-Page Set* dialog.

5. The new Sub-Page Set is indicated in the *Sub-Page Sets* list in the *Edit Sub-Page Sets* dialog (FIG. 110):
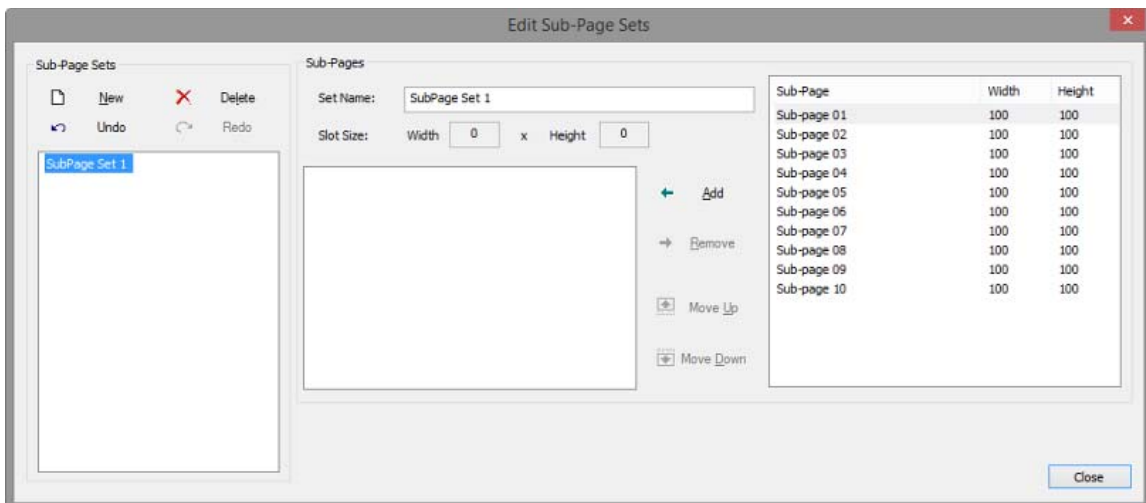
FIG. 110 Edit Sub-Page Sets dialog - "SubPage Set 1"

6. Select a Sub-Page in the Sub-Pages list (on the right side of the dialog), and click **Add** to add each Sub-Page to *Sub-Page Set 1*.
   - Alternatively, you can simply double-click on a Sub-Page to add it to the selected Sub-Page Set.
   - Repeat this action until all of the Sub-Pages have been added to Sub-Page Set 1 (FIG. 111):
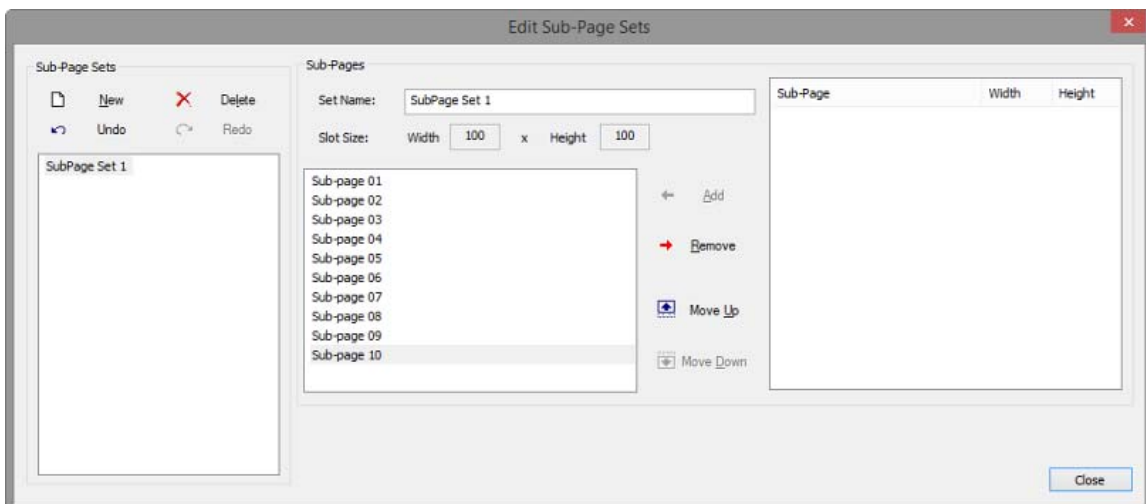
FIG. 111 Edit Sub-Page Sets dialog - "Sub-Pages 01 - 10" added to "SubPage Set 1"

The Sub-Pages will be displayed in the order that they are listed. Use the **Move Up** and **Move Down** buttons to set the order as necessary.

7. Click **Close** to save your changes and close the *Edit Sub-Page Sets* dialog.

Now that a Sub-Page Set has been defined to contain the new Sub-Pages, the Sub-Page Set (in this example, "Sub-Page Set 1") is available to be associated with a Sub-Page View button (next step).

Step 3 - Create a Sub-Page View Button

The third step of the basic workflow required to create a scrolling region is to create a Sub-Page View button which will serve as the container (or display area) for the specified Sub-Page Set.

For this example, we will create a Sub-Page View button to serve as a horizontal scrolling region. Sub-Page View buttons are created like any other button, but use button properties that are specific to configuring a scrolling region:

1. Select the Button Draw tool from the Button Selection/Draw toolbar and draw a button in the shape of a horizontal rectangle (FIG. 112):

FIG. 112 Sub-Page View Button

2. In the *General* tab of the Properties window - **Type** property, set **sub-page view** as the button type (FIG. 113):
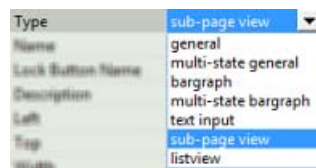
FIG. 113 Type (General) Property - sub-page view

Step 4 - Assign the Sub-Page Set to the Sub-Page View Button

The Sub-Page Set that was defined in Step 2 must now be assigned to the Sub-Page View button:

1. In the Design View, select the Sub-Page View button.

2. In the *General* tab of the Properties window, select "*Sub-Page Set 1*" from the *Sub-Page Set* drop-down menu (FIG. 114):
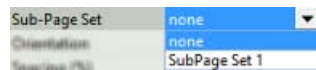
FIG. 114 Sub-Page Set (General) Property - "SubPage Set 1"

3. Once the Sub-Page Set has been assigned to the Sub-Page View button, the Sub-Pages that will be displayed on the touch panel page are represented with placeholders (FIG. 115):
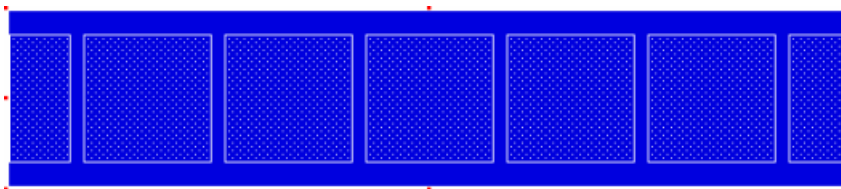
FIG. 115 Sub-Page View button - Sub-Pages indicated with placeholders

- These placeholders provide a visual indication of the current settings for size, anchor position and spacing of the Sub-Page Set contained by the Sub-Page View button.
- Note that by default, the Spacing (%) property is set to 0 (no spacing between the Sub-Pages), and Anchor Position is set to left (as represented above).
- See *Sub-Page View Buttons* on page 97 for details.

Step 5 - Set Other Scrolling-Related Properties for the Sub-Page View Button

Set the scrolling region-related (General) button properties for the Sub-Page View button:

- **Sub-Page Set** - Click the down arrow to select from a listing of all Sub-Page Sets that have been defined via the Edit Sub-Page Sets dialog. In this example, select Sub-Page Set 1.
- **Orientation** - Click the down arrow to select the orientation for the selected Sub-Page View button (Horizontal/Vertical, default = Horizontal). In this example, Orientation should be set to Horizontal.
- **Spacing (%)** - Enter an Integer (percentage) value to specify the amount of spacing between Sub-Pages when they are displayed within a Sub-Page View button (0-100, default = 0).

  This value represents the percentage of the Sub-Page's width (for Horizontal Sub-Page View buttons) or height (for Vertical Sub-Page View buttons) defined by the first Sub-Page in the Sub-Page Set associated with this Sub-Page View button.

For example, 0 (the default setting) will result in no spacing between the Sub-Pages displayed within a scrolling region. A value of 100 will insert a space that is equal to either the horizontal or vertical dimension (depending on whether the scrolling region is set to Horizontal or Vertical orientation) of the first Sub-Page in the Sub-Page Set.

- **Anchor Position**- Select an anchor position option to specify how the Sub-Pages associated with the selected Sub-Page View button are initially displayed and justified within the Sub-Page View button.

  The options in this menu depend on the Orientation setting of the selected Sub-Page View button:

  For *Horizontal* orientation:

  - Left: First Sub-Page is displayed aligned to the left side of the Sub-Page View button.
  - Middle: The middle Sub-Page is displayed positioned in the center of the Sub-Page View button (default setting).
  - Right: Last Sub-Page is displayed aligned to the right side of the Sub-Page View button.

  For *Vertical* orientation:

  - Top: First Sub-Page is displayed aligned to the top of the button.
  - Center: The middle Sub-Page is displayed positioned in the center of the button (default setting).
  - Bottom: Last Sub-Page is displayed aligned to the bottom of the button.

- **Show Sub-Pages** - This setting determines whether or not the Sub-Pages contained within a scrolling region are displayed on-screen (select Yes or No, default = Yes).

  If this property is set to No, then the Sub-Page View button will initially be displayed without sub-pages.

- **Reset View On Show** - This property determines whether to reset the positioning of the Sub-Pages displayed within a scrolling region (Yes/No, default = No), the next time the scrolling region is displayed.

Set Remaining Button Properties for the Sub-Page View Button

Finally, set the remaining button properties for the Sub-Page View Button, as you would for any other single-state button:

- Setting General Properties: Sub-Page View Buttons (see page 93)
- Setting Programming Properties: Sub-Page View Buttons (see page 93)
- Setting States Properties: Sub-Page View Buttons (see page 94)

The result as it will appear on the Touch Panel will be a horizontal scrolling region with 10 Sub-Pages, similar to the figure below (FIG. 116):
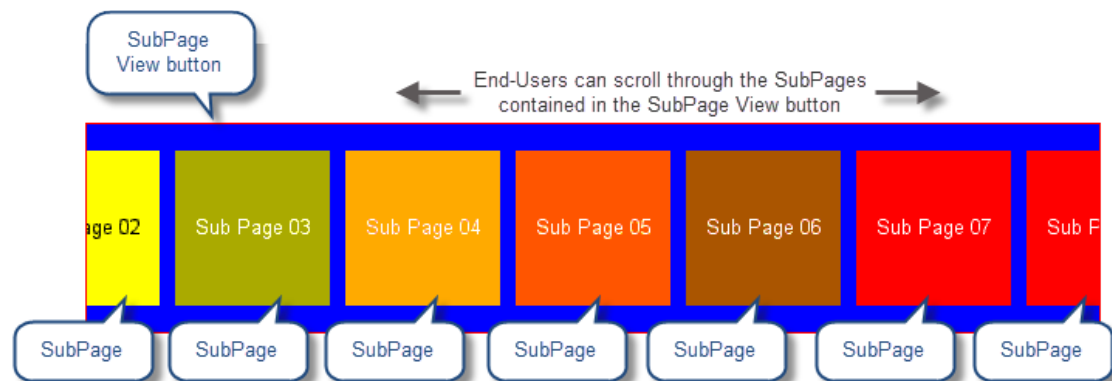
FIG. 116 Scrolling Region Example complete

- Keep in mind that each Sub-Page can be configured like any other (Standard) Popup Page.
- The motion of the Sub-Page Set is defined via the General button properties for the Sub-Page View button.
- The "clipping" of each Sub-Page as it enters and leaves the scrolling region is handled by the Sub-Page View button.

Listview Buttons & Dynamic Data

Overview

Modero X Series G5 touch panels and TPDesign5 (v1.2.0, build 65 or greater) support *Listview* buttons. Listview buttons provide the ability to display a listing of items from a dynamic data source on a G5 touch panel. Dynamic data can be created either using an XPort server, NetLinx code or a generic CSV file. The creator of the data can specify how many fields comprise a record and the format of those fields. As many records as necessary can be specified.

NOTE: *Dynamic data defines data files/feeds URL where the data can be loaded by the touch panel at runtime via HTTP (GET) or HTTPS (GET) transport protocols.*

This data can be used to populate a Listview button displayed on a G5 touch panel, where the end user can scroll or search through the list and make a selection. Once a selection has been made, a CUSTOM_EVENT is raised in the NX Master to retrieve the data fields comprising the selected record. An example Listview button is shown in FIG. 117:
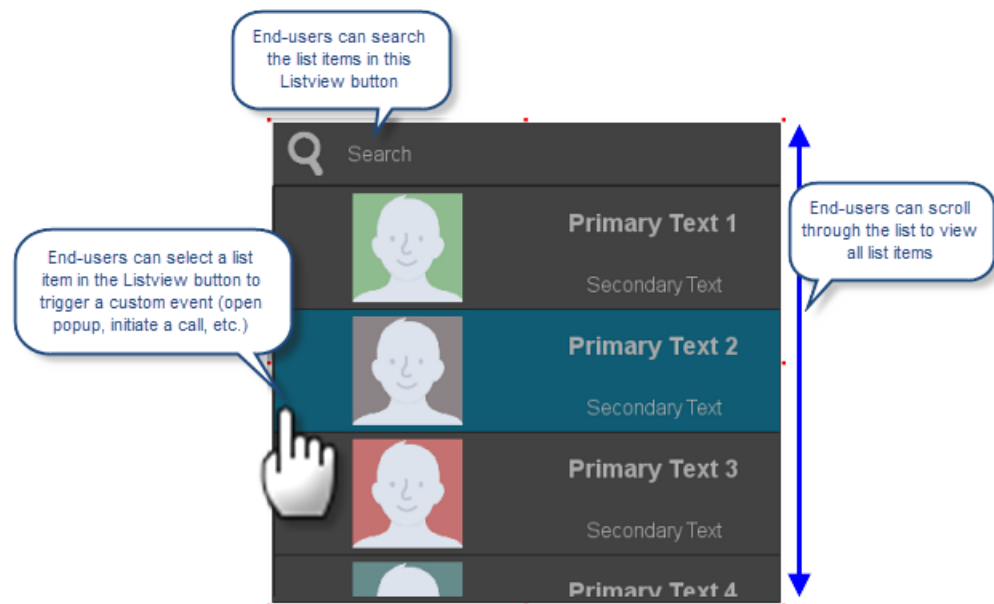
FIG. 117 Example Listview button (in TPDesign5)

End users can scroll through the items in the list, and select an item to initiate a custom event. Using the Outlook contacts list as an example, the end user could select a name in the Listview button to view contact information for the selected name, or call that contact directly from the panel, depending on NetLinx programming and settings in TPDesign5.

AMX System Requirements for Listview Buttons

The following software, hardware and firmware requirements must be met to support Listview buttons:

- TPDesign5 - version 1.2.0 build 65 (or greater)
- X Series G5 Touch Panels - panel firmware version 1.3.10 (or greater)
- NetLinx NX Series Masters - master firmware version 1.3.17 (or greater)

  NOTE: *Listview buttons will not work with NetLinx Masters that are in DoD Security Mode. Refer to the NX-Series Controllers, Enova DGX, Enova DVX, Massio WebConsole & Programming Guide for information on security mode settings on Central Controllers.*

- NetLinx.AXI file (version 1.55 or greater)

  NOTE: *To determine the version number of the NetLinx.AXI file currently loaded, refer to the NetLinx Studio 4 About dialog (Help > About NetLinx Studio). If your NetLinx.AXI file is older than version 1.55, use the NetLinx Support File Update Setup program to update the NetLinx.AXI file. See the Updating the NetLinx.AXI File to v1.55 section on page 106 for details.*

- A source for the data that will be presented in list form on the Listview button. See *Hosting a CSV Data Source File on the NX Master* section on page 108 for details.

Implementing Listview Buttons - Basic Workflow (CSV or XML)

Listview buttons are totally customizable, therefore there are no rules that dictate how to approach meeting the basic requirements of having a Listview button and a source for the dynamic data that will be displayed on the Listview button. However, a practical and logical workflow for implementing a Listview button with a hosted CSV or XPort-generated XML file as the data source is outlined below:

NOTE: *The data source for Listview buttons can also be defined via NetLinx Code. Refer to the Modero X G5 Programming and Configuration Guide for details.*

The example shown in these topics uses a CSV file named "*channelList.csv*", which contains a listing of TV channels and station icons that will be presented on the Listview button. Click here to view the data.

Implementing Listview Buttons - Basic Workflow (CSV or XML)			
Step		**Summary**	**Page Ref**
1	Create (draw) a Listview button	Listview buttons are drawn in the Design View, like any other button type - via the Button Draw tool.	page 106
2	Set Listview button properties	While Listview button properties are set via the Properties window like other button types, it is important to understand that Listview buttons have several unique properties, and others that work differently for Listview buttons than for other button types.	page 107
3	Host a Data Source file on the NX Master or XPort Server	If using a CSV file as the data source for the Listview button, the CSV file can be hosted on the NX Master. Use NetLinx Studio 4 to transfer the CSV to the NX Master. If using an XPort-generated XML file as the data source for the Listview button, the XML file is typically hosted on the XPort Server.	page 108
4	Add Dynamic Data Sources to the TPD5 Project	The data source file to be used to populate the data on the Listview button must be added to the TPD5 project via the Resource Manager.	page 109
5	Map the Data from the Data Source (CSV or XML) file to the fields in the Listview Button	With a data source associated with the Listview button, it is necessary to map the data in the CSV or XML file to the three fields (components) of the Listview button (Primary Text, Secondary Text and Image).	page 110
6	Assign a Data Source file to the Listview Button	Once you have created a Listview button, and a data source file is hosted on the NX Master, it is necessary to assign the data source to the Listview button, via the Dynamic Data Source (General) property.	page 112
7	Add Image Files to the TPD5 Project	If the Listview button needs to display images (for example, icons for TV stations), the images can be saved in your TPDesign5 project by importing them in to the Resource Manager (Images tab). The filenames of the imported images must match the filenames in the data source file. This step is optional, but precludes the need to retrieve image files from a separate server.	page 110
8	Configure the Response to a User Selection	Once these steps are complete, the project can be transferred to the panel.	page 113

Creating Listview Buttons - Examples

There are four demos at the end of this section that illustrate example workflows for configuring Listview buttons using four types of data source files:

1. a CSV file with headers (page 114)
2. a CSV file without headers (page 128)
3. an XPort-generated XML file (page 141)
4. NetLinx Data Source (page 154)

Updating the NetLinx.AXI File to v1.55

In order to support Listview buttons and dynamic data, **NetLinx.AXI file v1.55 (or higher)** is required. If your NetLinx.AXI file is older than v1.55, use the *NetLinx Support File Update Setup* program to update the NetLinx.AXI file.

Determining the Current Version of the NetLinx.AXI File

To determine the version number of the NetLinx.AXI file currently loaded, refer to the NetLinx Studio 4 *About* dialog (**Help > About NetLinx Studio**):
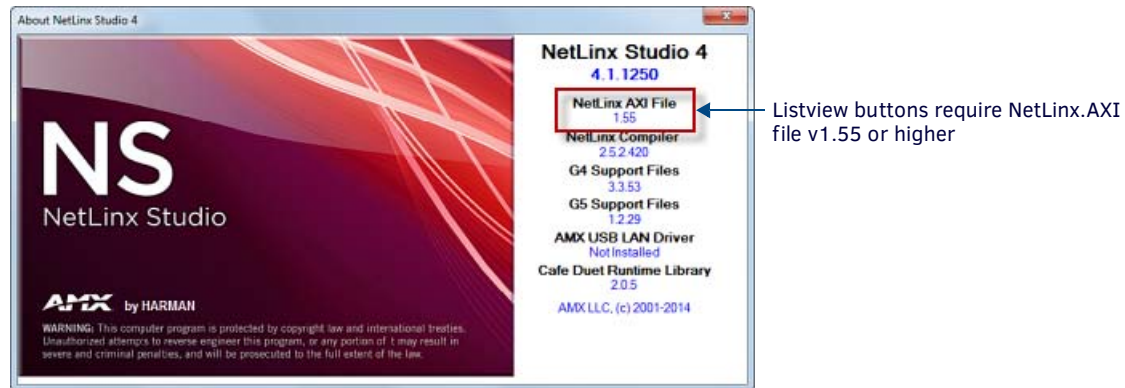
FIG. 118 About NetLinx Studio 4 dialog, indicating NetLinx.AXI file version 1.55

Updating the NetLinx.AXI File

1. Download the *NetLinx Support File Update Setup* file (**NetLinxSupportFileUpdateSetup.exe**) from www.amx.com (FIG. 119):

FIG. 119 NetLinx Support File Update installation program

2. Double-click the file to begin the installation, and click *Next* in each dialog to accept the default location for the updated files.

3. When the installation is complete, NetLinx Studio 4 is ready to support Listview buttons.

Creating Listview Buttons

Listview buttons are drawn via the Button Draw Tool, like any other button type. To set a button as a Listview button, select the button and choose "listview" as the Type in the *General* tab of the Properties window (FIG. 120):
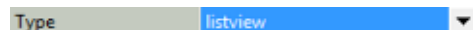
FIG. 120 General Property - Type (listview selected)

Refer to the *Creating New Buttons* section on page 66 for details.

With a Listview button selected in the Design View, several Listview-specific button properties are available via the Properties window.

Working With Listview Button Properties

While Listview button properties are the same as for other button types in many ways, it is important to understand that Listview buttons have several unique properties, and others that work differently for Listview buttons than for other button types:

Listview Buttons - General Properties

The following General properties are specific to Listview buttons:

- Listview Components (see page 239)
- Item Height (see page 238)
- Listview Columns (see page 239)
- Listview Item Layout (see page 241)
- Primary Partition (%) (see page 242)
- Secondary Partition (%) (see page 245)
- Filter Enabled (see page 236)
- Filter Height (see page 237)
- Alphabet Scrollbar (see page 235)
- Dynamic Data Source (see page 236)

Listview Buttons - Programming Properties

Once you have created a Listview button, you can use the Programming tab of the Properties window to set/edit programming-oriented button properties. To edit any of the properties, click in the right-hand table cell to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.

Listview buttons only use the Address Port and Address Code Programming properties:

- Address Port (see page 247)
- Address Code (see page 249)

  Channel Port and Channel Code are not supported for Listview buttons.

Listview Buttons - States Properties

Rather than the On/Off state options that apply to other button types, Listview buttons support the following two states:

- **Default** - The property values of the Default state will be used to render non-selected list items and also the button background, in the eventuality that there are not enough list items to fill the entirety of the Listview button.
- **Selected** - The property values of the Selected state will be used to render selected list items.

NOTE: *Note that in the TPD5 State Manager window,* Default *is labeled "Off" and* Selected *is labeled "On".*

Other than these new State types, the State properties supported by Listview buttons is basically the same as for other button types, with two properties that are specific to Listview buttons: *Secondary Font* and *Secondary Font Size*:

- Secondary Font (see page 253)
- Secondary Font Size (see page 253)

Note that the *Secondary Font* and *Secondary Font Size State* properties are available even if the selected Listview button only uses a single line of text. In this case, if the *List View Type* is changed to either two-line text or two-line text with icon, the second line of text will use these settings.

Several other States properties do not apply to Listview buttons:

- *Border Style* and *Border Name:* The Border Style and Border Name State properties are not configurable for Listview buttons. These values are fixed at "Single Line", which is used to draw both the outline of the list items and the Listview button itself.

  Note that the **Border Color** state property specifies the color used for the outlines.
- *Chameleon Images* are not supported for Listview buttons.

Listview Buttons - Events Properties

Use the *Events* tab of the TPD5 Properties window to set event properties for the selected Listview button. Listview buttons support the following three Events properties that are specific to Listview buttons. However, these Events support the same *actions* as existing events.

- Item Selected (see page 258)
- Scrollbar Begin (see page 258)
- Scrollbar End (see page 258)

Hosting a CSV Data Source File on the NX Master

NOTE: *Download and install the latest version of NetLinx Studio 4 from www.amx.com.*

The CSV file that will act as the data source for a Listview button can be hosted on the NX Master. CSV files can be used whether they use Headers or not. CSV files can be hosted on the NX Master. The example shown in these steps uses a CSV file named "channelList.csv", which contains a listing of TV channels and station icons that will be presented on the Listview button.

To host a CSV file on the NX Master:

1. Open NetLinx Studio 4 and establish communication with the Master (refer to NetLinx Studio online help for details on communication settings).

2. Select **Tools > File Transfer** to open the *File Transfer* dialog (FIG. 121):
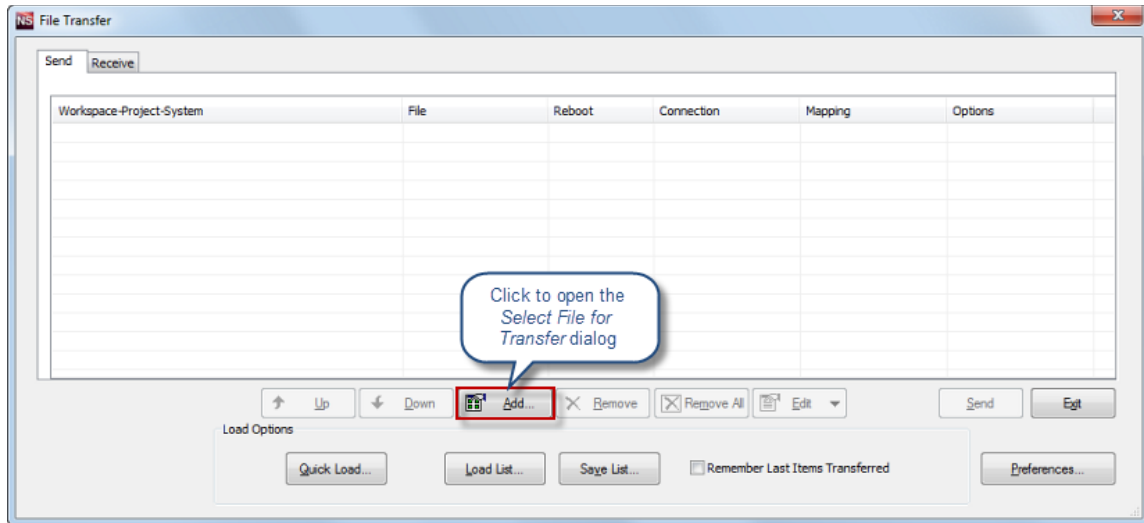
FIG. 121 NetLinx Studio 4 - File Transfer dialog

3. Click **Add** to open the *Select Files for File Transfer* dialog, and open the *Other* tab (FIG. 122):
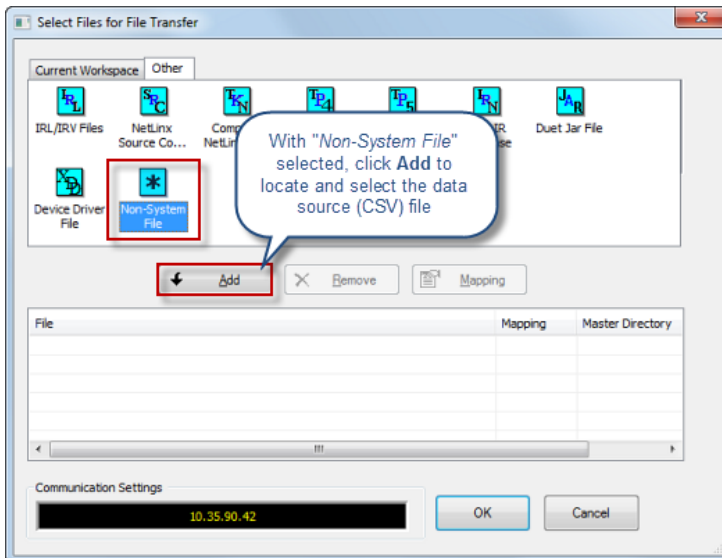
FIG. 122 NetLinx Studio 4 - Select Files for File Transfer dialog

4. Select **Non-System File**, then click **Add**.

5. In the *Open* dialog, locate and select the CSV file to use as a data source and click **Open** to access the *Enter Device Mapping Information* dialog (FIG. 123):
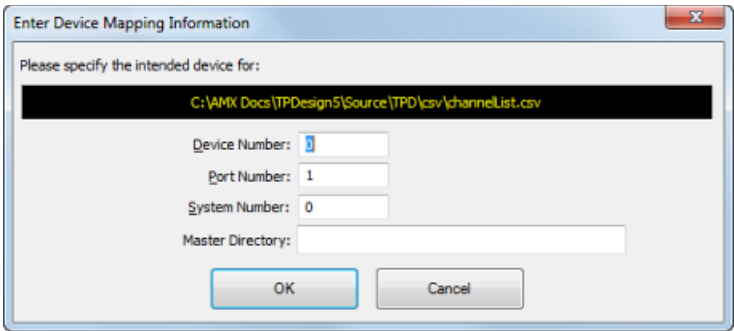
FIG. 123 NetLinx Studio 4 - Enter Device Mapping Information dialog

6. Enter the *Device*, *Port* and *System* Number for the target NX Master.

7. In the *Master Directory* field, enter the name of the directory on the NX Master that contains the data source file.

   NOTE: *If no directory is specified in the Master Directory field, the file will be copied to the root directory on the Master.*

8. Click **OK** to save changes and close the *Enter Device Mapping Information* dialog (and return to the *Select Files For File Transfer* dialog).

9. In the *Select Files For File Transfer* dialog, the selected file and it's device information are indicated in the *Files* list (FIG. 124):
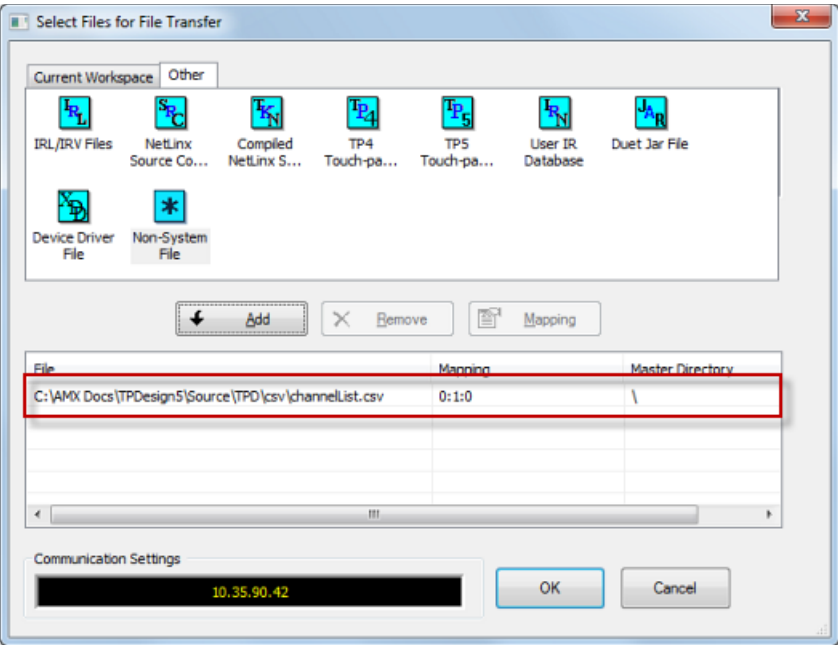
FIG. 124 NetLinx Studio5 - Select Files for File Transfer dialog indicating a CSV file for transfer

10. Click **OK** to close this dialog and return to the *File Transfer* dialog.

11. Click **Send** to initiate the file transfer. The Output Bar will indicate when the transfer is complete (FIG. 125):

FIG. 125 NetLinx Studio (Output Bar) - File Transfer complete

Adding Dynamic Data Sources to the Project

The following example describes adding a data source (CSV) file to the Resource Manager (for use with Listview buttons).

1. In the *Dynamic Data Sources* tab of the Resource Manager, click **New** to open the *Create Dynamic Data Source* dialog (FIG. 126):

2. In the *Name* field, enter a unique name for the data source. Note that the name entered here is used to represent this data source in the Resource Manager - Dynamic Data Sources tab.

3. In the *Host* field, enter the host name, which must be a fully qualified DNS or IP address.

4. In the *Path* field, enter the path to the source file. The path must be a valid HTTP URL minus the protocol and host. The only exception to this is the inclusion of special escape sequences and regular expressions.

5. In the *File* field, enter a file name that indicates the full path to the location of the source file.
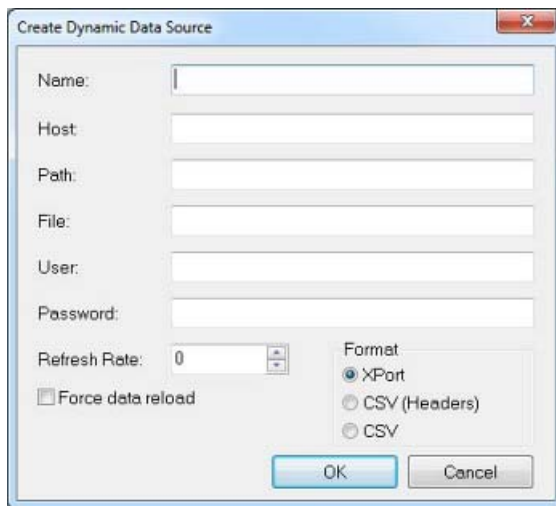
FIG. 126 Create Dynamic Data Source dialog

6.   In the *User* field, enter the user name required by the NX Master or server for authentication (if required).

7.   In the *Password* field, enter the password required by the NX Master or server for authentication (if required).

8.   In the *Refresh Rate* field, use the up/down arrows to adjust the number of seconds between refreshes in which the resource is downloaded again. Refreshing resources will cause the button displaying that resource to refresh as well. The default value is 0, which means that the resource is only downloaded once.

9.   Set the *Force Data Load* option. This option is only available if *Refresh Rate* is set to zero. If this option is selected, it will force a reload of the data file and images associated with data file. By default, this option is disabled.

10.   Under *Format*. specify the format of the source file:
    - **XPort** - Select if the data source file is XPort-generated XML.

        This option also applies when using NetLinx Data as the data source.

        Note that this is the default selection.
    - **CSV (Headers)** - Select if the data source is a CSV file with headers.
    - **CSV** - Select if the data source is a CSV file that does not have headers.

11.   Click **OK** to save changes and close this dialog. The new data source is indicated in the Dynamic Data Sources tab.

Adding Image Files to the Resource Manager

If there are images to be displayed on the Listview button (as Images), consider saving them to your TPDesign5 project.

The filenames of the images must match the filenames referenced in the data source file.

See *Importing Image Files Into the Project* on page 32.

Mapping the Data to Fields in the Listview button

It is necessary to map the data in the Data Source (CSV or XML) file to the three fields that comprise the Listview button layout. These three fields (or Components) are: *Primary Text, Secondary Text* and *Image* (FIG. 127):
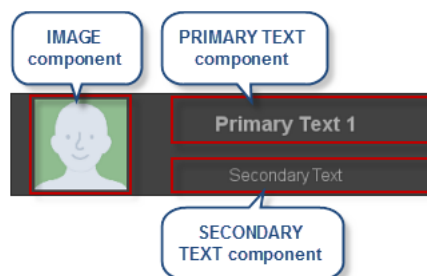
FIG. 127 Listview Button - Components

NOTE: *Listview buttons can use all of these components, or only the Primary Text component. Use the List View Components (General) property to specify which components will be displayed on the selected Listview button.*

Data-Mapping for content displayed on a Listview button is configured via the *Dynamic Data Mappings - Listview* dialog. This dialog is accessed via the Data button in the Dynamic Data Sources tab of the Resource Manager dialog.

To map data from a Dynamic Data Source to a Listview button, you must first have created a Listview button (see page 106), and specified a Data Source for the Listview (see page 112).

NOTE: *The example shown in these steps uses a CSV file named "channelList.csv", which contains a listing of TV channels and station icons that will be presented on the Listview button. Refer to page 126 to view the example CSV data.*

Step One: Analyze the Data Source

In this example the data source came from another source and is being reused for Listview UI and system control purposes. It must be analyzed in order to use it properly.

In the absence of headers, the columns will be named by default as: *column1*, *column2*, *column3*... (see the *Dynamic Data Mappings - Syntax Requirements* section on page 111 for details).

This information is necessary in order to map the images onto the listview button and retrieve the channel number in the NetLinx code.

Step Two: Map the Data to Fields (Components) of the Listview button

1. Select a Listview button in the Design View.
2. Select **Panel > Resource Manager**, and open the Resource Manager to the *Dynamic Data Sources* tab.
3. Select the Data Source that is assigned to the selected Listview button (i.e. "*channelList.csv*").
4. Click the **Data Maps** button to access the *Dynamic Data Mappings - Listview Buttons* dialog (FIG. 128):

(In the Resource Manager -
Dynamic Data Sources tab)
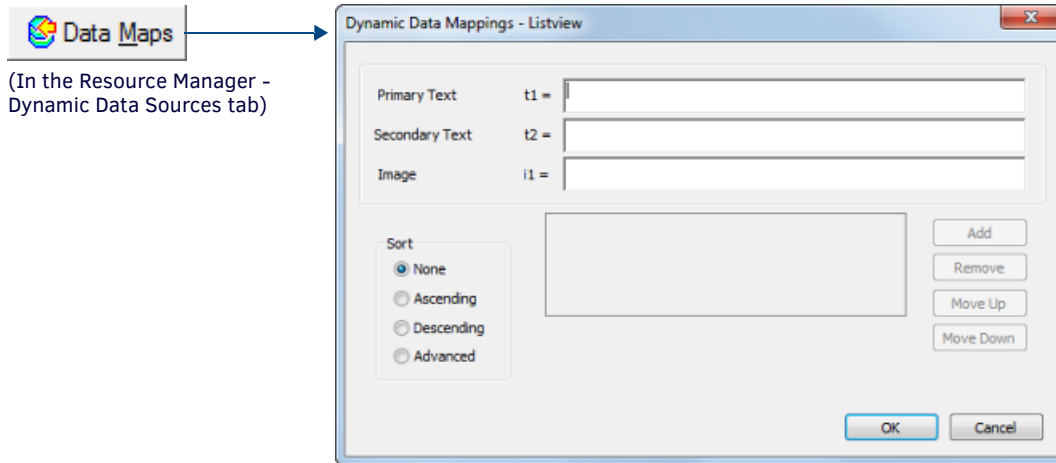
FIG. 128 Dynamic Data Mappings - Listview Buttons dialog

5. Use the fields in this dialog to specify the device mapping for the selected Listview button and the selected Data Source (see Dynamic Data Mappings - Syntax Requirements).

Dynamic Data Mappings - Syntax Requirements

Note that the syntax requirements for these fields depends on the type of file used as the data source (CSV without headers, CSV with headers or Xport-generated XML):

Dynamic Data Mappings - Syntax Requirements	
CSV Without Headers	**CSV With Headers or XPort-generated XML**

Primary Text:
 For CSV files without headers, the syntax is:
   ${column#}
 In the absence of headers, the columns will be named by default as: *column1*, *column2*, *column3*...
 Following this syntax, enter the column # in the data source file to be displayed as the Primary Text component of the Listview button.
 For example, for a CSV file with four columns for "NAME" (column 1), "CHANNEL" (column 2), "ICON" (column 3) and "RATING" (column 4), to display NAME as Primary Text, enter **${column1}**:
 	
 Primary Text   t1 = ${column1}
 	
 Multiple comma-separated fields can be included in any component. For example, to display both NAME and CHANNEL as Primary Text, the syntax would be:
   ${column1}, ${column2}
	Primary Text:
	 For CSV files with headers or XML, the syntax is:
	   ${header/field name}
	 Following this syntax, enter the name of the header (CSV) or field (XML) in the data source file to be displayed as the Primary Text component of the Listview button.
	 For example, for a CSV file that contains a listing of TV channels with the headers "NAME" (the channel names), "CHANNEL" (channel numbers), "ICON" (logo icons for each channel), and "RATING" (rating information for each channel), the syntax for mapping the Primary Text field to the "NAME" header would be **${NAME}**:
	
	 Primary Text t1 = ${NAME}
	
	 Multiple comma-separated fields can be included in any component. For example, to display both NAME and CHANNEL as Primary Text, the syntax would be:
	   ${NAME}, ${CHANNEL}

Dynamic Data Mappings - Syntax Requirements (Cont.)	
CSV Without Headers	**CSV With Headers or XPort-generated XML**
Secondary Text: For CSV files without headers, the syntax is: `${column#}` In the absence of headers, he columns will be named by default as: *column1*, *column2*, *column3*... Following this syntax, enter the column # in the data source file to be displayed as the Secondary Text component of the Listview button. For example, to display CHANNEL (column 2 in the CSV file) as Secondary Text, the syntax would be: **`${column2}`** Secondary Text t2 = \$\{column2\}	Secondary Text: For CSV files with headers or XML, the syntax is: `${header/field name}`. Following this syntax, enter the name of the header (CSV) or field (XML) in the data source file to be displayed as the Secondary Text component of the Listview button. For example, the syntax for mapping the Secondary Text field to the "RATING" header (using the same CSV file containing a listing of TV channels) would be `${RATING}` Secondary Text t2 = \$\{RATING\}

Image:
For CSV files without headers, the syntax is
`${column#}`
In the absence of headers, the columns will be named by default as: *column1*, *column2*, *column3*...
Following this syntax, enter the column # in the data source file to be displayed as the Image component of the Listview button.
To display ICON (column 3 in the CSV file) as the Image component, the syntax would be:
`${column3}`
 	
Image   i1 =   \$\{column3\}
	Image:
	For CSV files with headers or XML, the syntax is
	`${header/field name}`
	Following this syntax, enter the name of the header or field in the CSV or XML Data Source file that contains the file names of the image files to be displayed as the Image component of the Listview button. For example, the syntax for mapping the Image field to the "ICON" header (using the same CSV file containing a listing of TV channels) would be
	`${ICON}`
	
	Image i1 = \$\{ICON\}

Assigning a Data Source to a Listview button

Once a Listview button has been created and a data source has been hosted on the NX Master, the data source must be associated with the Listview button, via the *Dynamic Data Source* (General) property.

1. With a Listview button selected in the Design View, click on the **Dynamic Data Source** property in the *General* tab of the Properties window. This opens the *Select Resource* dialog (FIG. 129):

Dynamic Data Source   `...`

FIG. 129 Dynamic Data Source (General) Property

2. Select the file to use as the Data Source for the selected Listview button (see "*Channel List* in FIG. 130):
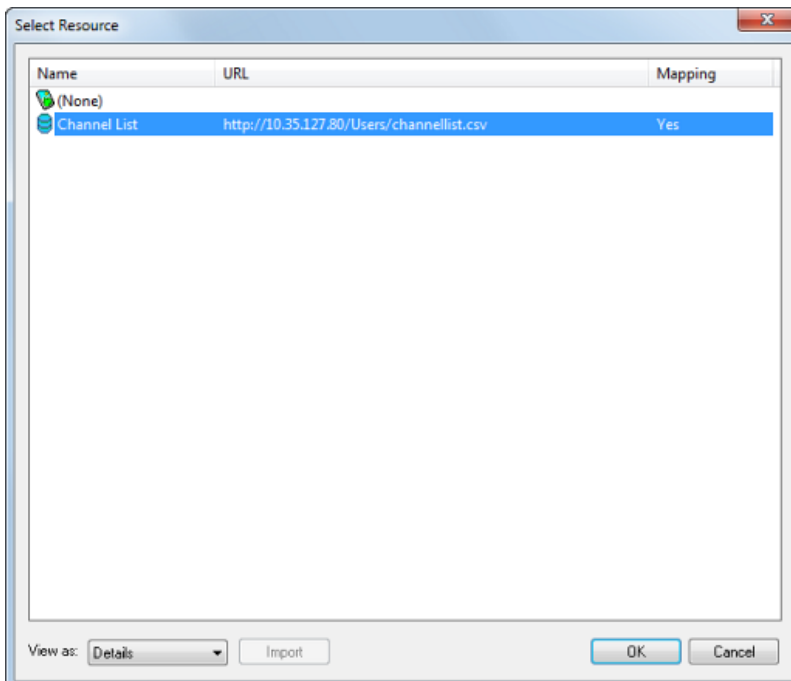
FIG. 130 Select Resource dialog.

3. Click **OK** to close this dialog.

4. The selected Data Source file is indicated in the *Dynamic Data Source* property (see "*channelList.csv*" in FIG. 131):

Dynamic Data Source channellist ...
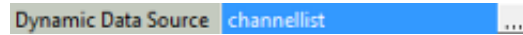
FIG. 131 Dynamic Data Source property indicating "channelList.csv"

Configuring the Response to a User Selection

When the user selects an item on the Listview button, the entire record for that selection is sent to the NX Master. A CUSTOM_EVENT is raised and within this function the desired information can be retrieved for the selection.

NOTE: *The example Custom Event shown below refers to a CSV file named "channelList.csv", which contains a listing of TV channels and station icons that will be presented on the Listview button. Refer to Listview Button/Dynamic Data Example 1: CSV File - With Headers section on page 114 for details.*

In this example, the channel number needs to be retrieved. The channel number can be used to initiate a channel change on the cable box:

```
DEFINE_EVENT

// The custom event that is raised whenever a TV listview item is selected on the panel.
// Custom event data has three integers, a data string and an
// encoding string. This is not enough to represent what could
// potentially be a very complex DATA_RECORD. So the listview
// custom event will include a payload ID that can then be used
// to retrieve the contents of the DATA_RECORD.

CUSTOM_EVENT[dvTP,btnTvListview,LISTVIEW_ON_ROW_SELECT_EVENT]
{
  // Variables to hold the ID and type for the payload
  SLONG payloadId
  SLONG payloadType
  // The function to retrieve the payload data takes an array of 1 or more
  // strings that specify which DATA_FIELDs we wish to retrieve. In our
  // example we're interested in 3 fields and 16 characters is long enough
  // to hold the IDs.
  CHAR fields[3][16]
  // Create a DATA_RECORD to hold the retrieved data
  DATA_RECORD record
  // Get the payload ID from the custom event
  payloadId = custom.value1
  // Get the data type from the custom event
  payloadType = custom.value2
  // Always check for a valid payload ID and check the payload
  // type. Future improvements to the feature may have other
  // payload types.
  if (payloadId > 0 && payloadType == DATA_STRUCTURE_DATARECORD)
  {
    // Specify which DATA_FIELD IDs we want to retrieve from the payload
    fields[1] = 'NAME'
    fields[2] = 'CHANNEL CODE'
    fields[3] = 'ICON'
    // When retrieving the data, always check the return value. If the
    // return value is greater than zero then the DATA_RECORD that was
    // passed in will be populated with the requested DATA_FIELDs.
    if (DATA_GET_EVENT_RECORD(dvTP, payloadId, fields, record) > 0)
    {
      // Put the channel number and name at the bottom of the TV
      // subpage and show the subpage
      SEND_COMMAND dvTP,"'^TXT-13,0,Channel ',record.content[2].value, ' - ',record.content[1].value"
      SEND_COMMAND dvTP,"'^BMX-77,0,',record.content[3].value,',1,10'"
    }
  }
}
```

Listview Button/Dynamic Data Example 1: CSV File - With Headers

The following instructions describe using the TV Guide demo for creating a Listview button with a dynamic data source in the form of a CSV file with headers that is hosted on an NX Master.

NOTE: *This set of instructions uses files that are included in the "TV Guide.ZIP" demo file which is available to download from the UI RESOURCE CENTER at www.amx.com.*

The resulting Listview button will display a listing of TV channels with each channel's station icon in a three-column grid layout (FIG. 132):

FIG. 132 Example - Listview button based on "channelList.csv"

Before You Begin

1. Download the *TVGuide.ZIP* file from the UI RESOURCE CENTER at www.amx.com and extract it's contents to a known location.
2. Open the *channelList.csv* file and analyze it's contents. It is a relatively simple csv file that consists of four columns with headers (NAME, CHANNEL CODE, ICON and RATING). See page 126 to view this file.

1) Create (draw) a Listview button

1. In TPDesign5, open a Page and use the Button Draw tool to create a new button.
2. With the new button selected, click the **Type** (General) property and select **Listview** from the drop-down of button types. This selection sets the new button as a Listview button, and enables a set of Listview-specific properties (FIG. 133):

Type (General) property set to Listview   Example Listview button
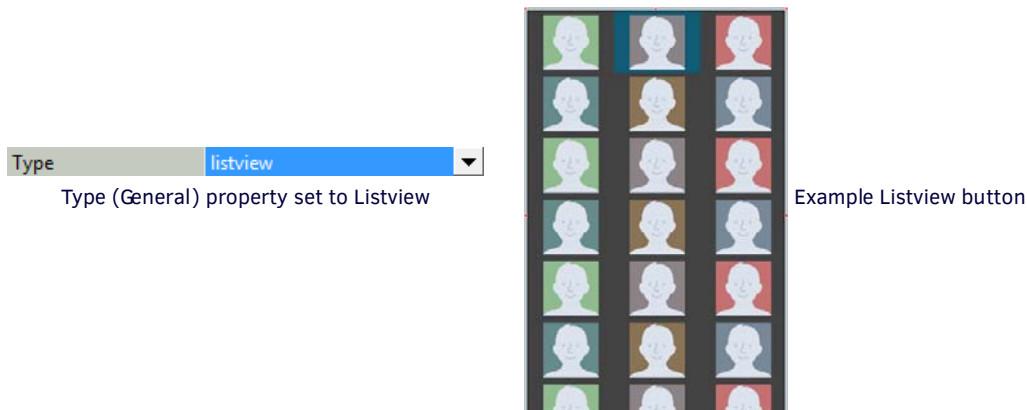
FIG. 133 Type (General) Property set to Listview

NOTE: *The "TVGuide.TP5" file included in the TV Guide demo has a Listview button already drawn on the "Main" page.*

2) Set the Listview Button Properties

Use the options in the Properties window to view/edit the *General*, *Programming* and *States* properties for the Listview button (this demo does not use *Events* properties). The settings used for the Listview button in the TV Guide demo are shown in FIG. 134:

| General Properties | Programming Properties | States Properties |
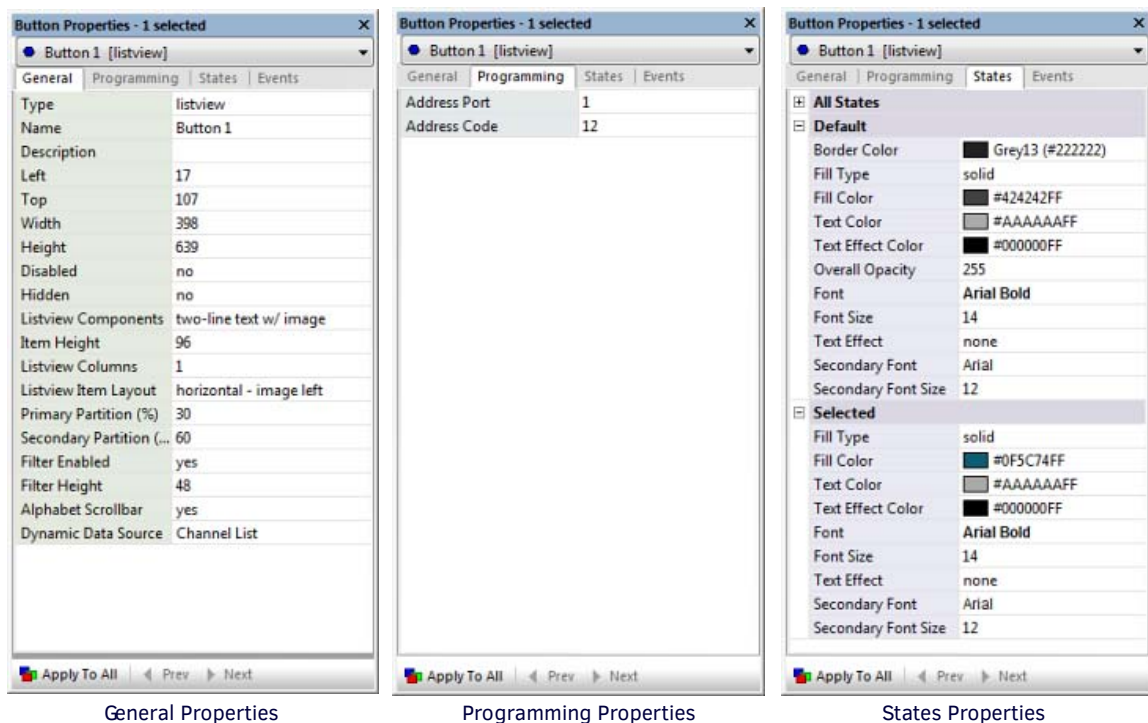
FIG. 134 Properties for the TV Guide Listview Button

NOTE: *The Listview button in the TV Guide demo is pre-configured with the General, Programming and States properties shown above.*

Refer to the *Working With Listview Button Properties* section on page 107 for details on Listview-specific button properties.

3) Host the Data Source File (CSV with Headers) on the NX Master

In this example, "channelList.csv" will be the data source for the Listview button. This CSV file will be hosted on the NX Master. "channeList.csv" contains a listing of TV channels and station icons that will be presented on the Listview button. FIG. 135 presents a sample of the first few rows of this file. Refer to page 126 to view the entire file.

FIG. 135 Data Source File - "channelList.csv" (CSV file with headers)

1. In NetLinx Studio 4, establish communication with the Master (refer to NetLinx Studio 4 online help for details).

2. Select **Tools > File Transfer** to open the *File Transfer* dialog (FIG. 136):

FIG. 136 NetLinx Studio 4 - File Transfer dialog

3. Click **Add** to open the *Select Files for File Transfer* dialog, and open the *Other* tab (FIG. 137):
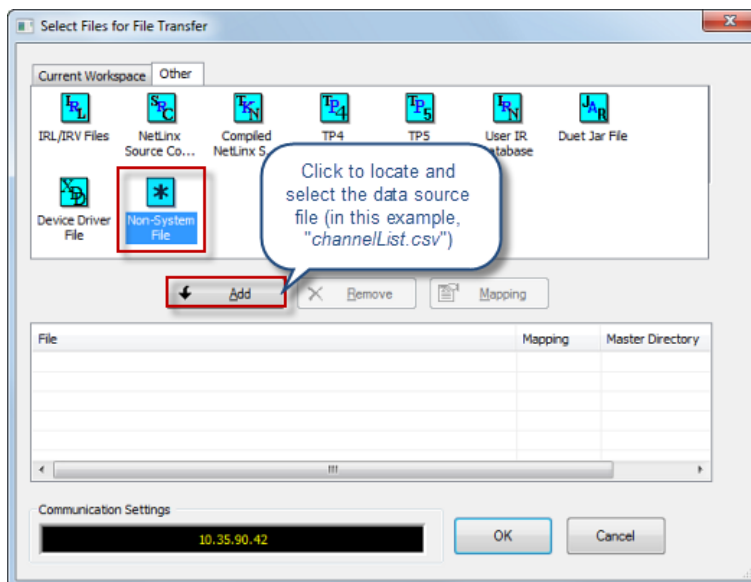
FIG. 137 NetLinx Studio 4 - Select Files for File Transfer dialog

4. Select **Non-System File**, then click **Add**.

5. In the *Open* dialog, locate and select the "*channelList.csv*" file and click **Open** to access the *Enter Device Mapping Information* dialog (FIG. 138).
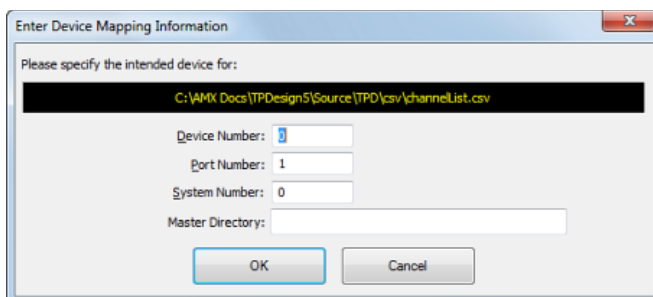
FIG. 138 NetLinx Studio 4 - Enter Device Mapping Information dialog

    a. Enter the *Device*, *Port* and *System* Number for the target NX Master.

    b. In the *Master Directory* field, enter the name of the directory on the NX Master that contains the data source file.

    NOTE: *If no directory is specified in the Master Directory field, the file will be copied to the root directory on the Master.*

    c. Click **OK** to save changes and return to the *Select Files For File Transfer* dialog.

6. In the *Select Files For File Transfer* dialog, the selected file and it's device information are indicated in the *Files* list (FIG. 124):
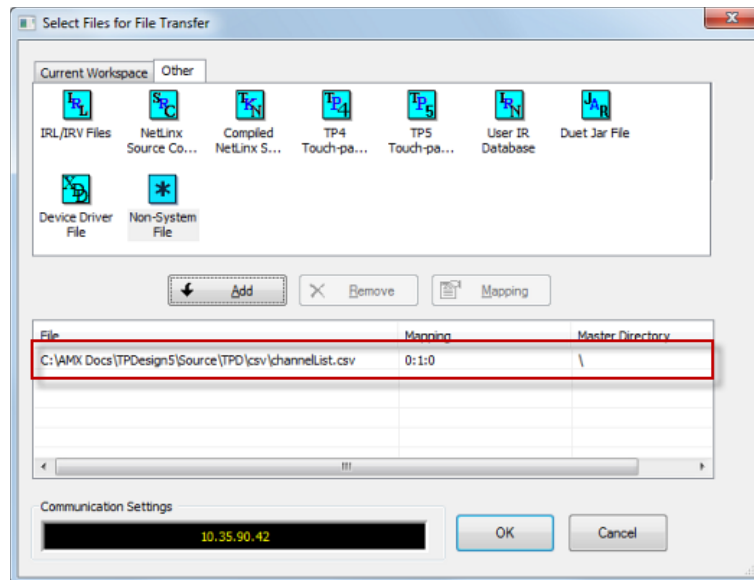
FIG. 139 NetLinx Studio 4 - Select Files for File Transfer dialog indicating a CSV file for transfer

7. Click **OK** to close this dialog and return to the *File Transfer* dialog.

8. Click **Send** to initiate the file transfer. The program will indicate when the transfer is complete.

4) Add the Dynamic Data Source to the Project

To add the data source file (channelList.csv) to the TPDesign5 project:

1. Open the Resource Manager to the *Dynamic Data Sources* tab and click **New** to open the *Create Dynamic Data Source* dialog (FIG. 140):
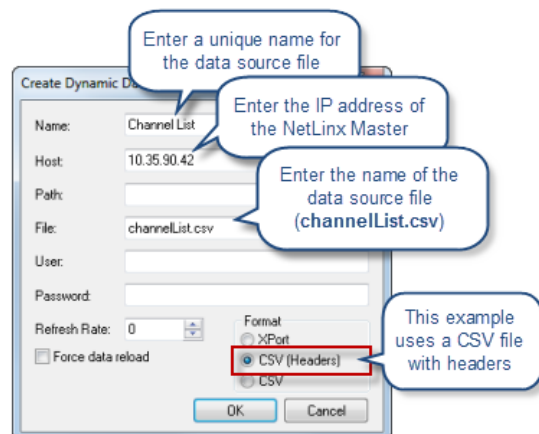
FIG. 140 Create Dynamic Data Source dialog with Example data (ChanelList.csv)

2. In the *Name* field, enter a unique friendly name for the data source. For this example, enter "**Channel List**".

3. In the *Host* field, enter the host name, which must be a fully qualified DNS or IP address.

4. In the *File* field, enter a file name that indicates the full path to the location of the source file.

5. In the *User* field, enter the user name required by the NX Master or server for authentication (if required).

6. In the *Password* field, enter the password required by the NX Master or server for authentication (if required).

7. In the *Refresh Rate* field, use the up/down arrows to adjust the number of seconds between refreshes in which the resource is downloaded again. Refreshing resources will cause the button displaying that resource to refresh as well. The default value is 0, which means that the resource is only downloaded once.

8. Under *Format*, select **CSV (Headers)**, since the data source file in this example (channelList.csv) uses a CSV file with headers.

9. Click **OK** to save changes and close this dialog. The new data source is indicated in the Resource Manager - Dynamic Data Sources tab (FIG. 141):
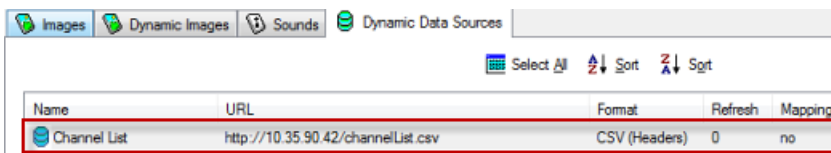
FIG. 141 Resource Manager - Dynamic Data Sources tab indicating "channelList.csv" as the data source

NOTE: *The Listview button in the TVGuide.TP5 file is pre-configured to use ChannelList (channelList.csv) as it's data source file. However, it is necessary to update the Host address with the IP address of your NX Master as shown above. Double-click on Channel List in the Resource Manager to open the Edit Dynamic Data Source dialog and update accordingly.*

5) Map the Data from the Data Source File to the Listview Button Components

It is necessary to map the data in the *channelList.csv* file to the three fields that comprise the Listview button layout. These three fields (called Components in TPDesign5) are: *Primary Text, Secondary Text* and *Image* (FIG. 142):
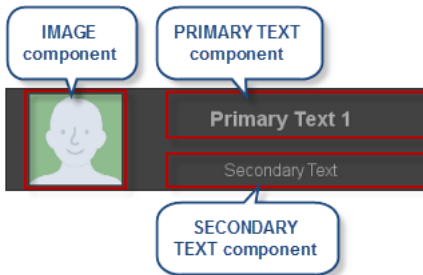
FIG. 142 Listview Button Components

Step One: Analyze the Data Source

It is necessary to understand the contents of the data source file in order to map the data to the Components in the Listview button. In this example, the *channelList.csv* file contains four columns with headers: *NAME*, *CHANNEL*, *ICON* and *RATING* (FIG. 143):

FIG. 143 Understanding the contents of the data source file - channelList.csv

In this example:

- The items in the **NAME** column will be mapped to display as the *Primary Text* component of the list items in the Listview button.
- The items in the **RATING** column will be mapped to display as the *Secondary Text* component of the list items in the Listview button.
- The items in the **ICON** column will be mapped to display as the *Image* component of the list items in the Listview button.
- The items in the **CHANNEL CODE** column will not be mapped to display in the Listview button. However, this data can still be put to use in a custom event - see *8) Write a Custom Event To Respond To User Selection* on page 122 for details.

Step Two: Map the Data to Components of the Listview button

1. With the Listview button selected, open the Resource Manager to the *Dynamic Data Sources* tab.
2. Select the data source (*channelList.csv*) that is assigned to the Listview button (as described on page 117):
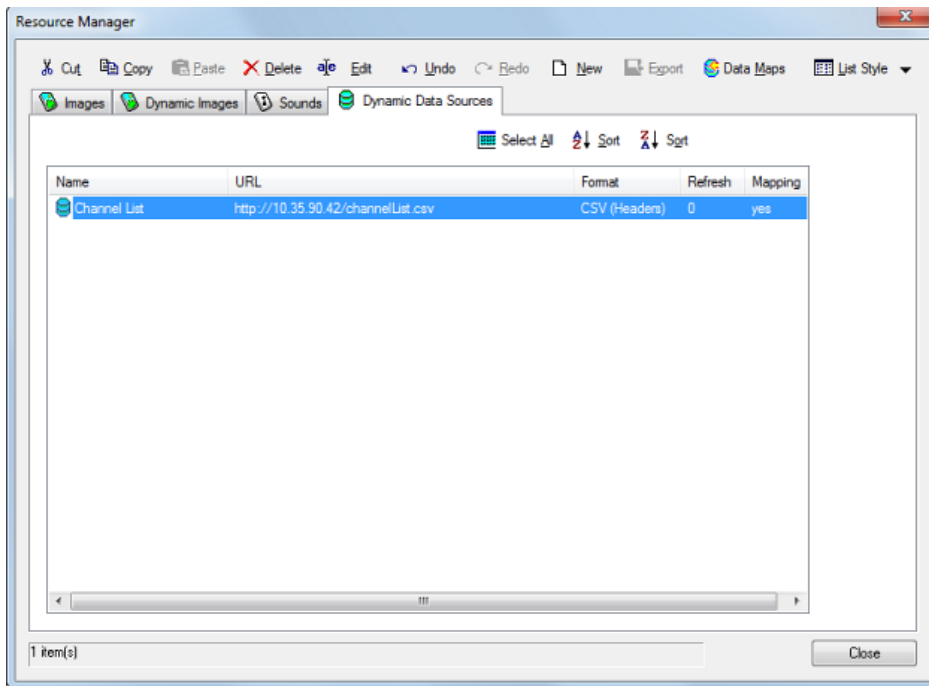
FIG. 144 Resource Manager - Dynamic Data Source tab

3.   Click the **Data Maps** button to access the *Dynamic Data Mappings - Listview Buttons* dialog (FIG. 145):
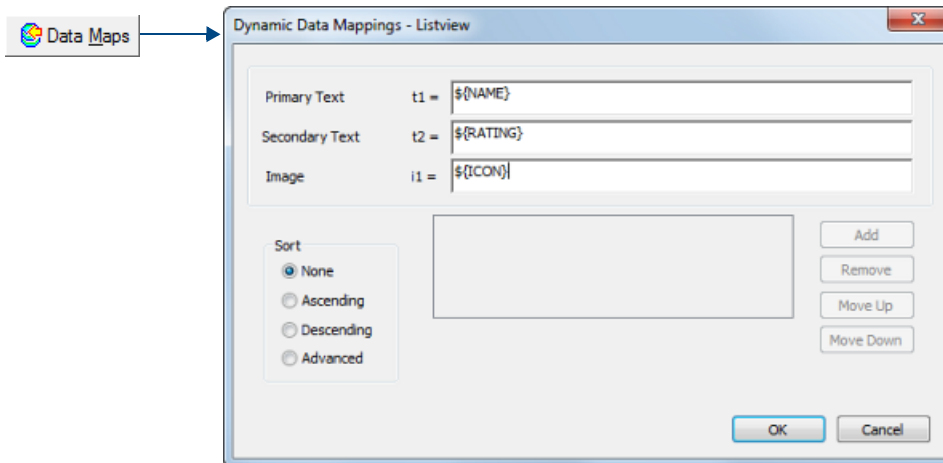
FIG. 145 Dynamic Data Mappings - Listview dialog (with example data indicated)

4.   Use the fields in this dialog to specify the device mapping for the selected Listview button and the selected Data Source (see *Dynamic Data Mappings - Syntax Requirements (CSV with Headers)* below).

NOTE: *The Listview button in the TV Guide demo is pre-configured with the data mapping settings shown above.*

Dynamic Data Mappings - Syntax Requirements (CSV with Headers)

Note that the syntax requirements for these fields depends on the type of file used as the data source. The data source file in this example uses a CSV file with headers. The syntax requirements for data mapping to a CSV with headers is described below:

Dynamic Data Mappings - Syntax Requirements (CSV with Headers)
Primary Text:
For CSV files with headers, the syntax is:
${header}
Following this syntax, enter the name of the header in the data source file to be displayed as the Primary Text component of the Listview button. In this example, channelList.csv lists TV channel names in the "NAME" column. To display the contents of the NAME column as the Primary Text component, enter **${NAME}** in the *Primary Text* field:

Dynamic Data Mappings - Syntax Requirements (CSV with Headers)

Secondary Text:

For CSV files with headers, the syntax is:

    ${header}

Following this syntax, enter the name of the header in the data source file to be displayed as the Secondary Text component of the Listview button. In this example, channelList.csv lists TV channel ratings in the "RATING" column. To display the contents of the RATING column as the Secondary Text component, enter **${RATING}** in the *Secondary Text* field:

Secondary Text	t2 =	${RATING}

Image:

For CSV files with headers, the syntax is

    ${header}

Following this syntax, enter the name of the header in the data source file to be displayed as the Image component of the Listview button. In this example, channelList.csv lists the file names of the image files associated with each TV channel (station icons) in the "ICON" column. To display the contents of the ICON column as the Image component, enter **${ICON}** in the *Image* field:

Image	i1 =	${ICON}

NOTE: *The fields in the Dynamic Data Mappings - Listview Buttons dialog are case-sensitive.*

6) Add Image Files to the Project

In the data source file for this example (channelList.csv), the *ICON* column lists image files associated with each TV channel in the list (FIG. 146):

	A	B	C	D
1	NAME	CHANNEL CODE	ICON	RATING
2	A&E	118	A&E.jpg	PG-13
3	ABC	8	ABC.jpg	PG-13
4	ABCFAM	180	ABCFAM.jpg	PG-13
5	ANIMAL	184	ANIMAL.jpg	PG-13
6	BBC	135	BBC.jpg	PG-13
7	BRAVO	129	BRAVO.jpg	PG-13
8	CBS	11	CBS.jpg	PG-13
9	CMDY-E	107	CMDY-E.jpg	PG-13
10	CMT	166	CMT.jpg	PG-13

FIG. 146 Data Source File - "channelList.csv"

In order to display these image files on the Listview button, the image files named in the data source file must be added to the project, via the Resource Manager - Images tab:

1. Open the Resource Manager to the *Images* tab.

2. Click **Import** to access the *Open* dialog. Locate and select all of the image files that are named in the data source file (channelList.csv).

3. Click **OK** to import the selected files and return to the Resource Manager (FIG. 147).
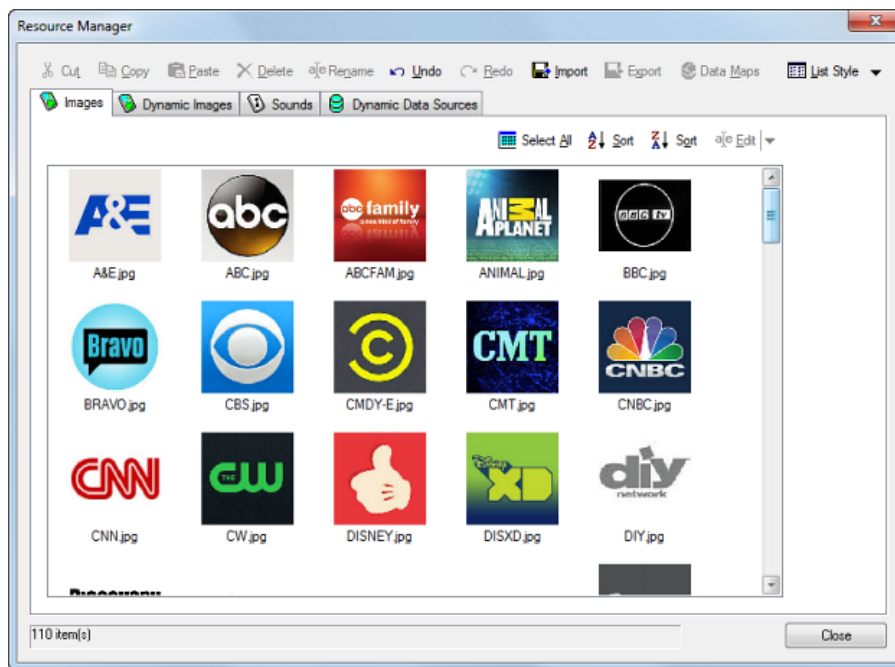
FIG. 147 Resource Manager Images tab - Channel images imported

4. Click **Close** to close the Resource Manager.

NOTE: *The TVGuide.TP5 file in the TV Guide demo has the channel images shown above already imported into the project. These image files are also available in the "Channel images" folder (included in the TV Guide.ZIP file).*

7) Assign a Data Source file to the Listview Button

The data source (channelList.csv) is associated with the Listview button via the *Dynamic Data Source* property (in the *General* tab of the Properties window):

1. With the Listview button selected, click the browse button in the **Dynamic Data Source** (*General*) property to open the *Select Resource* dialog (FIG. 148):
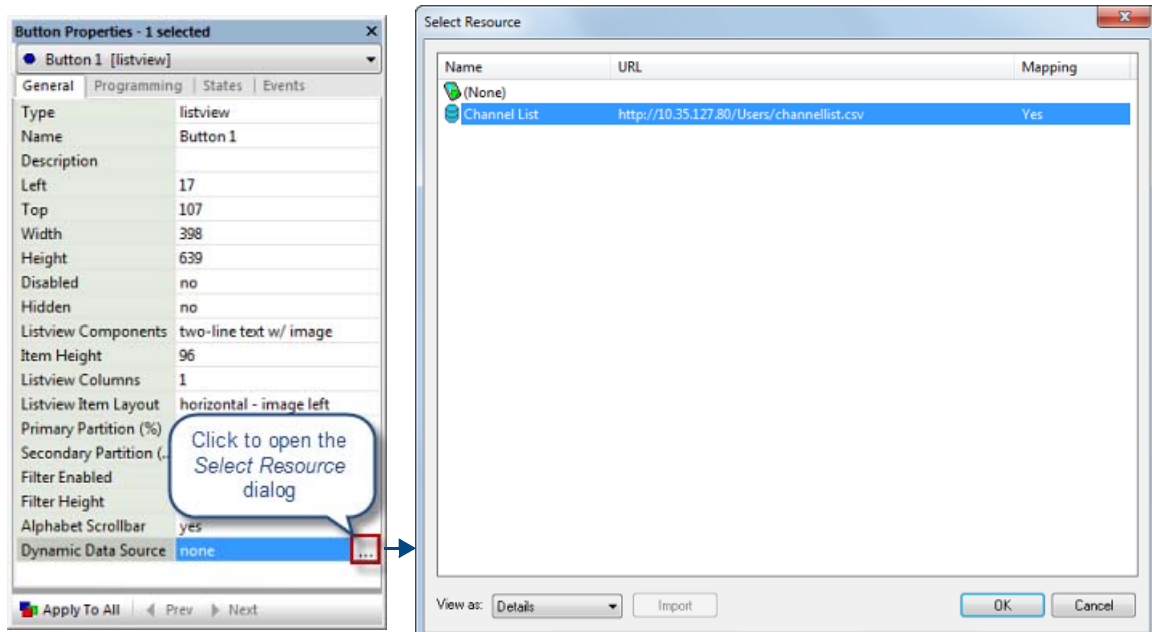
FIG. 148 Dynamic Data Source (General) Property and Select Resource dialog

2. Select the CSV file to use as the data source (in this example, "*channelList.csv*").

3. Click **OK** to close this dialog.

4. The selected Data Source file is indicated in the *Dynamic Data Source* property (see "*Channel List*" in FIG. 149):
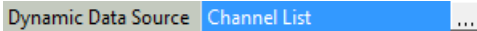
FIG. 149 Dynamic Data Source property indicating "channelList.csv"

NOTE: *The "TVGuide.TP5" file included in the TV Guide demo has "Channel List" already assigned as the Dynamic Data Source for the Listview button.*

8) Write a Custom Event To Respond To User Selection

When the user selects an item on the Listview button, the entire record for that selection is sent to the NX Master. A CUSTOM_EVENT is raised and within this function the desired information can be retrieved for the selection. In this example, the channel number needs to be retrieved. The channel number can then be used to initiate a channel change on the cable box. Note that in the *channelList.csv* file, the channel numbers are listed in the *CHANNEL CODE column* (FIG. 150):

FIG. 150 channelList.csv - CHANNEL CODE column

Listview buttons use the custom event parameter "LISTVIEW_ON_ROW_SELECT_EVENT" to provide the ability to configure a response to the selection of a list item in a Listview button in NetLinx code. This custom event must be added to the NetLinx code on the NX Master.

1.   Use NetLinx Studio 4 to add the following code to the CUSTOM EVENT section of the NetLinx program loaded on the Master:

```
PROGRAM_NAME='TVGuide_CUSTOM_EVENT'
(*********************************************************)
(*********************************************************)
(*  FILE_LAST_MODIFIED_ON: 04/05/2006  AT: 09:00:25       *)
(*********************************************************)
(* System Type : NetLinx                                 *)
(*********************************************************)
(* REV HISTORY:                                          *)
(*********************************************************)
(*
    $History: $
*)
DEFINE_DEVICE
dvTP = 10001:1:0

DEFINE_CONSTANT
// TV Channels Listview button address
INTEGER btnTvListview = 12

DEFINE_VARIABLE

DEFINE_EVENT

// The custom event that is raised whenever a TV listview item is selected on the panel.
// Custom event data has three integers, a data string and an
// encoding string. This is not enough to represent what could
// potentially be a very complex DATA_RECORD. So the listview
// custom event will include a payload ID that can then be used
// to retrieve the contents of the DATA_RECORD.
CUSTOM_EVENT[dvTP,btnTvListview,LISTVIEW_ON_ROW_SELECT_EVENT]
{
  // Variables to hold the ID and type for the payload
  SLONG payloadId
  SLONG payloadType
  // The function to retrieve the payload data takes an array of 1 or more
  // strings that specify which DATA_FIELDs we wish to retrieve. In our
  // example we're interested in 3 fields and 16 characters is long enough
  // to hold the IDs.
  CHAR fields[3][16]
  // Create a DATA_RECORD to hold the retrieved data
  DATA_RECORD record
```

```
    // Get the payload ID from the custom event
    payloadId = custom.value1
    // Get the data type from the custom event
    payloadType = custom.value2
    // Always check for a valid payload ID and check the payload
    // type. Future improvements to the feature may have other
    // payload types.
    if (payloadId > 0 && payloadType == DATA_STRUCTURE_DATARECORD)
    {
        // Specify which DATA_FIELD IDs we want to retrieve from the payload
        fields[1] = 'NAME'
        fields[2] = 'CHANNEL CODE'
        fields[3] = 'ICON'
        // When retrieving the data, always check the return value. If the
        // return value is greater than zero then the DATA_RECORD that was
        // passed in will be populated with the requested DATA_FIELDs.
        if (DATA_GET_EVENT_RECORD(dvTP, payloadId, fields, record) > 0)
        {
            // Put the channel number and name at the bottom of the TV
            // subpage and show the subpage
            SEND_COMMAND dvTP,"'^TXT-13,0,Channel ',record.content[2].value, ' - ',record.content[1].value"
            SEND_COMMAND dvTP,"'^BMX-77,0,',record.content[3].value,',1,10'"
        }
    }
}


DEFINE_PROGRAM


(************************************************************)
(*                      END OF PROGRAM                     *)
(*          DO NOT PUT ANY CODE BELOW THIS COMMENT         *)
(************************************************************)
```

2. Compile the code (select **Build > Compile**).

3. Use NetLinx Studio 4 to transfer the AXS file to the NX Master:
   a. Select **Tools > File Transfer** to open the *File Transfer* dialog.
   b. In the *Send* tab, click the **Add** button. This opens the *Select Files for File Transfer* dialog.
   c. In the *Other* tab, select **Non-System File** and click **Add**.
   d. Select the compiled NetLinx code (in this example, "*ISE_CUSTOM_EVENT.axs*") and click **Open**. This opens the *Enter Device Mapping* dialog.
   e. Review and edit the D:P:S settings for the target NX Master (leave the *Master Directory* field empty), and click **OK** to close the *Enter Device Mapping* dialog and return to the *Select Files for File Transfer* dialog.
   f. Select **OK** to return to the *File Transfer* dialog.
   g. In the *File Transfer* dialog, click **Send** to initiate the file transfer.
   h. The progress of the transfer is indicated in the Output Bar.

NOTE: *The custom event code shown above is included in the NetLinx Studio Workspace file (TV Guide.apw) that is in the TV Guide.ZIP file.*

9) Transfer the TPDesign5 Project to the Touch Panel

At this point, everything is ready to go: the NX Master has the code to handle custom events and the TPD5 project file is handling the data source/mapping for the Listview button. The only thing left to do is to transfer the TPD5 project containing the Listview button, data source reference and image references to the G5 touch panel:

1. In TPDesign5, select **Transfer > Connect** to open the *Connect* dialog (FIG. 151):
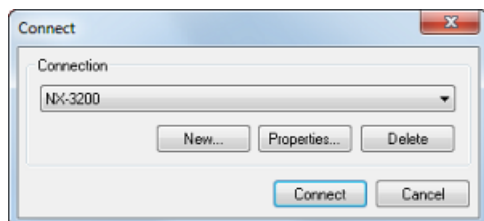
FIG. 151 Connect dialog

> **NOTE:** *If the Master has never been connected to before, a new connection will need to be configured. Refer to the File Transfer Operations section on page 285 for details.*

2. Select the connection configuration for the target NX Master from the *Connection* drop-down list, and click **Connect**. Once a connection has been established with the Master, select **Transfer > Send to Panel** to open the *Send to Panel* dialog (FIG. 152):

FIG. 152 Send To Panel dialog

3.   Click **Send** to begin the file transfer.

When the transfer is complete, the Listview button should appear on the Page it was added to.

Example 1 (CSV File - With Headers) - Results

FIG. 153 shows an example of a Listview button created by following these steps:

channelList.csv

Resource Manager (Images tab) showing channel icon images imported into the TPD5 project

Resource Manager (Dynamic Data Sources tab) showing the Channel List data source imported into theTPD5 project

Data from channelList.csv mapped to Listview components

Resulting Listview button as it is displayed on the panel

FIG. 153 Example Listview button based on "channelList.csv"

Using the *channelList.csv* file as it's data source:

- It displays each channel's name (based on the data in the NAME column) as the *Primary Text* component.
- It displays each channel's rating (based on the data in the RATING column) as the *Secondary Text* component.
- It displays each channel's station icon (based on the data in the ICON column) as the *Image* component.

NOTE: *While the Listview button shown in this example uses only basic design characteristics, note that Listview buttons support most of the same display options as other button types, including Radiant/Gradient fills, Text Effects, Opacity, etc... Use these options to create eye-catching designs, just like for any other button type.*

Reference: "channelList.csv" (CSV File With Headers)

NAME,   CHANNEL CODE, ICON, RATING
A&E, 118, A&E.jpg, PG-13
ABC, 8, ABC.jpg, PG-13
ABCFAM, 180, ABCFAM.jpg, PG-13
ANIMAL, 184, ANIMAL.jpg, PG-13
BBC, 135, BBC.jpg, PG-13
BRAVO, 129, BRAVO.jpg, PG-13
CBS, 11, CBS.jpg, PG-13
CMDY-E, 107, CMDY-E.jpg, PG-13
CMT, 166, CMT.jpg, PG-13
CNBC, 208, CNBC.jpg, PG-13
CNN, 200, CNN.jpg, PG-13
CW, 33, CW.jpg, PG-13
DISNEY, 172, DISNEY.jpg, G
DISXD, 174, DISXD.jpg, PG
DIY, 111, DIY.jpg, G
DSC, 182, DSC.jpg, PG
ENC, 340, ENC.jpg, PG-13
ESPN, 140, ESPN.jpg, PG-13
ESPN2, 144, ESPN2.jpg, PG-13
ESQTV, 191, ESQTV.jpg, PG-13
FOOD, 110, FOOD.jpg, PG
FOX, 205, FOX.jpg, PG-13
FUSE, 164, FUSE.jpg, PG-13
FXM, 384, FXM.jpg, R
FXX, 390, FXX.jpg, R
FYI, 119, FYI.jpg, PG-13
GOLF, 401, GOLF.jpg, G
GSN, 116, GSN.jpg, PG
HALMRK, 185, HALMRK.jpg, R
HBO2e, 301, HBO2e.jpg, R
HBOe, 300, HBOe.jpg, PG
HGTV, 112, HGTV.jpg, PG-13
ID, 192, ID.jpg, PG-13
IFC, 133, IFC.jpg, PG-13
ION, 216, ION.jpg, PG-13
LIF-E, 108, LIF-E.jpg, PG-13
LMN, 109, LMN.jpg, PG-13
MAXe, 315, MAXe.jpg, R
MNT, 27, MNT.jpg, PG-13
msnbc, 209, MSNBC.jpg, PG-13
MTV-E, 160, MTV-E.jpg, R
NBC, 5, NBC.jpg, PG-13
NGC, 186, NGC.jpg, PG-13
NIK, 170, NIK.jpg, PG-13
OWN, 186, OWN.jpg, R
OXYGN, 127, OXYGN.jpg, R
PBS, 13, PBS.jpg, PG-13
QVC, 137, QVC.jpg, G
REELZ, 299, REELZ.jpg, PG-13
SCI, 193, SCI.jpg, R
SHO, 318, SHO.jpg, R
SPIKE, 241, SPIKE.jpg, R
STARZ, 350, STARZ.jpg, PG-13
SUNDe, 358, SUNDe.jpg, PG-13
SYFY, 122, SYFY.jpg, PG-13
TBS, 139, TBS.jpg, PG-13

TCM, 132, TCM.jpg, PG-13

THC, 120, THC.jpg, PG-13

TLC, 183, TLC.jpg, PG-13

TMC, 132, TMC.jpg, PG-13

TNT, 138, TNT.jpg, PG-13

TOON, 176, TOON.jpg, PG-13

TRAVEL, 196, TRAVEL.jpg, PG-13

truTV, 149, TRU.jpg, PG-13

TVLAND, 106, TVLAND.jpg, PG-13

USA-E, 105, USA.jpg, R

VH1, 162, VH1.jpg, R

WE, 128, WE.jpg, PG-13

WGN, 239, WGN.jpg, PG-13

TV Guide Demo File ("TVGuide.ZIP")

Demo (ZIP) files for the Listview examples presented here are available to download from the UI RESOURCE CENTER at www.amx.com. The preceding example followed the *TV Guide* demo. The TV Guide demo ZIP file (**TVGuide.ZIP**) contains the following:

TVGuide.ZIP Contents	
File	Description
channelList.csv	This CSV file (with headers) will be used as the data source file for this example.
TVGuide.TP5	TPDesign5 project file that includes a Listview button pre-configured to use the layout properties and data source file shown in the *Listview Button/Dynamic Data Example 1: CSV File - With Headers example (see page 114)*.
"channel images" folder	This folder contains the images used for the Listview button in this example.
TVGuide.apw	NetLinx Studio 4 Workspace file, with the Listview demo custom event defined. This Workspace contains the following files: • TVGuide_CUSTOM_EVENT.axs • TVGuide_CUSTOM_EVENT.src • TVGuide_CUSTOM_EVENT.tkn • TVGuide_CUSTOM_EVENT.tko

To use this demo:

1. Download the *TVGuide.ZIP* file and extract it's contents to a known location.

2. Launch TPDesign5 and open the *TVGuide.TP5* project file. Use TPDesign5 to set to the Host (IP) address for the data source file:

   a. Open the Resource Manager to the *Dynamic Data Sources* tab, and double-click on the **channelList.csv** file to access the *Edit Dynamic Data Source* dialog.

   b. Edit the **Host** field with the IP address of the NX Master that will host the file. Click **OK** to save changes and close this dialog.

   c. Close the Resource Manager.

   d. Save changes and close the TP5 project.

3. Use NetLinx Studio 4 to transfer the *channelList.csv* data source file to the target NX Master.

   a. Launch NetLinx Studio 4 and select **Tools > File Transfer** to open the *File Transfer* dialog (*Send* tab).

   b. Click **Add** to open the *File Transfer* dialog, and open the *Other* tab.

   c. Select *Non-System File* and click **Add** to access the *Open* dialog.

   d. Locate and select the *channelList.csv* file and click **Open**

   e. In the *Enter Device Mapping Information* dialog, review (and edit if necessary) the mapping information for this file, and click **OK** to return to the *Select File for File Transfer* dialog.

   f. Click **OK** to return to the *File Transfer* dialog. Verify that the *channelList.csv* file is selected for transfer, and click **Send**. Refer to *3) Host the Data Source File (CSV with Headers) on the NX Master* on page 115 for more details.

4. In NetLinx Studio 4, open the *TVGuide.apw* workspace file (**File > Open Workspace)**.

   This Workspace contains NetLinx source code that is pre-configured with a Custom Event for user selection, as well as a TPDesign5 project that includes a pre-configured Listview button that uses *channelList.csv* as it's data source.

5. Build the Workspace: Select **Build > Build Active System**.

6. Transfer all files contained in the Workspace to the target NX Master:

   a. Select **Settings > Active System Communication Settings** to open the *Communication Settings* dialog. Use the options in this dialog to establish a connection to the target NX Master. Note that by default, the workspace is configured to use Serial communication (FIG. 154):
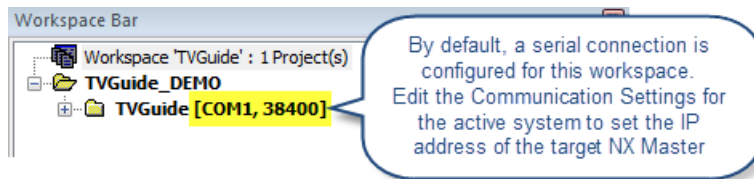
FIG. 154 NetLinx Studio 4 Workspace Bar - TV Guide demo (default communication settings)

This IP address should be the same as was specified for the data source file (see Step 3 above). See NetLinx Studio 4 online help for details on configuring communication settings.

b. Select **Tools > File Transfer** to open the *File Transfer* dialog (*Send* tab). Remove any files (from previous transfer operations) that may be in the list.

c. Click **Add** to open the *Select Files for File Transfer* dialog (*Current Workspace* tab).

d. Click the top-level *Projects* directory to auto-select all files in the Workspace.

e. Verify that the IP address indicated here indicates the correct NX Master, and click **OK** to save changes and return to the *File Transfer* dialog.

f. In the *File Transfer* dialog, click **Send** to transfer the Workspace files to the target NX Master.

Listview Button/Dynamic Data Example 2: CSV File - No Headers

The following instructions describe using the Conference Rooms demo for creating a Listview button with a dynamic data source in the form of a CSV file *without* headers.

NOTE: *This set of instructions uses files that are included in the "Conference Rooms.ZIP" demo file which is available to download from the UI RESOURCE CENTER at www.amx.com.*

The resulting Listview button will display a listing of Conference rooms with each room's name, phone number and room icons (FIG. 155):
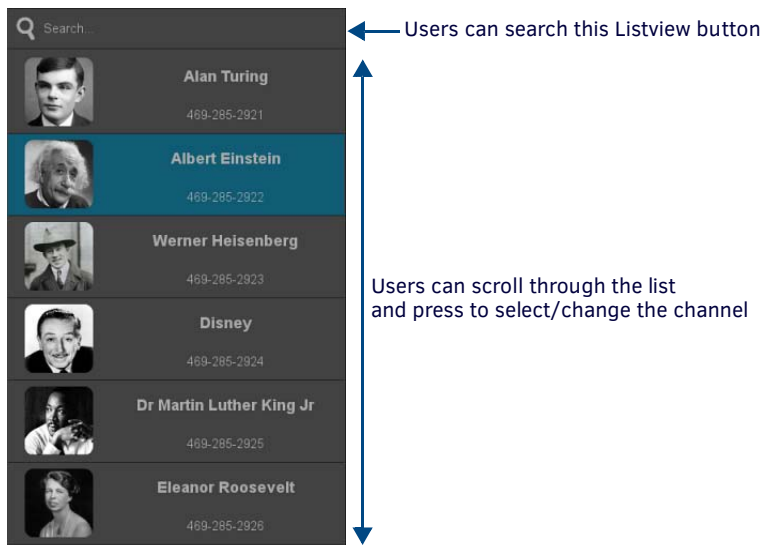
FIG. 155 Example - Listview button based on "conference.csv"

Before You Begin

1. Download the *Conference.ZIP f*ile from the UI RESOURCE CENTER at www.amx.com and extract it's contents to a known location.

2. Open the *conference.csv* file and analyze it's contents. It is a relatively simple csv file that consists of three columns without headers. See page 140 to view this file.

1) Create (draw) a Listview button

1. In TPDesign5, open a Page and use the Button Draw tool to create a new button.

2. With the new button selected, click the **Type** (General) property and select **Listview** from the drop-down of button types. This selection sets the new button as a Listview button, and enables a set of Listview-specific properties (FIG. 156):

Type (General) property set to Listview   Example Listview button
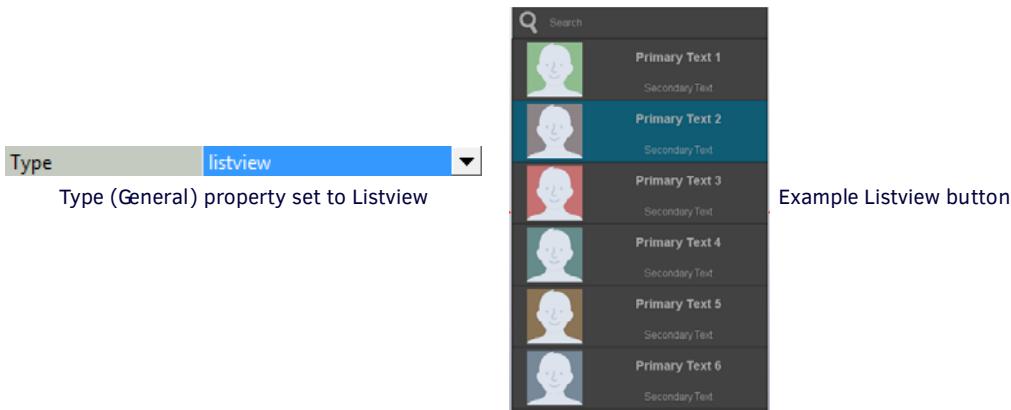
FIG. 156 Type (General) Property set to Listview

NOTE: *The "CSV.TP5" file included in the Conference Rooms demo has a Listview button already drawn on the "Main" page.*

2) Review the Listview Button Properties

Use the options in the Properties window to view/edit the *General*, *Programming* and *States* properties for the Listview button to match the settings shown in FIG. 157:

General Properties   Programming Properties   States Properties
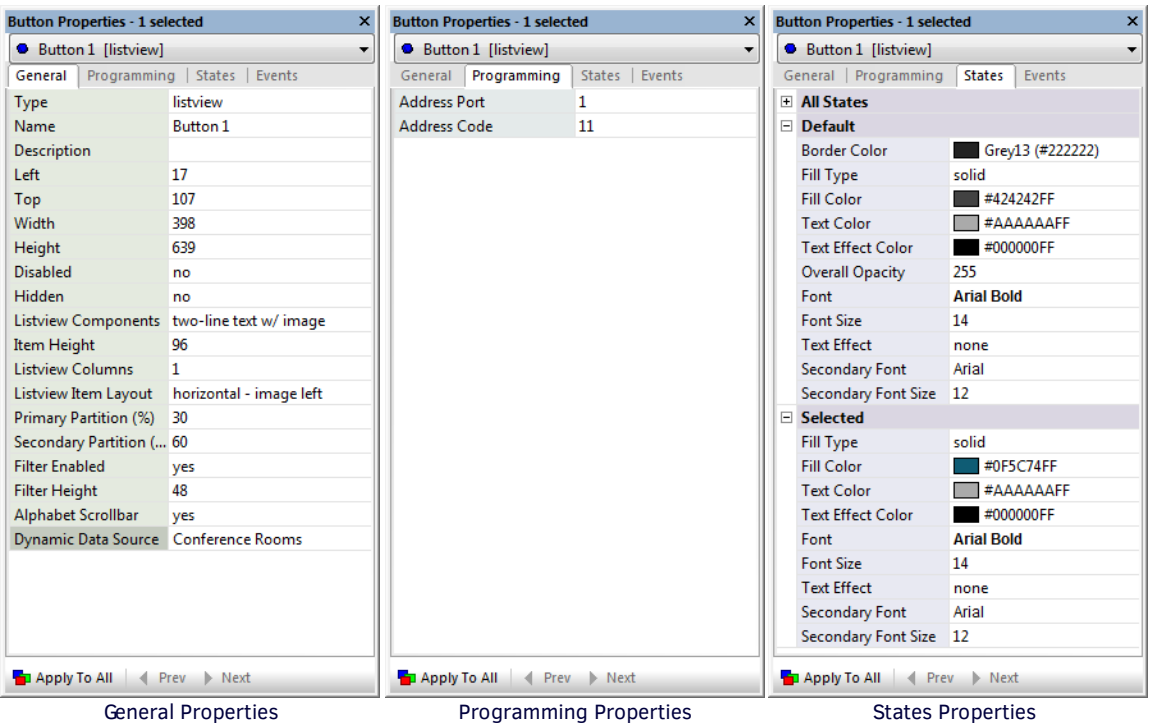
FIG. 157 Properties for the Conference Rooms Listview Button

NOTE: *The Listview button in the Conference Rooms demo is pre-configured with the General, Programming and States properties shown above.*

Refer to the *Working With Listview Button Properties* section on page 107 for details on Listview-specific button properties.

3) Host a Data Source File (CSV without Headers) on the NX Master

In this example, "conference.csv" will be the data source for the Listview button. This CSV file will be hosted on the NX Master. The "conference.csv" file contains a listing of conference rooms with phone numbers and room icons that will be presented on the Listview button. FIG. 158 presents a sample of the first few rows of this file. Refer to page 140 to view the entire file.

	A	B	C
1	Alan Turing	TURING.jpg	469-285-2921
2	Albert Einstein	EINSTEIN.jpg	469-285-2922
3	Werner Heisenberg	HEISENBERG.jpg	469-285-2923
4	Disney	WALTDISNEY.jpg	469-285-2924
5	Dr Martin Luther King	MARTINLUTHER.jpg	469-285-2925
6	Eleanor Roosevelt	ELEANOR.jpg	469-285-2926
7	Emmeline Pankhurst	PANKHURST.jpg	469-285-2927
8	Frank Sinatra	FRANKSINATRA.jpg	469-285-2928
9	Henry Ford	HENRYFORD.jpg	469-285-2929
10	Jackie Robinson	JACKIEROBINSON.jpg	469-285-2930
11	Jean Piaget	PIAGET.jpg	469-285-2931
12	Margaret Thatcher	MARGARETTHATCHER.i	469-285-2932

conference.csv has three columns with no headers.
Column1 contains the room names
Column 2 contains the room icons
Column 3 contains the room phone numbers

FIG. 158 Data Source File - "conference.csv" (CSV file without headers)

To host a CSV file on the NX Master:

1. In NetLinx Studio 4, establish communication with the Master (refer to NetLinx Studio 4 online help for details).

2. Select **Tools > File Transfer** to open the *File Transfer* dialog (FIG. 159):

Click to open the *Select File for Transfer* dialog
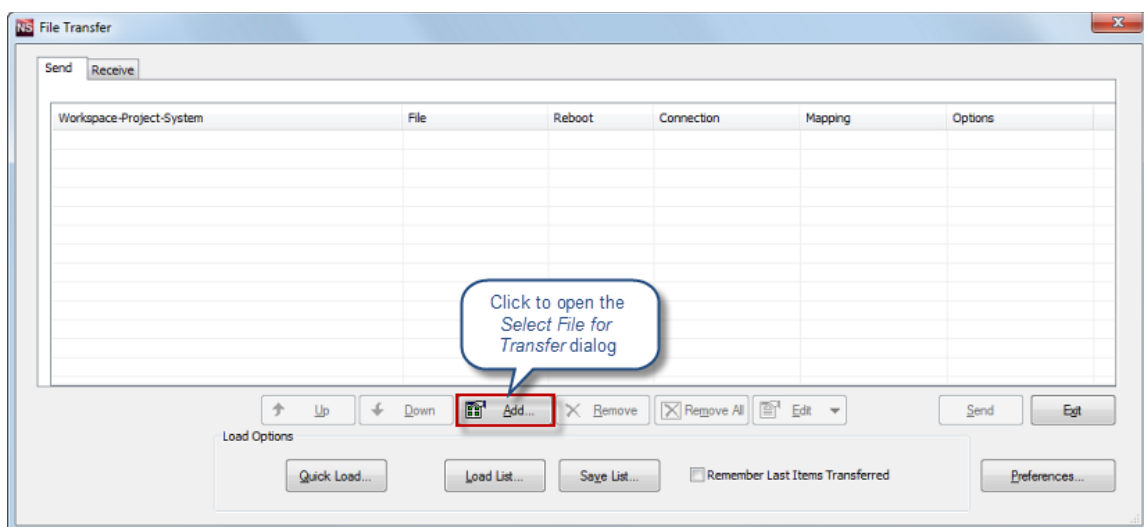
FIG. 159 NetLinx Studio 4 - File Transfer dialog

3. Click **Add** to open the *Select Files for File Transfer* dialog, and open the *Other* tab (FIG. 160):

Click to locate and the data source file (in this example, "*conference.csv*")
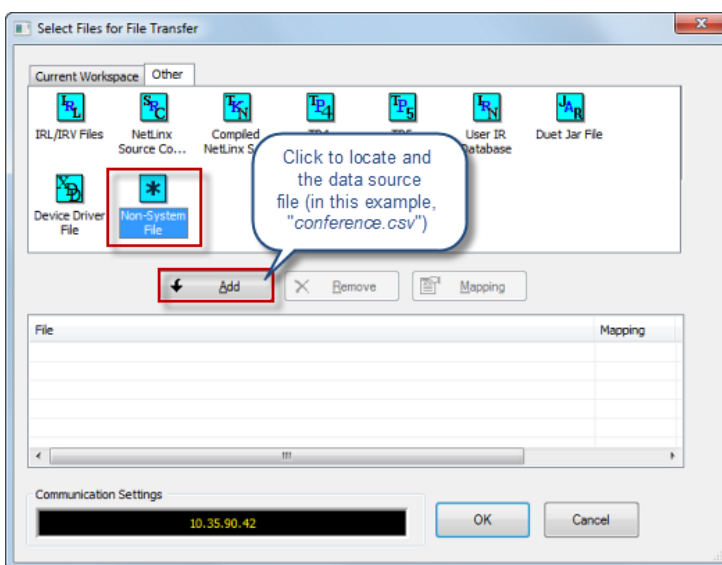
FIG. 160 NetLinx Studio 4 - Select Files for File Transfer dialog

4. Select **Non-System File**, then click **Add**.

5. In the *Open* dialog, locate and select the "*conference.csv*" file and click **Open** to access the *Enter Device Mapping Information* dialog (FIG. 161).
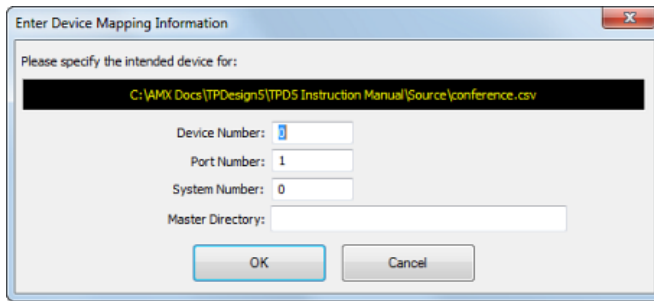
FIG. 161 NetLinx Studio 4 - Enter Device Mapping Information dialog

    a. Enter the *Device*, *Port* and *System* Number for the target NX Master.

    b. In the *Master Directory* field, enter the name of the directory on the NX Master that contains the data source file.

    NOTE: *If no directory is specified in the Master Directory field, the file will be copied to the root directory on the Master.*

    c. Click **OK** to save changes and close the *Enter Device Mapping Information* dialog (and return to the *Select Files For File Transfer* dialog).

6. In the *Select Files For File Transfer* dialog, the selected file and it's device information are indicated in the *Files* list (FIG. 162):
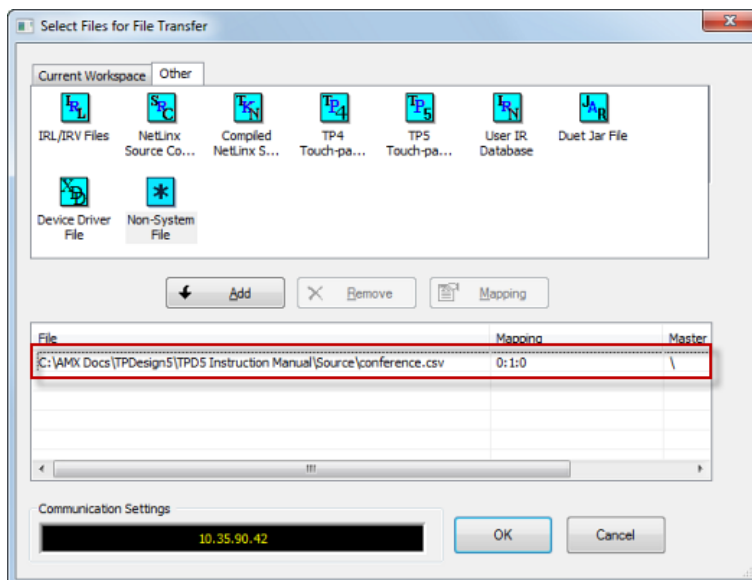
FIG. 162 NetLinx Studio5 - Select Files for File Transfer dialog indicating a CSV file for transfer

7. Click **OK** to close this dialog and return to the *File Transfer* dialog.

8. Click **Send** to initiate the file transfer. The program will indicate when the transfer is complete.

4) Add the Dynamic Data Source to the Project

To add the data source file (chanelList.csv) to the TPDesign5 project:

1. Open the Resource Manager to the *Dynamic Data Sources* tab and click **New** to open the *Create Dynamic Data Source* dialog (FIG. 163):
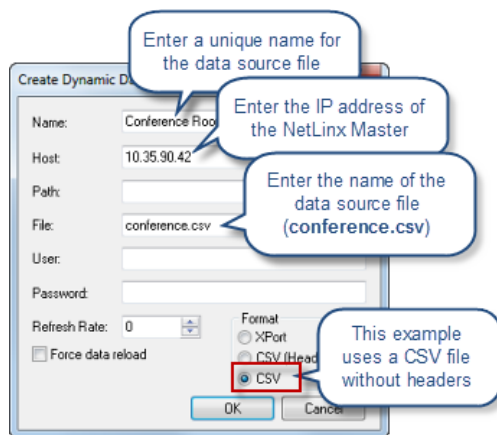
FIG. 163 Create Dynamic Data Source dialog with Example data (conference.csv)

2. In the *Name* field, enter a unique friendly name for the data source. For this example, enter "**Conference Rooms**".

3. In the *Host* field, enter the host name, which must be a fully qualified DNS or IP address.

4. In the *File* field, enter a file name that indicates the full path to the location of the source file.

5. In the *User* field, enter the user name required by the NX Master or server for authentication (if required).

6. In the *Password* field, enter the password required by the NX Master or server for authentication (if required).

7. In the *Refresh Rate* field, use the up/down arrows to adjust the number of seconds between refreshes in which the resource is downloaded again. Refreshing resources will cause the button displaying that resource to refresh as well. The default value is 0, which means that the resource is only downloaded once.

8. Under *Format*, select **CSV**, since the data source file in this example (conference.csv) uses a CSV file without headers.

9. Click **OK** to save changes and close this dialog. The new data source is indicated in the Resource Manager - Dynamic Data Sources tab (FIG. 164):
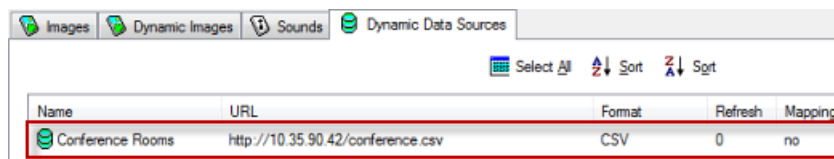
FIG. 164 Resource Manager - Dynamic Data Sources tab indicating "conference.csv" as the data source

NOTE: *The Listview button in the CSV.TP5 file is pre-configured to use Conference Rooms (conference.csv) as it's data source file. However, it is necessary to update the Host address with the IP address of your NX Master as shown above. Double-click on Conference Rooms in the Resource Manager to open the Edit Dynamic Data Source dialog and update accordingly.*

5) Map the Data from the Data Source File to the Listview Button Components

It is necessary to map the data in the *conference.csv* file to the three fields that comprise the Listview button layout. These three fields (called Components in TPDesign5) are: *Primary Text, Secondary Text* and *Image* (FIG. 165):
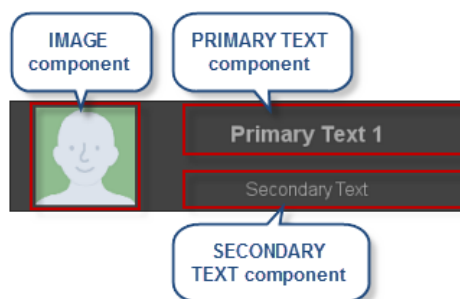
FIG. 165 Listview Button - Components

Step One: Analyze the Data Source

It is necessary to understand the contents of the data source file in order to map the data to the Components in the Listview button. In this example, the *conference.csv* file contains three columns with no headers.

Note that Column #1 (A) contains room names, Column #2 (B) contains room icons, and Column #3 (C) contains room phone numbers   (FIG. 166):

FIG. 166 Understanding the contents of the data source file - channelList.csv

In this example:

- The items in **Column #1** will be mapped to display as the *Primary Text* component of the list items in the Listview button.
- The items in **Column #2** will be mapped to display as the *Image* component of the list items in the Listview button.
- The items in **Column #3** will be mapped to display as the *Secondary Text* component of the list items in the Listview button.

Step Two: Map the Data to Components of the Listview button

1. With the Listview button selected, open the Resource Manager to the *Dynamic Data Sources* tab.

2. Select the data source (*conference.csv*) that is assigned to the Listview button (as described on page 131):
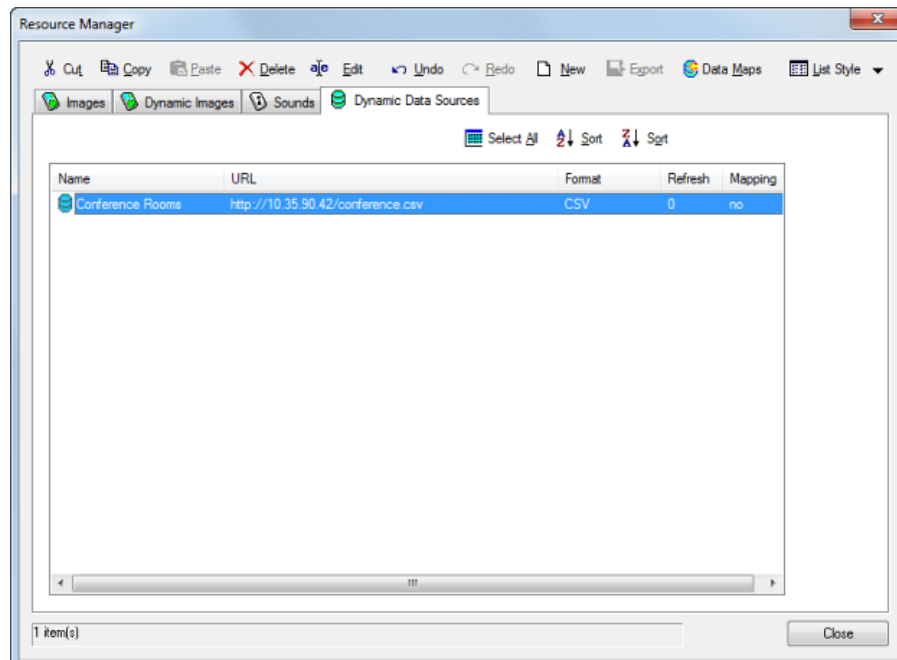
FIG. 167 Resource Manager - Dynamic Data Source tab

3. Click the **Data Maps** button to access the *Dynamic Data Mappings - Listview Buttons* dialog (FIG. 168):
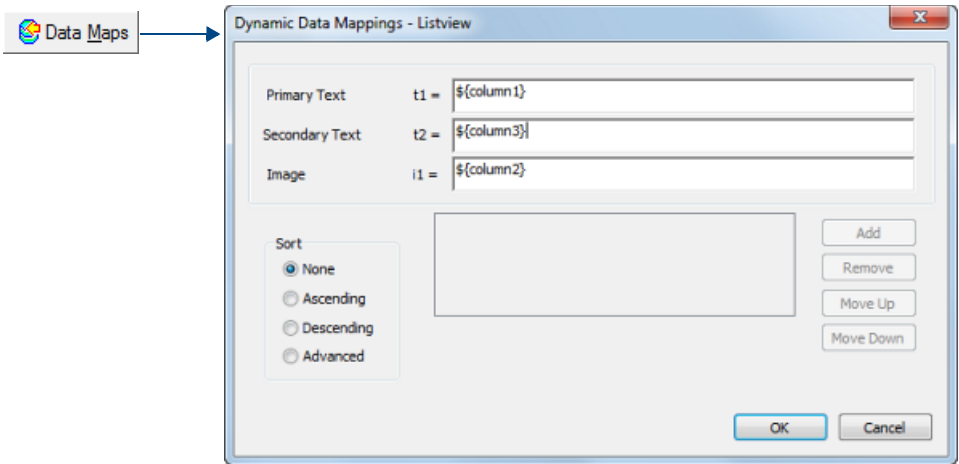
FIG. 168 Dynamic Data Mappings - Listview dialog (with example data indicated)

4. Use the fields in this dialog to specify the device mapping for the selected Listview button and the selected Data Source (see *Dynamic Data Mappings - Syntax Requirements (CSV with Headers)* below).

NOTE: *The Listview button in the Conference Rooms demo is pre-configured with the data mapping settings shown above.*

Dynamic Data Mappings - Syntax Requirements (CSV Without Headers)

Note that the syntax requirements for these fields depends on the type of file used as the data source. The data source file in this example uses a CSV file without headers. In the absence of headers, the columns will be named by default as: column1, column2, column3... The syntax requirements for data mapping to a CSV without headers is described below:

Dynamic Data Mappings - Syntax Requirements (CSV Without Headers)
Primary Text: For CSV files without headers, the syntax is: $\{column\#\}$ Following this syntax, enter the column # in the data source file to be displayed as the Primary Text component of the Listview button. In this example, conference.csv lists room names in Column #1. To display the contents of Column #1 as the Primary Text component, enter **${column1}** in the *Primary Text* field:
Secondary Text: For CSV files without headers, the syntax is: $\{column\#\}$ Following this syntax, enter the column # in the data source file to be displayed as the Secondary Text component of the Listview button. In this example, conference.csv lists room phone numbers in Column #3. To display the contents of Column #3 as the Secondary Text component, enter **${column3}** in the *Secondary Text* field:
Image: For CSV files without headers, the syntax is $\{column\#\}$ Following this syntax, enter the column # in the data source file to be displayed as the Image component of the Listview button. In this example, conference.csv lists room icons in Column #2. To display the contents of Column #2 as the Image component, enter **${column2}** in the *Image* field:

NOTE: *These fields in the Dynamic Data Mappings - Listview Buttons dialog are case-sensitive.*

6) Add Image Files to the Project

In the data source file for this example (conference.csv), column #2 lists image files associated with each conference room in the list (FIG. 169):

FIG. 169 Data Source File - "conference.csv"

In order to display these image files on the Listview button, the image files named in the data source file must be added to the project, via the Resource Manager - Images tab:

1. Open the Resource Manager to the *Images* tab.

2. Click **Import** to access the *Open* dialog. Locate and select all of the image files that are named in the data source file (conference.csv).

3. Click **OK** to import the selected files and return to the Resource Manager (FIG. 170):

FIG. 170 Resource Manager Images tab - Conference Room images imported

4. Click **Close** to close the Resource Manager.

NOTE: *The CSV.TP5 file in the Conference Rooms demo has the channel images shown above already imported into the project. These image files are also available in the "conference images" folder (included in the Conference Rooms.ZIP file).*

7) Assign a Data Source file to the Listview Button

The data source (conference.csv) is associated with the Listview button via the *Dynamic Data Source* property (in the *General* tab of the Properties window):

1. With the Listview button selected, click the browse button in the **Dynamic Data Source** (General) property to open the *Select Resource* dialog (FIG. 171):
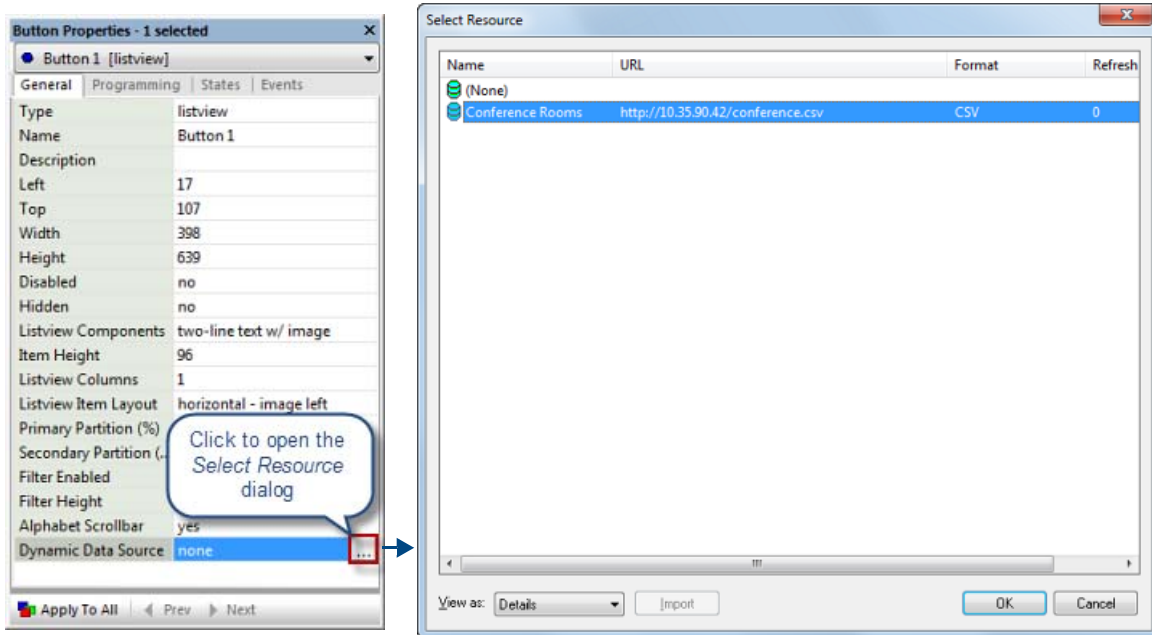
FIG. 171 Dynamic Data Source (General) Property and Select Resource dialog

2. Select the CSV file to use as the data source (in this example, "*conference.csv*").

3. Click **OK** to close this dialog.

4. The selected Data Source file is indicated in the *Dynamic Data Source* property (see "*Conference Rooms*" in FIG. 172):
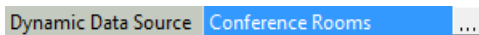
FIG. 172 Dynamic Data Source property indicating "conference.csv"

NOTE: *The "CSV.TP5" file included in the Conference Rooms demo has "Conference Rooms" already assigned as the Dynamic Data Source for the Listview button.*

8) Write a Custom Event To Respond To User Selection

When the user selects an item on the Listview button, the entire record for that selection is sent to the NX Master. A CUSTOM_EVENT is raised and within this function the desired information can be retrieved for the selection. In this example, the phone number needs to be retrieved. The phone number can then be used to initiate a call to the associated room. Note that in the *conference.csv* file, the phone numbers are listed in Column #3 (FIG. 173):

FIG. 173 conference.csv - Column #3 (phone numbers)

Listview buttons use the custom event parameter "LISTVIEW_ON_ROW_SELECT_EVENT" to provide the ability to configure a response to the selection of a list item in a Listview button in NetLinx code. This custom event must be added to the NetLinx code on the NX Master.

1. Use NetLinx Studio 4 to add the following code to the CUSTOM EVENT section of the NetLinx program loaded on the Master:

```
PROGRAM_NAME='ISE_CUSTOM_EVENT'
(***********************************************************)
(***********************************************************)
(*  FILE_LAST_MODIFIED_ON: 04/05/2006  AT: 09:00:25        *)
(***********************************************************)
(* System Type : NetLinx                                   *)
(***********************************************************)
(* REV HISTORY:                                            *)
(***********************************************************)
(*
     $History: $
*)
```

```
DEFINE_DEVICE
dvTP = 10001:1:0


DEFINE_CONSTANT
    // CONTACTS Listview button address
INTEGER btnListview = 11

DEFINE_VARIABLE

DEFINE_EVENT

    // The custom event that is raised whenever a contact on
    // the listview item is selected on the panel
CUSTOM_EVENT[dvTP,btnListview,LISTVIEW_ON_ROW_SELECT_EVENT]
{
    SLONG payloadId
    SLONG payloadType
    CHAR fields[3][16]
    CHAR name[DATA_MAX_VALUE_LENGTH]
    CHAR number[DATA_MAX_VALUE_LENGTH]
    CHAR image[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] = 'column1'
    fields[2] = 'column3'
    fields[3] = 'column2'

        // 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
            image = record.content[3].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,"'^BMX-52,0,',image,',1,10'"
            SEND_COMMAND dvTP,"'^PPN-Calling'"
            SEND_COMMAND dvTP, "'^PPT-Calling;50'"
        }
    }
}


DEFINE_PROGRAM

(*************************************************************)
(*                    END OF PROGRAM                     *)
(*        DO NOT PUT ANY CODE BELOW THIS COMMENT         *)
(*************************************************************)
```

2. Use NetLinx Studio 4 to compile the code (select **Build > Compile**).

3. Use NetLinx Studio 4 to transfer the AXS file to the NX Master:

   a. Select **Tools > File Transfer** to open the *File Transfer* dialog.

   b. In the *Send* tab, click the **Add** button. This opens the *Select Files for File Transfer* dialog.

   c. In the *Other* tab, select **Non-System File** and click **Add**.

   d. Select the compiled NetLinx code (in this example, "*ISE_CUSTOM_EVENT.axs*") and click **Open**. This opens the *Enter Device Mapping* dialog.

   e. Review and edit the D:P:S settings for the target NX Master (leave the *Master Directory* field empty), and click **OK** to close the *Enter Device Mapping* dialog and return to the *Select Files for File Transfer* dialog.

   f. Select **OK** to return to the *File Transfer* dialog.

   g. In the *File Transfer* dialog, click **Send** to initiate the file transfer.

h. The progress of the transfer is indicated in the Output Bar.

NOTE: *The custom event code shown above is included in the NetLinx Studio Workspace file (CSV.apw) that is in the Conference Rooms.ZIP file.*

9) Transfer the TPDesign5 Project to the Touch Panel

At this point, everything is ready to go: the NX Master has the code to handle custom events and the TPD5 project file is handling the data source/mapping for the Listview button.

The only thing left to do is to transfer the TPD5 project containing the Listview button, data source reference and image references to the G5 touch panel:

1.   In TPDesign5, select **Transfer > Connect** to open the *Connect* dialog (FIG. 174):
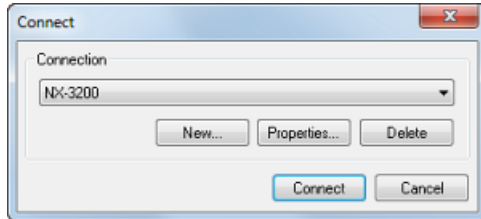
FIG. 174 Connect dialog

> **NOTE:** *If the Master has never been connected to before, a new connection will need to be configured. Refer to the File Transfer Operations* section on page 285 *for details.*

2.   Select the connection configuration for the target NX Master from the *Connection* drop-down list, and click **Connect**.

Once a connection has been established with the Master, select **Transfer > Send to Panel** to open the *Send to Panel* dialog (FIG. 175):
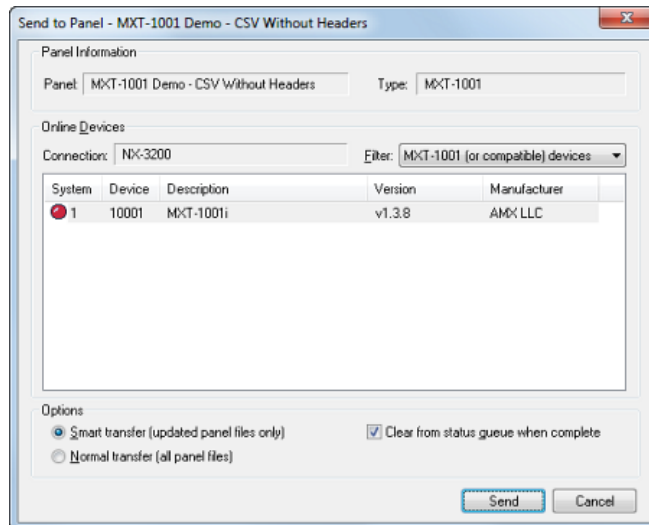
FIG. 175 Send To Panel dialog

3.   Click **Send** to begin the file transfer.

When the transfer is complete, the Listview button should appear on the Page it was added to.

Example 2 (CSV File - No Headers) - Results

FIG. 176 shows an example of a basic Listview button created by following these steps:

 conference.csv

Resource Manager (Images tab) showing room icon images imported into the TPD5 project

Resource Manager (Dynamic Data Sources tab) showing the Conference Rooms data source imported into theTPD5 project

Data from conference.csv mapped to Listview components

Resulting Listview button as it is displayed on the panel

FIG. 176 Example Listview button based on "conference.csv"

Using the *conference.csv* file as it's data source:

- It displays each room's name (based on the data in Column #1) as the *Primary Text* component.
- It displays each room's phone number (based on the data in Column #3) as the *Secondary Text* component.
- It displays each room's icon (based on the data in Column #2) as the *Image* component.

NOTE: *While the Listview button shown in this example uses only basic design characteristics, note that Listview buttons support most of the same display options as other button types, including Radiant/Gradient fills, Text Effects, Opacity, etc... Use these options to create eye-catching designs, just like for any other button type.*

Reference: "conference.csv" (CSV File Without Headers)

Alan Turing, TURING.jpg, 469-285-2921

Albert Einstein, EINSTEIN.jpg, 469-285-2922

Werner Heisenberg, HEISENBERG.jpg, 469-285-2923

Disney, WALTDISNEY.jpg, 469-285-2924

Dr Martin Luther King Jr, MARTINLUTHER.jpg, 469-285-2925

Eleanor Roosevelt, ELEANOR.jpg, 469-285-2926

Emmeline Pankhurst, PANKHURST.jpg, 469-285-2927

Frank Sinatra, FRANKSINATRA.jpg, 469-285-2928

Henry Ford, HENRYFORD.jpg, 469-285-2929

Jackie Robinson, JACKIEROBINSON.jpg, 469-285-2930

Jean Piaget, PIAGET.jpg, 469-285-2931

Margaret Thatcher, MARGARETTHATCHER.jpg, 469-285-2932

Muhammad Ali, MUHAMMADALI.jpg, 469-285-2933

Nelson Mandela, NELSONMANDELA.jpg, 469-285-2934

Rachel Carson, RACHELCARLSON.jpg, 469-285-2935

Scott Miller, MILLER.jpg, 469-285-2936

Sigmund Freud, SIGMUNDFREUD.jpg, 469-285-2937

Teddy Roosevelt, THEODOREROOSEVELT.jpg, 469-285-2938

The Beatles, THEBEATLES.jpg, 469-285-2939

TS Eliot, TSELIOT.jpg, 469-285-2940

Warren Buffet, WARRENBUFFET.jpg, 469-285-2941

Winston Churchill, CHURCHILL.jpg, 469-285-2942

Edwin Hubble, EDWINHUBBLE.jpg, 469-285-2943

Conference Rooms Demo File ("Conference.ZIP")

Demo (ZIP) files for the Listview examples presented here are available to download from the UI RESOURCE CENTER at www.amx.com. The preceding example followed the *Conference Rooms* demo. The Conference Rooms demo ZIP file (**Conference.ZIP**) contains the following:

Conference.ZIP Contents	
File	Description
CSV.TP5	TPDesign5 project file with the Listview button configured according to this example.
"conference images" folder	This folder contains the images used for the Listview button in this example.
CSV.apw	NetLinx Studio 4 Workspace file, with the Listview demo custom event defined. This Workspace contains the following files: • CSV_CUSTOM_EVENT.axs • CSV_CUSTOM_EVENT.src • CSV_CUSTOM_EVENT.tkn • CSV_CUSTOM_EVENT.tko
conference.csv	This CSV file (with no headers) will be used as the data source file for this example.

To use this demo:

1. Download the *Conference.ZIP* file and extract it's contents to a known location.
2. Launch TPDesign5 and open the *CSV.TP5* project file. Use TPDesign5 to set to the Host (IP) address for the data source file:
    a. Open the Resource Manager to the *Dynamic Data Sources* tab, and double-click on the **conference.csv** file to access the *Edit Dynamic Data Source* dialog.
    b. Edit the **Host** field with the IP address of the NX Master that will host the file. Click **OK** to save changes and close this dialog.
    c. Close the Resource Manager.
    d. Save changes and close the TP5 project.
3. Use NetLinx Studio 4 to transfer the *conference.csv* data source file to the target NX Master, so that it will be hosted on the Master:
    a. Launch NetLinx Studio 4 and select **Tools > File Transfer** to open the *File Transfer* dialog (*Send* tab).
    b. Click **Add** to open the *File Transfer* dialog, and open the **Other** tab.
    c. Select *Non-System File* and click **Add** to access the *Open* dialog.
    d. Locate and select the *channelList.csv* file and click **Open**
    e. In the *Enter Device Mapping Information* dialog, review (and edit if necessary) the mapping information for this file, and click **OK** to return to the *Select File for File Transfer* dialog.
    f. Click **OK** to return to the *File Transfer* dialog. Verify that the *conference.csv* file is selected for transfer, and click **Send**.
        Refer to *3) Host a Data Source File (CSV without Headers) on the NX Master* on page 129 for more details.

4. In NetLinx Studio 4, open the *CSV.apw* workspace file (**File > Open Workspace)**.

   This Workspace contains NetLinx source code that is pre-configured with a Custom Event for user selection, as well as a TPDesign5 project that includes a pre-configured Listview button that uses *conference.csv* as it's data source.

5. Build the Workspace: Select **Build > Build Active System**.

6. Transfer all files contained in the Workspace to the target NX Master:

   a. Select **Settings > Active System Communication Settings** to open the *Communication Settings* dialog. Use the options in this dialog to establish a connection to the target NX Master. Note that by default, the workspace is configured to use Serial communication (FIG. 177):
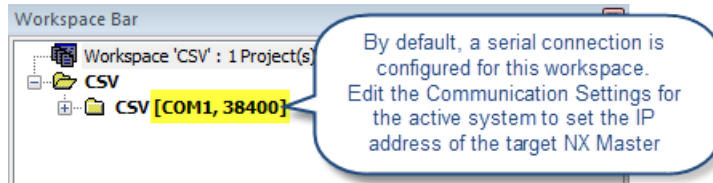
FIG. 177 NetLinx Studio 4 Workspace Bar - Conference demo (default communication settings)

   This IP address should be the same as was specified for the data source file (see Step 3 above). See NetLinx Studio 4 online help for details on configuring communication settings.

   b. Select **Tools > File Transfer** to open the *File Transfer* dialog (*Send* tab). Remove any files (from previous transfer operations) that may be in the list.

   c. Click **Add** to open the *Select Files for File Transfer* dialog (*Current Workspace* tab).

   d. Click the top-level *Projects* directory to auto-select all files in the Workspace.

   e. Verify that the IP address indicated here indicates the correct NX Master, and click **OK** to save changes and return to the *File Transfer* dialog.

   f. In the *File Transfer* dialog, click **Send** to transfer the Workspace files to the target NX Master.

Listview Button/Dynamic Data Example 3: XML File/XPort Server

The following instructions describe using the XML File/XPort Server demo for creating Listview buttons with a dynamic data source in the form of a Xport-generated XML file that is hosted on an NX Master.

For use as data source files for Listview buttons, XML files must be in the format that is exported by XPort servers - this represents the "AMX Standard" XML format.

NOTE: *Listview buttons do not support grouped data. If data is grouped, it cannot be displayed on a G5 Listview using the amxstandard.xml XPort output format.*

NOTE: *This set of instructions uses files that are included in the "Twitter.ZIP" demo file which is available to download from the UI RESOURCE CENTER at www.amx.com.*

The resulting Listview button will display a listing of messages from the "AMX Talk" Twitter account, with the Twitter user name (*@AMX Talk*), the text of each message in the feed, and an image if one is associated with the message (FIG. 178):
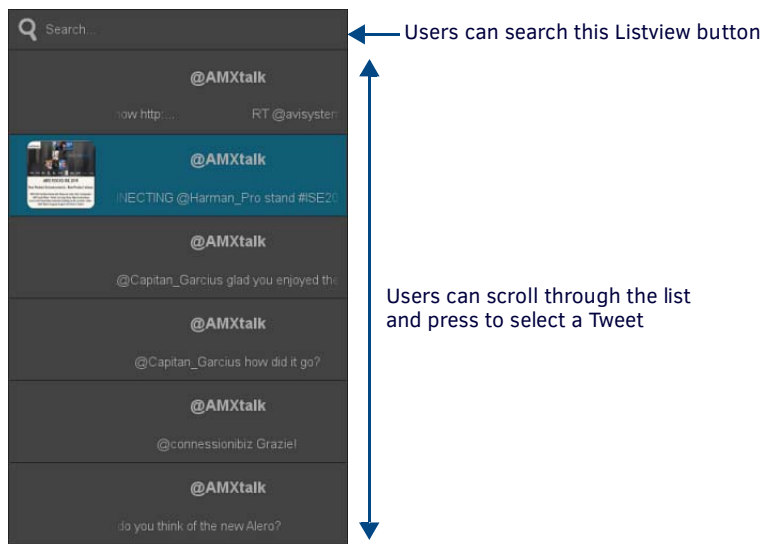
FIG. 178 Example - Listview button based on "amxstandard.xml" (AMX Talk Twitter account)

Before You Begin

Download the *Twitter.ZIP* file from the UI RESOURCE CENTER at www.amx.com and extract it's contents to a known location.

1) Create Twitter Feed on the XPort Server

The XML file that will serve as the data source file in this example is generated by an XPort server. The following instructions describe generating an XML file for a Twitter feed from the XPort server:

NOTE: *In order to create a new Twitter feed on the XPort server, a valid Twitter user account is required.*

Use a web browser to access the XPort server's configuration (Home) page: Enter the XPort server's IP address in the address bar, followed by "**/xport/**" (for example, "**10.35.90.45/xport/**").

1. In the XPort Server's Dashboard page under *External Feeds*, hover the cursor over the **Twitter** configuration icon to access the **[+] Create New Feed** option (FIG. 179):
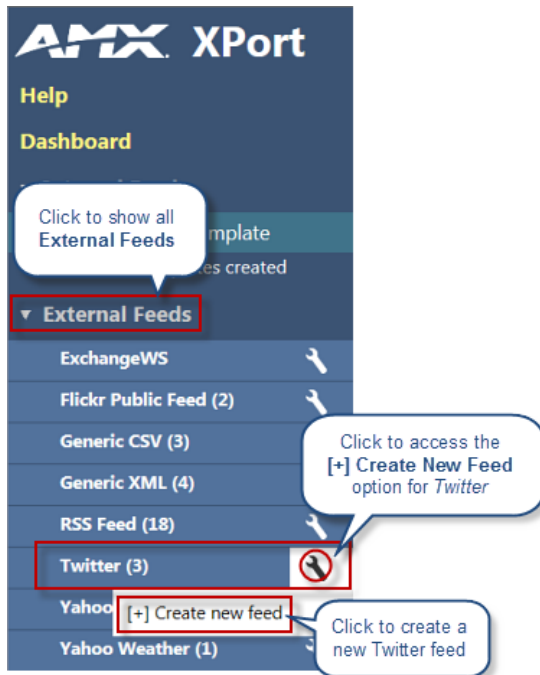
FIG. 179 XPort Dashboard page > External Feeds > Twitter [+] Create New Feed

2. Click on **[+] Create New Feed** to open the *Create a New Feed* page. This example will create a feed named "*AMXTalk*" for a Twitter user named "*@amxtalk*". Fill in the options on this page as shown in FIG. 180:

FIG. 180 Create New Feed page indicating the AMXTalk twitter feed

   a. In the *Name* field, enter "**AMXTalk**".
   b. In the *Description* field, enter "**amx**".
   c. Under *Feed Type*, select **User Tweets**.
   d. Under *User Name*, enter "**@amxtalk**".
   e. Under *Refresh Interval*, select **5 minutes**.

3. Click **Authorize** to grant access to an active Twitter account via Twitter.com (FIG. 181):
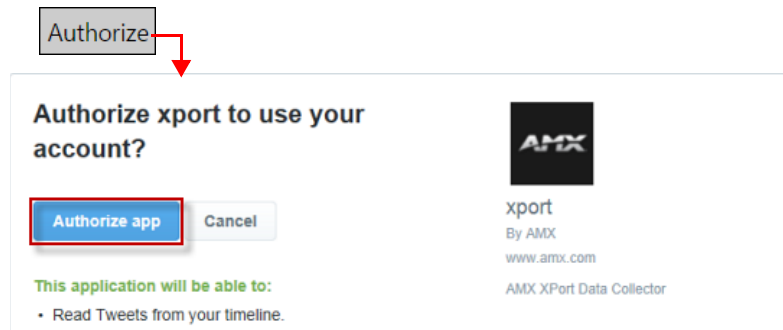
FIG. 181 Twitter.com - Authorize app page

4. In the *Create New Feed* page, click **Create Feed**. Xport will indicate that a new feed is being prepared (FIG. 182):
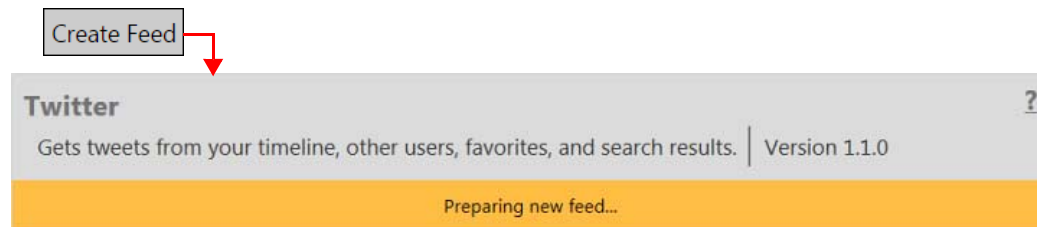
FIG. 182 XPort - Creating new Twitter feed

5. When the feed is ready, the Twitter feed page presents several **Output Feed** options (FIG. 183):
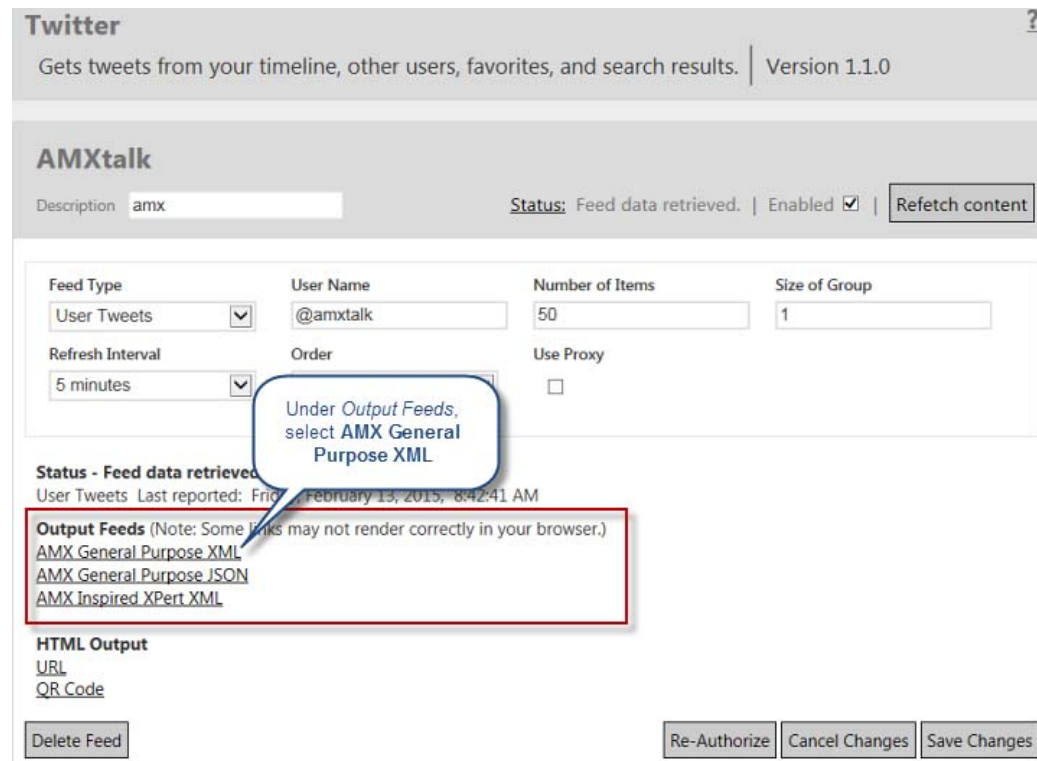
FIG. 183 XPort - Twitter (AMXTalk) feed page

2) Generate the "amxstandard.xml" file

In the Twitter (AMXTalk) feed page, under *Output Feeds*, click on the **AMX General Purpose XML** link to view the XML. The first few lines of the *amxstandard.xml* file are shown below (FIG. 184):

```
▼<data timestamp="Fri, 13 Feb 2015 10:52:43 GMT">
  ▼<feed>
      <name>AMXtalk</name>
      <description>amx</description>
      <source/>
      <lastupdate>2015-02-13 10:52:43Z</lastupdate>
    ▼<recordsets count="1">
      ▼<recordset id="AMXtalk">
        ▼<records count="50">
          ▼<record>
            ▼<metadata>
                <field id="timestamp" type="datetime" format="ISO-8601" label="Record timestamp">2015-02-12 20:02:38Z</field>
                <field id="date created" type="date" format="ISO-8601" label="date created">2015-02-12</field>
              </metadata>
            ▼<content>
                <field id="time created" type="time" format="ISO-8601" label="time created">8:02:38 PM</field>
              ▼<field id="profile image" type="image" format="url" label="profile image">
                  http://10.35.82.107/xport/feeds/twitter/amxtalk/cache_FdoIJ01i_normal.jpeg
                </field>
                <field id="photo" type="image" format="url" label="photo"/>
                <field id="user name" type="string" label="user name">AMX</field>
                <field id="screen name" type="string" label="screen name">@AMXtalk</field>
              ▼<field id="text" type="string" label="text">
                  RT @avisystems: Need a flexible platform for meeting rooms to share content? Check out the new Enzo from @AMXtalk at the #AVITechShow http:…
                </field>
              </content>
            </record>
          ▼<record>
            ▼<metadata>
                <field id="timestamp" type="datetime" format="ISO-8601" label="Record timestamp">2015-02-12 15:46:56Z</field>
                <field id="date created" type="date" format="ISO-8601" label="date created">2015-02-12</field>
              </metadata>
```

FIG. 184 AMX General Purpose XML

NOTE: *If using Internet Explorer, it is necessary to save the file, then open it (via File > Open) to view the XML.*

It will be necessary to understand the contents of the data source file in order to map the data to the Components in the Listview button later in this example. Also, note the path indicated in the address bar of the browser when the XML is displayed. This address will be used later in this example to identify this file as a dynamic data source. See page 153 to view this file.

3) Create (draw) a Listview button

1. In TPDesign5, open a Page and use the Button Draw tool to create a new button.
2. With the new button selected, click the **Type** (General) property and select **Listview** from the drop-down of button types. This selection sets the new button as a Listview button, and enables a set of Listview-specific properties (FIG. 185):

| Type | listview ▼ |

Type (General) property set to Listview

Example Listview button
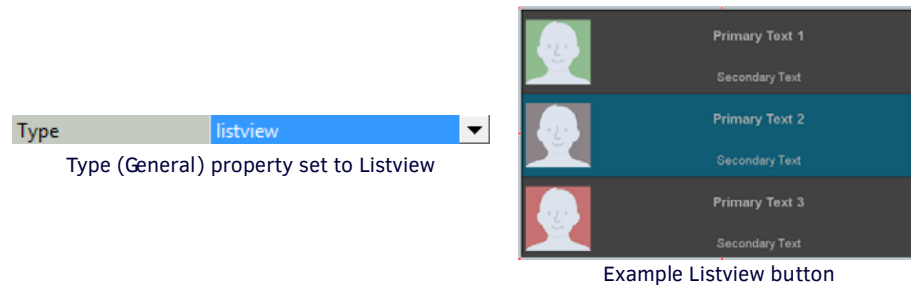
FIG. 185 Type (General) Property set to Listview

NOTE: *The "TWITTER.TP5" file included in the Twitter demo has a Listview button already drawn on the "Main" page.*

4) Set Listview Button Properties

Use the options in the Properties window to view/edit the *General*, *Programming* and *States* properties for the Listview button. The settings used in this demo are shown in FIG. 186:

| General Properties | Programming Properties | States Properties |
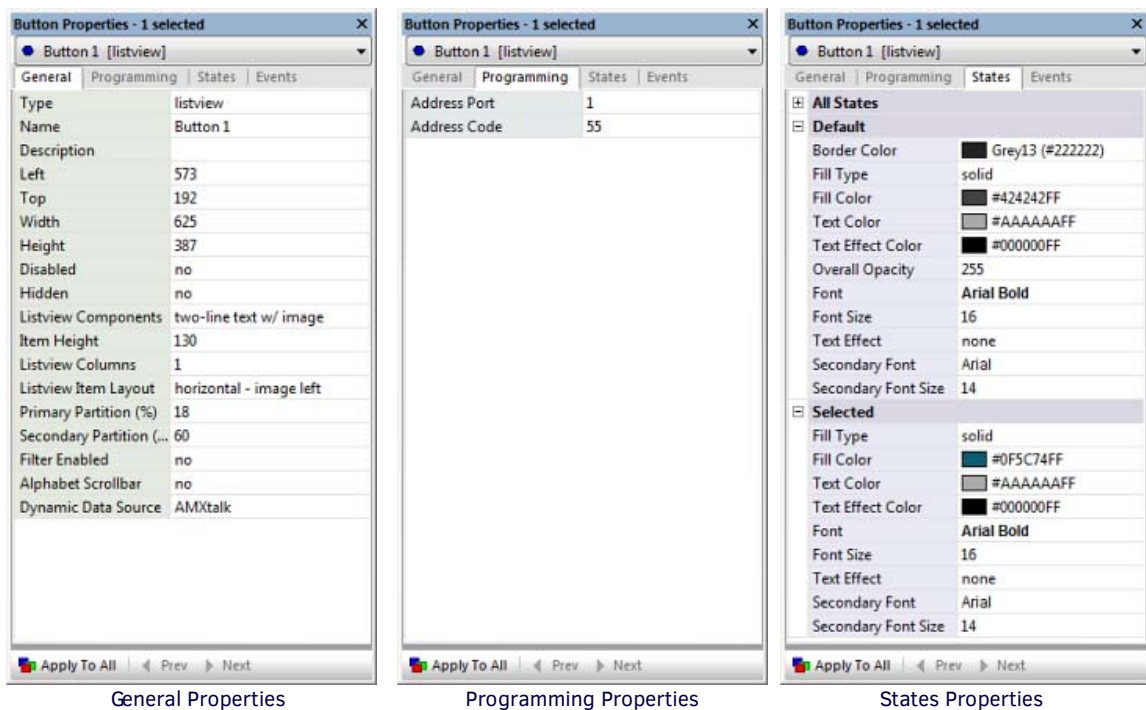
FIG. 186 Properties for the Listview Button

Refer to the *Working With Listview Button Properties* section on page 107 for details on Listview-specific button properties.

NOTE: *The Listview button in the Twitter demo is pre-configured with the General, Programming and States properties shown above.*

5) Add Dynamic Data Source to the Project

To add the data source file (amxstandard.xml) to the TPDesign5 project:

1. Open the Resource Manager to the *Dynamic Data Sources* tab and click **New** to open the *Create Dynamic Data Source* dialog (FIG. 187):
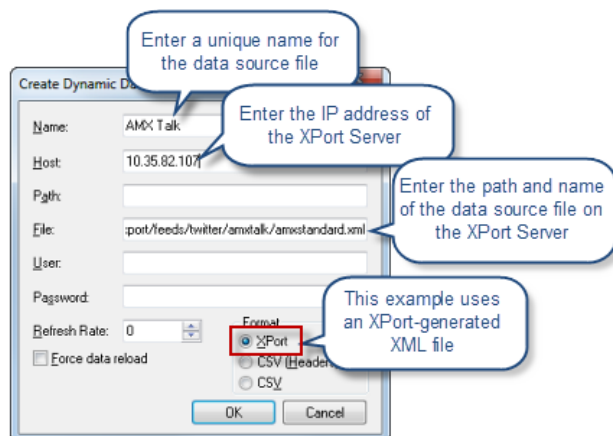
FIG. 187 Create Dynamic Data Source dialog with Example data (amxstandard.xml)

2. In the *Name* field, enter a unique friendly name for the data source. For this example, enter "**AMX Talk**".

3. In the *File* field, enter a file name that indicates the full path to the location of the source file on the XPort Server.

    This information can be retrieved from the browser window, when the *amxstandard.xml* file is opened via the AMX *General Purpose XML* link on the Twitter configuration page on the XPort server (FIG. 188):
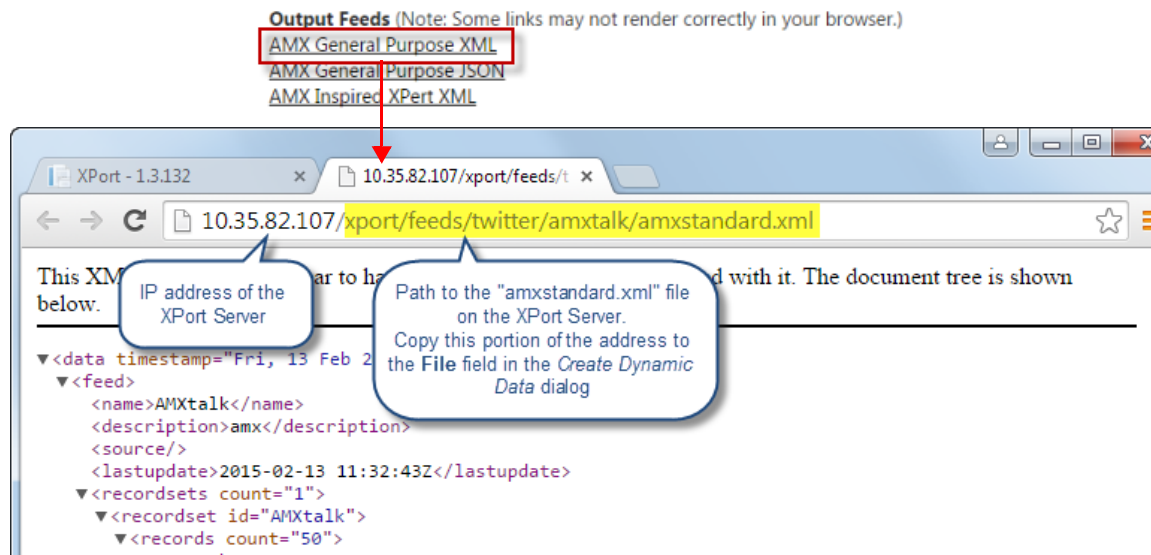
FIG. 188 Path to the "amxstandard.xml" file on the XPort Server

Note that the address/path displayed in the brower's address tab includes the IP address of the XPort Server (see "**10.35.82.107**" in the example above), and the path to the file ("**xport/feeds/twitter/amxtalk/amxstandard.xml**")

Enter the *IP address* in the **Host** field.

Enter the *path information* in the **File** field. Do not include the forward slash at the beginning of the path.

4. In the *User* field, enter the user name required by the NX Master or server for authentication (if required).

5. In the *Password* field, enter the password required by the NX Master or server for authentication (if required).

6. In the *Refresh Rate* field, use the up/down arrows to adjust the number of seconds between refreshes in which the resource is downloaded again. Refreshing resources will cause the button displaying that resource to refresh as well. The default value is 0, which means that the resource is only downloaded once.

7. Under *Format*, select **Xport**, since the data source file in this example (amxstandard.xml) uses an XPort-generated XML file.

8. Click **OK** to save changes and close this dialog. The new data source is indicated in the Resource Manager - Dynamic Data Sources tab (FIG. 189):
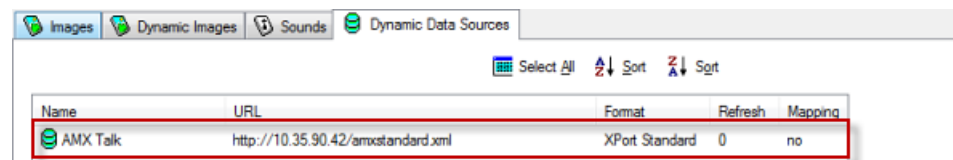
FIG. 189 Resource Manager - Dynamic Data Sources tab indicating "amxstandard.xml" as the data source

NOTE: *The Listview button in the TWITTER.TP5 file is pre-configured to use AMX Talk (amxstandard.xml) as it's data source file. However, it is necessary to update the Host address with the IP address of your NX Master as shown above. Double-click on Conference Rooms in the Resource Manager to open the Edit Dynamic Data Source dialog and update accordingly.*

6) Map the Data from the Data Source File to the Listview Button Components

It is necessary to map the data in the *amxstandard.xml* file to the three fields that comprise the Listview button layout. These three fields (called Components in TPDesign5) are: *Primary Text, Secondary Text* and *Image* (FIG. 190):
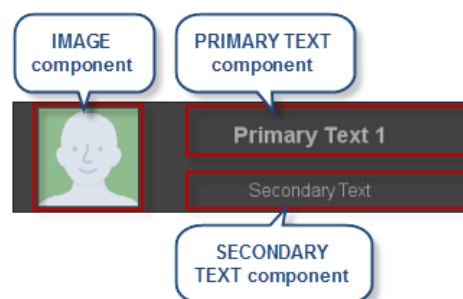
FIG. 190 Listview Button - Components

Step One: Analyze the Data Source

It is necessary to understand the contents of the data source file in order to map the data to the Components in the Listview button. In this example, the *amxstandard.xml* file contains several IDs that will be mapped to the Listview button: *screen name*, *text*, and *photo* (FIG. 191):
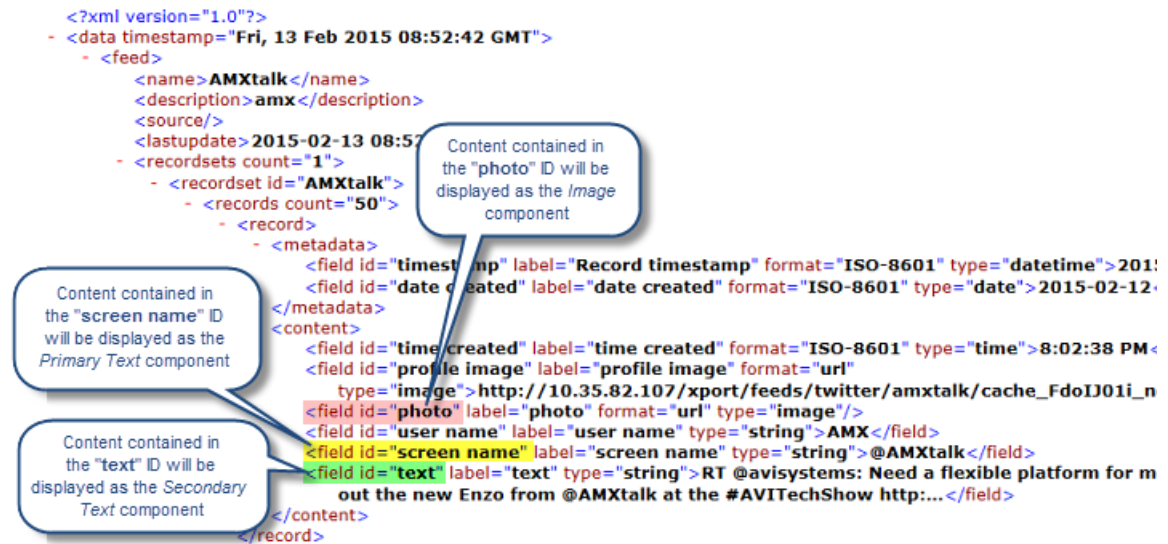
FIG. 191 Understanding the contents of the data source file - amxstandard.xml (AMXTalk Twitter feed)

In this example:

- The contents of the **"screen name"** ID will be mapped to display as the *Primary Text* component of the list items in the Listview button.
- The contents of the **"text"** ID will be mapped to display as the *Secondary Text* component of the list items in the Listview button.
- The contents of the **"photo"** ID will be mapped to display as the *Image* component of the list items in the Listview button.

Step Two: Map the Data to Components of the Listview button

1. With the Listview button selected, open the Resource Manager to the *Dynamic Data Sources* tab.

2. Select the data source (*AMX Talk)* that is assigned to the Listview button (as described on page 149):
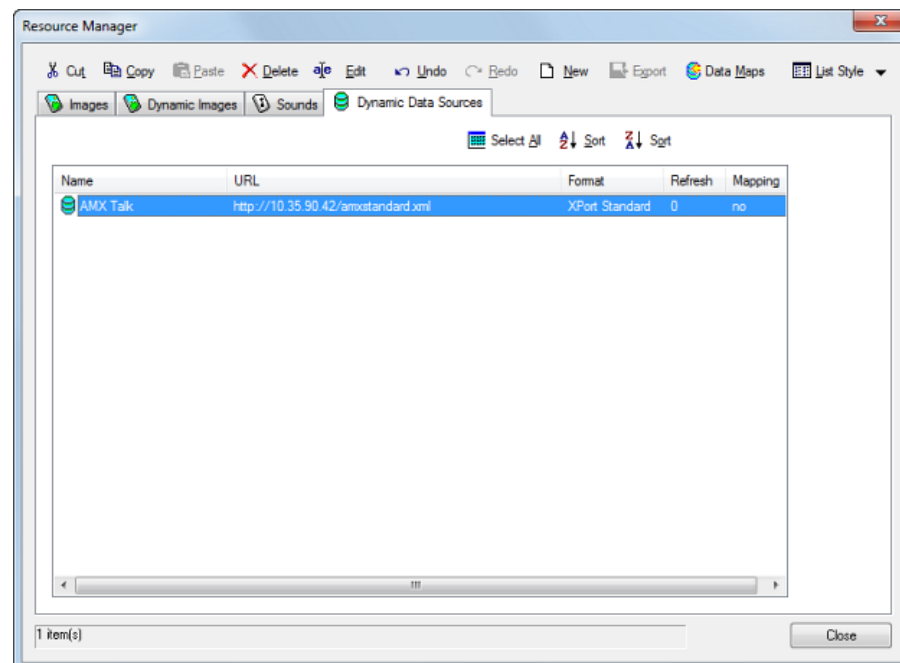
FIG. 192 Resource Manager - Dynamic Data Source tab

3. Click the **Data Maps** button to access the *Dynamic Data Mappings - Listview Buttons* dialog (FIG. 193):
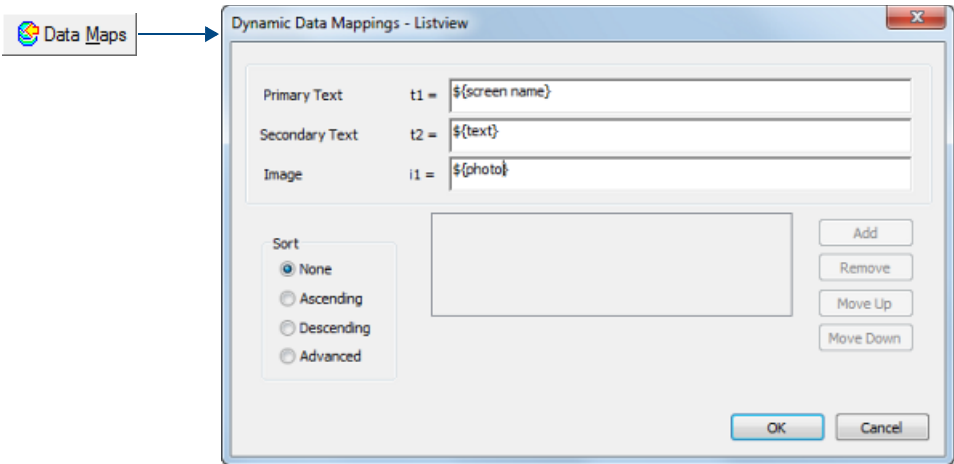
FIG. 193 Dynamic Data Mappings - Listview dialog (with example data indicated)

4. Use the fields in this dialog to specify the device mapping for the selected Listview button and the selected Data Source (see *Dynamic Data Mappings - Syntax Requirements (XML)* below).

NOTE: *The Listview button in the Twitter demo is pre-configured with the data mapping settings shown above.*

Dynamic Data Mappings - Syntax Requirements (XPort-Generated XML)

Note that the syntax requirements for these fields depends on the type of file used as the data source. The data source file in this example uses an XPort-generated XML file. The syntax requirements for data mapping to an XPort-generated XML file is described below:

Dynamic Data Mappings - Syntax Requirements (XPort-generated XML file)
Primary Text: For XPort-generated XML files, the syntax is: ${ID} Following this syntax, enter the name of the ID in the data source file to be displayed as the Primary Text component of the Listview button. In this example, *amxstandard.xml* lists the name associated with the Twitter user in the "*screen name*" ID. To display the contents of the "screen name" ID as the Primary Text component, enter **${screen name}** in the *Primary Text* field:
Secondary Text: For XPort-generated XML files, the syntax is: ${ID} Following this syntax, enter the name of the ID in the data source file to be displayed as the Secondary Text component of the Listview button. In this example, *amxstandard.xml* lists the caption associated with each "Tweet" in the "*text*" ID. To display the contents of the "text" ID as the Secondary Text component, enter **${text}** in the *Secondary Text* field:
Image: For XPort-generated XML files, the syntax is: ${ID} Following this syntax, enter the name of the ID in the data source file to be displayed as the Image component of the Listview button. In this example, *amxstandard.xml* lists the image associated with each "Tweet" in the "*photo*" ID. To display the contents of the "photo" ID as the Secondary Text component, enter **${photo}** in the *Image* field:

NOTE: *The fields in the Dynamic Data Mappings - Listview Buttons dialog are case-sensitive.*

7) Assign a Data Source file to the Listview Button

The data source (amxstandard.xml) is associated with the Listview button via the *Dynamic Data Source* property (in the *General* tab of the Properties window):

1. With the Listview button selected, click the browse button in the **Dynamic Data Source** (General) property to open the *Select Resource* dialog (FIG. 194):
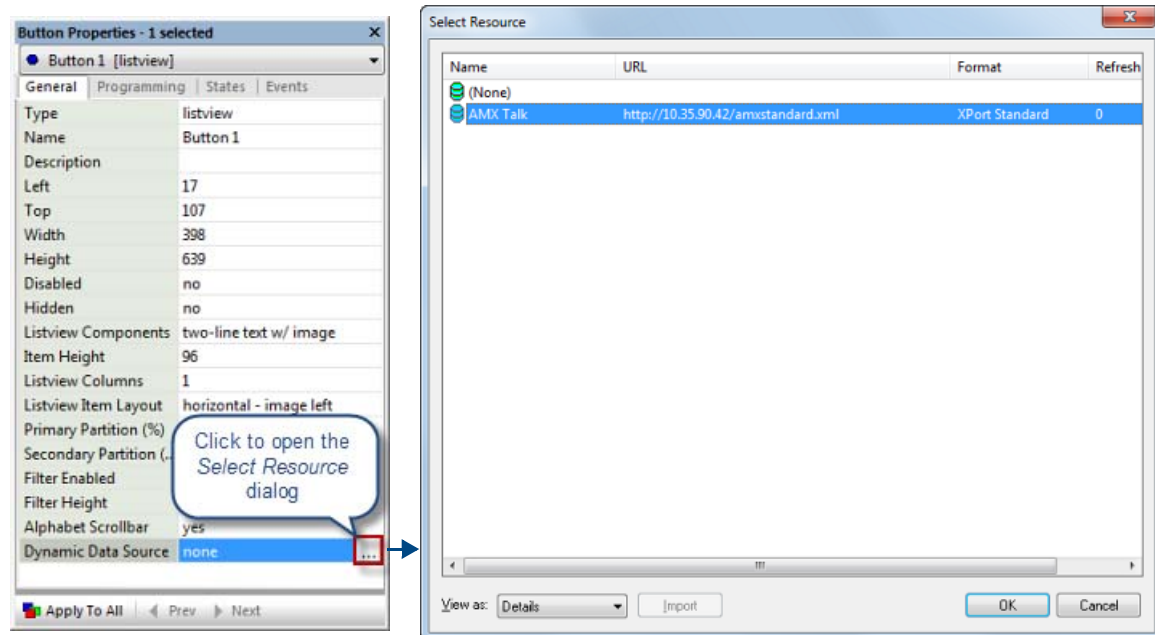
FIG. 194 Dynamic Data Source (General) Property and Select Resource dialog

2. Select the XML file to use as the data source (in this example, "*AMX Talk*").

3. Click **OK** to close this dialog.

4. The selected Data Source file is indicated in the *Dynamic Data Source* property (see "*AMX Talk*" in FIG. 195):
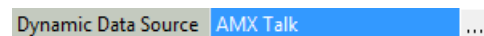
FIG. 195 Dynamic Data Source property indicating "AMX Talk"

NOTE: *The "TWITTER.TP5" file included in the Twitter demo has "AMX Talk" already assigned as the Dynamic Data Source for the Listview button.*

8) Write a Custom Event To Respond To User Selection

When the user selects an item on the Listview button, the entire record for that selection is sent to the NX Master. A CUSTOM_EVENT is raised and within this function the desired information can be retrieved for the selection. In this example, a popup window that displays the text and image for each message.

Listview buttons use the custom event parameter "LISTVIEW_ON_ROW_SELECT_EVENT" to provide the ability to configure a response to the selection of a list item in a Listview button in NetLinx code.

This custom event must be added to the NetLinx code on the NX Master.

1. Use NetLinx Studio 4 to add the following code to the CUSTOM EVENT section of the NetLinx program loaded on the Master:

```
PROGRAM_NAME='ISE_CUSTOM_EVENT'
(***********************************************************)
(***********************************************************)
(*  FILE_LAST_MODIFIED_ON: 04/05/2006  AT: 09:00:25       *)
(***********************************************************)
(* System Type : NetLinx                                  *)
(***********************************************************)
(* REV HISTORY:                                           *)
(***********************************************************)
(*
    $History: $
*)

DEFINE_DEVICE
dvTP = 10001:1:0

DEFINE_CONSTANT
```

```
// Twitter Listview button address
INTEGER btnTWTListview = 55

DEFINE_VARIABLE

DEFINE_EVENT


 // CUSTOM EVENT RAISED WHEN ITEM IN
 // TWITTER LISTVIEW WIDGET IS SELECTED
CUSTOM_EVENT[dvTP,btnTWTListview,LISTVIEW_ON_ROW_SELECT_EVENT]
{
    SLONG payloadId
    SLONG payloadType
    CHAR fields[2][16]
    CHAR screenName[DATA_MAX_VALUE_LENGTH]
    CHAR text[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] = 'screen name'
    fields[2] = 'text'
    // 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.
        screenName = record.content[1].value
        text = record.content[2].value

        // Put the name and number that was selected on a popup and
        // show the popup
        SEND_COMMAND dvTP,"'^TXT-56,0,',screenName"
        SEND_COMMAND dvTP,"'^TXT-57,0,',text"
        SEND_COMMAND dvTP,"'^PPN-Twitter'"
        SEND_COMMAND dvTP, "'^PPT-Twitter;50'"
    }
    }
}
DEFINE_PROGRAM
(***********************************************************)
(*                   END OF PROGRAM                       *)
(*        DO NOT PUT ANY CODE BELOW THIS COMMENT          *)
(***********************************************************)
```

2. Use NetLinx Studio 4 to compile the code (select **Build > Compile**).
3. Use NetLinx Studio 4 to transfer the AXS file to the NX Master:
   a. Select **Tools > File Transfer** to open the *File Transfer* dialog.
   b. In the *Send* tab, click the **Add** button. This opens the *Select Files for File Transfer* dialog.
   c. In the *Other* tab, select **Non-System File** and click **Add**.
   d. Select the compiled NetLinx code (in this example, "*ISE_CUSTOM_EVENT.axs*") and click **Open**. This opens the *Enter Device Mapping* dialog.
   e. Review and edit the D:P:S settings for the target NX Master (leave the *Master Directory* field empty), and click **OK** to close the *Enter Device Mapping* dialog and return to the *Select Files for File Transfer* dialog.
   f. Select **OK** to return to the *File Transfer* dialog.
   g. In the *File Transfer* dialog, click **Send** to initiate the file transfer.
   h. The progress of the transfer is indicated in the Output Bar.

NOTE: *The custom event code shown above is included in the NetLinx Studio Workspace file (TWITTER.apw) that is in the Twitter.ZIP file.*

9) Transfer the TPDesign5 Project to the Touch Panel

At this point, everything is ready to go: the NX Master has the code to handle custom events and the TPD5 project file is handling the data source/mapping for the Listview button.

The only thing left to do is to transfer the TPD5 project containing the Listview button, data source reference and image references to the G5 touch panel:

1. In TPDesign5, select **Transfer > Connect** to open the *Connect* dialog (FIG. 196):
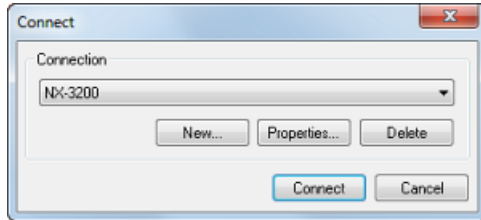
FIG. 196 Connect dialog

> **NOTE:** *If the Master has never been connected to before, a new connection will need to be configured. Refer to the File Transfer Operations section on page 285 for details.*

2. Select the connection configuration for the target NX Master from the *Connection* drop-down list, and click **Connect**.

    Once a connection has been established with the Master, select **Transfer > Send to Panel** to open the *Send to Panel* dialog (FIG. 197):
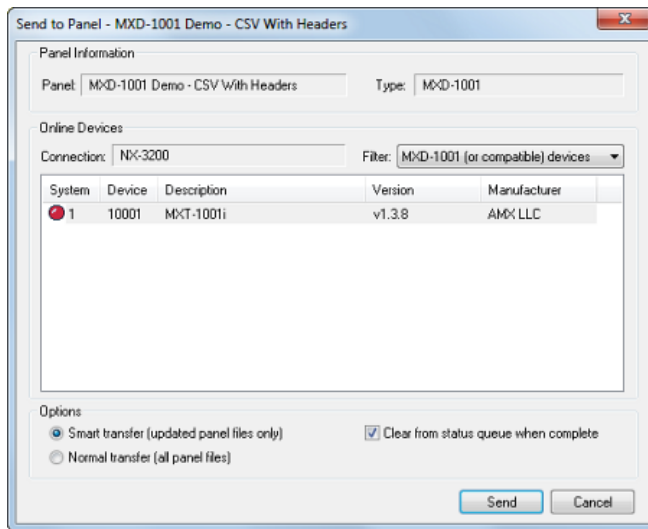
FIG. 197 Send To Panel dialog

3. Click **Send** to begin the file transfer.

When the transfer is complete, the Listview button should appear on the Page it was added to.

Example 3 (XML File/XPort Server) - Results

FIG. 198 shows an example of a basic Listview button created by following these steps:

amxstandard.xml

Resource Manager (Dynamic Data Sources tab) showing the Conference Rooms data source imported into theTPD5 project

Data from amxstandard.xml mapped to Listview components

Resulting Listview button as displayed on the panel

FIG. 198 Example Listview button based on "AMXTalk"

Using the *amxstandard.xml* file as it's data source:

- It displays the contents of the "screen name" ID as the *Primary Text* component.
- It displays the contents of the "text" ID as the *Secondary Text* component.
- It displays the contents of the "photo" ID as the *Image* component.

NOTE: *While the Listview button shown in this example uses only basic design characteristics, note that Listview buttons support most of the same display options as other button types, including Radiant/Gradient fills, Text Effects, Opacity, etc... Use these options to create eye-catching designs, just like for any other button type.*

Reference: "amxstandard.xml"

FIG. 199 provides a partial view of the "amxstandard.xml" file generated by the XPort server as "AMX General Purpose XML":
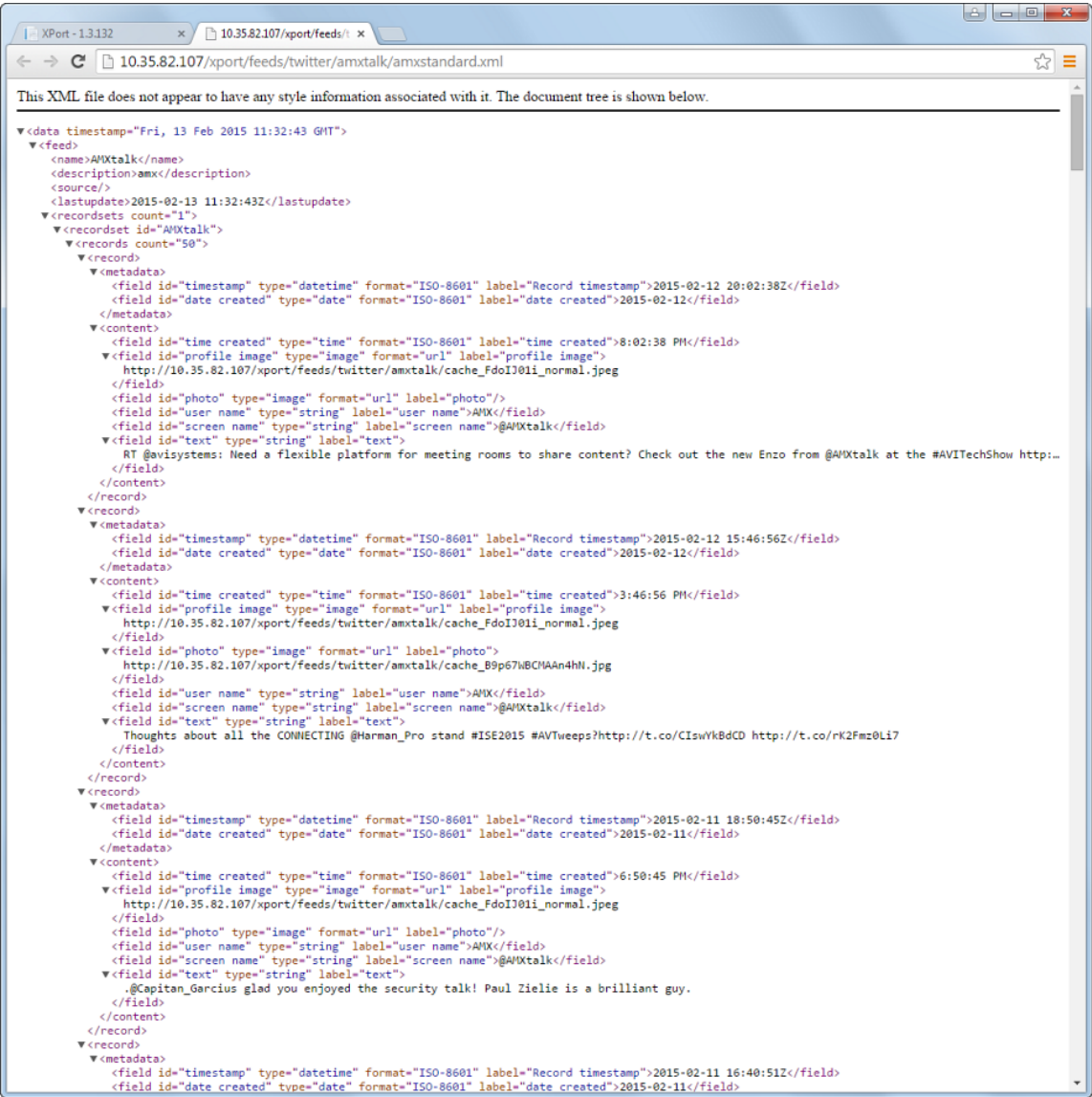
FIG. 199 amxstandard.xml file (Generated by an XPort server) - partial view

Twitter (XPort XML) Demo File ("Twitter.ZIP")

Demo (ZIP) files for the Listview examples presented here are available to download from the UI RESOURCE CENTER at www.amx.com. The preceding example followed the *Twitter* demo. The Twitter demo ZIP file (**Twitter.ZIP**) contains the following:

Twitter.ZIP Contents	
File	Description
TWITTER.TP5	TPDesign5 project file that includes a Listview button pre-configured to use the layout properties and data source file shown in the *Listview Button/Dynamic Data Example 3: XML File/XPort Server example.*
TWITTER.png	This image file is used as the Image component for the Listview button.
TWITTER.apw	NetLinx Studio 4 Workspace file, with the Listview demo custom event defined. This Workspace contains the TWITTER_CUSTOM_EVENT.axs file.

To use this demo:

1. Download the *TWITTER.ZIP* file and extract it's contents to a known location.
2. Launch TPDesign5 and open the *TWITTER.TP5* project file. Use TPDesign5 to set to the Host (IP) address for the data source file:
   a. Open the Resource Manager to the *Dynamic Data Sources* tab, and double-click on **AMXTalk** to access the *Edit Dynamic Data Source* dialog.

     b. Edit the **Host** field with the IP address of the XPort server that will provide the data. Click **OK** to save changes and close this dialog.

     c. Close the Resource Manager.

     d. Save changes and close the TP5 project.

3. In NetLinx Studio 4, open the *TWITTER.apw* workspace file (**File > Open Workspace)**.

    This Workspace contains NetLinx source code that is pre-configured with a Custom Event for user selection, as well as a TPDesign5 project that includes a pre-configured Listview button that uses *AMXTalk* as it's data source.

4. Build the Workspace: Select **Build > Build Active System**.

5. Transfer all files contained in the Workspace to the target NX Master:

     a. Select **Settings > Active System Communication Settings** to open the *Communication Settings* dialog. Use the options in this dialog to establish a connection to the target NX Master (FIG. 200):
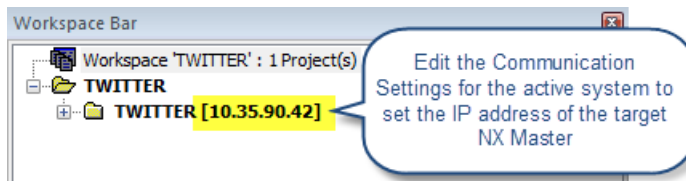
FIG. 200 NetLinx Studio 4 Select Files for File Transfer dialog (Current Workspace tab) - Projects directory selected

     See NetLinx Studio 4 online help for details on configuring communication settings.

     b. Select **Tools > File Transfer** to open the *File Transfer* dialog (*Send* tab). Remove any files (from previous transfer operations) that may be in the list.

     c. Click **Add** to open the *Select Files for File Transfer* dialog (*Current Workspace* tab).

     d. Click the top-level *Projects* directory to auto-select all files in the Workspace.

     e. Verify that the IP address indicated here indicates the correct NX Master, and click **OK** to save changes and return to the *File Transfer* dialog.

     f. In the *File Transfer* dialog, click **Send** to transfer the Workspace files to the target NX Master.

Listview Button/Dynamic Data Example 4: 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 selects a contact on the screen and then presses a call button to initiate the call. This is an example workflow for creating a Listview widget on a touchpanel page, creating a data source in NetLinx, configuring and populating the Listview and responding to a user selection.

NOTE: *This set of instructions uses files that are included in the "NetLinxAPI.ZIP" demo file which is available to download from the UI RESOURCE CENTER at www.amx.com.*

The resulting Listview button will display a listing of phone contacts with each contact's name and phone number (FIG. 201):
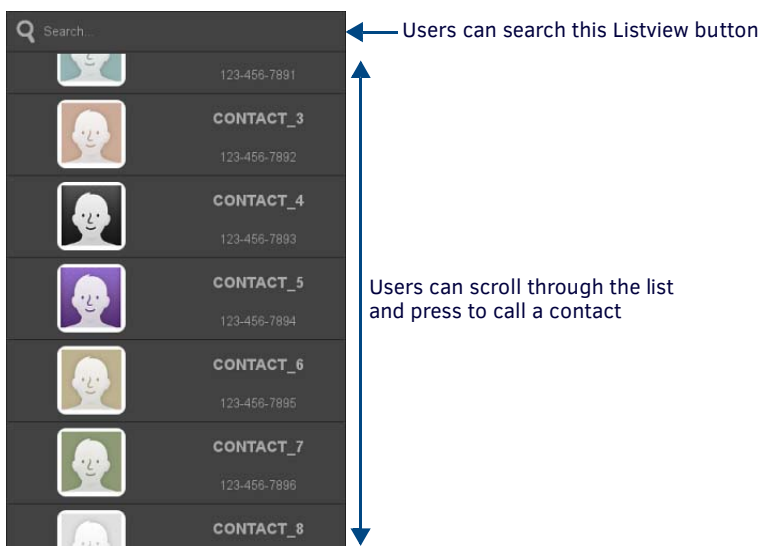
FIG. 201 Example - Listview button based on "NetLinxAPI.axs"

Before You Begin

Download the *NetLinxAXI.ZIP f*ile from the UI RESOURCE CENTER at www.amx.com and extract it's contents to a known location.

1) Create (draw) a Listview button

1.   In TPDesign5, open a Page and use the Button Draw tool to create a new button.

2.   With the new button selected, click the **Type** (General) property and select **Listview** from the drop-down of button types. This selection sets the new button as a Listview button, and enables a set of Listview-specific properties (FIG. 156):

 Type (General) property set to Listview	 Example Listview button
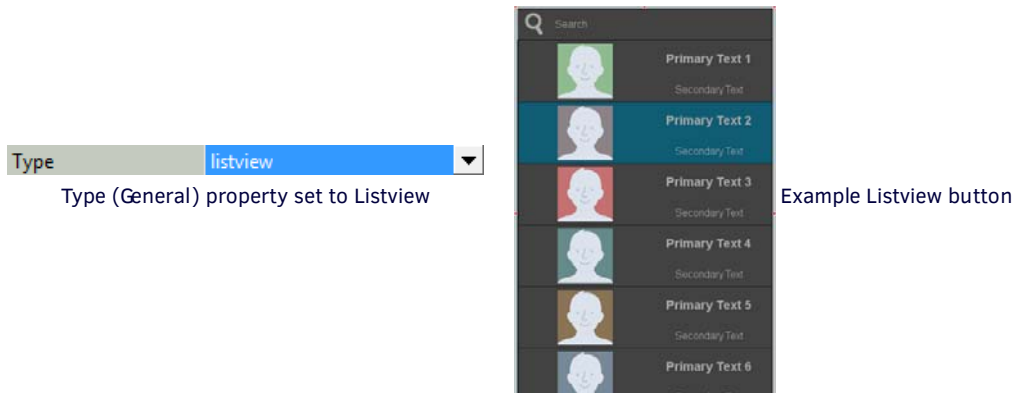
FIG. 202 Type (General) Property set to Listview

> **NOTE:** *The "NetLinxAXI.TP5" file included in the NetLinxAXI demo has a Listview button already drawn on the "Main" page.*

2) Set Listview Button Properties

Use the options in the Properties window to view/edit the *General*, *Programming* and *States* properties for the Listview button. The settings used in this demo are shown in FIG. 203:

General Properties   Programming Properties   States Properties
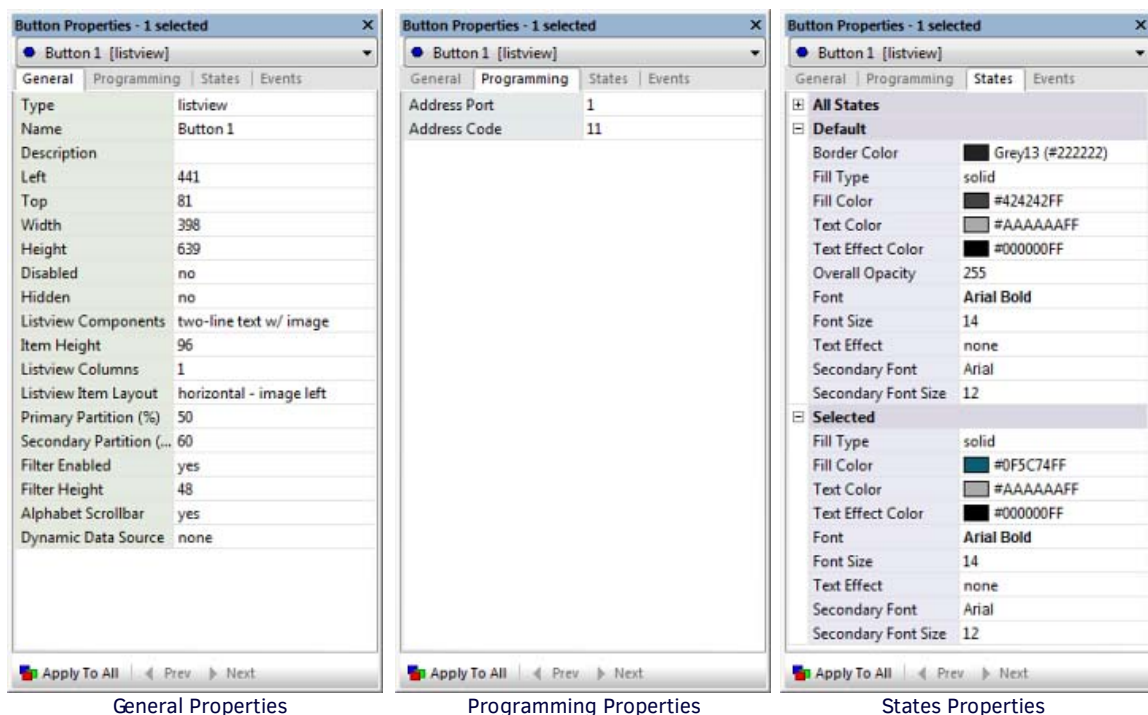
FIG. 203 Properties for the Listview Button

> **NOTE:** *The Listview button in the NetLinxAXI demo is pre-configured with the General, Programming and States properties shown above.*

Refer to the *Working With Listview Button Properties* section on page 107 for details on Listview-specific button properties.

3) Create the Data Source

Follow the example *NetLinx Usage Example - ASCII* (below) to create a data source in NetLinx and publish the data source to the NX Master's internal web server.

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

4) Configuring the Response to a User Selection

Follow the CUSTOM_EVENT example at the end of the example below to retrieve the phone number that was selected by the user.

NetLinx Usage Example - ASCII

Review the following code and read all comments to see how this file works:

```
PROGRAM_NAME='Listview Example'

DEFINE_DEVICE
dvTP = 10001:1:0

DEFINE_CONSTANT
// listview button address
INTEGER btnListview = 11

DEFINE_VARIABLE
//just a variable to hold our "published URL" value
CHAR publishedURL[DATA_MAX_VALUE_LENGTH]
//just a variable to hold our recordset ID
CHAR recordsetID[DATA_MAX_ID_LENGTH]

DEFINE_FUNCTION GenerateDataFeed()
{
    //we can't add fields to a record or a record to
    // a feed if they don't exist so lets define them here

    //DATA_FEED is a predefined structure in the NetLinx AXI
    //datafeed is our variable representing that structure
    STACK_VAR DATA_FEED datafeed

    //DATA_RECORD is a predefined structure in the NetLinx AXI
    //record is our variable representing that structure
    STACK_VAR DATA_RECORD record

// -------------------------------------------------
// CREATE A NEW DATA FEED
// -------------------------------------------------
    //set the characteristics of the dataFeed (these are just descriptive strings)
    //name/description/source are defined in the DATA_FEED structure
    datafeed.name = 'phonelist'
    datafeed.description = 'Some Harman Employees'
    datafeed.source = 'NetLinx PhoneList'

    //we've defined all the values for our DATA_FEED
    //now we need to "create" our DATA_FEED
    DATA_CREATE_FEED(datafeed)

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

// -------------------------------------------------
// 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, 1)
    // 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;
    // When mapping the data to a listview widget on the panel and when using
    // the data in the custom event, the id field is used
    // 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
    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_1.png'
    record.content[2].value = 'CONTACT_1'
    record.content[3].value = '123-456-7890'
    // 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://server-lin/ftp/listview/CONTACT_2.png'
    record.content[2].value = 'CONTACT_2'
    record.content[3].value = '123-456-7891'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_3.png'
    record.content[2].value = 'CONTACT_3'
    record.content[3].value = '123-456-7892'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_4.png'
    record.content[2].value = 'CONTACT_4'
    record.content[3].value = '123-456-7893'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_5.png'
    record.content[2].value = 'CONTACT_5'
    record.content[3].value = '123-456-7894'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_6.png'
    record.content[2].value = 'CONTACT_6'
    record.content[3].value = '123-456-7895'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_7.png'
    record.content[2].value = 'CONTACT_7'
    record.content[3].value = '123-456-7896'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_8.png'
    record.content[2].value = 'CONTACT_8'
    record.content[3].value = '123-456-7897'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_9.png'
    record.content[2].value = 'CONTACT_9'
    record.content[3].value = '123-456-7898'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    record.content[1].value = 'http://server-lin/ftp/listview/CONTACT_10.png'
    record.content[2].value = 'CONTACT_10'
    record.content[3].value = '123-456-7899'
    DATA_ADD_RECORD(datafeed.name, recordsetID, record)

    // The final step is to publish the feed
    publishedURL = DATA_PUBLISH_FEED(datafeed.name)
}

DEFINE_START
    GenerateDataFeed()

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=${number}'"
    // Sort ascending by name
    SEND_COMMAND dvTP,"'^LVS-',ITOA(btnListview),',${name};a'"
```

```
                // Command the listview to load the data from the master
                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
            //just a char array to hold the data we want to use in the custom event.
            CHAR fields[2][16]
            //char variables to hold our data for "name" & "number"
            CHAR name[DATA_MAX_VALUE_LENGTH]
            CHAR number[DATA_MAX_VALUE_LENGTH]

            //variable record, of type DATA_RECORD, to hold the record we retrieve from the custom event
            DATA_RECORD record
            // Get the data access ID from the custom event
            // variable is payloadID - custom.value1 is predefined
            payloadId = custom.value1
            // Get the data type from the custom event
            // variable is payloadType - custom.value2 is predefined
            payloadType = custom.value2

            if (payloadId > 0 && payloadType == DATA_STRUCTURE_DATARECORD)
            {
            // Specify which fields we want to retrieve from the payload
            // (these are the IDs we defined earlier)
            fields[1] = 'name'
            fields[2] = 'number'
            // Retrieve the record and get our requested fields
            if (DATA_GET_EVENT_RECORD(dvTP, payloadId, fields, record) > 0)
            {
                // The record existed and contained our fields
                // let's retrieve the values that we are interested in
                name = record.content[1].value
                number = record.content[2].value
                // Send the name & number that was retrieved to the appropriate buttons & 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           *)
(************************************************************)
```

NOTE: *The NetLinx code shown above is included in the NetLinx Studio Workspace file (NetLinxAPI.apw) that is in the NetLinxAPI.ZIP file.*

Update this code as necessary to reference your NX Master.

In order for this code to work with your Master, all instances of

`'http://server-lin/ftp/listview/CONTACT_1.png'`

must be updated to indicate the IP address of your NX Master.

For example, a Master with the IP address of "**10/35.90.42**" would require the following update

`'http://`**10.35.90.42**`/CONTACT_1.png'`

5) Compile the Code

In NetLinx Studio 4, select **Build > Build Active System** to compile the NetLinx code.

6) Transfer the Workspace to the NX Master

Use NetLinx Studio 4 to transfer the NetLinx code (*NetLinxAPI.tkn* and *NetLinxAPI.src*) files as well as the TPDesign5 project file (*NetLinxAPI.TP5*):

1.   Select **Tools > File Transfer** to open the *File Transfer* dialog.

2.   In the *Send* tab, click the **Add** button. This opens the *Select Files for File Transfer* dialog.

3.   In the *Current Workspace* tab, select the top-level *Projects* folder and click **OK** (FIG. 204):

FIG. 204 Select Files for File Transfer dialog - Current Workspace tab

4.   Select **OK** to return to the *File Transfer* dialog. The selected files are indicated in the *Send* tab (FIG. 205):
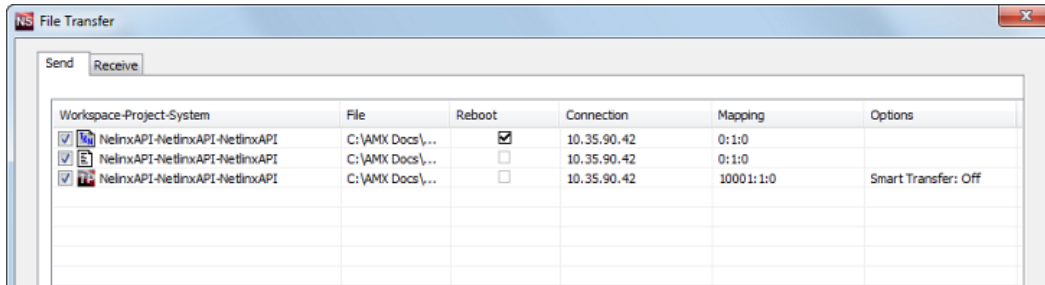
FIG. 205 File Transfer dialog - Send tab

5.   In the *File Transfer* dialog, click **Send** to initiate the file transfer.

6.   The progress of the transfer is indicated in the Output Bar.

When the transfer is complete, and the NX Master has completed a reboot, the Listview button should appear on the Page it was added to.

Example 4 (NetLinx Data Source) - Results

FIG. 206 shows an example of a basic Listview button created by following these steps:

NetLinxAXI.axs

Resulting Listview button as displayed on the panel
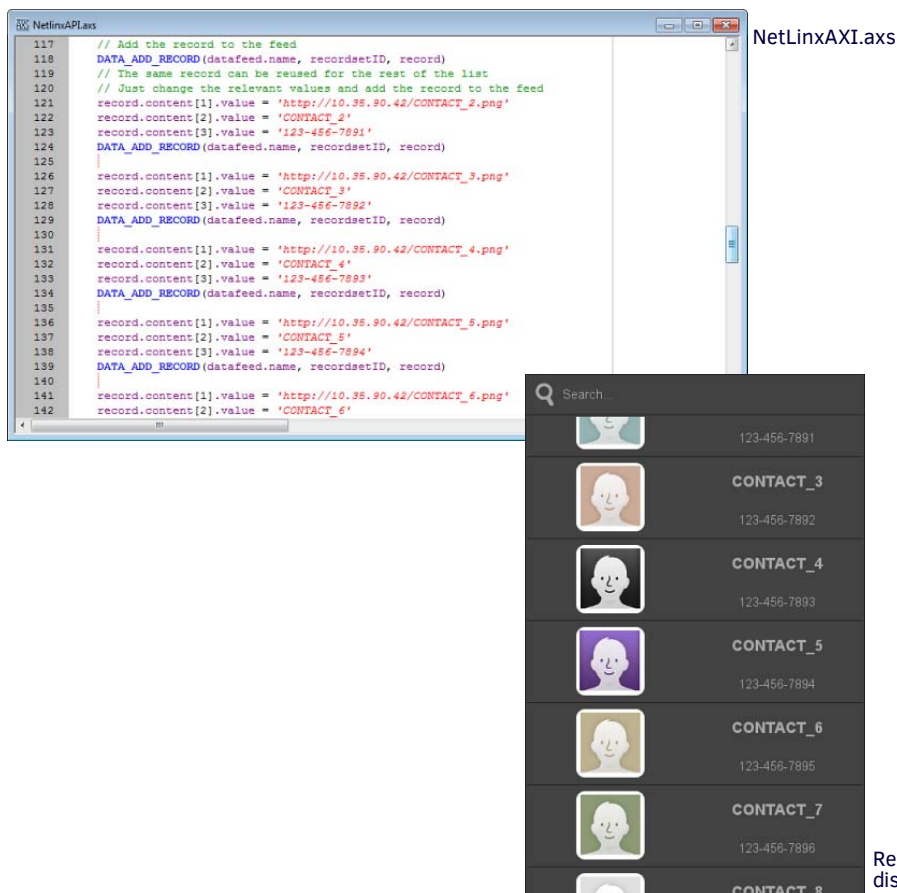
FIG. 206 Example Listview button based on "NetLinxAXI.axs"

Using *NetLinxAXI.axs* file as it's data source:
- It displays the contents of the "record.content[2].value" ID as the *Primary Text* component.
- It displays the contents of the "record.content[3].value" ID as the *Secondary Text* component.
- It displays the contents of the "record.content[1].value" ID as the *Image* component.

NOTE: *While the Listview button shown in this example uses only basic design characteristics, note that Listview buttons support most of the same display options as other button types, including Radiant/Gradient fills, Text Effects, Opacity, etc... Use these options to create eye-catching designs, just like for any other button type.*

NetLinxAPI Demo File ("NetLinxAPI.ZIP")

Demo (ZIP) files for the Listview examples presented here are available to download from the UI RESOURCE CENTER at www.amx.com. The preceding example followed the *NetLinxAPI* demo. The NetLinxAPI demo ZIP file (**NetLinxAPI.zip**) contains the following:

NetLinxAPI ZIP Contents	
File	**Description**
NetLinxAPI.TP5	TPDesign5 project file that includes a Listview button pre-configured to use the layout properties and data source file shown in the *Listview Button/Dynamic Data Example 4: NetLinx Data Source example (see page 154)*.
"CONTACT_IMAGES" folder	This folder contains the images used for the Listview button in this example.
NetLinxAPI.apw	NetLinx Studio 4 Workspace file, with the Listview demo custom event defined. This Workspace contains the following files: • NetLinxAPI.axs • NetLinxAPI.src

To use this demo:

1. Download the *NetLinxAPI.ZIP* file and extract it's contents to a known location.

2. In NetLinx Studio 4, open the *NetLinxAPI.apw* workspace file (**File > Open Workspace)**.

   This Workspace contains NetLinx source code that is pre-configured with a Custom Event for user selection, as well as a TPDesign5 project that includes a pre-configured Listview button that uses NetLinx data as it's data source.

3. Build the Workspace: Select **Build > Build Active System**.

4. Transfer all files contained in the Workspace to the target NX Master:

   a. Select **Settings > Active System Communication Settings** to open the *Communication Settings* dialog. Use the options in this dialog to establish a connection to the target NX Master. Note that by default, the workspace is configured to use Serial communication (FIG. 154):

FIG. 207 NetLinx Studio 4 Select Files for File Transfer dialog (Current Workspace tab) - Projects directory selected

   b. Select **Tools > File Transfer** to open the *File Transfer* dialog (*Send* tab). Remove any files (from previous transfer operations) that may be in the list.

   c. Click **Add** to open the *Select Files for File Transfer* dialog (*Current Workspace* tab).

   d. Click the top-level *Projects* directory to auto-select all files in the Workspace.

   e. Verify that the IP address indicated here indicates the correct NX Master, and click **OK** to save changes and return to the *File Transfer* dialog.

   f. In the *File Transfer* dialog, click **Send** to transfer the Workspace files to the target NX Master.

Listview (Data Access) Send Commands

The *Data Access Send Commands* described in the following table represent a new set of Button (^) Send Commands that support the use of dynamic data for Listview buttons in NetLinx code. Note that the *variable text address range* (`<vt addr 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	A 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.

^LVD	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 (page 165) must be issued to load the data.
	Syntax:
	`"'^LVD-<vt addr range>,<URL to data source or Dynamic Data Resource name>,`
	`<configuration_option=configuration_value>'"`
	Variables:
	• variable text address range = 1 - 4000.
	• URL to the data source/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.
	A file on the panel's local file system can be specified using the "**file:///**" option.
	Note: "***ftp://***" *is not a supported option.*
	• 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.
	• **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.
	• **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.

Listview Commands (Cont.)

^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-<vt addr range>,<filter character sequence>'"`
	Variables:
	• variable text address range = 1 - 4000.
	• filter character sequence. All characters including whitespace characters will be applied to the filter.
	Example:
	`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.

^LVL	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: Optional valid tags for the columns parameter are nc=, numcol=, and columns=).* • **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***: *Optional valid tags for the comp parameter are c= and comp=.*

Component Value	Description
1	The image (i) is used in the cell.
2	The primary text field (t1) is used in the cell.
4	The secondary text field (t2) is used in the cell

Not all variations of component values are valid. To have the secondary text field present, the primary text field must be preset as well.

Component Combinations	Description
0	Invalid. No component displayed.
1	The image (i) is the only component displayed.
2	The primary text field (t1) is the only component displayed.
3	The image (i) and the primary text field (t1) are displayed.
4	Secondary text (t2) only. Invalid. Secondary text (t2) cannot be displayed without the primary text (t1).
5	Secondary text (t2) and image (i). Invalid. Secondary text (t2) cannot be displayed without the primary text (t1).
6	The primary text (t1) and secondary text (t2) are displayed.
7	The image (i), primary text (t1), and secondary text (t2) are displayed

• **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=**.

• **layout** - An integer that sets the layout configuration of each cell.

  *Note: valid tags for the layout parameter are **l=** and **layout=**.*

Listview Commands (Cont.)

^LVF (Cont.)	Layout Value	Description
	1	Horizontal layout with image on the left and text(s) on the right. If multiple texts are selected then the texts are stacked vertically
	2	Horizontal layout with image on the right and text(s) on the left. If multiple texts are selected then the texts are stacked vertically.
	3	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.
	4	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.
	5	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.
	6	Vertical layout with text1 on top, the image below text1, and text2 below the image.

- **layout** - An integer that sets the layout configuration of each cell.
  Note: valid tags for the layout parameter are l= and layout=.

Layout Value	Description
1	Horizontal layout with image on the left and text(s) on the right. If multiple texts are selected then the texts are stacked vertically
2	Horizontal layout with image on the right and text(s) on the left. If multiple texts are selected then the texts are stacked vertically.
3	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.
4	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.
5	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.
6	Vertical layout with text1 on top, the image below text1, and text2 below the image.

- **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'.
  Note: Valid tags for the filter parameter are f= and filter= .
- **filterheight** - 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.
  Note: Valid tags for the filterheight param is fh= and filterheight= .
- **alphascroll** - 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'. (NOTE: Valid tags for the alphascroll parameter are as= and alphascroll=).

Examples:
```
SEND_COMMAND Panel,"'^LVL-42, layout=1, comp=7, columns=1, cellheight=120, p1=40%, p2=66%'"
```
   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.
```
SEND_COMMAND Panel,"'^LVL-42,l=4, c=3, ch=150, nc=4, p1=70'"
```
   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.
```
SEND_COMMAND Panel,"'^LVL-42,layout=3, comp=6, ch=100, numcol=1, p1=50'"
```
   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.

Listview Commands (Cont.)

^LVF (Cont.)	`SEND_COMMAND Panel,"'^LVL-42,filter=1, fh=10%, as=false'"` 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.

^LVM	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.
	
	Syntax:
	`"'^LVM-<vt addr range>,<display_element=field_expression| <display_element=field_expression>|...'"`
	Variables:
	• variable text address range = 1 - 4000.
	• 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...
	Field Expressions:
	An expression that can be used to map field values to display elements. Any time a field name is used, it follows the form **${field_name}.** Other text characters can be used to construct a more complex string using multiple fields.
	Examples:
	`SEND_COMMAND Panel,"'^LVM-42,i1=${image}'"`
	    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.
	`SEND_COMMAND Panel,"'^LVM-42,i1=${image}|t1=${lname}, ${fname}|t2=${number}'"`
	    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.
	`SEND_COMMAND Panel,"'^LVM-42,t1=${column2}, ${column1}|t2=${column3}|i1=${column4}'"`
	    This is the same example as the one above it but the source of the data was a csv file that didn't have any headers. The csv columns were laid out as first name, last name, number, URL to image.

^LVN	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.
	
	Syntax:
	`"'^LVN-<vt addr range>,<navigation_command>,[optional boolean_select_param]'"`
	Variables:
	• variable text address range = 1 - 4000.
	• navigation command.
	• optional select boolean
	Navigation Commands:
	• t or T - move to the top of the list (supports an optional select boolean).
	• b or B - move to the bottom of the list (supports an optional select boolean).
	• d or D - page down (DOES NOT support the optional select boolean. A select boolean will be ignored if present).
	• n - 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.)
	• u or U - page up (DOES NOT support the optional select boolean. A select boolean will be ignored if present).
	Examples:
	`SEND_COMMAND Panel,"'^LVN-42,B'"`
	    Move to the bottom of the list.
	`SEND_COMMAND Panel,"'^LVN-42,d'"`
	    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.

Listview Commands (Cont.)

^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-<vt addr range>,[optional refresh_interval],[optional force_reload]'"`
	 Variables:
	 • variable text address range = 1 - 4000.
	 • 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.
	
	 Example:
	 `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-<vt addr range>,<primary sort column name, secondary sort column name,..., final sort column name>,[optional sort type],[optional override sort syntax]'"`
	 Variables:
	 • variable text address range = 1 - 4000.
	 • 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 master Datafeed XML file generated by the master. Metadata fields are prepended with "meta" in front of the "ID" 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.
	
	 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 (Cont.)

^LVS (Cont.)	`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

Using Resource Images from TPDesign5 Resource Manager

In addition to using URLs to retrieve images via http from a Web server (as indicated in the previous examples), TPD5 Panel File resource images can be used as an image in a Listview item:

In a .csv file, the amxstandard.xml format, or via NetLinx data APIs, if the URL for an image that has been set via the **^LVM** command **[i1=$columnX]** (see page 164) does *not* contain a valid scheme (i.e http://, https://, or file://), then the text in the column mapped to the image field is assumed to be the name of a image in the TPD Resource Manager for a TPD5 file.

If the URL has a valid scheme, then the file is retrieved from the URL (via the server or filesystem).

Example - CSV Contents with URL Set to Retrieve Images via HTTP

The following is an example of CSV contents that use URLs to retrieve images via http on the *MediaServer* Web server:

ABC	1108	http://MediaServer/images/abc.png	free	PG-13
CBS	1111	http://MediaServer/images/cbs.png	free	PG-13
CNN	1124	http://MediaServer/images/cnn.jpg	free	PG-13
FOX	1104	http://MediaServer/images/fox.png	free	PG-13
HBO	1140	http://MediaServer/images/hbo.jpg	premium	R
NBC	1105	http://MediaServer/images/nbc.png	free	PG-13
SHO	1148	http://MediaServer/images/sho.png	premium	R

Example - CSV Contents with URL Set to Retrieve Images via HTTP

The following is an example of CSV contents with images set to be retrieved from TPD panel file Resource Manager. Note that since the second file does not have a valid scheme in the URL (http://, https://, file://), the images are assumed to be part of the TPD file:

ABC	1108	abc.png	free	PG-13
CBS	1111	cbs.png	free	PG-13
CNN	1124	cnn.jpg	free	PG-13
FOX	1104	fox.png	free	PG-13
HBO	1140	hbo.jpg	premium	R
NBC	1105	nbc.png	free	PG-13
SHO	1148	sho.png	premium	R

Drag and Drop

Overview

G5 Panels and TPDesign5 support "drag-and-drop" functionality for General and Multi-State General buttons. This function allows the end-user to initiate a drag on a button with a "long press", then drag and release (or "drop") the button onto a drop target (FIG. 208):
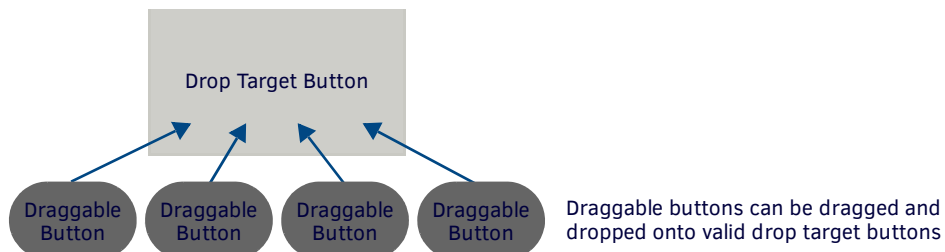
FIG. 208 Draggable buttons and Drop Target button

AMX System Requirements for Listview Buttons

The following software, hardware and firmware requirements must be met to support Drag and Drop functionality:

- TPDesign5 - version 1.3 (or higher)
- X Series G5 Touch Panels - panel firmware v1.3.23 (or higher)
- NetLinx Masters - master firmware v1.3.17 (or higher)

Draggable Buttons and Drop Target Buttons

To use the drag-and-drop function, the TPDesign5 project must include at least one "draggable" button, and at least one "drop target" button. *General* and *Multi-State General* Buttons (only) can be set as either a *Draggable* or as a *Drop Target* button, via the Drag/Drop Type (General) button property.

- **"Draggable"** buttons are buttons that can be long-pressed and dragged onto a drop target button.
- **"Drop Target"** buttons are buttons that serve as potential targets for draggable buttons.

Using Draggable Buttons (on the Touch Panel)

To use draggable buttons on a G5 touch panel (FIG. 209):
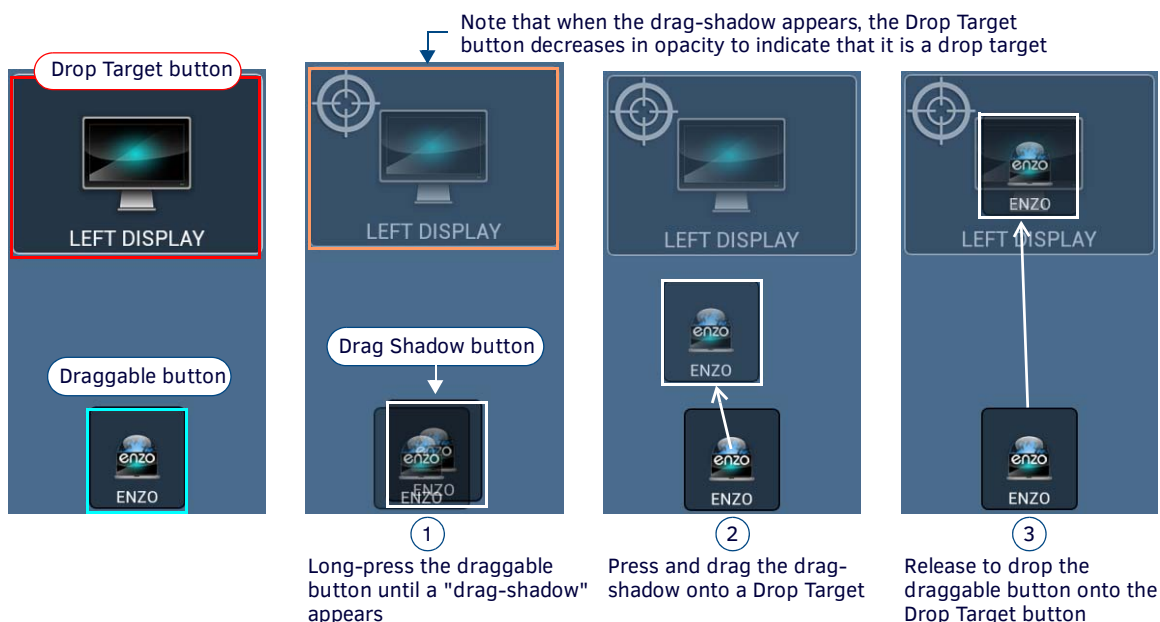
FIG. 209 Using Draggable Buttons on the Touch Panel

1. Long-press (*press and hold for 1 second*) the draggable button (1).
2. In approximately 1 second, a transparent copy of the button appears on the screen (2).
3. Drag the button onto a valid Drop Target button, and release to "drop" the draggable button (3).

NOTE: *As shown in FIG. 209, when the drag shadow appears, the target will decrease opacity to indicate it is a drop target. See page 183 for details on details on States properties for Drop Target buttons.*

Creating Drag and Drop Buttons - Examples

There are two demos at the end of this section that illustrate example workflows for configuring drag and drop buttons:

1. **Basic Demo - No Drop Groups**: This demo illustrates creating a set of draggable buttons that represent input devices, and a drop target button that represents an output device (a VTC) - see page 181.
2. **Advanced Demo - Three Drop Groups**: This demo illustrates creating a set of draggable buttons that represent input devices, and a set of drop target buttons that represent three output device (Displays) - see page 190.

Drag/Drop Type Button (General) Property

A new General property called "*Drag/Drop Type*" is available in TPDesign5 that sets the selected General or Multi-State General button as either "draggable" or as a "drop target" . By default, this property is set to "none" (FIG. 210):
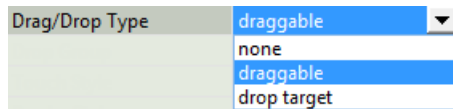
FIG. 210 General Property - Drag/Drop Type

General Property - Drag/Drop Type	
none:	The selected button is neither draggable or a drop target (default setting).
draggable:	With draggable selected, the user can drag the button on the touchpanel.
drop target:	When drop target is selected, the button acts as a target for a draggable button to be dropped on.

Drop Groups

Drop groups provide a means of a validity check for drop targets - they allow you to control which drop target buttons will serve as valid targets for draggable buttons.

Drop Groups are assigned to draggable buttons, and determine which Drop Target buttons are considered to be valid targets for each draggable button. Once a draggable button has a Drop Group assigned to it, only those drop targets that exist within the assigned Drop Group are valid targets. Conversely, draggable buttons are not allowed to be dragged and dropped onto an invalid drop target.

NOTE: *While Drop Groups are not a requirement for drag and drop functionality, they provide a powerful method of limiting drag and drop functionality to ensure an optimal user experience.*

Example - Grouping By Connection Type

FIG. 211 on page 169 provides an example of three Drop Groups being used to organize the source (input) devices that can be dragged onto each of four Video Output devices:

Drop Target buttons representing four Video Output devices

Drop Group A
accepts HDMI

Drop Group B
accepts Component

Drop Group C
accepts 4K

No Drop Group
Assigned

DROP TARGET:
Video Output Device 1

DROP TARGET:
Video Output Device 2

DROP TARGET:
Video Output Device 3

DROP TARGET:
Video Output Device 4

Accepts:
HDMI source

Accepts:
HDMI sources
Component source

Accepts:
HDMI source
Component source
4K source

Accepts:
HDMI source

Source Device:
HDMI

Source Device:
Component

Source Device:
4K

Source Device:
HDMI, 4K

Source Device:
HDMI, Component

Source Device:
HDMI

Draggable buttons representing six Source (Input) devices

FIG. 211 Using Drop Groups to Control targets for Draggable Buttons

- Each Video Output device is represented by a *Drop Target* button - each one supports a different set of inputs.
- Each Source (Input) device is represented by a *Draggable* button - each one provides a different type of source.

This example indicates three Drop Groups:

- **Drop Group A**: This group accepts all source inputs that provide HDMI input. Note that Drop Group A includes Video Output devices 1, 2 and 3, as all of these devices support HDMI.
- **Drop Group B**: This group accepts all source inputs that provide Component input. Note that Drop Group B includes Video Output devices 2 and 3, since both devices support Component.
- **Drop Group C**: This group accepts all source inputs that provide 4K input. Note that Drop Group C includes only Video Output device 3, since it is the only one that supports 4K.
- Note that Video Output Device 4 has no Drop Group assignment. Therefore, the only source inputs allowed to be dragged and dropped on this Drop Target are those that also have no Drop Group assignment.

In this example, Drop Groups prevent Input devices from being dragged and dropped onto incompatible video output devices. With the configuration indicated in FIG. 211, HDMI sources are only allowed to be dragged and dropped onto Video Output devices that support HDMI. Likewise, 4K sources are not allowed to be dragged and dropped onto Video Output devices that support do not support 4K input.

NOTE: *Use multiple states on drop target buttons to display a specific bitmap on the drop target button based on Drop Group assignments. For example, when a draggable button is dragged onto a valid drop target, a bitmap can be displayed on the drop target button to indicate that it is a valid target for the selected button. Conversely, a different image can be used to indicate that the drop target is invalid for the selected button.*

Drop Group Button (General) Property

A new General property called "*Drop Group*" is available in TPDesign5 (v1.3 or higher) that associates the selected Draggable button with a specific Drop Group (FIG. 212):

Drop Group	group_1 ▼
Touch Style	none
Border Style	group_1
Disabled	group_2
	group_3

In this example, three Drop Groups have been created
(via the Drop-Target Groups dialog)
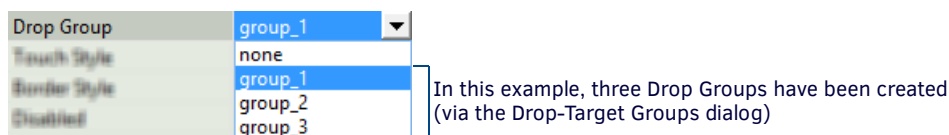
FIG. 212 General Property - Drop Group

General Property - Drop Group	
none:	The selected button is not associated with a Drop Group
Drop Groups:	Select the Drop Group to which the selected draggable button will be associated. Drop Groups are created in the *Drop-Target Groups* dialog (see page 198).

Note that this property is only available for General and Multi-State General buttons that have been set as *Draggable* via the *Drag/Drop Type* (General) property (see page 168).

Drop Groups - Notes

- Drop Group names are case-insensitive.
- Only drop targets can be grouped.
- Drop targets can exist in multiple Drop Groups.
- A draggable button can only have 1 Drop Group assigned to it.
- If no group is assigned to a draggable button, then only drop targets that are not assigned to Drop Groups are valid targets. Note that this is the default (and simplest) use case: it allows designers to quickly create a page with draggables that can be dropped on any drop target with no additional configuration required.

Drag and Drop-Specific Events

TPDesign5 (1.3.23 or higher) supports a set of new Events for Draggable and Drop Target buttons:

Events for Draggable Buttons

Draggable buttons support two drag-specific Events (FIG. 213):
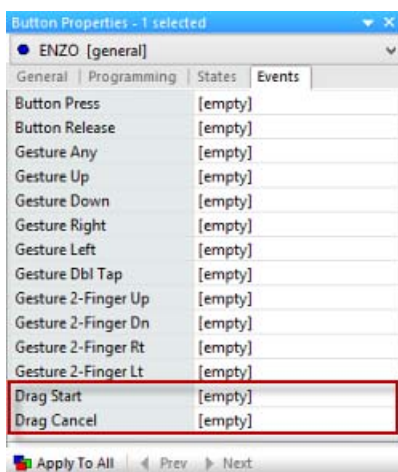
FIG. 213 Drag-Specific Events

- **Drag Start**: The event will occur when the specified draggable button has initiated a drag. Drag starts are initiated by a long press on a draggable button.
- **Drag Cancel**: The event will occur when the specified draggable button has been dropped outside of a valid drop target.

Events for Drop Target Buttons

Drop Target buttons support three drop-specific Events (FIG. 214):
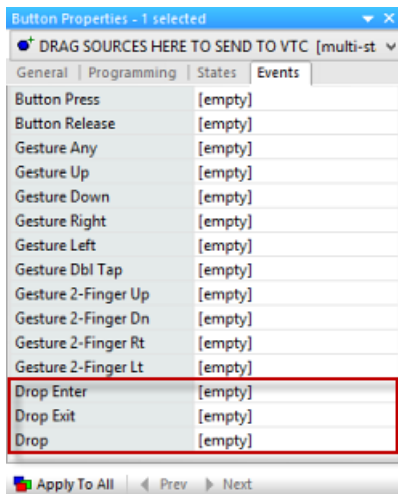
FIG. 214 Drop-Specific Events

- **Drop Enter**: The event will occur when a draggable button has entered a valid drop target.
- **Drop Exit**: The event will occur when a draggable button has exited a valid drop target.
- **Drop**: The event will occur when a draggable button has been dropped onto a valid drop target.

Custom Event Parameters for Drag and Drop Events

The events are:

- ActionDragStarted - a draggable button has initiated a drag
- ActionDragCancel - a draggable button has been dropped outside of a valid target
- ActionDropEntered - a draggable button has entered a valid target
- ActionDrop Exited - a draggable button has exited a valid target
- ActionDrop - a draggable button has been dropped on a valid target

Also, the Drag/Drop events provide predefined variables that are populated when action event occurs. These values can be used in the custom event definition:

DragEvent and DropEvent Parameters	
DragEvent Parameters	**DropEvent Parameters**
• ${dragChannelPort}	• ${dropChannelPort}
• ${dragChannelCode}	• ${dropChannelCode}
• ${dragAddressPort}	• ${dropAddressPort}
• ${dragAddressCode}	• ${dropAddressCode}
• ${dragGroupName}	• ${dragChannelPort}
• ${dragButtonName}	• ${dragChannelCode}
• ${dragPageName}	• ${dragAddressPort}
• ${dragInfo}	• ${dragAddressCode}
• ${dropTargetsValid}	• ${dragGroupName}
• ${dropTargetsInvalid}	• ${dragButtonName}
	• ${dragPageName}
	• ${dragInfo}
	• ${dropTargetsValid}
	• ${dropTargetsInvalid}

Creating Draggable Buttons

"Draggable" buttons are buttons that can be long-pressed and dragged onto a drop target button.

To create a Draggable button:

1. Create or select a General or Multi-State General button.
2. In the Properties window (General tab), click on the Drag/Drop Type property to open the drop-down menu (FIG. 215): .
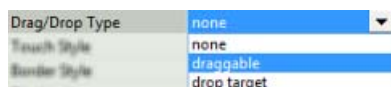
FIG. 215 Drag/Drop Type (General) property

3. Select **draggable**.

Creating Drop Target Buttons

"Drop Target" buttons are buttons that serve as potential targets for draggable buttons.

To create a Drop Target button:

1. Create or select a General or Multi-State General button.
2. In the Properties window (General tab), click on the Drag/Drop Type property to open the drop-down menu (FIG. 216):
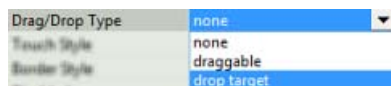
FIG. 216 Drag/Drop Type (General) property

3. Select **drop target**.

Creating Drop Groups

Drop Groups are created, edited and deleted via the Drop-Target Groups dialog.

To create a new Drop Group:

1. Select **Panel > Edit Drop-Target Groups** (or click the toolbar button) to open the *Drop-Target Groups* dialog.
2. Click **New Group** to open the *Create Drop Target Group* dialog.
3. Enter a unique name for the new Drop Group in the *Group Name* field (FIG. 217):
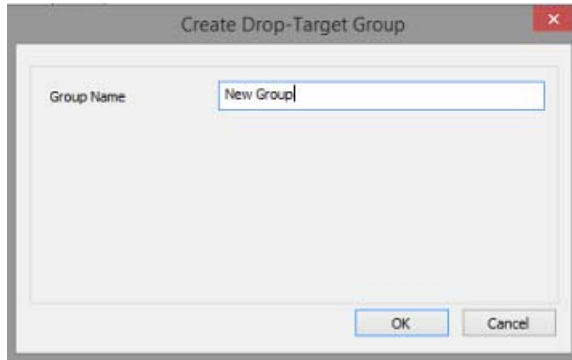
FIG. 217 Create Drop Target Group dialog

4. Click **OK** to save changes and close the *Create Drop-Target Groups* dialog.
    - The new Drop Group is listed in the *Drop Target Groups* window of the *Drop-Target Groups* dialog (FIG. 218):
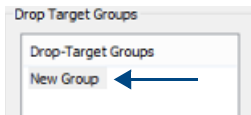
FIG. 218 Drop-Target Groups dialog - Drop Target Groups window

    - Once a Drop Group has been created, it is available for selection for Draggable buttons, via the Drop Group (General) property (FIG. 219):
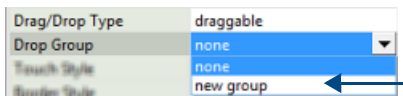
FIG. 219 Drop-Target Groups dialog - Drop Target Groups window

Editing Drop Groups

Drop Groups are created, edited and deleted via the Drop-Target Groups dialog.

To edit an existing Drop Group:

1. Select **Panel > Edit Drop-Target Group**s (or click the toolbar button) to open the *Drop-Target Groups* dialog (FIG. 220):
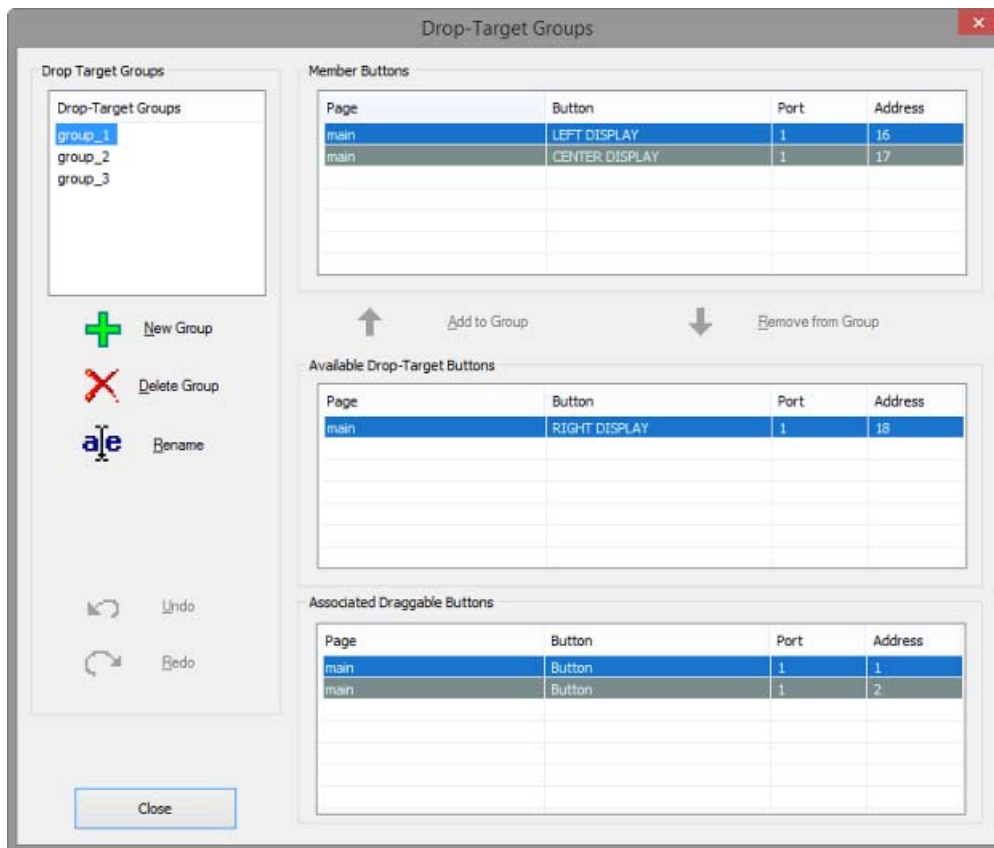
FIG. 220 Drop-Target Groups dialog

2. Select a Drop Group in the Drop Target Groups window. This populates the Member Buttons and Available Drop-Target Buttons windows:
    - The Member Buttons window indicates all Drop Target buttons that are currently members of the selected Drop Group. These Drop Target buttons can be removed from the selected Drop Group.
    - The Available Drop-Target Buttons window indicates all Drop Target buttons that are not currently members of the selected group. These Drop Target buttons are available to add to the selected Drop Group.

Adding Member Buttons to a Drop Group

1. In the *Drop-Target Groups* dialog - *Drop Target Groups* window, select the Drop Group to which you want to add one or more Member Buttons.

2. All Drop Target buttons that have been created in this project (and that are not already member buttons for the selected Drop Group) are listed in the Available Drop-Target Buttons window (FIG. 221):
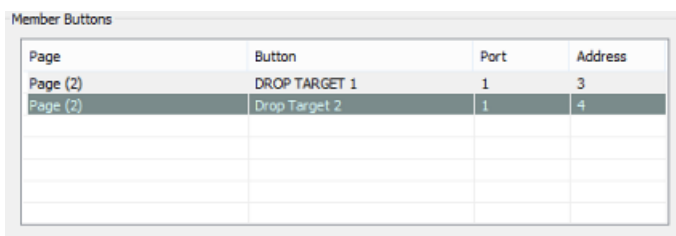
FIG. 221 Drop-Target Groups dialog - Available Drop Target Buttons window

3. Select a Drop Target button and click **Add to Group** to add the selected Drop Target button to the selected Drop Group.

4. Click **Close** to save changes and close the *Drop-Target Groups* dialog.

Deleting Member Buttons from a Drop Group

1. In the *Drop-Target Groups* dialog - *Drop Target Groups* window, select the Drop Group from which you want to delete one or more Member Buttons.

2. In the *Member Buttons* window, select the Drop Target button that you want to delete from the selected Drop Group (FIG. 222):

FIG. 222 Drop-Target Groups dialog - Member Buttons window

3. Click **Remove From Group**. The selected Drop Target button is removed from the Member Buttons list and re-added to the *Available Drop-Target Buttons* list.

4. Click **Close** to save changes and close the *Drop-Target Groups* dialog.

Deleting a Drop Group

1. In the *Drop-Target Groups* dialog - *Drop Target Groups* window, select the Drop Group that you want to delete from the project.

2. Click **Delete Group**. The selected Drop Group is removed from the project (and from the *Drop Target Groups* list).

3. Click **Close** to save changes and close the *Drop-Target Groups* dialog.

Renaming a Drop Group

1. In the *Drop-Target Groups* dialog - *Drop Target Groups* window, select the Drop Group that you want to rename.

2. Click **Rename Group**. The selected Drop Group is renamed. The new name is indicted in the *Drop Target Groups* list.

3. Click **Close** to save changes and close the *Drop-Target Groups* dialog.

^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 master whenever certain operations occur. For example, the system programmer may want to be notified whenever a drag button enters an acceptable target.

NOTE: *When using the ^BDC command, it is not necessary to assign button specific event actions. These can be empty if the ^BDC command is used. If these are defined, the action generated does not have to conform to the custom event definition as set in the ^BDC command. If the ^BDC command events are enabled, and button specific actions (i.e. custom event action) as both defined, then both will be sent when an event occurs.*

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 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 defaults when the panel restarts.

By default the ^BDC command is *enabled*, and the default values are:

- DragStartedEvent = 1410
- DropEnterEvent = 1411
- DropExitEvent = 1412
- DropEvent = 1413
- DragCancelEvent = 1414

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

Syntax

```
"'^BDC-[optional DragStarted event num], [optional DropEntered event num],
  [optional DropExited event num], [optional Drop event num], [optional DragCancel event num]'"
```

Variables

- DragStarted Event Number = 0 for no event or a value from 32001 to 65535.
- DropEntered Event Number = 0 for no event or a value from 32001 to 65535.
- DropEntered Event Number = 0 for no event or a value from 32001 to 65535.
- Drop Event Number = 0 for no event or a value from 32001 to 65535.
- DragCancel Event Number = 0 for no event or a value from 32001 to 65535.

Events

- DragStarted - a draggable button has initiated a drag
- DropEntered - a draggable button has entered a valid target
- DropExited - 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

In response to any or all of the above events, the panel will create a custom event which is then sent to the master. The format of **START** custom events transmitted to the master are as follows:

Format - START custom events	
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)
• ch	channel (port,channel)
• ad	address (port,address)
• nm	button name

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}

A NetLinx .AXI file that can provide routines to parse the drag and drop info strings can be found on page 176.

The format of **ENTER/EXIT/CANCEL** custom events transmitted to the master are as follows:

Format - ENTER/EXIT/CANCEL custom events	
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	0
CUSTOM.VALUE3	0
CUSTOM.TEXT	""

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

Format - DROP custom event	
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 master.

```
CUSTOM.TYPE = 10   (the dropTarget receives the drop event)
CUSTOM.ID = 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
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)
```

DragDrop.axi

The NetLinx .AXI file below provides routines to parse the drag and drop info strings:

```
PROGRAM_NAME='DragDrop'

(***********************************************************)
(*          DEVICE NUMBER DEFINITIONS GO BELOW           *)
(***********************************************************)
DEFINE_DEVICE

(***********************************************************)
(*              CONSTANT DEFINITIONS GO BELOW            *)
(***********************************************************)
DEFINE_CONSTANT

#IF_NOT_DEFINED __DRAG_DROP_MAX_TARGETS__
#DEFINE __DRAG_DROP_MAX_TARGETS__ '__DRAG_DROP_MAX_TARGETS=100'
INTEGER __DRAG_DROP_MAX_TARGETS = 100;
#END_IF

#IF_NOT_DEFINED __DRAG_DROP_NUM_PANELS__
#DEFINE __DRAG_DROP_NUM_PANELS__ '__DRAG_DROP_NUM_PANELS=1'
INTEGER __DRAG_DROP_NUM_PANELS = 1;
#END_IF


(***********************************************************)
(*              DATA TYPE DEFINITIONS GO BELOW           *)
```

```
(***********************************************************)
DEFINE_TYPE
STRUCTURE __DRAG_DROP_sDragObject
{
    INTEGER chanPort;
    INTEGER chan;
    INTEGER addrPort;
    INTEGER addr;
    char    groupName[100];
    char    buttonName[100];
}

STRUCTURE __DRAG_DROP_sDropTargetObject
{
    INTEGER valid;
    INTEGER chanPort;
    INTEGER chan;
    INTEGER addrPort;
    INTEGER addr;
    char    buttonName[100];
}

(***********************************************************)
(*              VARIABLE DEFINITIONS GO BELOW             *)
(***********************************************************)
DEFINE_VARIABLE

VOLATILE __DRAG_DROP_sDragObject __DRAG_DROP_current_drag[__DRAG_DROP_NUM_PANELS];
VOLATILE __DRAG_DROP_sDropTargetObject __DRAG_DROP_current_targets[__DRAG_DROP_NUM_PANELS]
                                                     [__DRAG_DROP_MAX_TARGETS];
VOLATILE INTEGER __DRAG_DROP_target_count[__DRAG_DROP_NUM_PANELS];
VOLATILE INTEGER __DRAG_DROP_panel_devices[__DRAG_DROP_NUM_PANELS]


(***********************************************************)
(*        SUBROUTINE/FUNCTION DEFINITIONS GO BELOW        *)
(***********************************************************)
(* EXAMPLE: DEFINE_FUNCTION <RETURN_TYPE> <NAME> (<PARAMETERS>) *)
(* EXAMPLE: DEFINE_CALL '<NAME>' (<PARAMETERS>) *)

DEFINE_FUNCTION __DRAG_DROP_SET_PANELS(INTEGER panels[])
{
    if(LENGTH_ARRAY(panels) <= __DRAG_DROP_NUM_PANELS)
    {
        __DRAG_DROP_panel_devices = panels;
    }
    else
    {
        STACK_VAR INTEGER count;

        for(count = 1 ; count <= __DRAG_DROP_NUM_PANELS; count++)
        {
            __DRAG_DROP_panel_devices[count] = panels[count];
        }
        SET_LENGTH_ARRAY(__DRAG_DROP_panel_devices,count);
    }
}

DEFINE_FUNCTION __DRAG_DROP_CLEAR_DATA(INTEGER panel)
{
    STACK_VAR INTEGER count;
    __DRAG_DROP_current_drag[panel].chanPort = 0;
    __DRAG_DROP_current_drag[panel].chan     = 0;
    __DRAG_DROP_current_drag[panel].addrPort = 0;
    __DRAG_DROP_current_drag[panel].addr     = 0;
    __DRAG_DROP_current_drag[panel].buttonName = '';
    __DRAG_DROP_current_drag[panel].groupName  = '';

    count = LENGTH_ARRAY(__DRAG_DROP_current_targets[panel]);
    if(count > 0)
    {
        STACK_VAR INTEGER x;
        for(x = 1; x <= count; x++)
        {
            __DRAG_DROP_current_targets[panel][x].chanPort = 0;
            __DRAG_DROP_current_targets[panel][x].chan = 0;
            __DRAG_DROP_current_targets[panel][x].addrPort = 0;
```

```
            __DRAG_DROP_current_targets[panel][x].addr = 0;
            __DRAG_DROP_current_targets[panel][x].buttonName = '';
            __DRAG_DROP_current_targets[panel][x].valid = 0;
        }
    }
    __DRAG_DROP_target_count[panel] = 0;
}

DEFINE_FUNCTION INTEGER __DRAG_DROP_PARSE_PORT_VALUE(
                          CHAR line[],
                          INTEGER start,
                          INTEGER port,
                          INTEGER value)
{
    STACK_VAR INTEGER x, run, state;
    STACK_VAR char ch;

    x = start;
    run = 1;
    state = 0;
    ch = 0;

    port = 0;
    value = 0;
    while(run)
    {
        ch = line[x];
        switch(state)
        {
            case 0:
            {
                if(ch >= '0' && ch <= '9')
                {
                    port = port * 10 + (ch-'0');
                }
                else if(ch == ',')
                {
                    state = 1;
                }
            }
            case 1:
            {
                if(ch >= '0' && ch <= '9')
                {
                    value = value * 10 + (ch-'0')
                }
                else if(ch == ':')
                {
                    run = 0;
                }
                else if(ch == '}')
                {
                    run = 0;
                }
            }
        }
        x++;
    }
    return x;
}

DEFINE_FUNCTION INTEGER __DRAG_DROP_PARSE_NAME(CHAR line[],
                                                INTEGER start,
                                                CHAR value[])
{
    STACK_VAR INTEGER end;

    value = '';
    end = FIND_STRING(line,':',start);
    if(end <= 0)
    {
        end = FIND_STRING(line,'}',start);
    }
    if(end > start)
    {
        value = MID_STRING(line,start,end-start);
        return end+1;
```

```
        }
        return start+1;

}

DEFINE_FUNCTION INTEGER __DRAG_DROP_PARSE_VALUE(CHAR line[],
                                                INTEGER start,
                                                INTEGER value)
{
    STACK_VAR INTEGER x, run;
    STACK_VAR INTEGER ch;

    x = start;
    run = 1;
    ch = 0;
    value = 0;
    while(run)
    {
        ch = line[x]
        if(ch >= '0' && ch <= '9')
        {
            value = value * 10 + (ch-'0');
        }
        else if(ch == ':')
        {
            run = 0;
        }
        else if(ch == '}')
        {
            run = 0;
        }
        x++;
    }
    return x;
}

DEFINE_FUNCTION __DRAG_DROP_PARSE_DRAG_START(INTEGER panel, TCUSTOM s)
{
    STACK_VAR char line[200],text[2000];
    STACK_VAR INTEGER length,index;

    length = 0;
    __DRAG_DROP_CLEAR_DATA(panel);

    text = s.text;
    line = REMOVE_STRING(text,"10",1);
    length = LENGTH_STRING(line);
    while( length > 0)
    {
        if(FIND_STRING(line,'dr{',1) == 1)
        {
            index = 4;
            while (index < length)
            {
                SELECT
                {
                    ACTIVE(FIND_STRING(line,'ch=',index) == index) :
                    {
                        index = __DRAG_DROP_PARSE_PORT_VALUE(line,index+3,
                            __DRAG_DROP_current_drag[panel].chanPort,
                            __DRAG_DROP_current_drag[panel].chan);
                    }
                    ACTIVE(FIND_STRING(line,'ad=',index) == index) :
                    {
                        index = __DRAG_DROP_PARSE_PORT_VALUE(line,index+3,
                            __DRAG_DROP_current_drag[panel].addrPort,
                            __DRAG_DROP_current_drag[panel].addr);
                    }
                    ACTIVE(FIND_STRING(line,'gp=',index) == index) :
                    {
                        index = __DRAG_DROP_PARSE_NAME(line,index+3,
                            __DRAG_DROP_current_drag[panel].groupName);
                    }


                    ACTIVE(FIND_STRING(line,'nm=',index )== index) :
```

```
                    {
                        index = __DRAG_DROP_PARSE_NAME(line,index+3,
                            __DRAG_DROP_current_drag[panel].buttonName);
                    }
                    ACTIVE(1) :
                    {
                        index = length;
                    }
                }
            }
        }
        else if(FIND_STRING(line,'dt{',1) == 1)
        {
            index = 4;
            __DRAG_DROP_target_count[panel]++;

            while (index < length)
            {
                SELECT
                {
                    ACTIVE(FIND_STRING(line,'vl=',index) == index) :
                    {
                        if(line[index+3] == '1')
                        {
                            __DRAG_DROP_current_targets[panel][__DRAG_DROP_target_count[panel]].valid = 1;
                        }
                        else
                        {
                            __DRAG_DROP_current_targets[panel][__DRAG_DROP_target_count[panel]].valid = 0;
                        }
                        index = index+5;
                    }
                    ACTIVE(FIND_STRING(line,'ch=',index) == index) :
                    {
                        index = __DRAG_DROP_PARSE_PORT_VALUE(line,index+3,
                            __DRAG_DROP_current_targets[panel][__DRAG_DROP_target_count[panel]].chanPort,
                            __DRAG_DROP_current_targets[panel][__DRAG_DROP_target_count[panel]].chan);
                    }
                    ACTIVE(FIND_STRING(line,'ad=',index) == index) :
                    {
                        index = __DRAG_DROP_PARSE_PORT_VALUE(line,index+3,
                            __DRAG_DROP_current_targets[panel][__DRAG_DROP_target_count[panel]].addrPort,
                            __DRAG_DROP_current_targets[panel][__DRAG_DROP_target_count[panel]].addr);
                    }
                    ACTIVE(FIND_STRING(line,'nm=',index) == index) :
                    {
                        index = __DRAG_DROP_PARSE_NAME(line,index+3,
                            __DRAG_DROP_current_targets[panel][__DRAG_DROP_target_count[panel]].buttonName);
                    }
                    ACTIVE(1) :
                    {
                        index = length;
                    }
                }
            }
        }
        line = REMOVE_STRING(text,"10",1);
        length = LENGTH_STRING(line);
    }
    SET_LENGTH_ARRAY(__DRAG_DROP_current_targets[panel],__DRAG_DROP_target_count[panel]);

}

DEFINE_FUNCTION __DRAG_DROP_PRINT_DATA(INTEGER panel)
{
    STACK_VAR INTEGER x;

    SEND_STRING 0,"FORMAT('drag ch=%d',__DRAG_DROP_current_drag[panel].chanPort),
                   FORMAT(',%-5d',__DRAG_DROP_current_drag[panel].chan),
                   FORMAT('ad=%d',__DRAG_DROP_current_drag[panel].addrPort),
                   FORMAT(',%-5d',__DRAG_DROP_current_drag[panel].addr),
                   'gp=''',__DRAG_DROP_current_drag[panel].groupName,''' bn=''',
                       __DRAG_DROP_current_drag.buttonName,''''";
    for(x = 1; x <= __DRAG_DROP_target_count[panel]; x++)


    {
```

```
        SEND_STRING 0,"FORMAT('target ch=%d',__DRAG_DROP_current_targets[panel][x].chanPort),
                       FORMAT(',%-5d',__DRAG_DROP_current_targets[panel][x].chan),
                       FORMAT('ad=%d',__DRAG_DROP_current_targets[panel][x].addrPort),
                       FORMAT(',%-5d',__DRAG_DROP_current_targets[panel][x].addr),
                       'bn=''',__DRAG_DROP_current_targets[panel][x].buttonName,
                       FORMAT(''' valid=%d',__DRAG_DROP_current_targets[panel][x].valid)";
    }
}
```

Basic Demo - No Drop Groups

The following instructions illustrate creating a set of *draggable* buttons that represent input devices, and a *drop target* button that represents an output device (a VTC).

NOTE: *This set of instructions uses files that are included in the "DragAndDropNoGroups" demo file which is available to download from the UI RESOURCE CENTER at www.amx.com.*

The resulting demo page will provide four draggable buttons that represent source (input) devices, and one drop target button representing an output (VTC) device. End users will be able to switch sources on the VTC by dragging and dropping a draggable button within the bounds of a the drop target button (FIG. 223):
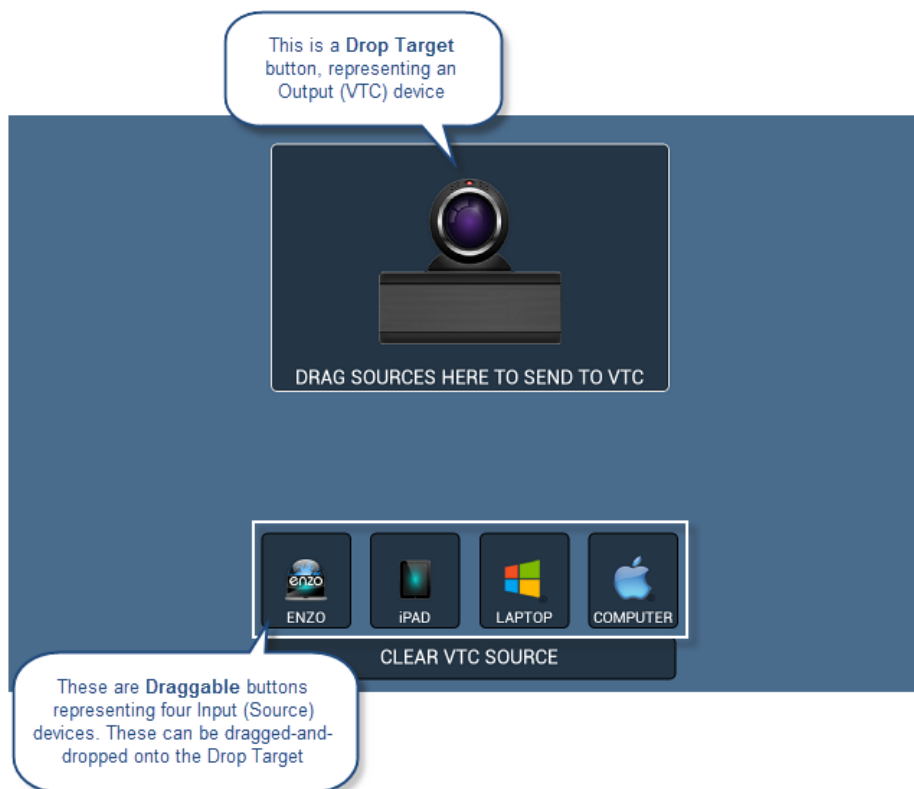
FIG. 223 Drag and Drop Demo - Four (draggable) Input buttons and one (drop target) Output button

Before You Begin

Download the *DragAndDropNoGroups.zip* file from www.amx.com and extract its contents to a known location. This ZIP file contains the following files, all of which are required for the demo described in this manual:

- **DragAndDropNoGroups.TP5** - A TPDesign5 project file, as well as all of the image files used by the Page and Buttons in this project:

  - icon-apple.png
  - icon-enzo.png
  - icon-iPad.png
  - icon-windows8.png
  - vtc.png

- **DragAndDropNoGroups.apw** - A NetLinx Studio workspace file that contains the NetLinx code:

  - DragAndDropNoGroups.axs
  - DragAndDropNoGroups.src
  - DragAndDropNoGroups.tkn
  - DragAndDropNoGroups.tko

1) Create a TPDesign5 Project/Import Images

In order to display the images on the page and buttons shown in this demo, the image files must be added to the project, via the Resource Manager - *Images* tab:

1. Open TPDesign5 (v1.3 or higher) and start a new Project (**File > New**).

2. Open the Resource Manager to the *Images* tab.

3. Click **Import** to locate and select all of the image files that were included in the *DragAndDropNoGroups.ZIP* file.

4. Click **OK** to import the selected files and return to the Resource Manager (FIG. 224):
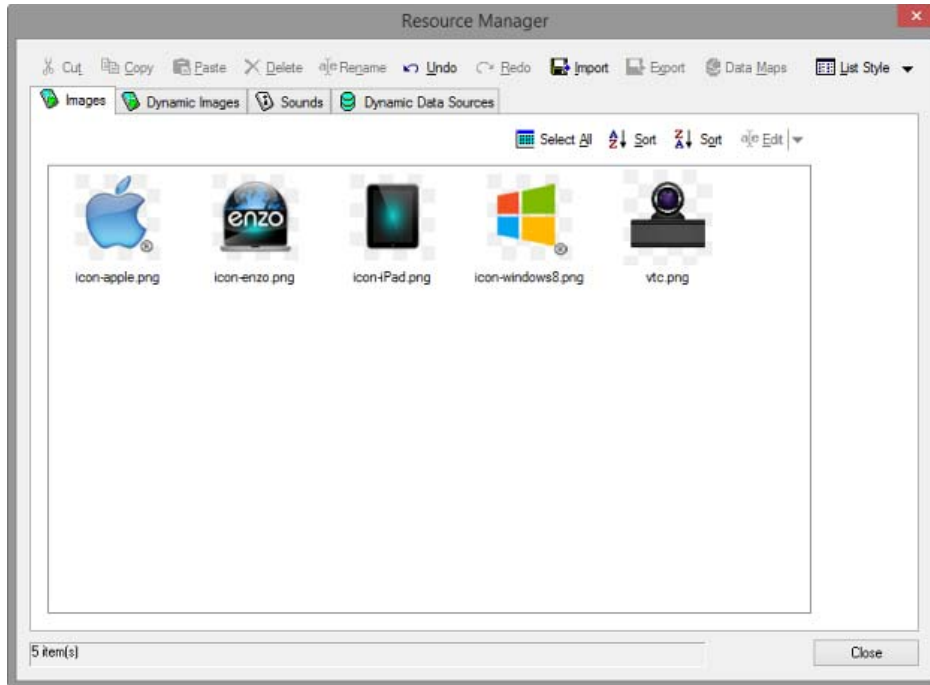
FIG. 224 Resource Manager Images tab - DragAndDropNoGroups Demo images imported

5. Click **Close** to close the Resource Manager.

> **NOTE:** *The DragAndDropNoGroups.TP5 file in the Drag and Drop demo has the images shown above already imported into the project.*

2) Create & Configure a Drop Target Button

In this example, there is only a single Drop Target button that will represent the Output Device (VTC) that can accept input from the source devices represented by the draggable buttons.

Create a Drop Target Button

1. Use the Button Draw tool to create a new button.

2. Set the button's *Type* (General) property to **multi-state general** (FIG. 225):

Only General and Multi-State General buttons can be used as Draggable or Drop Target buttons

FIG. 225 TPDesign5 General Properties - Type set to "general"

3. Set the button's *Drag/Drop Type* (General) property to **drop target** (FIG. 226):

FIG. 226 TPDesign5 General Properties - Drag/Drop Type set to "drop target"

Set Drop Target Button Properties - General

Set the remaining *General* properties for the Drop Target button as shown in FIG. 227:
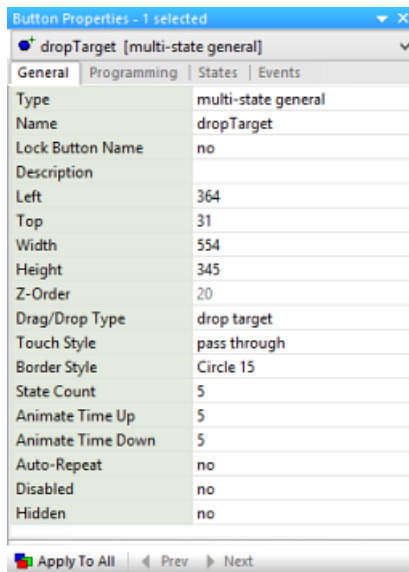
FIG. 227 Drop Target Button – General Properties

Set Drop Target Button Properties - Programming

Set the *Programming* properties for the Drop Target button as shown in FIG. 228:
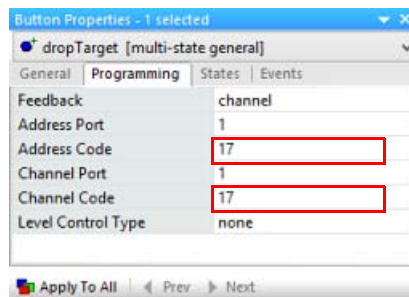
FIG. 228 Drop Target Button – Programming Properties

On the Drop Target button, set the *Address Code* to **17** and set the *Channel Code* to **17**.

Set Drop Target Button Properties - States

In this example, this button will represent the output (VTC) device. Use the *Text* (States) property to add the following labels to the button (*All States*): **DRAG SOURCES HERE TO SEND TO VTC**.

Use the *Bitmaps* (States) property to add the VTC bitmap to the button (*All States*):

- Note that all images must first be imported in to the project via the Resource Manager in order to be available to apply to buttons or pages in the project. The images used in this demo are pre-loaded in the TP5 project file.
- Select the drop target button and under *All States*, apply the bitmap: **VTC.png**.
- In this example, the *Bitmap Justification* is set to **center-middle**.

Drop target buttons can use states to provide a visual indication of target validity for draggable buttons. In this example, if a drag is started on a draggable button, the opacity of the drop target button is reduced to indicate that it is a drop target.

1. Select the Drop Target button and open the State Manager window.
2. In the State Manager window, select *State 1*, and set the State properties as shown in FIG. 229:
3. In the State Manager window, select *State 2*, and set the State properties as shown in FIG. 229:
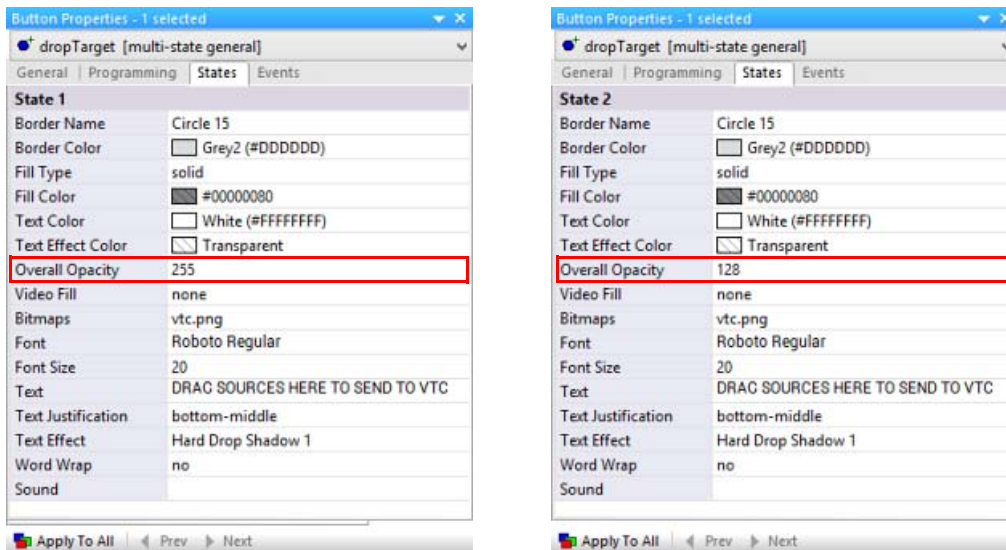
FIG. 229 State Manager context menu - Insert States

> **NOTE:** *In this example the only difference between the two states is the Overall Opacity property setting: State one uses "255" (totally opaque), and State 2 uses "128" (half-opacity). This provides a visual indication that this button is a drop target.*

4. The two states are indicated in the State Manager window (FIG. 230):

FIG. 230 State Manager window indicating five states

3) Create & Configure Draggable Buttons

In this example, four draggable buttons represent four source (input) devices that are used as the input for the VTC Output device represented by the Drop Target button.

Create Four Draggable Buttons

1. In TPDesign5, open a Page and use the Button Draw tool to create a new button.

2. Set the button's *Type* (General) property to **general** (FIG. 231):

Only General and Multi-State General buttons can be set as draggable or Drop Target
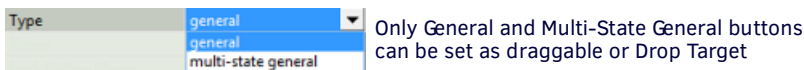
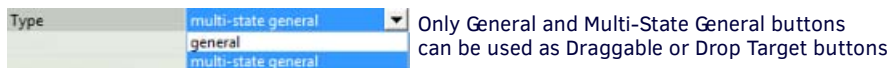**FIG. 231** TPDesign5 General Properties - Type set to "general"

3. Set the button's *Drag/Drop Type* (General) property to **draggable** (FIG. 232):
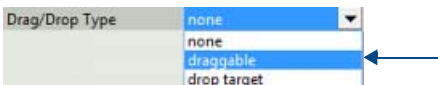
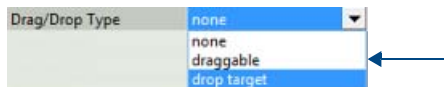**FIG. 232** TPDesign5 General Properties - Drag/Drop Type set to "draggable"

4. Repeat these steps to create a total of four draggable buttons. Alternatively, copy and paste the new button three times (FIG. 233):

FIG. 233 Draggable Buttons

Set Draggable Button Properties - General

Set the remaining *General* properties for the draggable buttons as shown in FIG. 234:

Replace with "iPAD", "LAPTOP", and "COMPUTER" for the other three draggable (source) buttons
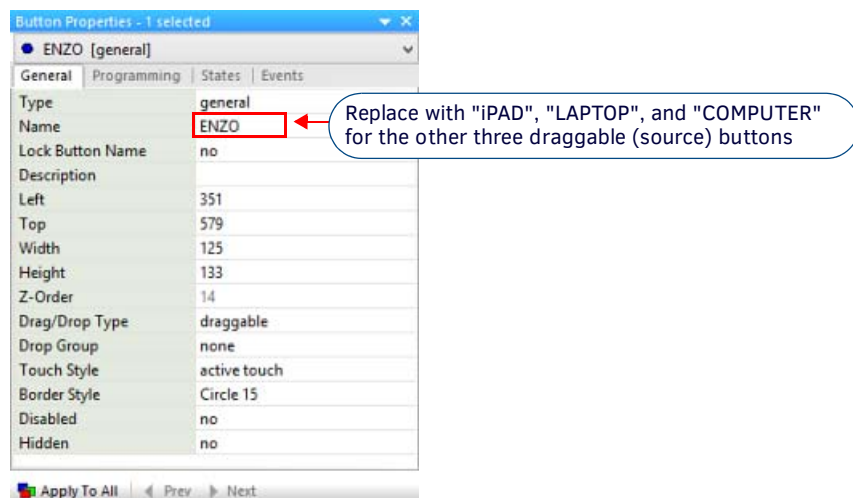
FIG. 234 Draggable Buttons - General Properties

Set Draggable Button Properties - Programming

Each of the draggable buttons needs to be configured with unique Address and Channel Codes. For this example, set the Address/ Channel Codes as shown below (FIG. 235):

ENZO button selected

iPAD button selected

LAPTOP button selected

COMPUTER button selected
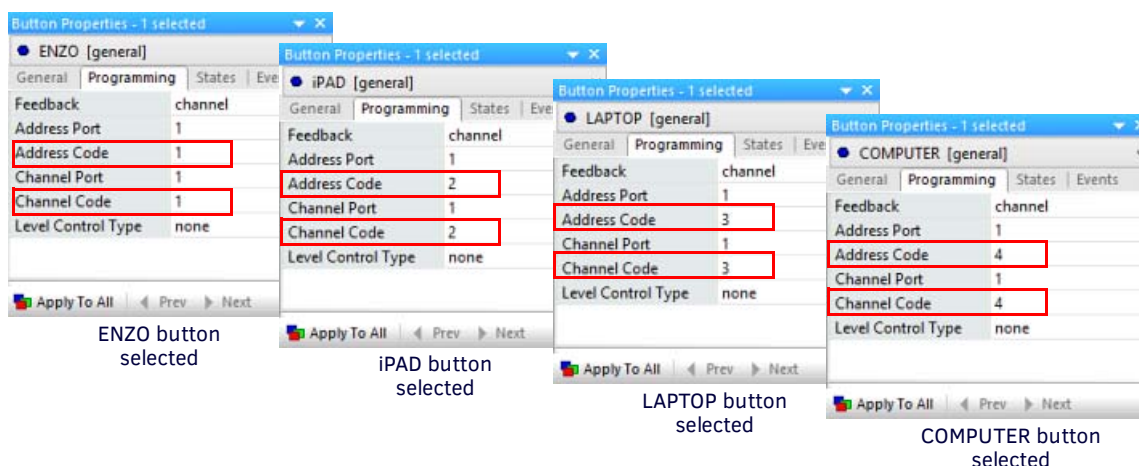
FIG. 235 Draggable Buttons - Programming Properties

- On the ENZO draggable button, set the *Address Code* to **1** and set the *Channel Code* to **1**.
- On the iPAD draggable button, set the *Address Code* to **2** and set the *Channel Code* to **2**.
- On the LAPTOP draggable button, set the *Address Code* to **3** and set the *Channel Code* to **3**.
- On the COMPUTER draggable button, set the *Address Code* to **4** and set the *Channel Code* to **4**.

Set Draggable Button Properties - States

In this example, each of these buttons will represent a different type of input (source) device. Edit the buttons to add text and icons to indicate the specific device represented by each button:

1. Use the *Text* (States) property to add labels to each of the buttons. Select each button and under *All States*, enter the following labels: **ENZO**, **iPAD**, **LAPTOP** and **COMPUTER**.

2. Use the *Bitmaps* (States) property to apply an appropriate icon to each of the buttons.

   NOTE: *All images must first be imported in to the project via the Resource Manager in order to be available to apply to buttons or pages in the project. The images used in this demo are pre-loaded in the TP5 project file.*

   - Select each button and under *All States*, apply the following bitmaps: **icon-enzo.png**, **icon-iPad.png**, **icon-windows8.png** and **icon-apple.png**.
   - In this example, the *Bitmap Justification* is set to **top-middle** for all four draggable buttons.

Set the remaining *States* properties for the draggable buttons as shown in FIG. 236:
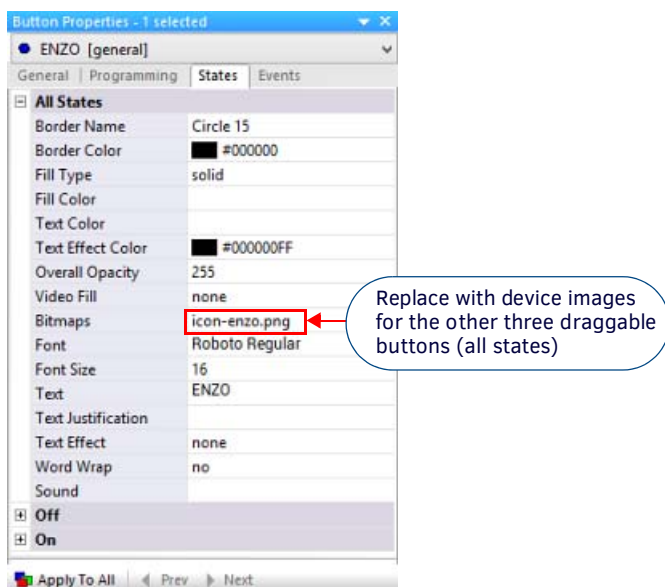
FIG. 236 Draggable Buttons - States Properties (All States shown)

For this example, the draggable buttons should look similar to the buttons shown below (FIG. 237):

FIG. 237 Draggable Buttons (Representing four Input Devices)

4) Create and Configure a "CLEAR VTC SOURCE" Button

This example includes the option for the user to "clear" the current input (Source) device setting on the VTC (FIG. 238):

FIG. 238 CLEAR VTC SOURCE button

To add a button that supports this option:

Create a "CLEAR VTC SOURCE" Button

1. Use the Button Draw tool to create a new button.
2. Set the button's *Type* (General) property to **general**.

Set "CLEAR VTC SOURCE" Button Properties - General

Set the remaining *General* properties for the "CLEAR VTC SOURCE" buttons as shown in FIG. 239:
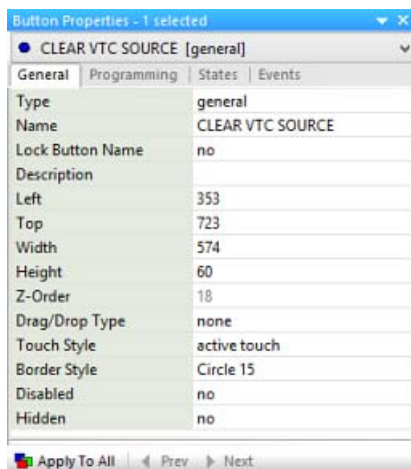
FIG. 239 "CLEAR VTC SOURCE" Button - General Properties

Set "CLEAR VTC SOURCE" Button Properties - Programming

Set the *Programming* properties for the "CLEAR VTC SOURCE" button as shown in FIG. 240:
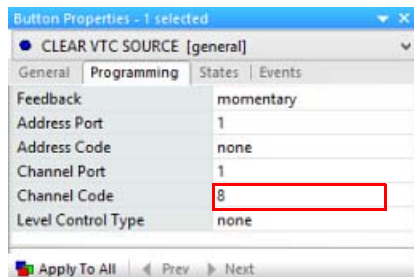
FIG. 240 "CLEAR VTC SOURCE" Button - Programming Properties

On the "CLEAR VTC SOURCE" button, set the *Channel Code* to **8**.

5) Write NetLinx Code To Respond To Custom Event

The NetLinx Code below utilizes the custom events that were configured in the TP file for "behavior" changes on the drop target buttons via the states configured earlier in this section.

1. Use NetLinx Studio 4 to add the following code to the NetLinx program loaded on the Master:

```
PROGRAM_NAME='MASTER'

DEFINE_DEVICE
dvTP = 10001:1:0

DEFINE_CONSTANT
//dropTargets
INTEGER btnDT  = 17
//draggables
INTEGER btnDG1 = 1
INTEGER btnDG2 = 2
INTEGER btnDG3 = 3
INTEGER btnDG4 = 4

DEFINE_VARIABLE
//an array to store our draggable buttons
INTEGER dgBTNS[] = {btnDG1 ,btnDG2 ,btnDG3 ,btnDG4}
//to store draggable address from start event
INTEGER nDragAddress = 0

DEFINE_MUTUALLY_EXCLUSIVE
([dvTP,1]..[dvTP,4])
//In this example the groups are defined as follows
//      - buttonAddresses 1,2, are assigned: group_1
//      - buttonAddresses 3,4 are assigned: group_2
//      - btnDT [17] will accept draggables from: group_2

DEFINE_EVENT

DATA_EVENT[dvTP]
{
    ONLINE:
    {
    //Let's make sure we are starting in state 1
    SEND_COMMAND dvTP,"'^ANI-',ITOA(btnDT),',1,1,0'"
    }
}

//Custom event for START [1410]
//Any time a draggable is initiated (long press, dragShadow appears)
//a START event is sent.
//CUSTOM_EVENT[dvTP,ID,Type]
CUSTOM_EVENT[dvTP,dgBTNS,1410]
{
    //Get the dragButtonAddress from the customEvent
    nDragAddress = custom.value1
    SEND_COMMAND dvTP,"'^ANI-',ITOA(btnDT),',2,2,0'"
}
//Custom event for ENTER [1411]
//Once the dragShadow enters the boundaries of a valid dropTarget
//a ENTER event is sent
```

```
CUSTOM_EVENT[dvTP,btnDT,1411]
{
    SEND_COMMAND dvTP,"'^ANI-',ITOA(btnDT),',2,2,0'"
}
//Custom event for EXIT [1412]
//Once the dragShadow leaves the boundaries of a valid dropTarget
//a EXIT event is sent

CUSTOM_EVENT[dvTP,btnDT,1412]
{
    SEND_COMMAND dvTP,"'^ANI-',ITOA(btnDT),',1,1,0'"
}

//Custom event for DROP [1413]
//A DROP event occurs when a draggable has been released within the boundaries
//of a valid dropTarget. A valid dropTarget is a dropTarget that has a group
//which the draggable is assigned to.
CUSTOM_EVENT[dvTP,btnDT,1413]
{
    SEND_COMMAND dvTP,"'^ANI-',ITOA(btnDT),',1,1,0'"
    //turn on the source(draggable)
    ON[dvTP,nDragAddress]
}
//Custom event for CANCEL [1414]
//A CANCEL event occurs when a draggable has been released over anything that
//is not a VALID dropTarget.
CUSTOM_EVENT[dvTP,dgBTNS,1414]
{
    SEND_COMMAND dvTP,"'^ANI-',ITOA(btnDT),',1,1,0'"
}

BUTTON_EVENT[dvTP,8] //CLEAR VTC SOURCES
{
    PUSH:
    {
    OFF[dvTP,1]
    OFF[dvTP,2]
    OFF[dvTP,3]
    OFF[dvTP,4]
    SEND_COMMAND dvTP,"'^ANI-',ITOA(btnDT),',1,1,0'"
    }
}
```

2.   Save changes.

NOTE: *The NetLinx code shown above is included in the NetLinx Studio Workspace file (DragAndDropNoGroups.apw) that is in the DragAndDropNoGroups.ZIP file.*

6) Use NetLinx Studio 4 to Compile and Transfer the Project Files

Use NetLinx Studio 4 to compile the code and transfer the project files to the Master:

1.   At the top of the *DragAndDropNoGroups.axs* source code file, change the *dvTP* value to match the device number of your touch panel (FIG. 241):

```
PROGRAM_NAME='MASTER'

DEFINE_DEVICE
dvTP = 10001:1:0
```

FIG. 241 dvTP Device Number value - Change to match the device number of your Touch Panel

2.   Compile the code (select **Build > Build Active System**).

3.   Transfer the *DragAndDropNoGroups.apw* workspace file to the NetLinx Master:
     a.   Select **Tools > File Transfer** to open the *File Transfer* dialog.
     b.   Open the *Send* tab and clear any files that are listed by clicking **Remove All**.
     c.   Click **Add** to open the *Select Files for File Transfer* dialog.
     d.   Select the top-level **Projects** folder to select all files in the workspace for transfer (FIG. 242):
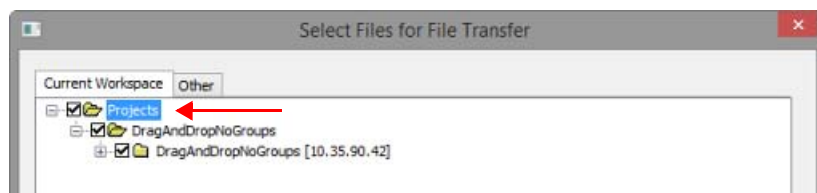
FIG. 242 Select Files for File Transfer dialog

e. Select **OK** to return to the *File Transfer* dialog (FIG. 243):
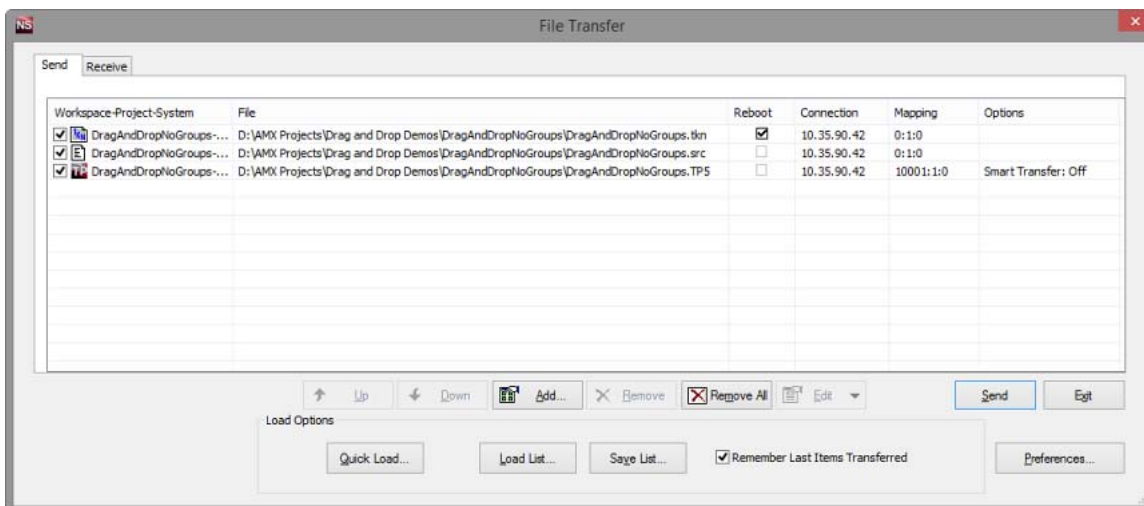
FIG. 243 File Transfer dialog – indicating files in the DragAndDropNoGroups.apw workspace queued for transfer

f. Click **Send** to initiate the file transfer.

g. The progress of the transfer is indicated in the Output Bar.

End Result

The result of this demo is a touch panel page with four draggable buttons representing source (input) devices and one drop target button representing an output device (VTC):
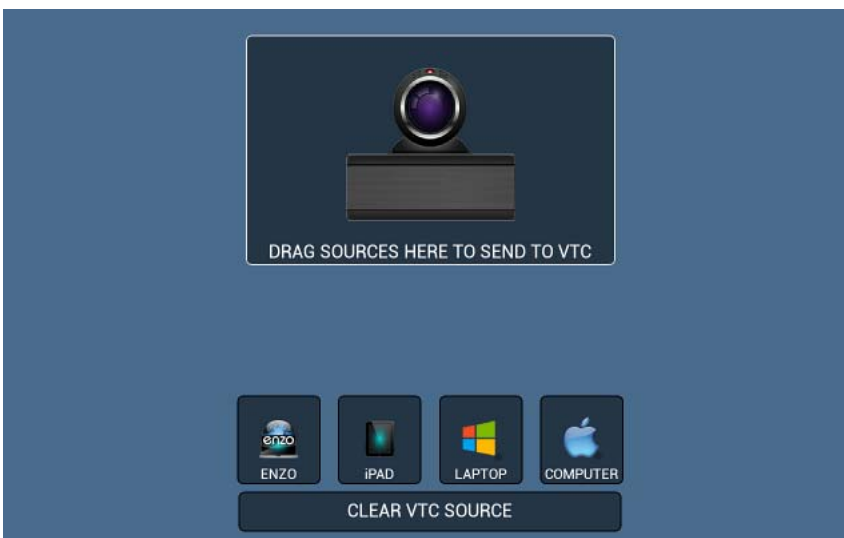
FIG. 244 Drag and Drop Demo – Page Layout

● The VTC button ("DRAG SOURCES HERE TO SEND TO VTC") is a drop target button representing an output device (VTC) that can accept any of the sources represented by the four draggable buttons. Note that in this example the VTC is a valid target for all sources, since no Drop Groups have been defined.

● Source buttons can each be dragged onto the VTC button individually. When one of the draggable buttons is released within the bounds of the VTC button, NetLinx code receives the custom events and turns on the source represented by the draggable button that was dropped.

● Press the CLEAR VTC SOURCE button to clear the current input setting.

Advanced Demo - Three Drop Groups

The following instructions illustrate creating a set of *draggable* buttons that represent input devices, and a set of *drop target* buttons that represent three output device (Displays).

NOTE: *This set of instructions uses files that are included in the "Drag-and-Drop " demo file which is available to download from the UI RESOURCE CENTER at www.amx.com.*

The resulting demo page will provide five draggable buttons that represent source (input) devices, and three drop target buttons representing three output (Display) devices. End users will be able to switch sources on the Displays by dragging and dropping a draggable button within the bounds of the drop target buttons (FIG. 223):
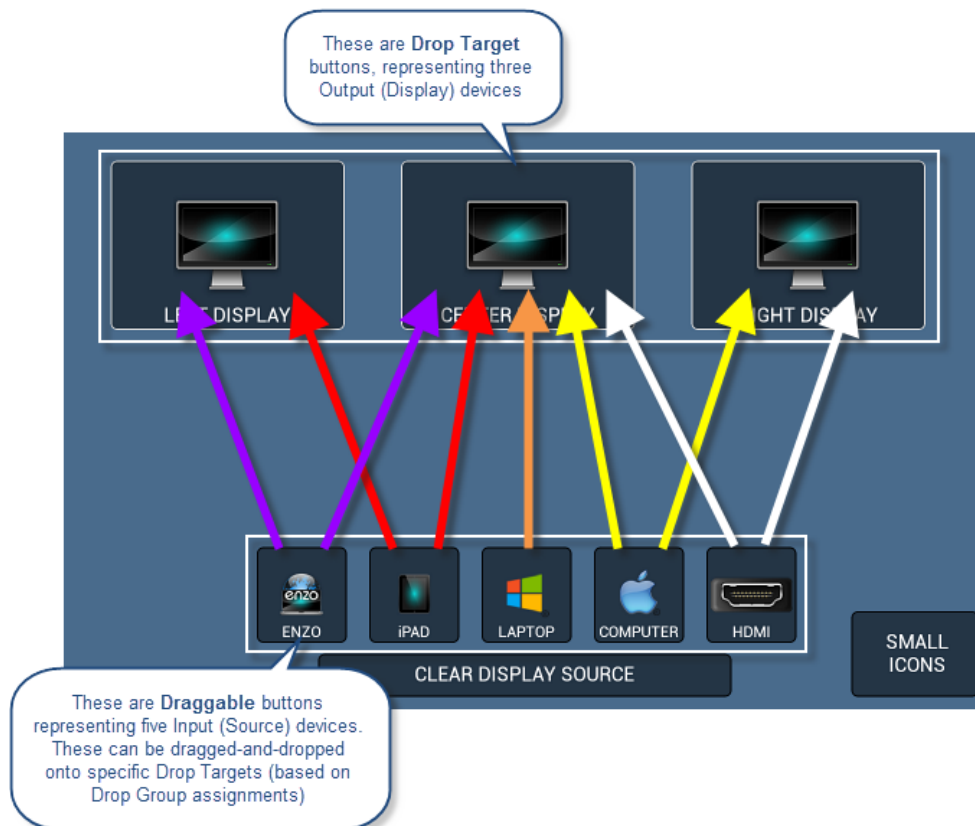
FIG. 245 Drag and Drop Demo - Five Draggable (Inputs) buttons and three Drop Target (Outputs) button

Before You Begin

Download the *AdvancedDragAndDropExample.ZIP* file from www.amx.com and extract its contents to a known location. This ZIP file contains the following files, all of which are required for the demo described in this manual:

- **AdvancedDragAndDropExample.TP5** - A TPDesign5 project file, as well as all of the image files used by the Page and Buttons in this project:

• amxicons_target-invalid.png	• icon-apple.png
• amxicons_target-invalid-small.png	• icon-enzo.png
• amxicons_target-valid.png	• icon-iPad.png
• amxicons_target-valid-small.png	• icon-windows8.png
	• SZ9_icon display.png

- **AdvancedDragAndDropExample.APW** - A NetLinx Studio workspace file that contains the NetLinx code:

• AdvancedDragAndDropExample.axs	• AdvancedDragAndDropExample.tkn
• AdvancedDragAndDropExample.src	• AdvancedDragAndDropExample.tko

1) Create a TPDesign5 Project/Import Images

In order to display the images on the page and buttons shown in this demo, the image files must be added to the project, via the Resource Manager - *Images* tab:

1. Open TPDesign5 (v1.3 or higher) and start a new Project (**File > New**).
2. Open the Resource Manager to the *Images* tab.
3. Click **Import** to locate and select all of the image files that were included in the *Drag and Drop Demo.ZIP* file.
4. Click **OK** to import the selected files and return to the Resource Manager (FIG. 224):
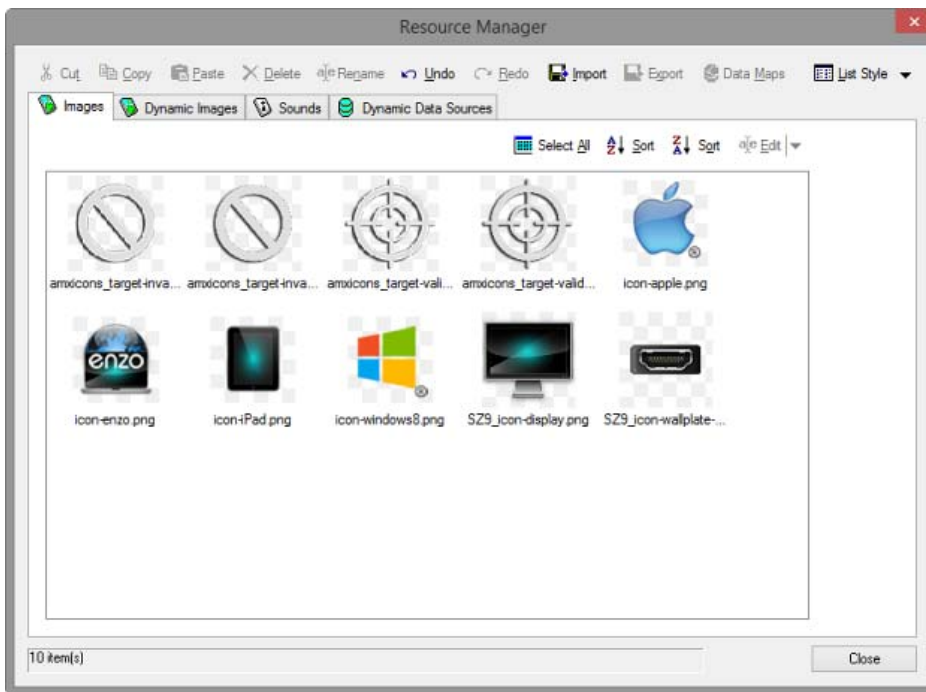
FIG. 246 Resource Manager Images tab - AdvancedDragAndDropExample Demo images imported

5. Click **Close** to close the Resource Manager.

> **NOTE:** *The AdvancedDragAndDropExample.TP5 file in the Drag and Drop demo has the images shown above already imported into the project.*

2) Create & Configure Drop Target Buttons

In this example, there are three Drop Target buttons that will represent the Output devices (Displays) that can accept input from the source devices represented by the draggable buttons.

Create Three Drop Target Buttons

1. Use the Button Draw tool to create three new buttons.

2. Arrange them horizontally on the top half of the page, and enter the following names for each button, in the *Name* (General) property:
   - LEFT DISPLAY
   - CENTER DISPLAY
   - RIGHT DISPLAY

3. Set each button's *Type* (General) property to **multi-state general** (FIG. 225):

Only General and Multi-State General buttons can be used as Draggable or Drop Target buttons
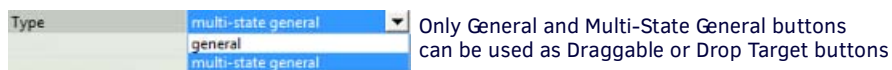
FIG. 247 TPDesign5 General Properties - Type set to "general"

4. Set each button's *Drag/Drop Type* (General) property to **drop target** (FIG. 226):
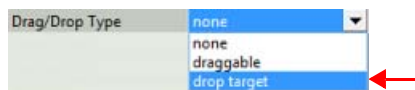
FIG. 248 TPDesign5 General Properties - Drag/Drop Type set to "drop target"

Set Drop Target Button Properties - General

Set the remaining *General* properties for the Drop Target buttons as shown in FIG. 227:
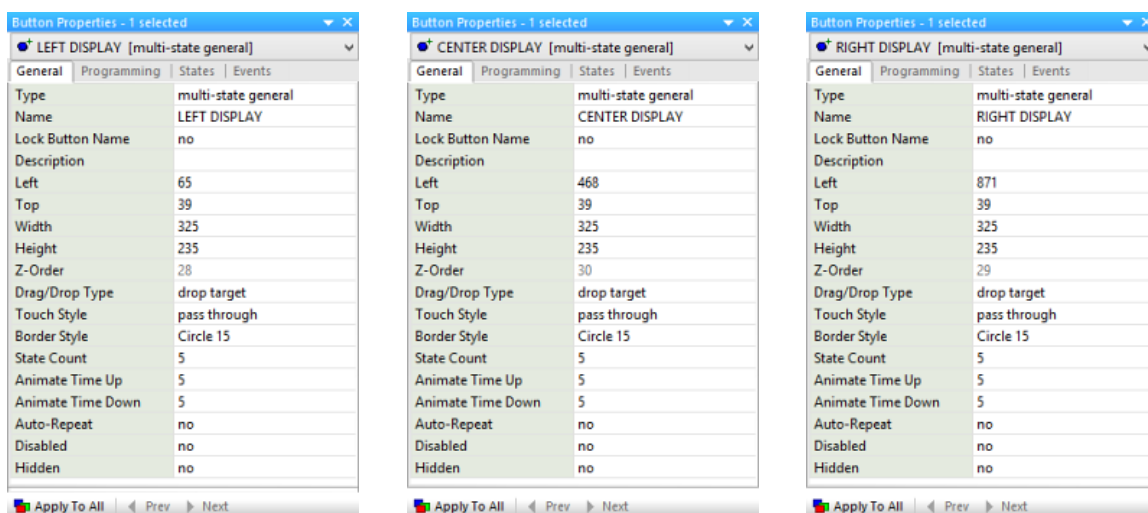
FIG. 249 Drop Target Buttons – General Properties

Set Drop Target Button Properties - Programming

Set the *Programming* properties for each of the Drop Target buttons as shown in FIG. 228:
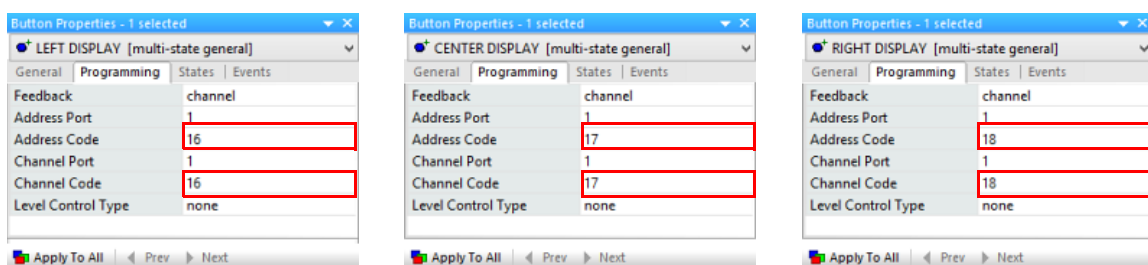
FIG. 250 Drop Target Button - Programming Properties

- On the LEFT DISPLAY button, set the *Address Code* to **16** and set the *Channel Code* to **16**.
- On the CENTER DISPLAY button, set the *Address Code* to **17** and set the *Channel Code* to **17**.
- On the RIGHT DISPLAY button, set the *Address Code* to **18** and set the *Channel Code* to **18**.

Set Drop Target Button Properties - States

In this example, these button will represent output (Display) devices. Use the *Text* (States) property to add the following labels to each button (*All States)*:

- On the LEFT DISPLAY button, set the *Text* to LEFT DISPLAY.
- On the CENTER DISPLAY button, set the *Address Code* to **17** and set the *Channel Code* to **17**.
- On the RIGHT DISPLAY button, set the *Address Code* to **18** and set the *Channel Code* to **18**.

NOTE: *As explained below, the drop-targets will use multiple states to provide visual feedback to the user in terms of whether each drop target is valid or invalid for a selected draggable button. While each of these states includes a bitmap that specifically indicates whether the target is valid or not, the button text remains the same across all States. Add and configure each button's text before duplicating states to avoid having to add the text to each individual state.*

Use the *Bitmaps* (States) property to add the Display bitmap to each button (*All States)*:

- Note that all images must first be imported in to the project via the Resource Manager in order to be available to apply to buttons or pages in the project. The images used in this demo are pre-loaded in the TP5 project file.
- Select each drop target button and under *All States*, apply the bitmap: **SZ9_icon-display.png**.
- In this example, the *Bitmap Justification* is set to **center-middle**.

Add States to each Drop Target Button

In this example, the drop target buttons will use multiple states to provide a visual indication of whether it is a valid target for draggable buttons. For example, if a drag is started on a draggable button, a bitmap featuring either a "Target-Valid" or "Target-Invalid" icon is displayed on Drop Target buttons, depending on whether each Drop Target button is a valid or invalid target for the selected draggable button.

NOTE: *Use Drop Groups to configure valid/invalid drop targets for draggable buttons. See the Drop Groups section on page 168 for details.*

Additionally, this example includes the option to use either "small icons" or large icons", so there are large versions of the "Target-Valid" and "Target-Invalid" icons as well. Therefore, there are five potential bitmap arrangements that need to be available to the drop target button - these are all configured as separate *States* for these buttons.

By default, multi-state general buttons have two states. To add three states:

1. Select a Drop Target button and open the State Manager window.
2. In the General tab of the Properties window, click in the State Count property and change the value to **5** (FIG. 251):

State Count	5

FIG. 251 State Count (General) Property – set to "5"

3. Press **Enter** to save changes. The new states are indicated in the State Manager window (FIG. 230):
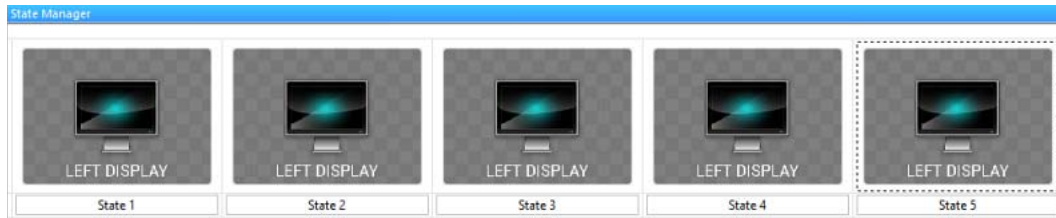
FIG. 252 State Manager window indicating five states

4. Repeat these steps for the CENTER DISPLAY and RIGHT DISPLAY Drop Target buttons.
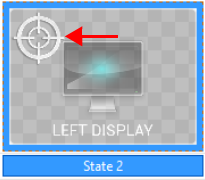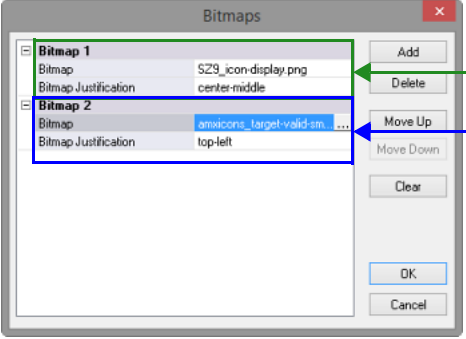
Add a "Target-Valid" or "Target-Invalid" Icon to each State of each Drop Target Button

As described on page 192, each of the five States for the drop target buttons are used to indicate whether each drop target is a valid target for a selected draggable button:
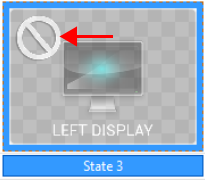
Additionally, this example includes the option to use either small or large icons, so there are large versions of the "Target-Valid" and "Target-Invalid" icons for each drop target button as well. Therefore, there are five potential bitmap arrangements that need to be available to the drop target button - these are all configured as separate *States* for each button.

NOTE: *This example employs specific "Target-Valid" and "Target-Invalid" icons, as well as a change in opacity to indicate the fact that there are drop-targets, and whether they are valid targets for each draggable button. However, the icons and opacity settings are optional. Any bitmaps (or no bitmaps) can be used; any change in opacity (or no change) can be used. There are many possible ways to indicate the presence of drop target buttons as well as the validity of each drop target relative to a selected draggable*

State 1 is used to when the drop target is being displayed (with no drag and drop action). Therefore, bitmaps must be added to states 2-5 to indicate a valid target or invalid target:

Drop Target Button - States 1-5	
State 1 	State 1 (no feedback) State 1 doesn't display either the "Target-Valid" or "Target-Invalid" icon, so no bitmap is added. This state includes the Display icon (only), and is used when the drop target is simply being displayed (without any drag and drop action).
State 2 	State 2 (small "Target-Valid" icon) State 2 displays the *small* "Target-Valid" icon in the upper-left corner of the button. This state is used to indicate that the drop target is valid for the selected button. 1. In the State Manager window, select **State 2**. 2. Set the *Overall Opacity* (States) property to **128**. 3. Click the browse (...) button in the *Bitmaps* (States) property to open the *Bitmaps* dialog. Note that *Bitmap 1* is already set to **SZ9_icon-display.png**: 4. Click **Add** to add the *Bitmap 2* field, and select the appropriate bitmap in the *Select Resource* dialog: In this example, all states use "SZ9-icon-display.png" as Bitmap 1 (Bitmap Justification = center-middle) In each State (2-5) use Bitmap 2 to show the appropriate feedback icon: "Target Valid" - small or large, or "Target Invalid" - small or large • Set *Bitmap 2* to "**amxicons_target-valid-small.png**". • Set *Bitmap Justification* for Bitmap 2 to **top-left.** 5. Click **OK** to close the *Bitmaps* dialog. The image for State 2 is updated in the State Manager.

Drop Target Button - States 1-5 (Cont.)	
State 3	State 3 (small "Target-Invalid" icon) State 3 displays the small "Target Invalid" icon in the upper-left corner of the button. This state is used to indicate that the drop target is *not* valid for the selected button. 1. In the State Manager window, select **State 3**. 2. Set the *Overall Opacity* (States) property to **128**. 3. Click the browse (...) button in the *Bitmaps* (States) property to open the *Bitmaps* dialog. Note that *Bitmap 1* is already set to **SZ9_icon-display.png**. 4. Click **Add** to add the *Bitmap 2* field, and select the appropriate bitmap in the *Select Resource* dialog. • Set *Bitmap 2* to "**amxicons_target-invalid-small.png**". • Set *Bitmap Justification* for Bitmap 2 to **top-left.** 5. Click **OK** to close the *Bitmaps* dialog. The image for State 3 is updated in the State Manager.
State 4	State 4 (large "Target Valid" icon) State 4 includes the VTC icon and a (large) "Target-Valid" icon. This state is used to indicate that the drop target is valid for the selected button (and that the LARGE ICONS option has been selected on the touch panel). 1. In the State Manager window, select **State 4**. 2. Set the *Overall Opacity* (States) property to **128**. 3. Click the browse (...) button in the *Bitmaps* (States) property to open the *Bitmaps* dialog. Note that *Bitmap 1* is already set to **SZ9_icon-display.png**. 4. Click **Add** to add the *Bitmap 2* field, and select the appropriate bitmap in the *Select Resource* dialog. • Set *Bitmap 2* to "**amxicons_target-valid.png**". • Set *Bitmap Justification* for Bitmap 2 to **scale to fit**. 5. Click **OK** to close the *Bitmaps* dialog. The image for State 4 is updated in the State Manager.
State 5	State 5 (large "Target Invalid" icon) State 5 includes the VTC icon and a (large) "Target-Invalid" icon in the middle of the button. This state is used to indicate that the drop target is *not* valid for the selected button (and that the LARGE ICONS option has been selected on the touch panel). 1. In the State Manager window, select **State 5**. 2. Set the *Overall Opacity* (States) property to **128**. 3. Click the browse (...) button in the *Bitmaps* (States) property to open the *Bitmaps* dialog. Note that *Bitmap 1* is already set to **SZ9_icon-display.png**. 4. Click **Add** to add the *Bitmap 2* field, and select the appropriate bitmap in the *Select Resource* dialog. • Set *Bitmap 2* to "**amxicons_target-valid.png**". • Set *Bitmap Justification* for Bitmap 2 to **scale to fit**. 5. Click **OK** to close the *Bitmaps* dialog. The image for State 2 is updated in the State Manager.

The result of these updates, as indicted in the State Manager window are shown in FIG. 253:

FIG. 253 State Manager - Drop Target button states with icons placed

Set Drop Target Button Properties - Events

TPDesign5 (1.3.23 or higher) supports three new Events for drop target buttons: **Drop Enter, Drop Exit** and **Drop** (see the *Drag and Drop-Specific Events* section on page 170 for details):
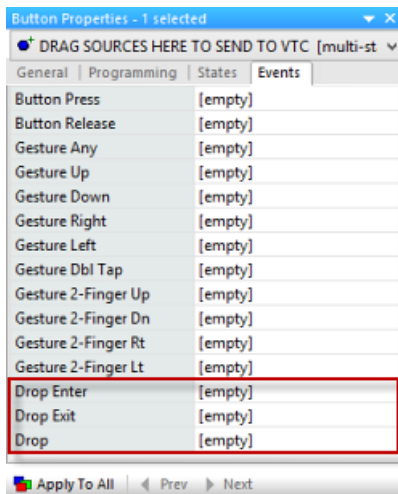
FIG. 254 Events for Drop Target Buttons

Button Event Actions are listed in the *Edit Event Actions* dialog. Use the **Add Action** option in this dialog to create new custom (event) actions via the *Edit Custom Action* dialog (FIG. 255):
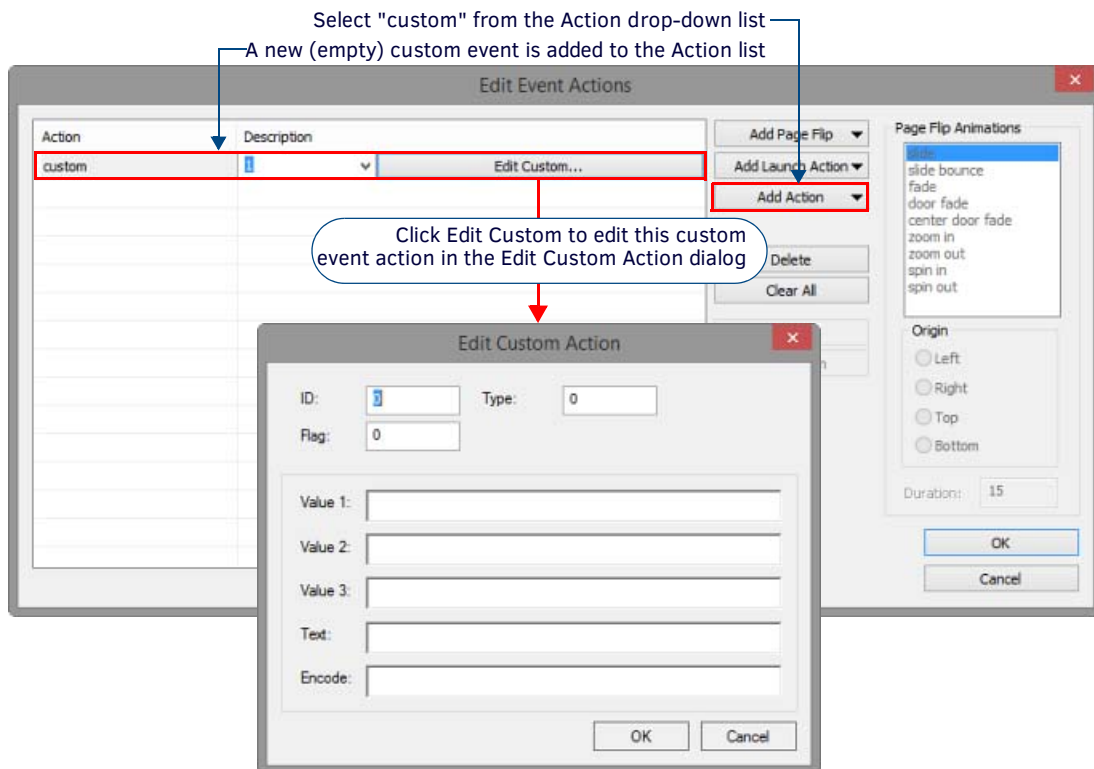
FIG. 255 Edit Event Actions dialog indicating a new (empty) custom event

Configure the "Drop Enter" Event for All Drop Target Buttons

The following steps describe how to configure custom events that will be sent to the Master, and picked up with our NetLinx code (which then handles the visual changes). Refer to page 208 to view the accompanying code.

1. Select a drop target button (in this example, start with the "LEFT DISPLAY" button).

2. In the *Events* tab of the Properties window, select the **Drop Enter** event and click the browse (...) button to open the *Edit Event Actions* dialog.

3. Click **Add Action**, and select "*custom*" from the drop-down list - this adds a new (empty) custom event action to the *Action* list.

4. Click **Edit Custom** to open the *Edit Custom Action* dialog. Use the fields in this dialog to define the event action for the selected drop target button event.

To configure the *Drop Enter* event for the drop target buttons, enter the **ID**, **Type**, **Flag** and **Value 1** fields according the table below. These fields must be configured for each drop target button in the demo. Note that in this case, the Custom Action Settings values are the same for all three drop target buttons:

Custom Action Settings for "Drop Enter" Event	
"LEFT DISPLAY", CENTER DISPLAY," and "RIGHT DISPLAY" drop target buttons	• ID: **17** • Type: **2** • Flag: **0** • Value 1: **${dropChannelCode}**

Once they have been configured, the custom event properties are displayed in the *Drop Enter* property for each drop target button (FIG. 256):

| Drop Enter | Port: 1;ID: 17;Type: 2;Val 1: ${dropChannelCode} | ... |

FIG. 256 Drop Enter Event indicating sample data

Configure the "Drop Exit" Event for All Drop Target Buttons

The following steps describe how to configure custom events that will be sent to the Master, and picked up with our NetLinx code (which then handles the visual changes). Refer to page 208 to view the accompanying code.

1. Select a drop target button (in this example, start with the "LEFT DISPLAY" button).
2. In the *Events* tab of the Properties window, select the **Drop Exit** event and click the browse (...) button to open the *Edit Event Actions* dialog.
3. Click **Add Action**, and select "*custom*" from the drop-down list.
4. This adds a new (empty) custom event action to the *Action* list.
5. Click **Edit Custom** to open the *Edit Custom Action* dialog. Use the fields in this dialog to define the event action for the selected drop target button event.

To configure the *Drop Exit* event for the drop target buttons, enter the **ID**, **Type**, **Flag** and **Value 1** fields according the table below. These fields must be configured for each drop target button in the demo.

Note that in this case, the Custom Action Settings values are the same for all three drop target buttons:

Custom Action Settings for "Drop Exit" Event	
"LEFT DISPLAY", CENTER DISPLAY," and "RIGHT DISPLAY" drop target buttons	• ID: **17** • Type: **3** • Flag: **0** • Value 1: **${dropChannelCode}**

Once they have been configured, the custom event properties are displayed in the *Drop Exit* property for each drop target button (FIG. 257):

| Drop Exit | Port: 1;ID: 17;Type: 3;Val 1: ${dropChannelCode} | ... |

FIG. 257 Drop Exit Event indicating sample data

Configure the "Drop" Event for All Drop Target Buttons

The following steps describe how to configure custom events that will be sent to the Master, and picked up with our NetLinx code (which then handles the visual changes). Refer to page 208 to view the accompanying code.

1. Select a drop target button (in this example, start with the "LEFT DISPLAY" button).
2. In the *Events* tab of the Properties window, select the **Drop Exit** event and click the browse (...) button to open the *Edit Event Actions* dialog.
3. Click **Add Action**, and select "*custom*" from the drop-down list.
4. This adds a new (empty) custom event action to the *Action* list.
5. Click **Edit Custom** to open the *Edit Custom Action* dialog. Use the fields in this dialog to define the event action for the selected drop target button event.

To configure the *Drop* event for the drop target buttons, enter the **ID**, **Type**, **Flag** and **Value 1** fields according the table below. These fields must be configured for each drop target button in the demo.

Note that in this case, the Custom Action Settings values are the same for all three drop target buttons:

Custom Action Settings for "Drop" Event	
"LEFT DISPLAY", CENTER DISPLAY," and "RIGHT DISPLAY" drop target buttons	• ID: **17** • Type: **4** • Flag: **0** • Value 1: **${dropChannelCode}**

Once they have been configured, the custom event properties are displayed in the *Drop* property for the selected button (FIG. 258):

| Drop | Port: 1;ID: 17;Type: 4;Val 1: ${dropChannelCode} | ... |

FIG. 258 Drop Event indicating sample data

When all of the Events have been configured for the drop target buttons, save your project file.

Add Each Drop Target Button to a Drop Group

Each Drop Target button will be added a Drop Group. The Drop Groups that have been created in this demo are named "*group_1*", "*group_2*" and "*group_3*" (see page 198 for details on creating these Drop Groups).

This will allow draggable buttons associated each group to use specific drop target buttons as a "valid" targets.

For example, only draggable buttons that are associated with "group_1" will be able to use the *group 1* drop target button as a valid target. Only draggable buttons that are associated with "group_2" will be able to use the *group 2* drop target button as a valid target. Only draggable buttons that are associated with "group_3" will be able to use the *group 3* drop target button as a valid target.

Drop Group assignments are managed in the *Drop-Target Groups* dialog (FIG. 259):
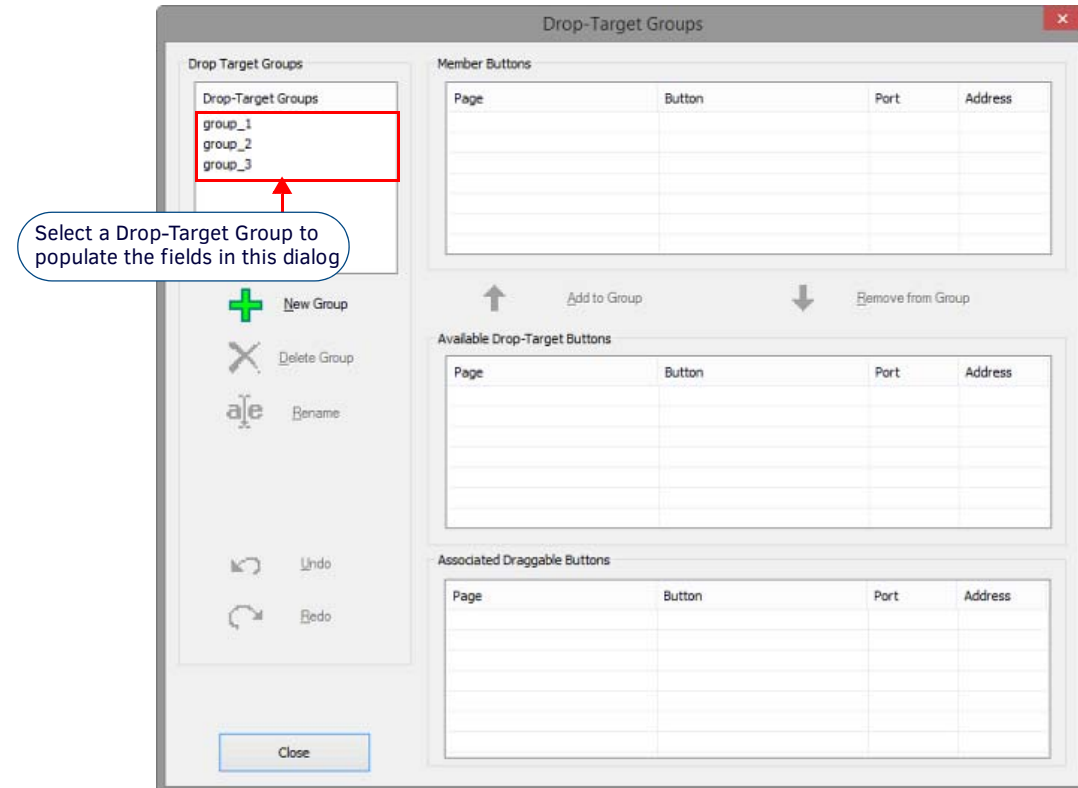
FIG. 259 Drop-Target Groups dialog (detail)

Add the LEFT DISPLAY and CENTER DISPLAY Drop Target Buttons To "group_1"

1. Select **Panel > Edit Drop Target Groups** to open the *Drop-Target Groups* dialog.

2. In the *Drop Target Groups* window, select "**group_1**".

3. The Drop Target buttons created in Step 3 (see page 198) are indicated in the *Available Drop Target Buttons* window. Select LEFT DISPLAY and click **Add to Group** to move it into the *Member Buttons* window.

4. Select CENTER DISPLAY and click **Add to Group** to move it into the *Member Buttons* window.

5. The *Member Buttons* window indicates that the LEFT DISPLAY and CENTER DISPLAY Drop Target buttons are "members" of the "*group_1*" Drop-Target Group (FIG. 260):
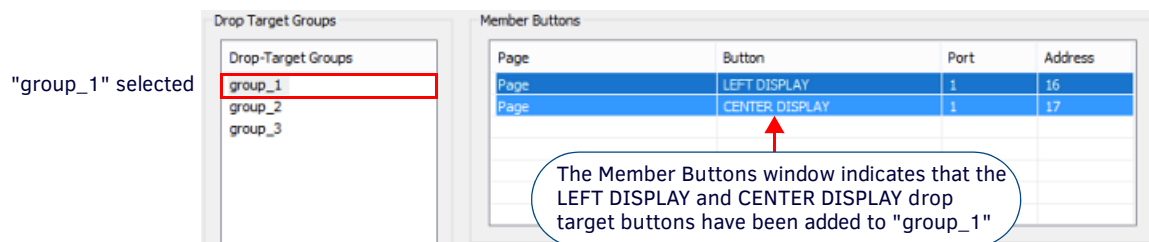
FIG. 260 Drop-Target Groups dialog - the LEFT DISPLAY dropTarget button is a member of "group_1"

Add the CENTER DISPLAY Drop Target Button To "group_2"

1. Select **Panel > Edit Drop Target Groups** to open the *Drop-Target Groups* dialog.

2. In the *Drop Target Groups* window, select "**group_2**".

3. Select CENTER DISPLAY and click **Add to Group** to move it into the *Member Buttons* window. This indicates that the CENTER DISPLAY Drop Target button is now a "member" of the "*group_2*" Drop-Target Group (FIG. 261):

"group_2" selected
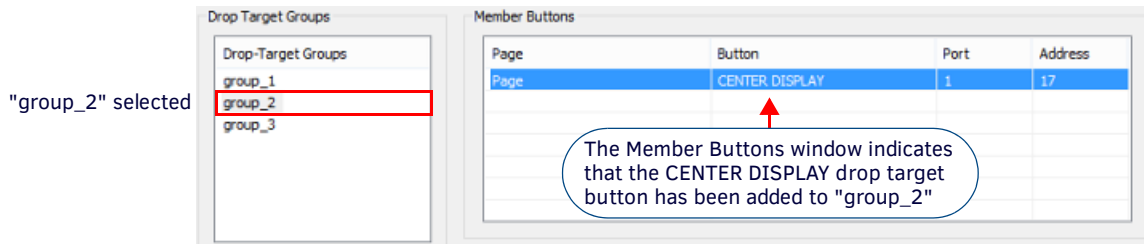
FIG. 261 Drop-Target Groups dialog - the CENTER DISPLAY dropTarget button is a member of "group_2"

Add the RIGHT DISPLAY Drop Target Button To "group_3"

1. Select **Panel > Edit Drop Target Groups** to open the *Drop-Target Groups* dialog.

2. In the *Drop Target Groups* window, select "**group_3**".

3. Select RIGHT DISPLAY and click **Add to Group** to move it into the *Member Buttons* window. This indicates that the RIGHT DISPLAY Drop Target button is now a "member" of the "*group_3*" Drop-Target Group (FIG. 262):

"group_3" selected
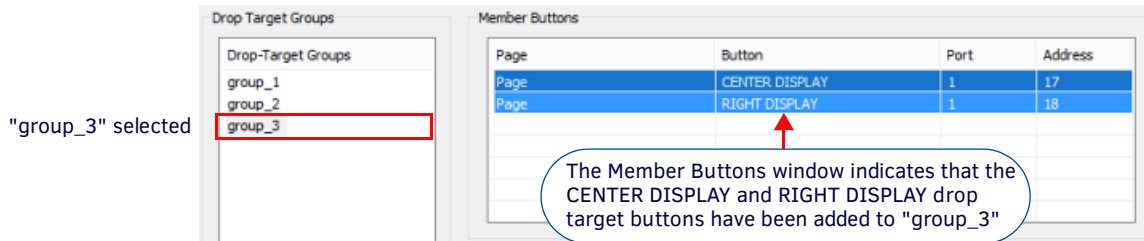
FIG. 262 Drop-Target Groups dialog - the RIGHT DISPLAY dropTarget button is a member of "group_3"

After all three drop target buttons have been assigned to their Drop Groups, click **Close** to save changes and close the *Drop-Target Groups* dialog.

3) Create Drop Groups

Drop Groups are created via the *Drop-Target Groups* dialog. After creating a drop group, drop target buttons will then be available for assignment.

- Drop Target buttons are added as "Members" of a specific Drop Group.
- Draggable buttons are associated with a specific Drop Group via the Drop Group (General) button property.

Once a Drop Target button has been added to a Drop Group (as a "member"), only draggable buttons that are associated with that Drop Group can be dragged and dropped onto the Drop Target button. Refer to the *Drop Groups* section on page 168 for a more detailed explanation of Drop Groups. In this example there are three Drop Target buttons, each of which represents a Display as an output device (see FIG. 223 on page 181):

- LEFT DISPLAY - This will be a valid drop target for the *ENZO* and *iPAD* draggable buttons.
- CENTER DISPLAY - This will be a valid drop target for all of the draggable buttons (*ENZO*, *iPAD*, *LAPTOP*, *COMPUTER* and *HDMI*).
- RIGHT DISPLAY - This will be a valid drop target for the *COMPUTER* and *HDMI* draggable buttons .

To create new Drop Groups:

1. Select **Panel > Edit Drop-Target Groups** to open the *Drop-Target Groups* dialog (FIG. 263):

Click to define 3 new Drop Groups
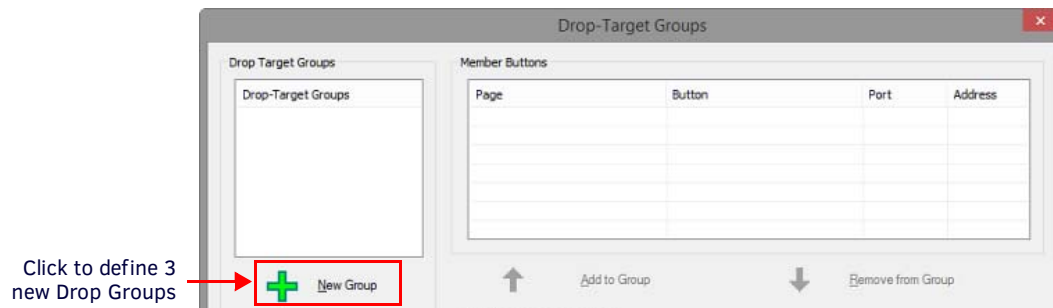
FIG. 263 Drop-Target Groups dialog

2. Click **New Group** to add a new Drop Group, via the *Create Drop-Target Group* dialog.
    a. Create a new group named "**group_1**", (FIG. 264):
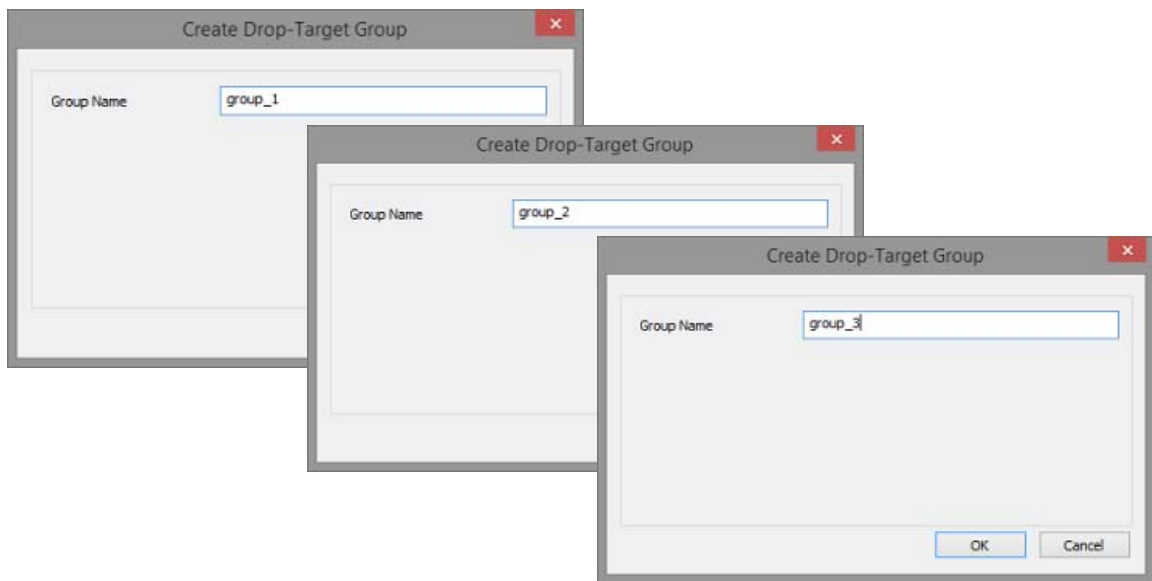
FIG. 264 Create Drop-Target Group dialog

    b.   Click **OK** to save changes and close the *Create Drop Group Target* dialog.

    c.   Create two additional groups named "**group_2**" and "**group_3**".

    d.   The new Groups are indicated in the *Drop Target Groups* window (FIG. 265):
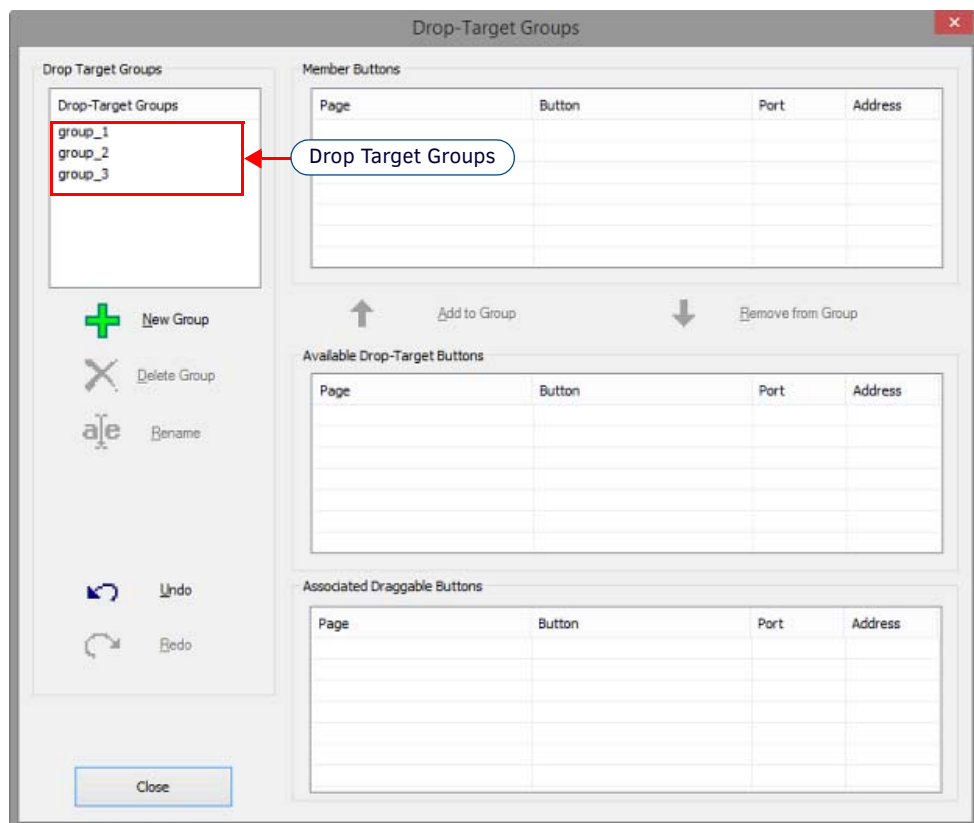
FIG. 265 Drop–Target Groups dialog - indicating "group_1", "group_2" and "group_3"

3.   Click **Close** to save changes and close the *Drop-Target Groups* dialog.

4) Create & Configure Draggable Buttons

In this example, four draggable buttons represent four source (input) devices that are used as inputs for the Output (Display) devices represented by the three Drop Target buttons.

Create Five Draggable Buttons

1. Use the Button Draw tool to create a new button.

2. Set the button's *Type* (General) property to **general** (FIG. 231):

Only General and Multi-State General buttons can be set as draggable or Drop Target
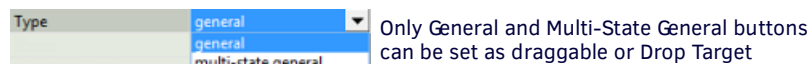
FIG. 266 TPDesign5 General Properties - Type set to "general"

3. Set the button's *Drag/Drop Type* (General) property to **draggable** (FIG. 232):
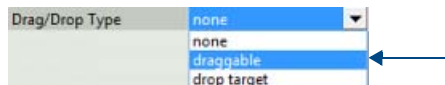
FIG. 267 TPDesign5 General Properties - Drag/Drop Type set to "draggable"

4. Repeat these steps to create a total of five draggable buttons. Alternatively, copy and paste the new button four times (FIG. 233):
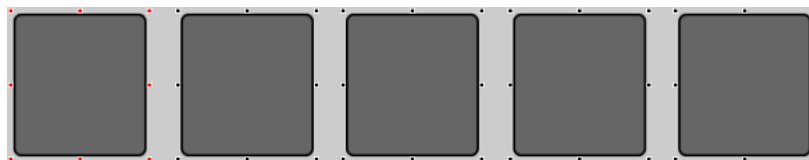
FIG. 268 Draggable Buttons

Set Draggable Button Properties - General

Set the remaining *General* properties for the draggable buttons as shown in FIG. 234:
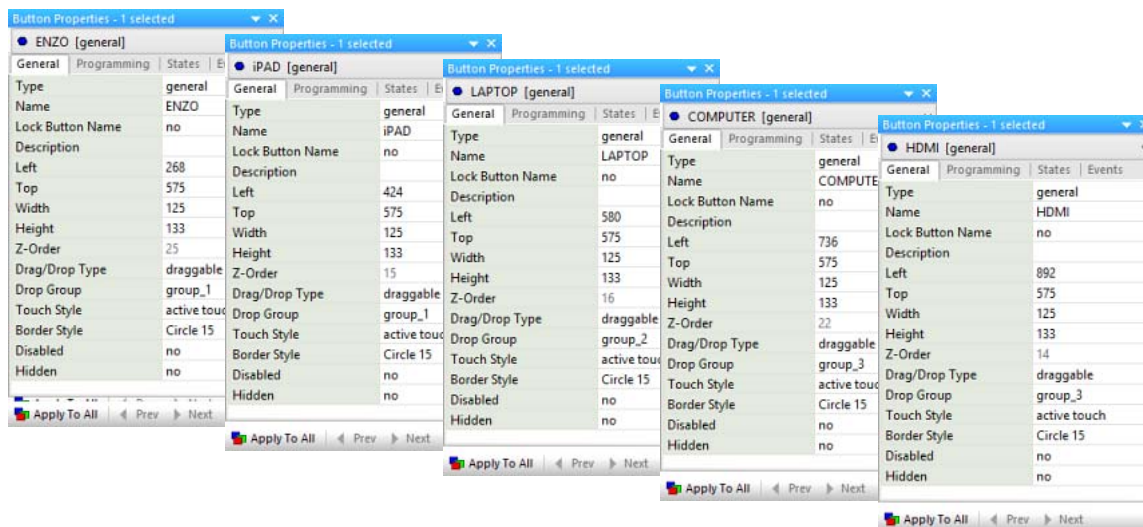
FIG. 269 Draggable Buttons - General Properties

Associate Draggable Buttons With a Drop Group

In this example, three Drop Groups have been created. Draggable buttons are associated with a specific Drop Group via the **Drop Group** (General) property (FIG. 270):

FIG. 270 General Properties - Drop Group

FIG. 270 indicates that three Drop Groups have been created: "group_1", "group_2" and "group_3". In this example, the ENZO and iPAD draggable buttons will be assigned to "*group_1*". The LAPTOP draggable button will be assigned to "*group_2*". The COMPUTER and HDMI draggable buttons will be assigned to "*group_3*":

1. Select the ENZO draggable button.
   a. In the *General* tab of the Properties window, click on **Drop Group** to access the drop-down list of all drop groups currently defined in this project.
   b. Select "**group_1**". This selection associates the ENZO draggable button with the "group_1" Drop Group.

As a result, the ENZO button will treat the LEFT DISPLAY and CENTER DISPLAY drop target buttons as valid targets (since they are both members of "*group_1*").

2.   Select the iPAD draggable button.
    a.   In the *General* tab of the Properties window, click on **Drop Group** to access the drop-down list of all drop groups currently defined in this project.
    b.   Select "**group_1**". This selection associated the selected draggable button with the "group_1" Drop Group.

As a result, the iPAD button will treat the LEFT DISPLAY and CENTER DISPLAY drop target buttons as valid targets (since they are both members of "*group_1*").

3.   Select the LAPTOP draggable button.
    a.   In the *General* tab of the Properties window, click on **Drop Group** to access the drop-down list of all drop groups currently defined in this project.
    b.   Select "**group_2**". This selection associated the selected draggable button with the "group_2" Drop Group.

As a result, the LAPTOP button will treat the CENTER DISPLAY drop target button as it's only valid targets (since it is a the only member of "*group_2*").

4.   Select the COMPUTER draggable button.
    a.   In the *General* tab of the Properties window, click on **Drop Group** to access the drop-down list of all drop groups currently defined in this project.
    b.   Select "**group_3**". This selection associated the selected draggable button with the "group_2" Drop Group - the same Drop Group that includes the Drop Target button in this example.

As a result, the COMPUTER button will treat the CENTER DISPLAY and RIGHT DISPLAY drop target buttons as valid targets (since they are both members of "*group_3*").

5.   Select the HDMI draggable button.
    a.   In the *General* tab of the Properties window, click on **Drop Group** to access the drop-down list of all drop groups currently defined in this project.
    b.   Select "**group_3**". This selection associated the selected draggable button with the "group_2" Drop Group - the same Drop Group that includes the Drop Target button in this example.

As a result, the HDMI button will treat the CENTER DISPLAY and RIGHT DISPLAY drop target buttons as valid targets (since they are both members of "*group_3*").

group_1 members:
LEFT DISPLAY, CENTER DISPLAY

group_2 members:
CENTER DISPLAY

group_3 members:
CENTER DISPLAY, RIGHT DISPLAY

FIG. 271 Drop Target and Draggable Buttons - Drop Group assignments

Set Draggable Button Properties - Programming

Each of the draggable buttons needs to be configured with unique Address and Channel Codes. For this example, set the Address/Channel Codes as shown below (FIG. 235):
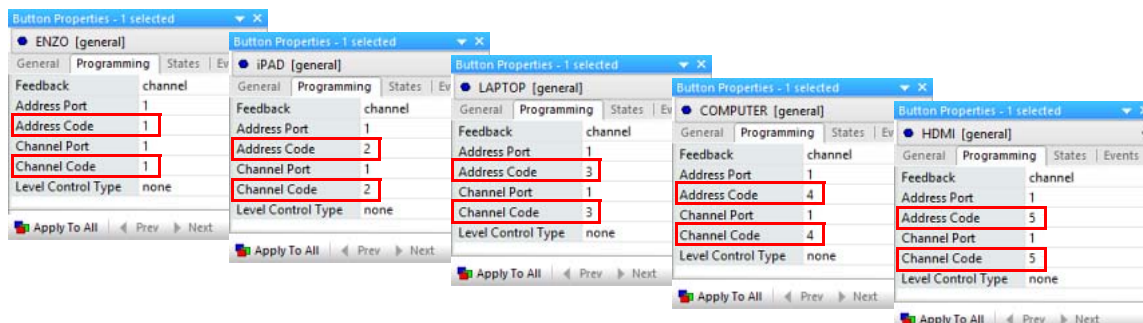
FIG. 272 Draggable Buttons - Programming Properties

- On the ENZO draggable button, set the *Address Code* to **1** and set the *Channel Code* to **1**.
- On the iPAD draggable button, set the *Address Code* to **2** and set the *Channel Code* to **2**.
- On the LAPTOP draggable button, set the *Address Code* to **3** and set the *Channel Code* to **3**.
- On the COMPUTER draggable button, set the *Address Code* to **4** and set the *Channel Code* to **4**.
- On the HDMI draggable button, set the *Address Code* to **5** and set the *Channel Code* to **5**.

Set Draggable Button Properties - States

In this example, each of these buttons will represent a different type of input (source) device. Edit the buttons to add text and icons to indicate the specific device represented by each button:

1. Use the *Text* (States) property to add labels to each of the buttons. Select each button and under *All States*, enter the following labels: **ENZO**, **iPAD**, **LAPTOP**, **COMPUTER** and **HDMI**.

2. Use the *Bitmaps* (States) property to apply an appropriate icon to each of the buttons.
   - Note that all images must first be imported in to the project via the Resource Manager in order to be available to apply to buttons or pages in the project. The images used in this demo are pre-loaded in the TP5 project file.
   - Select each button and under *All States*, apply the following bitmaps: **icon-enzo.png**, **icon-iPad.png**, **icon-windows8.png**, **icon-apple.png** and **SZ9_icon-wallplate-hdmi.png**.
   - In this example, the *Bitmap Justification* is set to **top-middle** for all five draggable buttons.

Set the remaining *States* properties for the draggable buttons as shown in FIG. 236:
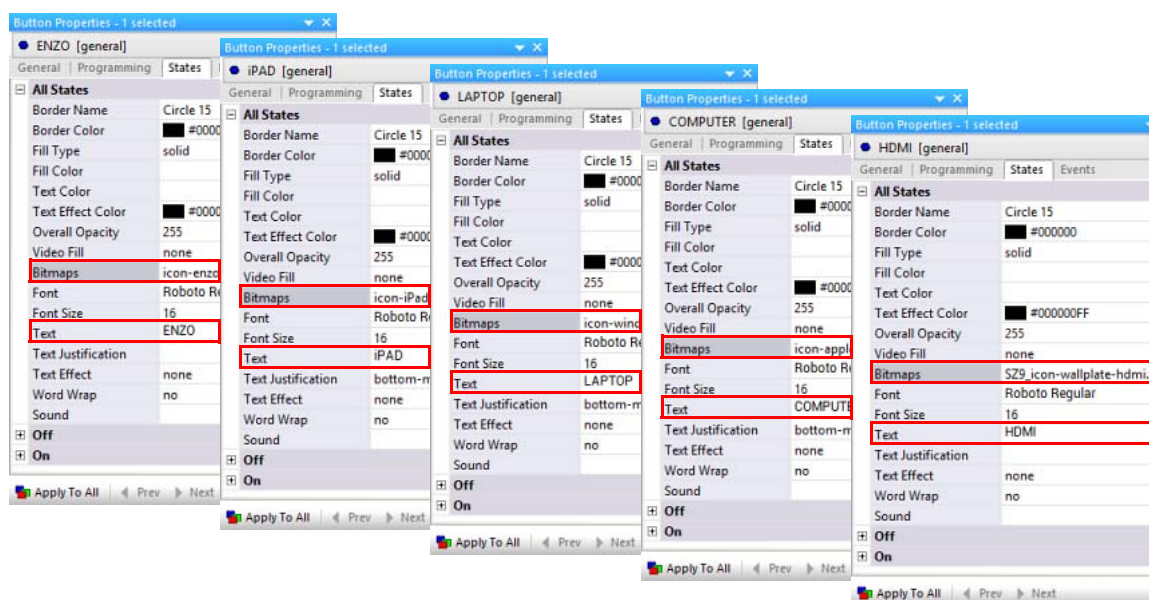
FIG. 273 Draggable Buttons - States Properties (All States shown)

For this example, the draggable buttons should look similar to the buttons shown below (FIG. 237):

FIG. 274 Draggable Buttons (Representing four Input Devices)

Set Draggable Button Properties - Events

TPDesign5 (1.3.23 or higher) supports a set of new Events for Draggable buttons: **Drag Start** and **Drag Cancel** (see the *Drag and Drop-Specific Events* section on page 170 for details):
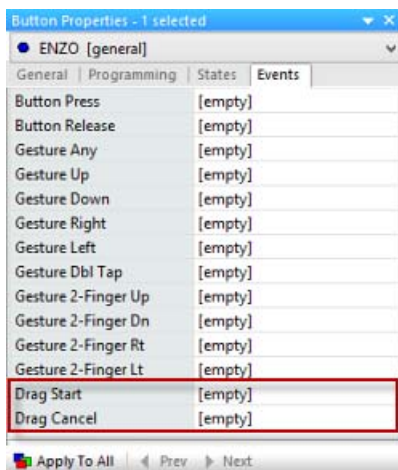
FIG. 275 Events for Draggable Buttons

Button Event Actions are listed in the *Edit Event Actions* dialog. Use the **Add Action** option in this dialog to create new custom (event) actions via the *Edit Custom Action* dialog (FIG. 276):

Select "custom" from the Action drop-down list

A new (empty) custom event is added to the Action list

Click Edit Custom to edit this custom event action in the Edit Custom Action dialog
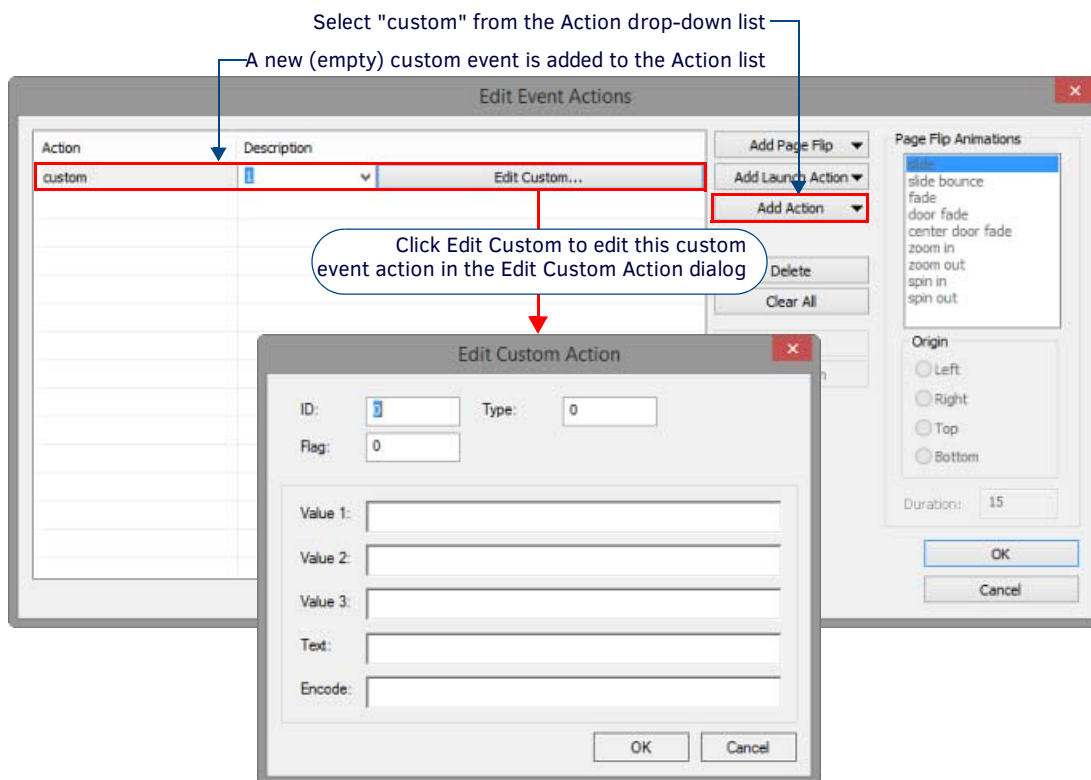
FIG. 276 Edit Event Actions dialog indicating a new (empty) custom event

Configure the "Drag Start" Event for Draggable Buttons

The following steps describe how to configure custom events that will be sent to the Master, and picked up with our NetLinx code (which then handles the visual changes). Refer to page 208 to view the accompanying code.

1. Select a draggable button (for example, the "ENZO" button).
2. In the *Events* tab of the Properties window, select the **Drag Start** event and click the browse (...) button to open the *Edit Event Actions* dialog.
3. Click **Add Action**, and select "*custom*" from the drop-down list (FIG. 277):
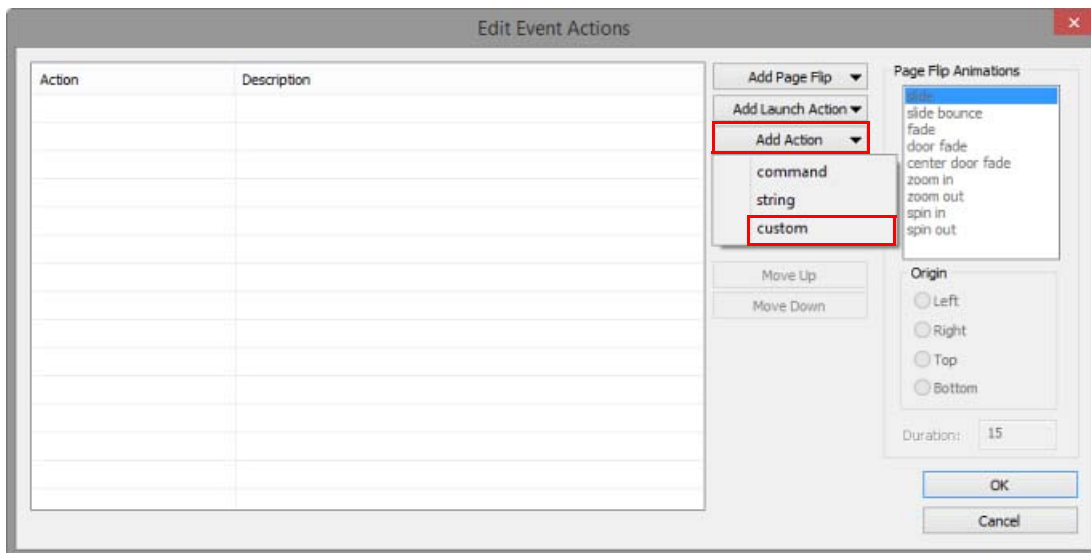
FIG. 277 Edit Event Actions dialog - Add Action drop-down list

4. This adds a new (empty) custom event action to the *Action* list (FIG. 278):

FIG. 278 Edit Event Actions dialog - new (empty) custom event action

5. Click **Edit Custom** to open the *Edit Custom Action* dialog (FIG. 279). Use the fields in this dialog to define the event action for the selected button/draggable button event.
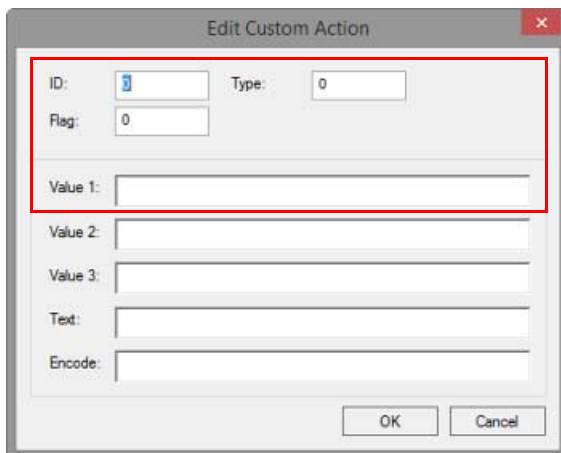
Note: In this example, the Value 2, Value 3, Text and Encode fields are not used.

FIG. 279 Edit Custom Action dialog

6. To configure the *Drag Start* Event for the selected button, enter the **ID**, **Type**, **Flag** and **Value 1** fields according the table below. These fields must be configured for each draggable button.

7. Click **OK** to save changes and close the *Edit Custom Action* dialog and return to the Edit Event Actions dialog.

8. Press **OK** to save changes and close this dialog.

The Custom Action settings for each draggable button in this demo are provided in the table below. Note that the *Drag Start* settings are identical for all buttons, with the exception of the **ID** value, which identifies each button:

Custom Action Settings for "Drag Start" Event	
"ENZO" button	• ID: **1** • Type: **1** • Flag: **0** • Value 1: **${dragChannelCode}** • Text: **${dragGroupName}**
"iPAD" button	• ID: **2** • Type: **1** • Flag: **0** • Value 1: **${dragChannelCode}** • Text: **${dragGroupName}**

Custom Action Settings for "Drag Start" Event (Cont.)	
"LAPTOP" button	• ID: **3** • Type: **1** • Flag: **0** • Value 1: **${dragChannelCode}** • Text: **${dragGroupName}**
"COMPUTER" button	• ID: **4** • Type: **1** • Flag: **0** • Value 1: **${dragChannelCode}** • Text: **${dragGroupName}**
"HDMI" button	• ID: **5** • Type: **1** • Flag: **0** • Value 1: **${dragChannelCode}** • Text: **${dragGroupName}**

Once they have been configured The custom event properties are displayed in the *Drag Start* property for the selected button (FIG. 280):

Drag Start — Port: 1;ID: 1;Type: 1;Val 1: ${dragChannelCode};Txt: ${dragGroupName}   ...

FIG. 280 Drag Start Event indicating sample data

Configure the "Drag Cancel" Event for Draggable Buttons

The following steps describe how to configure custom events that will be sent to the Master, and picked up with our NetLinx code (which then handles the visual changes). Refer to page 208 to view the accompanying code.

1. Select a draggable button (for example, the "ENZO" button).
2. In the *Events* tab of the Properties window, select the **Drag Start** event and click the browse (...) button to open the *Edit Event Actions* dialog.
3. Click **Add Action**, and select "*custom*" from the drop-down list (see FIG. 277 on page 204):
4. This adds a new (empty) custom event action to the *Action* list (see FIG. 278 on page 204):
5. Click **Edit Custom** to open the *Edit Custom Action* dialog (see FIG. 279 on page 204). Use the fields in this dialog to define the event action for the selected button/draggable button event.

To configure the *Drag Cancel* Event for the selected button, enter the **ID**, **Type**, **Flag** and **Value 1** fields according the table below. These fields must be configured for each draggable button.

The Custom Action settings for each draggable button in this demo are provided in the table below. Note that the *Drag Cancel* settings are identical for all buttons, with the exception of the **ID** value, which identifies each button:

Custom Action Settings for "Drag Cancel" Event	
"ENZO" button	• ID: **1** • Type: **5** • Flag: **0** • Value 1: **${dragChannelCode}**
"iPAD" button	• ID: **2** • Type: **5** • Flag: **0** • Value 1: **${dragChannelCode}**
"LAPTOP" button	• ID: **3** • Type: **5** • Flag: **0** • Value 1: **${dragChannelCode}**
"COMPUTER" button	• ID: **4** • Type: **5** • Flag: **0** • Value 1: **${dragChannelCode}**
"HDMI" button	• ID: **5** • Type: **5** • Flag: **0** • Value 1: **${dragChannelCode}**

Once they have been configured The custom event properties are displayed in the *Drag Cancel* property for the selected button (FIG. 281):

Drag Cancel — Port: 1;ID: 1;Type: 5;Val 1: ${dragChannelCode}   ...

FIG. 281 Drag Cancel Event indicating sample data

5) Add a "SMALL/LARGE ICONS" Button

This example includes a button that allows the end-user to toggle between small and large "target valid" and "target invalid" icons on the drop target buttons, when a draggable button has started a drag (FIG. 282):

The end-user can press the SMALL ICONS button to display small icons for "target-valid" and "target-invalid"

valid target - small icon

invalid target - small icon

The end-user can press the LARGE ICONS button to display large icons for "target-valid" and "target-invalid"

valid target - large icon

invalid target - large icon
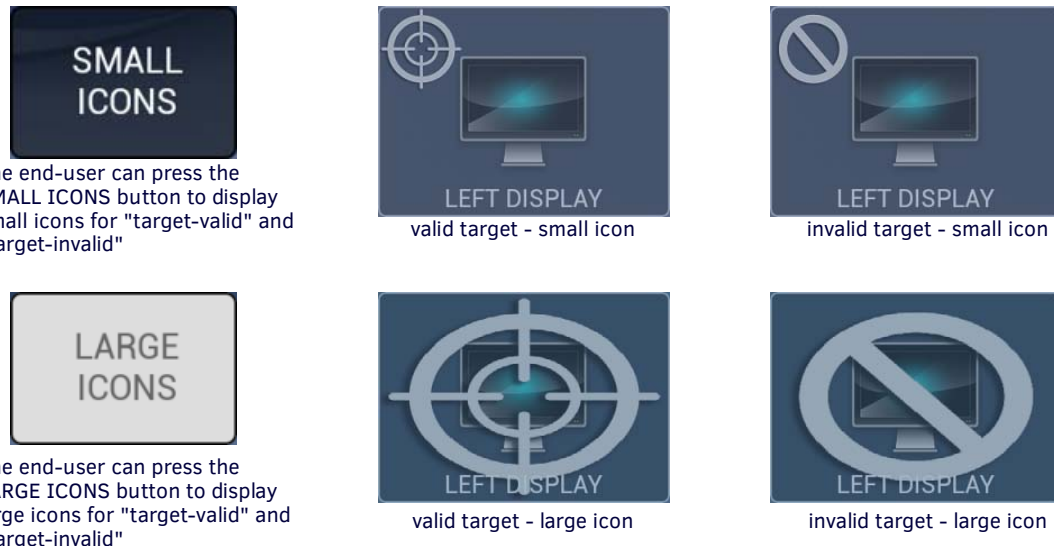
FIG. 282 SMALL ICONS and LARGE ICONS - as they will appear on the touch panel

Create a "SMALL/LARGE ICONS" Button

1. Use the Button Draw tool to create a new button.

2. Set the button's *Type* (General) property to **general**.

Set the remaining *General* properties for the "SMALL/LARGE ICONS" as shown in FIG. 283:
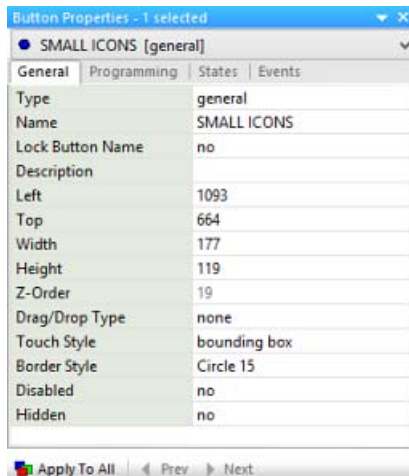
FIG. 283 "SMALL/LARGE ICONS" Button - General Properties

Set "SMALL/LARGE ICONS" Button Properties - Programming

Set the *Programming* properties for the "SMALL/LARGE ICONS" button as shown in FIG. 284:
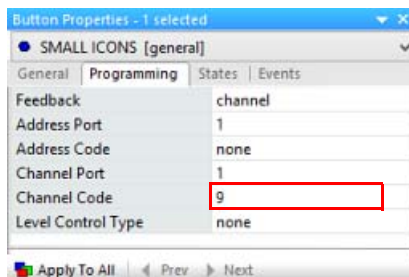
FIG. 284 "SMALL/LARGE ICONS" Button - Programming Properties

On the "SMALL/LARGE ICONS" button, set the *Channel Port* to **9**.

Set "SMALL/LARGE ICONS" Button Properties - States

Set the *States* properties for the "SMALL/LARGE ICONS" button as shown in FIG. 285:

Off State Selected   On State Selected
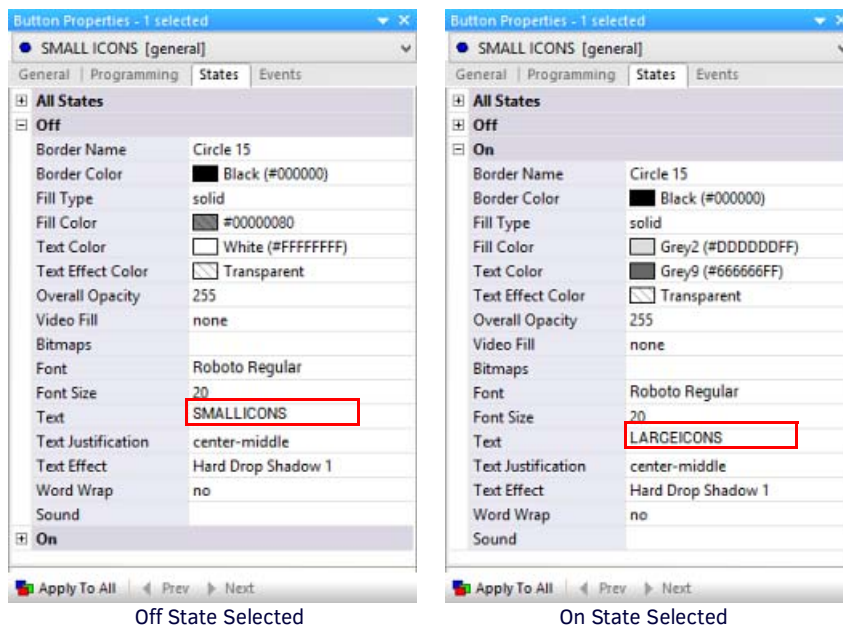
FIG. 285 "SMALL/LARGE ICONS" Button - States Properties

- Set the *Text* property on State 1 to "SMALL ICONS"
- Set the *Text* property on State 2 to "LARGE ICONS"

6) Add a "CLEAR DISPLAY SOURCE" Button

This example includes the option for the user to "clear" the current input (Source) device setting on the Displays (FIG. 238):

FIG. 286 CLEAR DISPLAY SOURCE button

To add a button that supports this option:

Create a "CLEAR DISPLAY SOURCE" Button

1. Use the Button Draw tool to create a new button.
2. Set the button's *Type* (General) property to **general**.

Set "CLEAR DISPLAY SOURCE" Button Properties - General

Set the remaining *General* properties for the "CLEAR DISPLAY SOURCE" buttons as shown in FIG. 239:
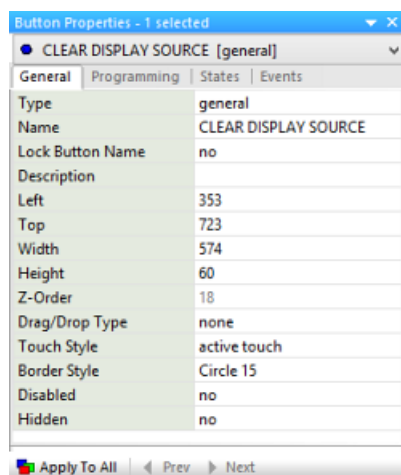
FIG. 287 "CLEAR DISPLAY SOURCE" Button - General Properties

Set "CLEAR DISPLAY SOURCE" Button Properties - Programming

Set the *Programming* properties for the "CLEAR DISPLAY SOURCE" button as shown in FIG. 240:
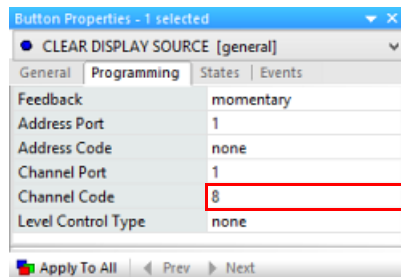
FIG. 288 "CLEAR DISPLAY SOURCE" Button - Programming Properties

On the "CLEAR DISPLAY SOURCE" button, set the *Channel Code* to **8**.

7) Write NetLinx Code To Respond To Custom Event

The NetLinx Code below utilizes the custom events that were configured in the TP file for "behavior" changes on the drop target buttons via the states configured earlier in this section.

1. Use NetLinx Studio 4 to add the following code to the NetLinx program loaded on the Master:

```
PROGRAM_NAME='MASTER'

DEFINE_DEVICE
dvTP = 10001:1:0

DEFINE_CONSTANT
//dropTargets
INTEGER leftDT   = 16
INTEGER centerDT = 17
INTEGER rightDT  = 18
//draggables
INTEGER btnDG1 = 1
INTEGER btnDG2 = 2
INTEGER btnDG3 = 3
INTEGER btnDG4 = 4
INTEGER btnDG5 = 5

DEFINE_VARIABLE
//an array to store our dropTarget buttons
INTEGER dTBTNS[] = {leftDT ,centerDT ,rightDT}
//an array to store our draggable buttons
INTEGER dgBTNS[] = {btnDG1 ,btnDG2 ,btnDG3 ,btnDG4,btnDG5}
//an array to store draggable buttons belonging to group_1
INTEGER dgG1[] = {btnDG1,btnDG2}
//an array to store draggable buttons belonging to group_2
INTEGER dgG2[] = {btnDG3}
//an array to store draggable buttons belonging to group_3
INTEGER dgG3[] = {btnDG4,btnDG5}
//to track small/large Icon
INTEGER nLargeIcon = 0
//to store draggable address from start event
INTEGER nDragAddress = 0
//to store which group we are in
INTEGER ngroupID = 0

DEFINE_MUTUALLY_EXCLUSIVE
([dvTP,1]..[dvTP,5])

//In this example the groups are defined as follows
//       - buttonAddresses 1,2  are assigned: group_1
//       - buttonAddress   3    are assigned: group_2
//       - buttonAddresses 4,5  are assigned: group_3
//       - leftDT   [16] will accept draggables from: group_1
//       - centerDT [17] will accept draggables from: group_1, group_2, group_3
//       - rightDT  [18] will accept draggables from: group_3

DEFINE_EVENT

DATA_EVENT[dvTP]
{
    ONLINE:
    {
        nLargeIcon = 0
```

```
                //By default ^BDC is enabled, let's disable it
            SEND_STRING dvTP,"'BDC-0,0,0,0,0'"

            //Let's make sure we are starting in state 1
            SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',1,1,0'"
            SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',1,1,0'"
            SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',1,1,0'"
        }
    }

    //Custom event for START [1]
    //Any time a draggable is initiated (long press, dragShadow appears)
    //a START event is sent.
    //CUSTOM_EVENT[dvTP,ID,Type]

    CUSTOM_EVENT[dvTP,dgBTNS,1]
    {
        //to store the draggable group name
        CHAR cDragGroup[DATA_MAX_VALUE_LENGTH]

        //Get the dragButtonAddress from the customEvent
        nDragAddress = custom.value1
        cDragGroup = custom.text

        if(COMPARE_STRING(cDragGroup,'group_1'))
        {
            ngroupID = 1
            if(nLargeIcon)
            {
                SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',4,4,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',4,4,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',5,5,0'"
            }
            else
            {
                SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',2,2,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',2,2,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',3,3,0'"
            }
        }
        else if(COMPARE_STRING(cDragGroup,'group_2'))
        {
            ngroupID = 2
            if(nLargeIcon)
            {
                SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',5,5,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',4,4,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',5,5,0'"
            }
            else
            {
                SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',3,3,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',2,2,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',3,3,0'"
            }
        }
        else
        {
            ngroupID = 3
            if(nLargeIcon)
            {
                SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',5,5,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',4,4,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',4,4,0'"
            }
            else
            {
                SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',3,3,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',2,2,0'"
                SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',2,2,0'"
            }
        }
    }

    //Since we are signaling what groups are valid on the START event,
    //there is no need to handle ENTER or EXIT events.
```

```
//Custom event for DROP [4]
//A DROP event occurs when a draggable has been released within the boundaries
//of a valid dropTarget. A valid dropTarget is a dropTarget that has a group
//which the draggable is assigned to.

CUSTOM_EVENT[dvTP,dTBTNS,4]
{
    SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',1,1,0'"
    SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',1,1,0'"
    SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',1,1,0'"
    //turn on the source(draggable)
    ON[dvTP,nDragAddress]
}
//Custom event for CANCEL [5]
//A CANCEL event occurs when a draggable has been released over anything that
//is not a VALID dropTarget.
CUSTOM_EVENT[dvTP,dgBTNS,5]
{
    SEND_COMMAND dvTP,"'^ANI-',ITOA(leftDT),',1,1,0'"
    SEND_COMMAND dvTP,"'^ANI-',ITOA(centerDT),',1,1,0'"
    SEND_COMMAND dvTP,"'^ANI-',ITOA(rightDT),',1,1,0'"
}

BUTTON_EVENT[dvTP,8] //CLEAR DISPLAY SOURCES
{
    PUSH:
    {
        OFF[dvTP,1]
        OFF[dvTP,2]
        OFF[dvTP,3]
        OFF[dvTP,4]
        OFF[dvTP,5]
    }
}

BUTTON_EVENT[dvTP,9] //SMALL/LARGE ICON DEMO MODE
{
    PUSH:
    {
    [dvTP,9] = ![dvTP,9]
    nLargeIcon = [dvTP,9]
    }
}
```

2.   Save changes.

NOTE: *The NetLinx code shown above is included in the NetLinx Studio Workspace file (AdvancedDragAndDropExample.apw) that is in the Drag and Drop Demo.ZIP file.*

8) Use NetLinx Studio 4 to Compile and Transfer the Project Files

Use NetLinx Studio 4 to compile the code and transfer the project files to the Master:

1.   At the top of the AdvancedDragAndDropExample.axs source code file, change the *dvTP* value to match the device number of your touch panel (FIG. 241):

```
PROGRAM_NAME='MASTER'

DEFINE_DEVICE
dvTP = 10001:1:0
```

FIG. 289 dvTP Device Number value - Change to match the device number of your Touch Panel

2.   Compile the code (select **Build > Build Active System**).
3.   Transfer the *DragAndDropNoGroups.apw* workspace file to the NetLinx Master:
    a.   Select **Tools > File Transfer** to open the *File Transfer* dialog.
    b.   Open the *Send* tab and clear any files that are listed by clicking **Remove All**.
    c.   Click **Add** to open the *Select Files for File Transfer* dialog.
    d.   Select the top-level **Projects** folder to select all files in the workspace for transfer (FIG. 242):
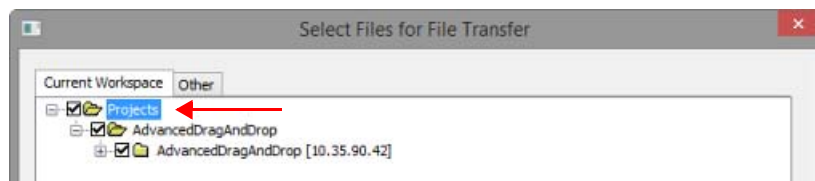
FIG. 290 Select Files for File Transfer dialog

e. Select **OK** to return to the *File Transfer* dialog (FIG. 243):
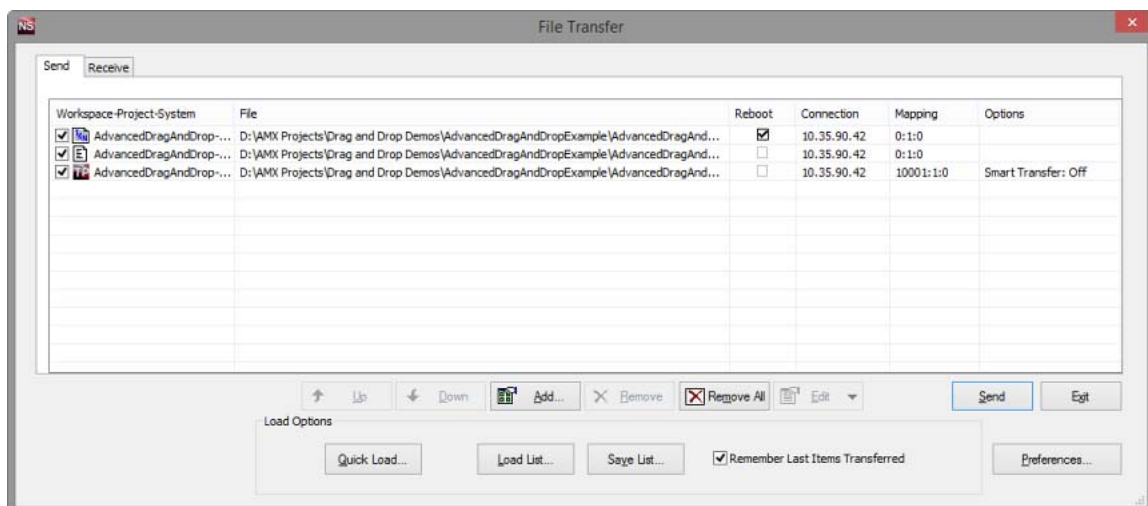
FIG. 291 File Transfer dialog - indicating files in the DragAndDropNoGroups.apw workspace queued for transfer

f. Click **Send** to initiate the file transfer.

g. The progress of the transfer is indicated in the Output Bar.

End Result

The result of this demo is a touch panel page with five draggable buttons representing source (input) devices and three drop target buttons representing output devices (Displays):

FIG. 292 Drag and Drop Demo - Page Layout

- The LEFT DISPLAY, CENTER DISPLAY and RIGHT DISPLAY buttons are drop target buttons representing three output devices (Displays) that can potentially accept the sources represented by the five draggable buttons as inputs:
  - The ENZO and iPAD draggable (source) buttons are configured to use LEFT DISPLAY and CENTER DISPLAY as valid drop targets.
  - The LAPTOP draggable (source) button is configured to use CENTER DISPLAY (only) as valid drop target.
  - The COMPUTER and HDMI draggable (source) buttons are configured to use CENTER DISPLAY and RIGHT DISPLAY as valid drop targets.
- These buttons can each be dragged onto the drop target buttons individually. When one of the draggable buttons is released within the bounds of a drop target, NetLinx code receives the custom events and makes visual changes to reflect the validity of the drag. If the drop target is valid for the selected draggable button, the target Display switches to the source represented by the draggable button that was dropped.

  Note that the LEFT DISPLAY (drop target) button indicates that it is a *valid* drop target for the ENZO and iPAD draggable buttons (FIG. 293):
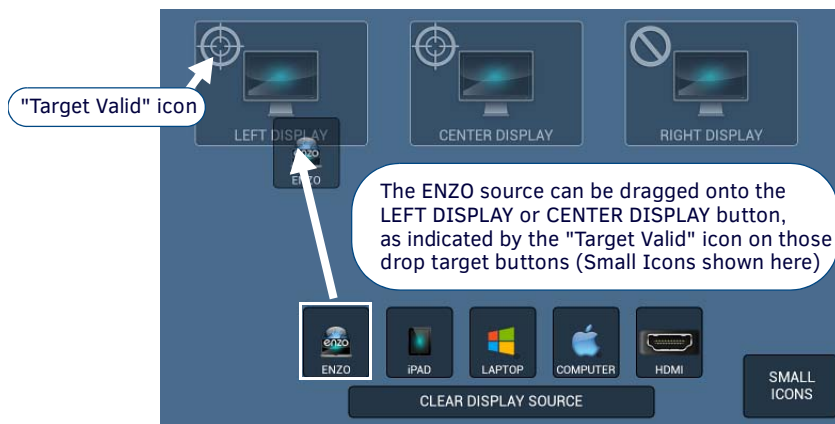
FIG. 293 Drag and Drop Demo - Dragging the COMPUTER Source button onto the LEFT DISPLAY Drop Target button

When the ENZO button is released, the LEFT DISPLAY uses it as it's new input. Note that the currently selected input is indicated with a highlighted source button (FIG. 294):

FIG. 294 Drag and Drop Demo - COMPUTER is the currently selected Source for the LEFT DISPLAY

● The LAPTOP draggable button is configured such that the LEFT DISPLAY button is an *invalid* target, therefore this buttons cannot be released on the LEFT DISPLAY drop target button.

Note that the LEFT DISPLAY and RIGHT DISPLAY drop targets indicates that they are both in*valid* drop targets for the LAPTOP draggable button (FIG. 295):
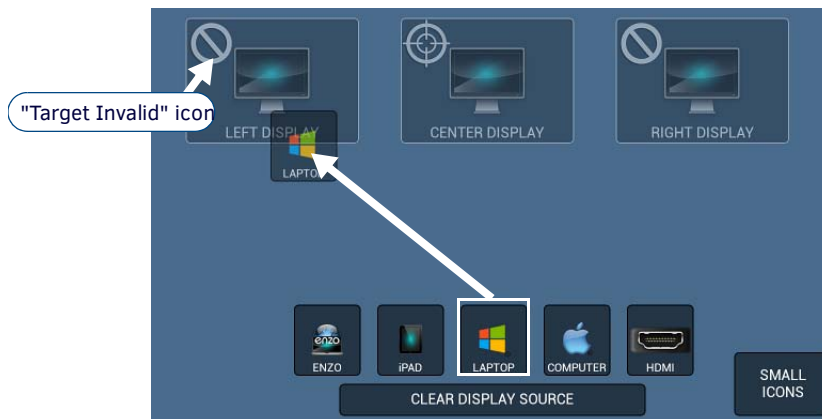
FIG. 295 Drag and Drop Demo - iPAD not allowed as a Source for the LEFT DISPLAY

● The CLEAR VTC SOURCE button will clear the current input setting on all displays when pressed.
● The SMALL ICONS button will toggle the "valid target" and "invalid target" icons (on the Display buttons) from small icons to large icons (FIG. 296):
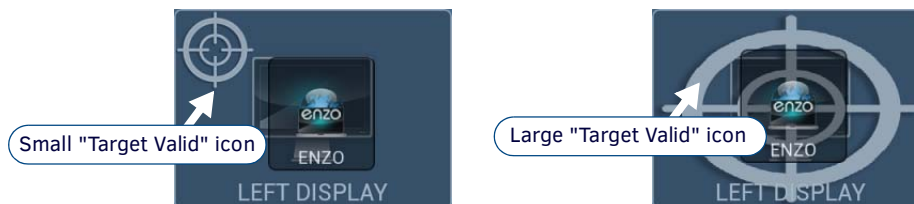
FIG. 296 Target Valid Icons (Small/Large)

Fills, Text Effects, Animation Effects & Tweening

Gradient Fills

Gradient Fills allow you assign gradient color fills using up to 10 colors to Pages, Popups, Sub-pages, and Buttons. Gradient Fills are managed via States properties.

- Gradient fills utilize a minimum of two colors to create a multi-color graded fill effect. Gradient fills can use up to ten colors.
- When any *Fill Type* other than **Solid** is chosen, the colors used for the gradient are selected via the Fill Gradient Colors (State) property.

NOTE: *The transparency mask (alpha channel) color used for Pages is not supported as a gradient fill color. All other elements support the transparency mask.*

Gradient Fill Types

TPD5 provides the following types of Gradient Fills, selected via the *Fill Type* (State) Property (FIG. 297).

Radial	Top-Left to Bottom-Right	Right to Left	Bottom-Left to Top-Right
Sweep	Top to Bottom	Bottom-Right to Top-Left	
Left to Right	Top-Right to Bottom-Left	Bottom to Top	

FIG. 297 Gradient Fill Types

Radial Fills

Radial is a radial gradient fill pattern starting at the center of a specified point blending in circular fashion out to the edges of the element. There are specific (State) properties associated with Radial gradient fills.

Note that If **Radial** is selected as the *Fill Type*, the following additional State Properties are provided:

- State Properties - Gradient Radius (see page 253)
- State Properties - Gradient Center X% (see page 253)
- State Properties - Gradient Center Y% (see page 253)

Sweep Fills

Sweep is a gradient fill pattern blending colors counter-clockwise in radial sweep fashion around the center of the element.

The starting point of the sweep is on the center-right-half of the element.

NOTE: *In order to create a complete blending of colors (i.e. without a hard transition on the right) the start and end colors must be the same.*

Selecting Colors for a Gradient Fill

1. With any gradient Fill Type selected, click on the **Fill Gradient Colors** (State) Property and click the browse (...) button to open the *Fill Colors* dialog. Use this dialog to include up to 10 colors in the gradient fill for the selected element.

2. In the *Fill Colors* dialog, click **Add** to select the first color, via the *Colors* dialog. With a color selected, click **OK** to close the *Colors* dialog and return to the *Fill Colors* dialog.

3. Click **Add** again, and select a second color via the *Colors* dialog. Click **OK** to close the *Colors* dialog and add the second color to the list in the *Fill Colors* dialog.

4. Repeat this process to add up to 10 colors to this gradient fill.

The example below indicates a gradient fill with five colors assigned (FIG. 298):.

FIG. 298 Fill Colors dialog

- To change a color in this list, select the color entry and click the browse (...) button to open the *Colors* dialog, to select a different color.
- To delete a color from this gradient fill, select a color and click **Delete**.
- Use the **Move Up** and **Move Down** buttons to arrange to order of the colors as desired.

NOTE: *Regardless of type, gradient blending starts with the first color defined in the Fill Gradient Colors list and ends with the last color defined.*

5.   Click **OK** to save changes and close this dialog.

Text Effects

Text effects are graphic effects that can applied to button/page/popup text. Each text effect is available in several variations:

Text Effect - none   Text Effect - Drop Shadow
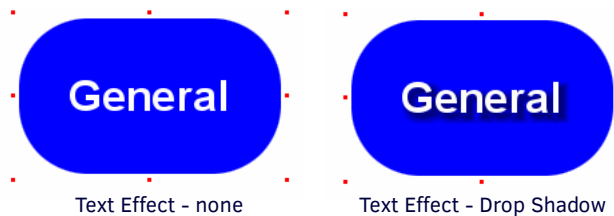
FIG. 299 Fill Colors dialog

To apply a text effect to the button text, click the browse button (...) to access the *Text Effect* sub-menu. This sub-menu presents all available text effects, sorted by type.

- Click the + symbol next to any effect type in the sub-menu to see all of the variations on that effect.
- Once you have selected a text effect, use the Text Effect Color field to specify a color for the effect.

Animation Effects

TPD5 utilizes two main concepts for supporting animation effects in your projects:

- Animation Wizard (see below)
- Tweening (see page 219)

Animation Wizard

The *Animation Wizard* is a powerful tool included with TPD5. It guides you through the steps of generating an animation sequence that can be applied to a multi-state button to apply impressive visual effects to your buttons. Multi-state buttons can have up to 256 states, all of which are available to be used as "frames" in an animation.

You can create an animation sequence for a multi-state button manually, by creating a series of states and applying a different bitmap to each state and treating each state as an individual frame. This however, would be a very tedious and time-intensive process. The Animation Wizard automates most of the process and makes the task easy.

NOTE: *If you select a General (two-state) button to use with the animation wizard, the wizard will allow you to add the necessary number of states to the button to accommodate the animation sequence (and automatically change the button type to Multi-State General).*

To create a simple button animation using the Animation Wizard:

1.   Select **Button > Animation Wizard** to start the wizard. This opens the first dialog: *Animation Wizard - Select Type (Step 1 of 6)*. Use the radio buttons in this dialog to specify the type of animation to create (*Bitmap* or *Chameleon Image*).
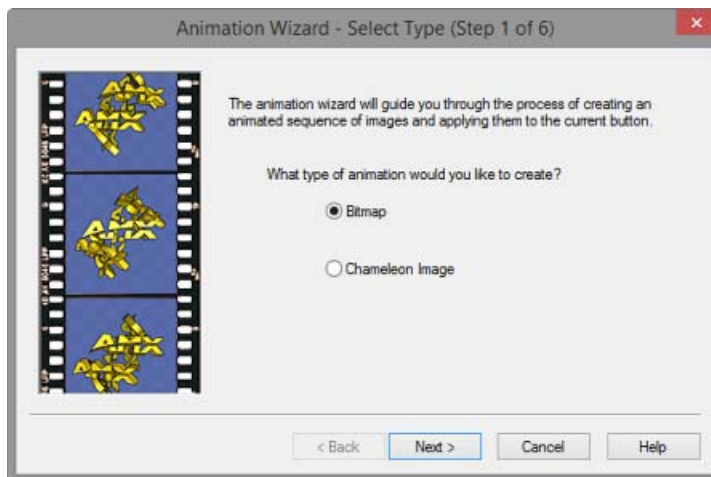
FIG. 300 Animation Wizard - Select Type (Step 1 of 6) dialog

- Select *Bitmap* to use bitmaps in the animation.
- Select *Chameleon Image* to use (32-bit PNG) chameleon images in the animation.

2. Click **Next** to proceed to the *Animation Wizard - Create Sequence (Step 2 of 6)* dialog (FIG. 301):
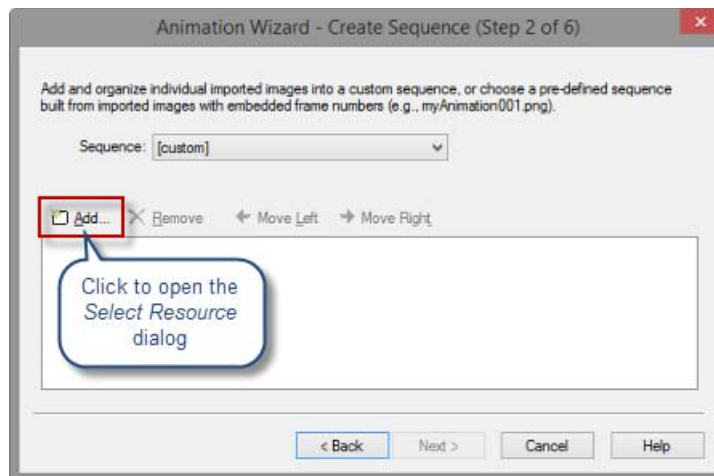
FIG. 301 Animation Wizard - Create Sequence (Step 2 of 6) dialog

Use this dialog to select the images to be included in the animation, and set the sequence of the images.

The *Sequence* selection is set to "**[custom]**" by default, and there are no other options in this drop-down list until at least one sequence has been created. Once you import a series of images with identical names except for a post-fixed numeric indicator, they will be available via this drop-down list.

3. Click the **Add** button to open the *Select Resource* dialog, where you can select which images to include in the animation (FIG. 302):
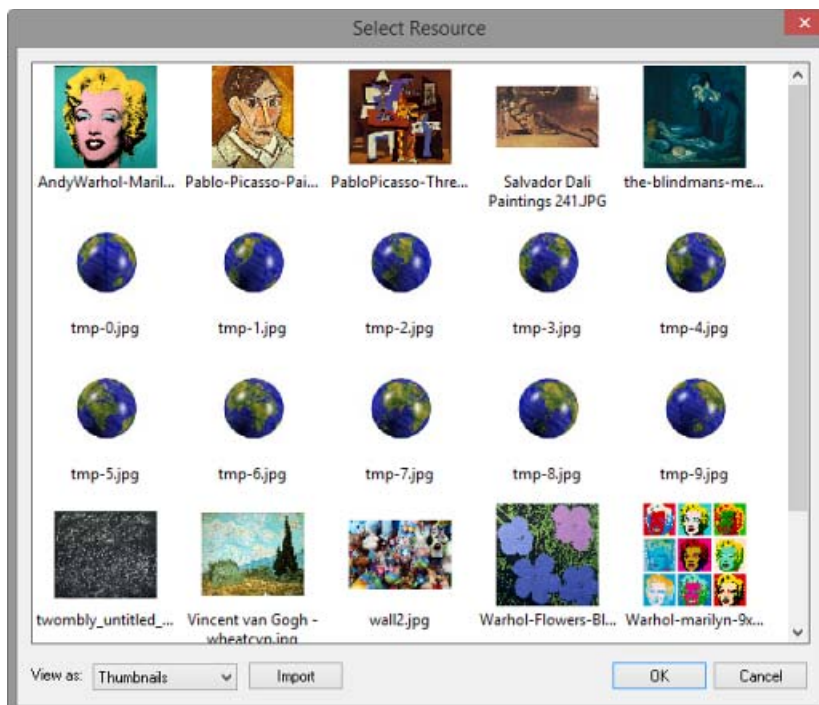
FIG. 302 Select Resource dialog

- Ctrl+click to select multiple files, or Shift+click to select the range of files between two selections. Note that the only images that are available to choose from are those that have been imported into the project, through the **File > Import Resources** option, or via the **Import** button on the *Resource Manager* dialog.
- In order for images to appear in the Sequence drop down for selection, the images that you use to create the animation must be named in sequence (i.e. frame_000.jpg, frame_001.jpg, frame_002.jpg, frame_003.jpg, etc...). Also, make sure the numbering convention used for the files is consistent (i.e. don't mix "1" and "01" or "001"). Although you can edit the sequence of the images in the animation manually, it makes things easier if you take naming into consideration before importing the files.
- There are many third-party application available that can extract individual frames from an existing animated GIF. Once extracted, these files can be imported into your TPD5 project, and the original GIF animation can be simulated. Note that since TPD5 does not support the (proprietary) .GIF file type, you will have to save the frames out to a supported file type.

4. Click **OK** in the *Select Resource* dialog to return to the *Animation Wizard - Create Sequence (Step 2 of 6)* dialog. The image files you selected now appear in alpha-numeric order in the preview window of this dialog. If the images were named consecutively, you shouldn't have to modify the sequence at all (FIG. 303):
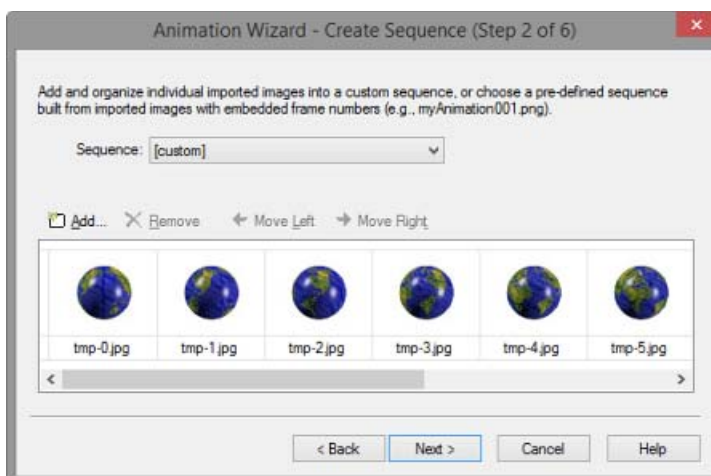
FIG. 303 Animation Wizard - Select Type (Step 2 of 6) dialog - with images added

- Use the **Move Left** and **Move Right** buttons to re-arrange the image sequence (if necessary).
- Use the **Add** and **Remove** buttons to add/remove selected images from the sequence (if necessary).

5. Click **Next** to proceed to the *Animation Wizard - Size & Position (Step 3 of 6)* dialog. Use this dialog to specify the size and position of the animation relative to the button that will contain it.
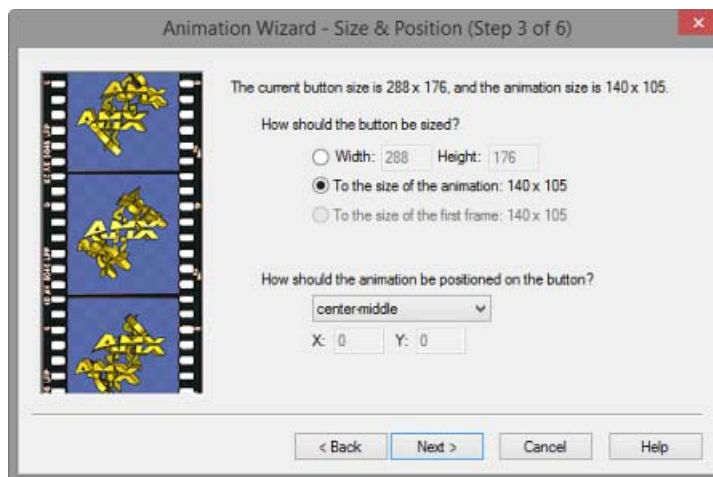
FIG. 304 Animation Wizard - Select Type (Step 3 of 6) dialog

This dialog tells you two important pieces of information: the size of the animated image, and the size of the button that will contain it. Use the radio buttons in the top-half of the dialog to specify whether to scale the button to fit the animated image, or to simply apply the animation to the button at its current size. If you are applying an animation that is bigger than the containing button, a warning message is displayed along the bottom of the dialog, and the animated image will be clipped (cropped) to fit in the button.

Use the options in the lower-half of this dialog to specify the positioning of the animated image relative to the button containing it. Select the desired position option from the drop-down list. If you select Absolute, use the X and Y fields to indicate the desired position.

NOTE: *All position (X-Y) values in TPD5 are zero-based (measured in pixels), meaning that the upper-left corner of each page is represented by the X-Y value of 0, 0.*

6. Click **Next** to proceed to the *Animation Wizard - Assign Frames (Step 4 of 6)* dialog. Use this dialog to create enough states to accommodate the number of frames in the animation sequence.
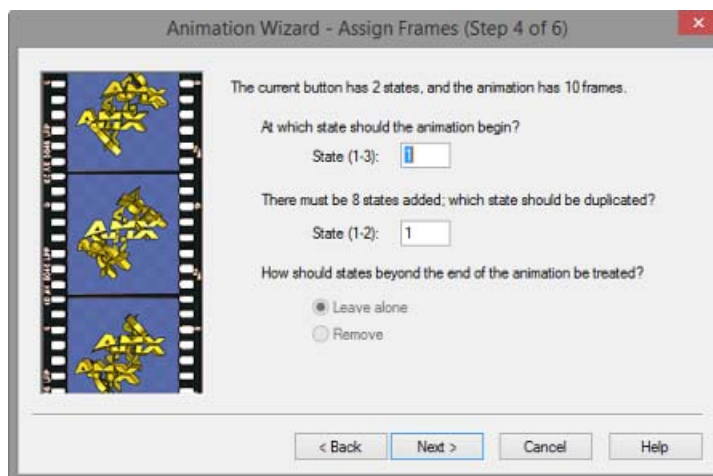
FIG. 305 Animation Wizard - Select Type (Step 4 of 6) dialog

For example, if you create a multi-state button, and open the Animation Wizard before adding any states, you have only two states on the button. Obviously, the animation sequence will have more than two frames. Lets say that your sequence has ten frames.

The options in this dialog allow you to automatically detect the number of frames in the animation sequence, and the number of states available on the button to which the animation is to be applied. So, for a ten-frame animation, you'll have to add at least eight states to the button (one frame per button state).

- Use the first field to indicate which state to start the animation on (default = 1, the first state).
- Use the next field to indicate which of the existing states to duplicate, if more states are required for the animation.
- If button states exist beyond the end of the animation, use the radio buttons to indicate how to handle them (*Leave Alone* or *Remove*).

7. Click **Next** to proceed to the *Animation Wizard - Appearance (Step 5 of 6)* dialog. Use this dialog to specify how treat the Button Border Style, and Button Fill Color on the resulting button.
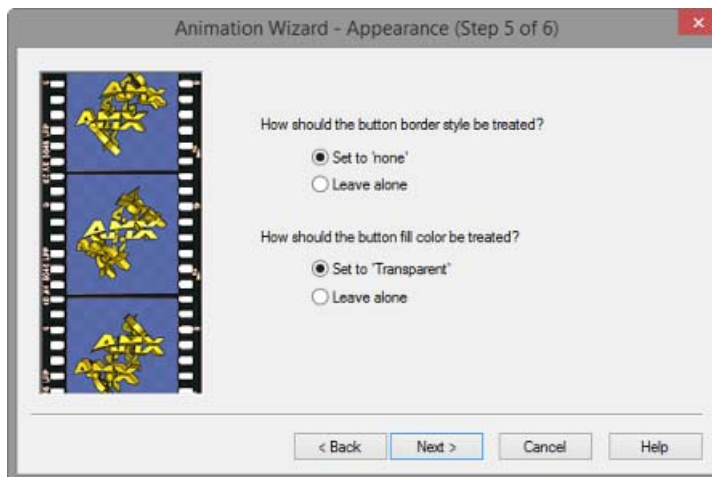
FIG. 306 Animation Wizard - Select Type (Step 5 of 6) dialog

These options default to no border (Set to "none") and transparent fill (Set to "transparent").

8. Click **Next** to proceed to the *Animation Wizard - Finish (Step 6 of 6)* dialog. This dialog lists the actions to be taken to generate the button animation.
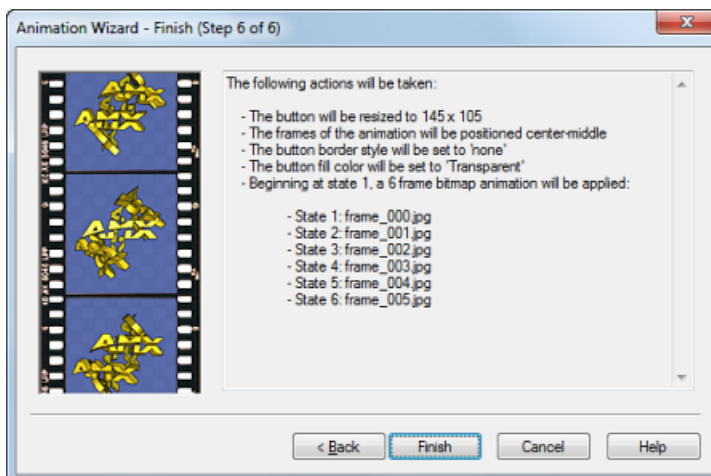
FIG. 307 Animation Wizard - Select Type (Step 6 of 6) dialog

- The number of states to be added to the button, and which state to duplicate.
- The position of the animation on the button.
- The starting state for the animation, the number of frames in the animation, and the animation type (Bitmap).
- A listing of each state and the image file associated with it.

9. Click **Finish** to generate the animation. Note that the animation is displayed in the State Manager window (if it is open).
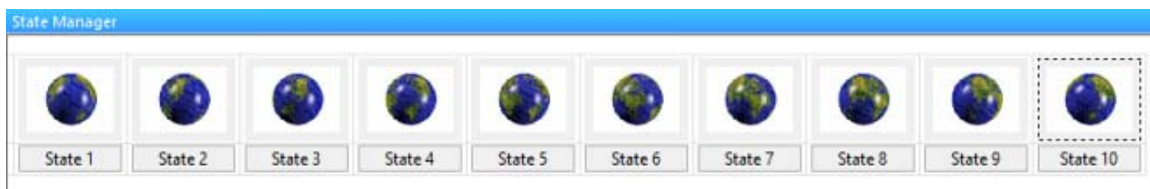
FIG. 308 State Manager window showing an animated bitmap

- Use the Button Preview window to view the animation (select **View > Button Preview,** and click **Push**).
- Remember, many other animation effects can be achieved via the Tweening tools of TPD5 (see the *Tweening* section on page 219).

Tweening

Short for "in-betweening", *tweening* is the process of altering the display properties of intermediate frames between two images to give the appearance that the first image evolves smoothly into the second image. Tweening is a common concept in all sorts of animation software.

TPD5 supports multi-state buttons, which can have up to 256 states which are used to animate a button push from Off to On and back again to Off. In TPD5, each state of a multi-state button can be thought of as an individual frame. So, if you create a multi-state button with 256 states, you might say that you have 256 frames available for the animation.

When the multi-state button is turned On it will display all the assigned states from first to last within a specified time interval. This is called "Animate Time Up", and is definable in 1/10th second increments. When the button is turned back Off, the states will be displayed in reverse order. This is called "Animate Time Down", also definable in 1/10th second increments. The Animate Time Up and Animate Time Down values are set in the General tab of the Properties Control window.

Provided you are not creating an image-based animation, the tweening process greatly simplifies the process of generating each state individually by automatically creating a gradual transition across all states based on the state properties of the first and last states.

Beyond simplifying the process of creating motion animations, tweening also generates very smooth color transition effects that would be difficult or impossible to do any other way. When used in combination with TPD5's ability to handle RGB colors (including the opacity setting), it is also possible to use tweening to make buttons fade in and out on the page.

NOTE: *Since transparent borders are not supported in TPD5, to make a button fade completely in/out on a page, you would have to create the button without borders.*

To illustrate, here's an example of how to create a simple button animation using just some of the tweening options available in TPD5:

1. Create a new multi-state button. Note that even multi-state buttons initially have only two states, as indicated by the State Manager window (FIG. 309):
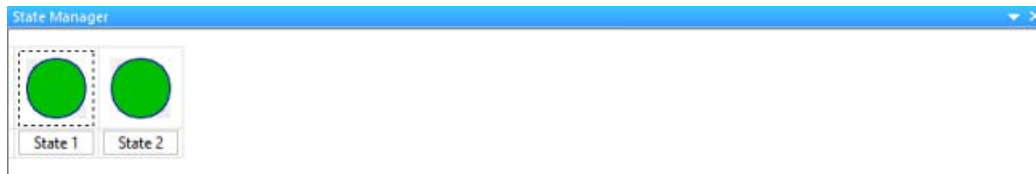
FIG. 309 State Manager window showing a button with two states

2. Select **Button > Add States**, and add 10 states, for a total of 12 (FIG. 310):

FIG. 310 State Manager window showing a button with 12 states

3. Change the *Fill* and *Border* colors on the last state in the series (in this case State 12), via the Properties Control window (*States* tab). In this example, text was also added to the first and last states in the series - "ON" was applied to the first state in yellow, "OFF" was applied in black to the last state in red (FIG. 311):

FIG. 311 State Manager window - first and last states selected for tweening

4. Select all states in the State Manager window, and right-click to open the State Manager context menu (FIG. 312):
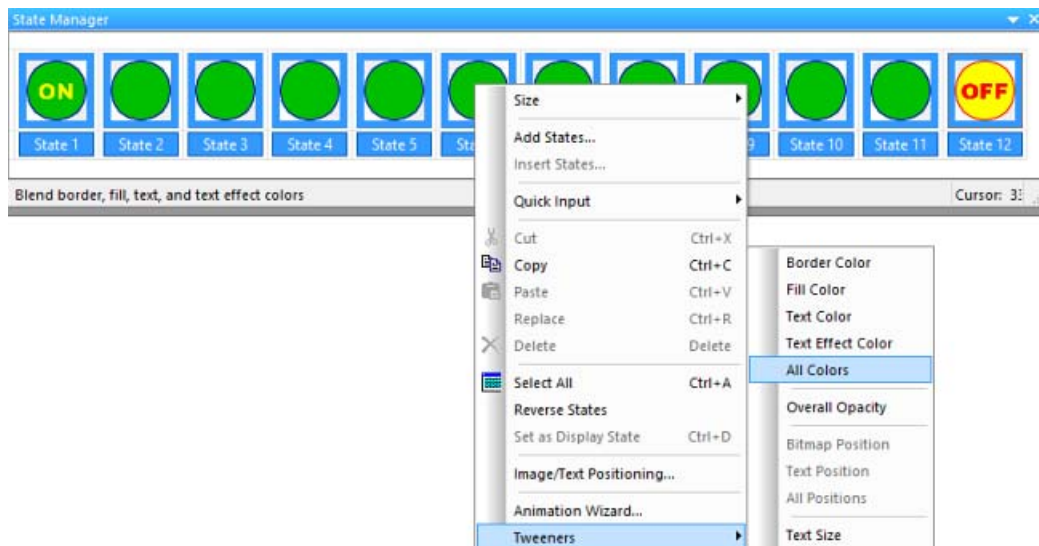
FIG. 312 State Manager - Tweeners sub-menu

5. Select **Tweeners** to open the Tweeners sub-menu, and select *All Colors*.
    - Note that the Fill Color tween option only works if the *Fill Type* (States property) is set to **Solid**.
    - The *Bitmap Position* and *Text Position* tween options only work when the Bitmap and Text Justification settings are set to Absolute positioning for the two states selected in the tweening operation.

6. The thumbnails in the State Manager change to show the results of the tween options applied. Notice the gradual transition in border and fill colors (FIG. 313):

FIG. 313 State Manager window showing a button with 12 states and Tweened colors

7. To preview the animation as it appears when the button is pushed, select **View > Button Preview** to open the Button Preview window, and click **Push**.

Creating Color Transition Effects

Use the *Border Color*, *Fill Color* and *Text Color* tweeners to easily apply color transition effects to multi-state buttons. The color tweeners can be used individually, or in combinations to create smooth fades from one color to another when the button is pressed. With 256 states available to use as "frames" in a tweened animation, these effects can be made to be very subtle and smooth. However, most color transition effects don't require that many frames to produce a very smooth fade.

When used in combination with TPD5's ability to handle RGB colors (including the opacity setting), it is also possible to use tweening to make buttons fade in and out on the page.

Note that to make buttons fade in/out completely, you would have to create the button without a border assignment, since transparent borders are not supported in this version of TPD5.

When the button is turned On it will display all the assigned states from first to last within a specified time interval. This is called "Animate Time Up", and is definable in 1/10th second increments.

When the button is turned back Off, the states will be displayed in reverse order. This is called "Animate Time Down", also definable in 1/10th second increments. The *Animate Time Up* and *Animate Time Down* values are set in the General tab of the Properties Control window.

To create a color transition effect:

The following steps apply to all three button color attributes (Border Color, Fill Color, Text Color and Text Effect Color):

1. Select (or create a new) multi-state button with at least three states. Note that the more states you use, the smoother the transitions will appear (FIG. 314):

FIG. 314 State Manager window - multi-state button with four states

2. Apply a color (Fill Color, Border Color, Text Color, Text Effect Color or any combination) to the last state that is different from that of the first state (FIG. 315):

FIG. 315 State Manager window - colors changed on last state

> **NOTE:** *To create a fade effect, leave the colors the same for the first and last states, but change the opacity on one of them to zero (via the Colors dialog, set to RGB colors).*

- An opacity setting of zero makes the button totally transparent.
- An opacity setting of 255 (max) makes the button totally opaque.

3. Ctrl+click to select two states in the State Manager window that are separated by at least one state (do not select the intermediate states). The color tweener(s) will generate a transition effect that fades the first color into the second (FIG. 316):

FIG. 316 State Manager window - all states selected

> **NOTE:** *The most basic type of color transition effect starts at the first state (or frame) and ends at the last. However, the TPD5 tweening tools are not limited to one tween effect per multi-stage button press. Experiment with applying multiple color tweens to the same button, and with different combinations of tweeners.*

4. Right-click on one of the highlighted states in the State Manager window, and select *Tweeners* from the context menu to open the Tweeners sub-menu.

5. Select one or more of the color tweeners to apply effects. The results are displayed immediately in the State Manager window (FIG. 317):

FIG. 317 State Manager window - all colors tweened

> **NOTE:** *Use the All Colors Tweener to tween all colors applied to the button.*

Creating Animated Bitmap and Text Effects

Use the *Bitmap Position* and *Text Position* tweeners to apply animated bitmap and animated text effects to multi-state buttons. These tweeners allow you to cause a bitmap or text to move around the button area when it is pressed. Animated bitmap and text effects can be used alone or in conjunction with the other tweeners to create all sorts of eye-catching visual effects.

The following steps describe how to create a basic animated bitmap effect, but note that the same method is used for animated text:

1. Select (or create a new) multi-state button (FIG. 318):
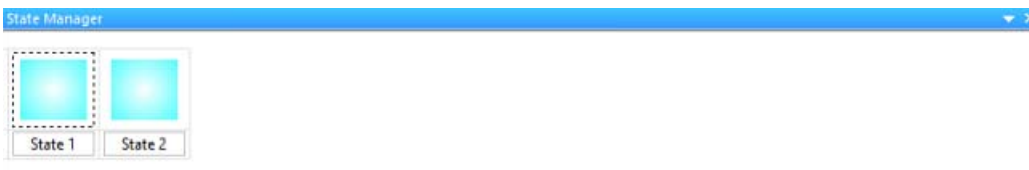
FIG. 318 State Manager window - multi-state button with 2 states

2. In the States tab of the Properties window, select State 1 and click in the **Bitmaps** property to add a bitmap via the *Bitmaps* dialog. For text effects, enter text for State 1 via the *Text* (State) property

3. In the *Bitmaps* dialog, set *Bitmap Justification* to **Absolute**. For this example, leave the *Bitmap X Offset* and B*itmap Y Offset* values at their default settings: **0, 0** (FIG. 319):
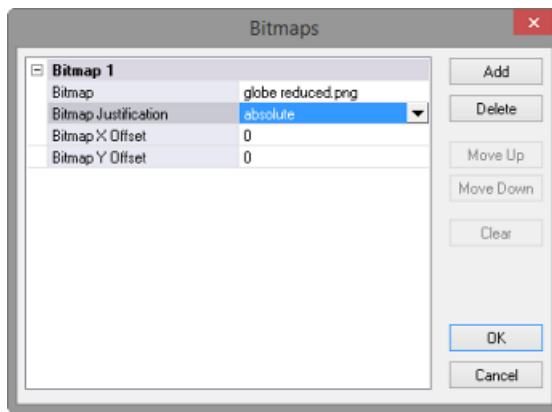
FIG. 319 Bitmaps dialog - Bitmap Justification set to Absolute

4.   Click **OK** to close the *Bitmaps* dialog. The State Manager window now shows the button with the bitmap in the 0,0 position in State 1 (FIG. 320):
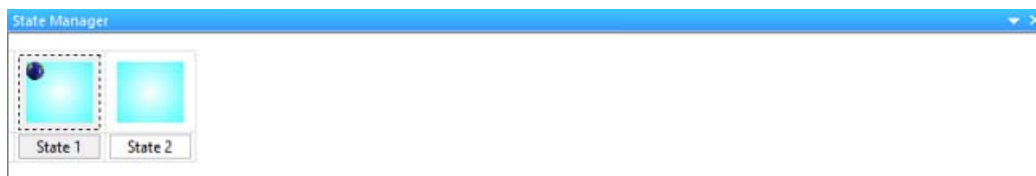
FIG. 320 State Manager window - multi-state button with bitmap applied to State 1

5.   In the States tab of the Properties window, select State 2 and click in the **Bitmaps** property to add the same bitmap via the *Bitmaps* dialog.

6.   In the State Manager window, right-click on the first state and select **Image/Text Positioning** to open the *Image/Text Positioning* dialog (FIG. 321):
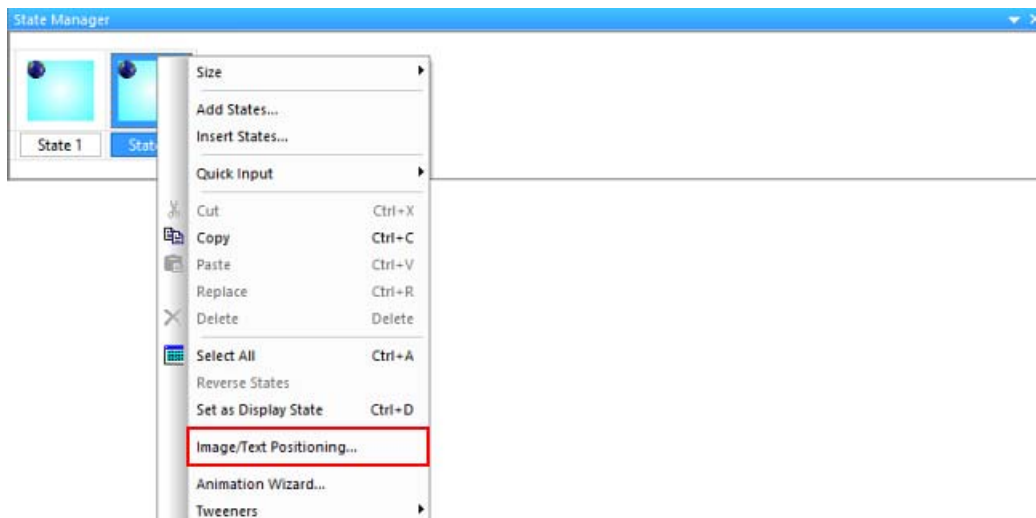
FIG. 321 State Manager context menu - Image/Text Positioning

7.   Initially, the *Image/Text Positioning* dialog shows the bitmap (or text) in it's current position on the button (FIG. 322):
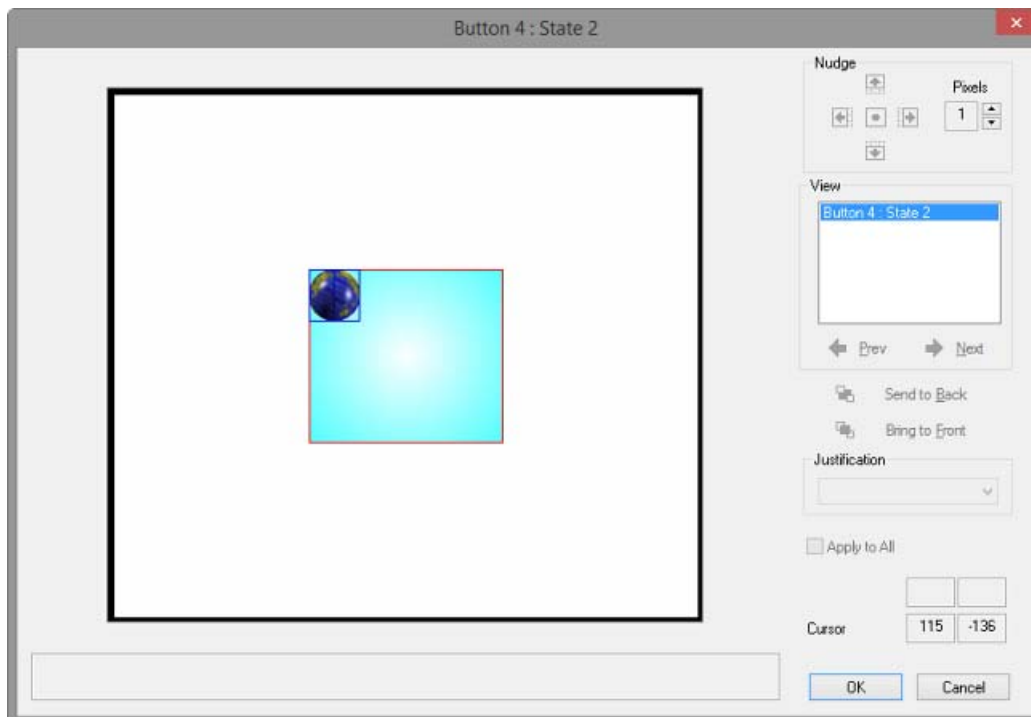
FIG. 322 State Manager context menu - Image/Text Positioning

8. Click on the bitmap in the preview window and drag the bitmap (or text) into it's start position (FIG. 323):
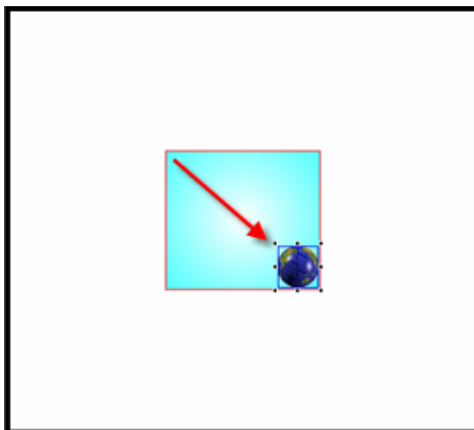
FIG. 323 Image/Text Positioning dialog - set the bitmap's end position

Alternatively, use the *Nudge* and *Justification* controls to position the bitmap (or text).

9. Click **OK** to close the *Image/Text Positioning* dialog. The State Manager shows the button with the bitmap (or text) in it's start position (State 1), and in it's end position (State 2):
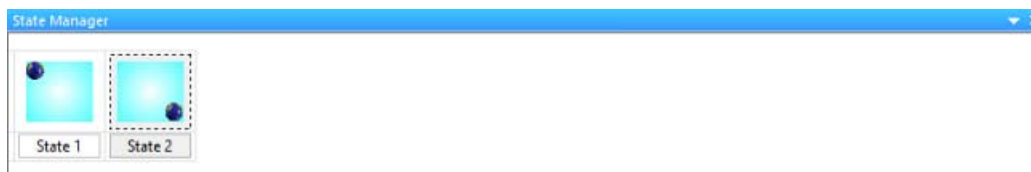
FIG. 324 Insert States dialog

10. In the State Manager window, right-click on the first state and select **Insert States** to open the *Insert States* dialog (FIG. 325):
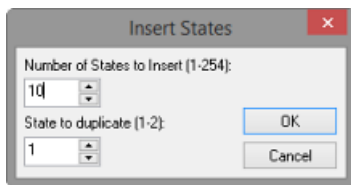
FIG. 325 Insert States dialog

11. Insert at least one state (tweeners require at least three states to work). In this example, 10 states will be added for a total of 12 states. Duplicate the first state.

12. Click **OK** to close the *Insert States* dialog. The State Manager window now shows the button with 12 states, and the bitmap is in the same position for all states (FIG. 326):

FIG. 326 State Manager window - multi-state button with 12 states with bitmaps

13. Select all states in the State Manager, and right-click to open the State Manager context menu, and select **Bitmap Position** (FIG. 327):
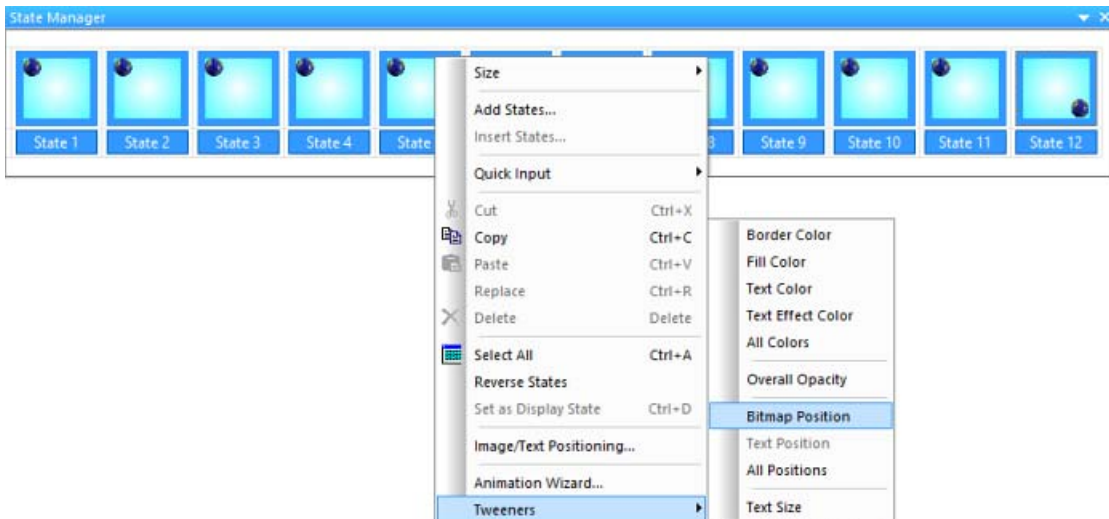
FIG. 327 State Manager context menu - Bitmap Position

14. In the State Manager window, select all states (Ctrl+A), then right-click and select **Tweeners > Bitmap Position**.

15. The result of the *Bitmap Position* tween is shown in the State Manager window (FIG. 328):
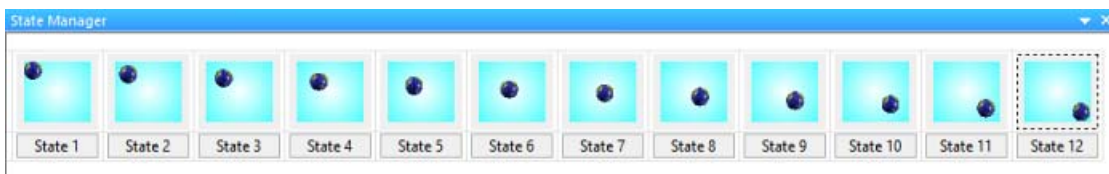
FIG. 328 State Manager window - multi-state button with 12 states with Bitmap Position tweened

NOTE: *Use Button Preview to see how this animation will appear on a button push.*

Application Windows

Overview

TPD5 supports *Application windows*, which allow you to display "windowed" applications that operate outside of TPD5 on a G5 touch panel. These include applications for email, weather updates, Web and file browsing, calendars and calculators, and other functions. The figure below shows a set of Application windows, as they appear in TPD5s Design View (FIG. 329):
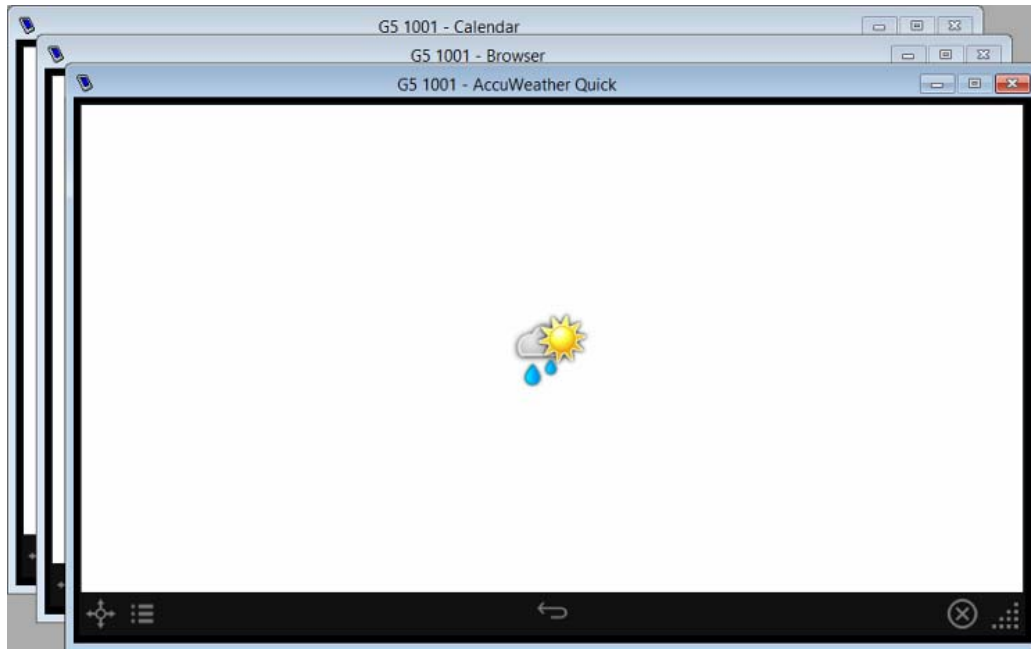
FIG. 329 Application windows

When one or more Application windows have been added to a project, they are indicated in the Application windows folder in the Workspace windows - Pages tab (FIG. 330):
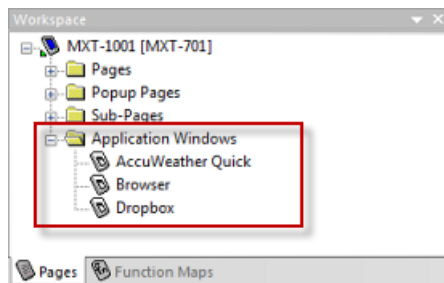
FIG. 330 Workspace windows (Pages tab) - Application windows

Note that (like popup pages), just because an Application window is listed does not mean it will appear on the panel. It is necessary to tie an action such as a button press/release or gesture swipe etc. to launch the Application window. See Events Properties for details.

- On the panel, multiple apps can be launched in separate windows, but only one instance of a particular app type (i.e. VNC, Browser, Dropbox etc.) is allowed.
- TPD5 supports defining multiple windows (layout, position, and parameter settings) for a single application type (i.e. Dropbox for example). In this case, only one instance of Dropbox will run on the panel, but its window and parameter settings can be modified by launching different Application window definitions via buttons (that is, different buttons tied to different Dropbox Application windows via the button press event property).

Opening Application Windows

Application windows can be opened into a self-contained view by double-clicking the entry in the Application windows folder in the Workspace window (Pages tab). Because applications are pre-packaged android apps, your cannot design anything within the view. The view is just a way to open an application window and edit the properties like window type and size to name a few.

Showing/Hiding Application Windows on Pages

Like popup pages, you can show and hide application windows on a page. To show an Application window on a Page: Open the page you want the application window to appear in, then right click on the Application window in the Workspace window and select Show Application window.

- Note that you can change the size and location of the Application window within the page by selecting the Application window in the Workspace window and adjusting the Left, Top, Width and Height (General) properties.
- Application windows are always drawn above other popups and buttons on a page.
- Like popup pages, showing application windows on a page is a design-time feature only, to allow you to mock up the Application window's size and position for your needs. Shown application windows have no runtime effect.

Application Window Properties

Application windows have General Properties (only) that can be configured via the Properties window (when an Application window is opened and has focus, or when shown on a page and selected).

Adding Applications

1. Select **Panel > Add Application window** to open the *Add Application window* dialog.
2. In the left side of the dialog, select the Application that you wish to add to the project.
3. In the right side of the dialog, adjust the Application properties as needed.
4. Click **OK** to add the selected Application to the current project.

Note that once an Application is added to the project, the new Application is indicated in the *Application windows* folder in the Workspace window (Pages tab). This folder lists all of the Application windows that have been added to the project (FIG. 331):
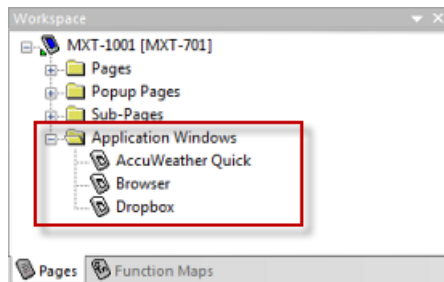
FIG. 331 Workspace window - Pages tab (Application windows)

Double-click on any Application window in the Workspace window to display the Properties (General only) of the selected Application (FIG. 332):
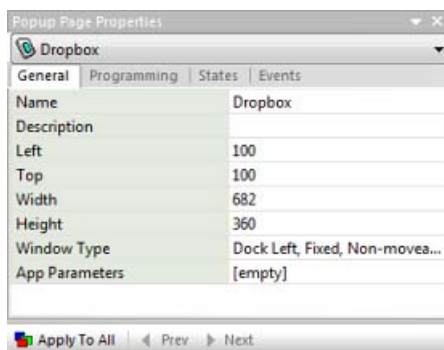
FIG. 332 General Properties - Application windows

Click on the Browse button (...) in **App Parameters**, to open the *Edit Parameter List* dialog. Use this dialog to add/edit application parameters for the selected Application window. See *Editing Application Parameters* on page 227 for details.

Setting Application Windows Properties

Application windows have *General* Properties (only) that can be configured via the fields in the Properties window. Note that Application windows do not support *Programming*, *States* or *Events* properties. To set Application window properties, select an Application window in the Workspace window (Pages tab). With the Application window selected, the Properties window displays the properties available for the Popup (General tab).

NOTE: *To edit any of the listed properties, click on an item in the right-hand column to activate the field. Depending on the item selected, you can either set the item manually, select from a drop-down menu, or both.*

Application Windows - General Properties

The following general properties are supported for Application windows:

Application windows - General Properties		
	• Name	see page 242
	• Description	see page 236
	• Left	see page 238
	• Top	see page 246
	• Width	see page 247
	• Height	see page 237
	• window Type	see page 247
	• App Parameters	see page 236

Editing Application Parameters

Some Applications require or support configuration parameters at startup that must be configured. For example, a browser application supports a page or startup page parameter so that when the browser is launched it opens to the configured page.

With an Application window selected, the *App Parameters* (General) Property is available. Use this property to enter/edit parameters for specific application types, via the *Edit Parameter List* dialog.

FIG. 333 Edit Parameter List dialog

Adding Stock Parameters

Stock parameters are application parameters that are pre-defined for a given application window. Available stock parameters vary depending on the type of application window selected.

NOTE: *Many application windows do not have any stock parameters, in which case the Available Parameters window is empty.*

Unused stock parameters for the selected application window are listed in the *Available Parameters* window of the *Edit Parameter List* dialog (FIG. 334):
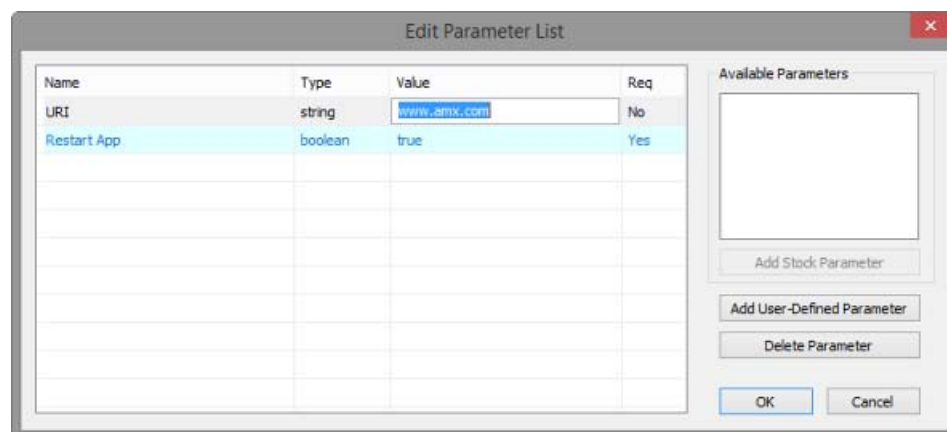
FIG. 334 Edit Parameter List dialog (for a Browser Application Window)

For example, the *Browser* application window uses two available stock parameters: *URI* and *RestartApp*.

- The *URI* parameter is indicated in the Parameters list, as shown in FIG. 334. Since this is not a required parameter, it can be removed from the Parameter list (via the **Delete Parameter** button), in which case it is moved to the Available Parameters list as an unused available parameter. It can be added to the *Parameters* list via the **Add Stock Parameter** button.
- The *Restart App* parameter cannot be removed, because it is a required parameter (see **Yes** in the *Req* column).

NOTE: *Stock parameters are pre-defined and cannot be deleted, renamed, or type-changed - only the value is editable. User defined parameters will appear under any default definitions, and are displayed in the order in which they are defined.*

Adding User-Defined Parameters

User-Defined application parameters can be added and configured via the **Add User-Defined Parameter** option in the *Edit Parameter List* dialog:

1. Click the Browse (...) button in the **App Parameters** property (*General* tab of the Properties window) to open the *Edit Parameter List* dialog.
2. Click **Add User-Defined Parameter** to add a new parameter to the parameter list.
3. Use the fields in the parameter list to edit the parameters as desired. For example, the *Restart App* parameter supports an editable boolean value - select either **true** or **false** from the *Value* drop-down (FIG. 335):
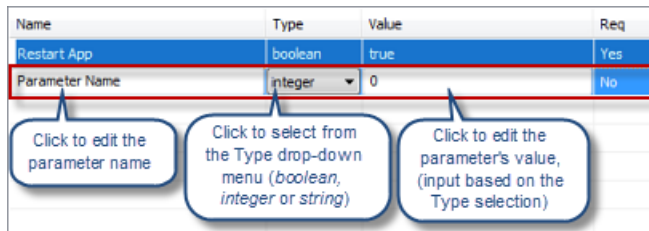
FIG. 335 Edit Parameter List dialog - user-defined parameter

- **Name**: Indicates the name of each parameter
- **Type**: Indicates the parameter type (*Boolean*, *Integer*, or *String*)
- **Value**: Enter or select the desired value based on the parameter type/function.
- **Req**: Indicated whether each parameter is required by the application. Note that user-defined parameters are not required.

4. Click **OK** to save changes and close this dialog.

Notes:

- User definitions must fully match the name, value type and value expected by the application - otherwise they will be ignored.
- User-defined parameter names must be unique within the parameter list.
- To delete a user-defined parameter, select the parameter and then click **Delete**.
- Drag-and-Drop as well as Copy/Paste functionality is supported for the entire parameter list, but only functions between Applications of the same type (i.e. Browser apps).

NOTE: *See the Working with Browser Application Windows section on page 230 for details on working the Browser application window type to set a Default URL, and switching between desktop and mobile content.*

Deleting Parameters

In the *Edit Parameter List* dialog, select any optional parameter from the parameter list, and click **Delete Parameter**.

- Parameters that are required (as indicted in the **Req** column of the parameter list) cannot be deleted.
- When stock parameters are deleted, they are re-added to the Available Parameters list.

Launch Actions

Launch Actions provide the ability to open an Application window on the panel, based on either a button press or release, a gesture, or when a specific Page is either opened or closed. In TPD5, Launch Actions are Events that can be assigned to Pages or Buttons. Events are defined via the Events tab of the Properties window.

NOTE: *In addition to Launch Actions, Page Flips and/or Actions (NetLinx commands and strings) can be assigned as Events to Pages and Buttons. See Working With Events for details.*

Creating a Launch Action Event on a Button

1. In the Design View, select a button to populate the Properties window with the selected button's properties.
   Note that Events (including Launch Actions) can also be applied to Pages.
2. In the *Events* tab, click on the Event that you want to use to trigger this Launch Action. For example, select the *Button Press* event (FIG. 336):
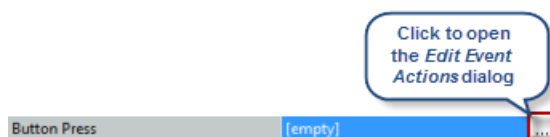
FIG. 336 Button Press (Event) Property

A button press event will cause the Launch Action to occur when the User presses the selected button.

3. Click the Browse (...) button to open the *Edit Event Actions* dialog (FIG. 337):
4. In the *Edit Event Actions* dialog, click on **Add Launch Action** to select from a listing of available Launch Action types (for example *show*):
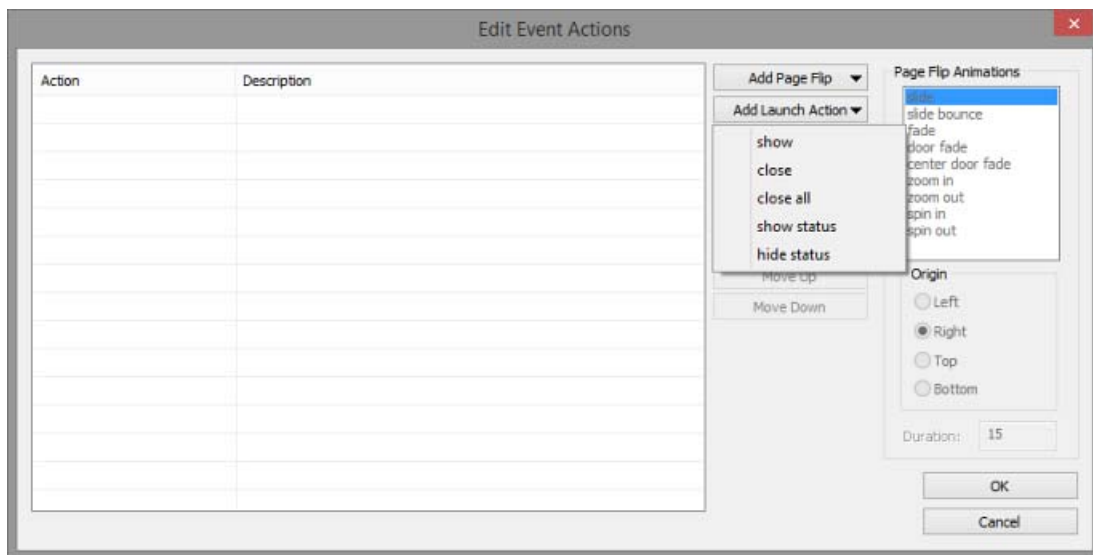
FIG. 337 Edit Event Actions dialog - Add Launch Action drop-down menu

> **NOTE:** *As it's name implies, a "show" launch action will show an Application window when the User presses the button on the panel. The other launch action type options are also self-explanatory and provide the ability to close Application windows, or show/hide Application status information.*

5. This adds **show** as an *Action*: select the target Application window from the drop-down list of all Application windows in the project, for example *Dropbox* (FIG. 338).
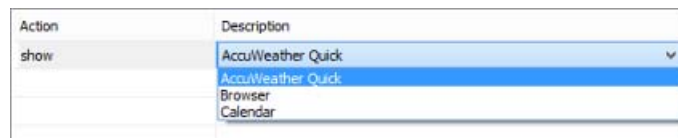
FIG. 338 Button Press (Event) Property

In this example, when the User presses the button configured here, the *AccuWeather Quick* Application window will open on the panel.

6. Click **OK** to save changes and close the *Edit Event Actions* dialog.

Cut, Copy and Paste - Application Windows

1. Cut or Copy an Application window to clipboard memory:
    *   To cut an Application window to the clipboard, select an Application window in the Workspace window (Pages tab) and select **Cut**. The program will prompt you to verify this action before the Application window is removed from the project.
    *   To copy an Application window to the clipboard, select an Application window in the Workspace window (Pages tab) and select **Copy**.
2. Select the target project for the Application window in the Workspace window (Pages tab). Application windows can be pasted into the current project, or into any other project that is open in the Workspace window.
3. Select **Paste** to paste a copy of the Application window into the selected project. If an Application window with the same name already exists in the target project, the Application window's name will be modified to indicate that it is a copy of another Application window. This prevents existing Application window from being overwritten by a Paste operation.

Working with Browser Application Windows

One of the Application window types that can be added to the TPD5 project is *Browser.* Use this application window to provide a fully functional web browser on the touch panel (FIG. 339):
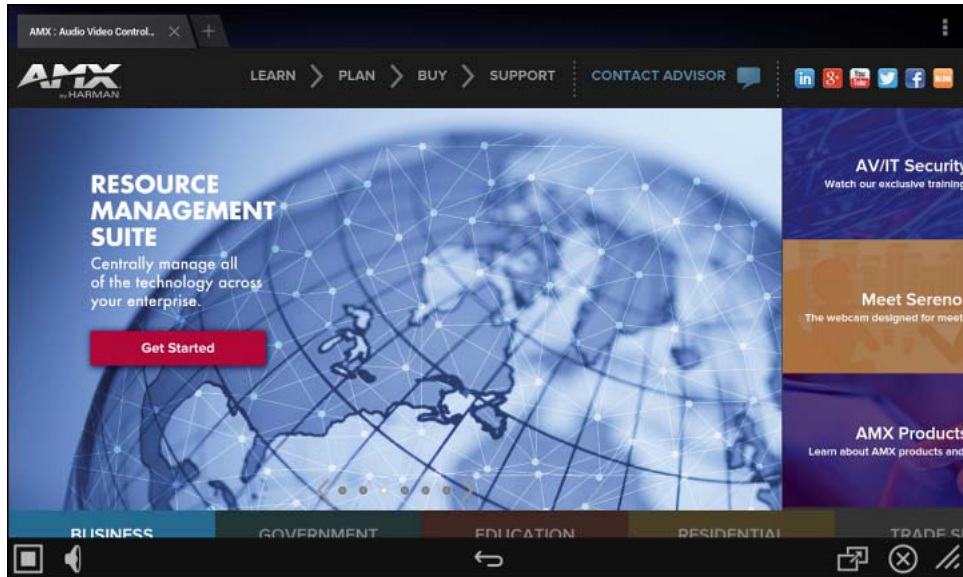
FIG. 339 Browser Application Window displayed on a G5 touch panel

Setting a Default URL for Browser Application Windows

The *App Parameters* (General) property for Application windows provides the ability to have a Browser window open to a specific URL when it is displayed on the panel:

1. Select **Panel > Add Application window** to open the *Add Application window* dialog.

2. Select **Browser**, set the *Name*, *Type*, *Position* and *Size* options as desired, and click **OK** to add the Application window to the project.

3. With the Browser Application window active in the Design View, select the **App Parameters** (General) property.

4. Click the browse (**...**) button to access the *Edit Parameter List* dialog (FIG. 340):

FIG. 340 App Parameters property (General tab)

5. Enter the desired default URI for this browser window in the "URI" text input field. Note that the URI must be an absolute URI (i.e. fully qualified, with scheme specified). If not, the browser window may not appear. As an example, to set the URI to open the AMX home page, enter the string "*http://www.amx.com*" (FIG. 341):

FIG. 341 App Parameters property (General tab)

> **NOTE:** *Consider naming each Browser application window that is set to open a specific URL with a descriptive name. For example, in this case the Application window could be named "Browser - AMX" to provide an easy way to differentiate it from other browser windows with specific default URIs.*

6. Click **OK** to save changes and close the *Edit Parameter List* dialog. The URI is now indicated in the App Parameters (General) property):

FIG. 342 App Parameters property (General tab)

7. Create or select a button and set a Launch Action to open this browser window:
   a. In the *Events* tab of the Properties window, select an event (i.e. Button Press) and click the browse (...) button to open the *Edit Event Actions* dialog.
   b. Click **Add Launch Action**, and select **show**.
   c. Open the drop-down menu of application windows that are in this project, in the *Description* column (FIG. 343):
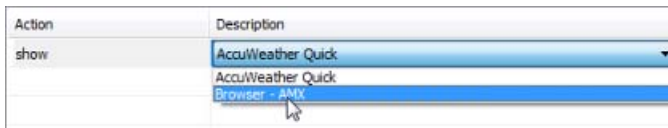
FIG. 343 Edit Event Actions dialog (Launch Action p- show menu example)

d. Select the Browser application window that was configured with a default URI (in this example the browser application window is named, "*Browser - AMX*").

e. Click **OK** to save changes and close the *Edit Event Actions* dialog.

f. At this point, when the project is loaded to the touch panel, a button press on the button configured in Step 7 will launch a Browser window, and it will open to the URI specified in Step 5 (in this example, "http:\\www.amx.com").

Switching Between Desktop and Mobile Content

Many webpages have two modes for viewing: desktop and mobile. Desktop is optimized for a desktop PC, while mobile is optimized for viewing on a mobile device. By default, all Browser application windows are displayed on the touch panel as "mobile" content. Use options in the touch panel's on-board *Settings* menu to request "desktop" content for a specific URL, if desired:

1. Press and hold the Settings/Sleep button on the G5 panel to access the *Power off/Settings* menu, and select **Settings** to access the Settings menu (FIG. 344):

FIG. 344 G5 Touch Panel - Power off/Settings menu

2. Scroll down to the CONNECTIONS section, and select **Browser**.

3. The panel will prompt you to enter a password in order to access the Browser page (default = **1988**).

4. In the Browser page, press **Add a url** to open the *Enter URL* window - use the on-screen keyboard to enter the URL of a website for which you want to request desktop (rather than mobile) content (FIG. 344):
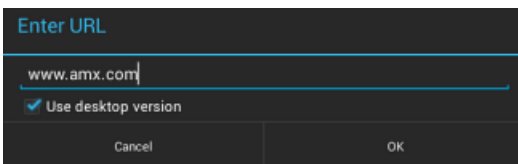
FIG. 345 G5 Touch Panel - Settings > CONNECTIONS > Browser > Enter URL

Note that by default, when URLs are added here, the Use desktop content option is pre-selected.

5. Press **OK** to save the URL and content setting.

At this point, the next time that the URL is accessed in a Browser application window, the touch panel will request desktop content.

NOTE: *Refer to the Modero X Series® G5 Programming Guide for details on using the Settings application on G5 touch panels.*

Properties

Overview

The Properties window is used to view/edit page, popup page and button properties, and to view/edit the states information associated with each element in your project. It is typically located on the right side of the screen (although you may move it anywhere you like).

Select **View > Properties** (or click the toolbar button) to display the Properties window (FIG. 346):
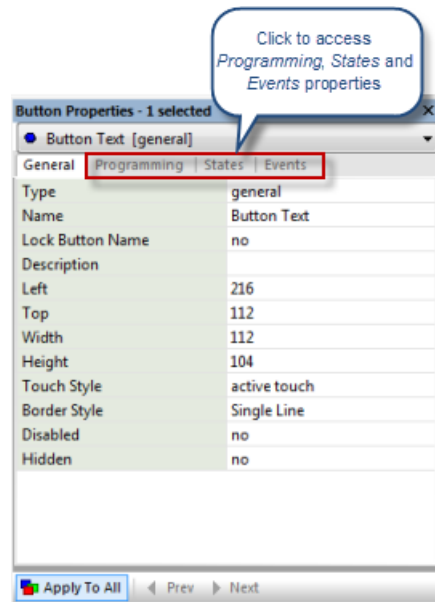
FIG. 346 Properties window (General tab shown)

The Properties window contains four tabs: *General*, *Programming*, *States* and *Events*. The properties presented in these tabs varies depending on the TPD5 element currently selected in either the Workspace window or the Design View. Note that the Events tab is empty when Standard Popup pages, Sub-page popups or Application windows are selected, since Events d not apply to these TPD5 element types.

The title bar of the Properties window indicates which element of the workspace is currently selected. If one or more buttons are selected in the Workspace, the title bar of the Properties window indicates the number of buttons selected.

Button names and types are displayed in the text box above the tabs (in the example below, the selected button is named "Home", and the button type is general. Click the down arrow next to this field to view a list of all buttons on the active page. Selecting a button from this list is the same as selecting it in a Design View window: the edit focus shifts to the selected button, and the Properties window reflects the properties of the newly selected button.

Apply To All

The **Apply To All** toggle button is located at the bottom of the Properties window. Use this option to edit properties on multiple buttons simultaneously.

- If the *Apply To All* button is not depressed, and you select more than one button to act on, the Prev or Next buttons will activate on the Properties window. Also note that while you have multiple buttons selected, only one of them has the edit focus at any given time. Use the Prev and Next buttons to cycle through the selected buttons to view each button's properties in the Properties window. The button represented in the Properties window is the one with edit focus. All modifications are always on the button with the edit focus.
- If the *Apply To All* button is depressed and you select more than one button to act on, the Prev or Next buttons are unavailable and every selected button has the edit focus. You may also notice that one or more (if not all) of the property values in the grid are blank. The only values that will display in the grid while the Apply To All button is depressed are those values that are common among all selected items. Typing in or changing a value in any property box will immediately affect all selected buttons, provided that the change can be applied to them all. In the event that a value is appropriate for one (or more) buttons but inappropriate for others, you will see a message that states that the value was only applied to those buttons for which it was valid.

NOTE: *This Apply To All button works on multiple button selections, but not on multiple states for a single button. To make state-oriented changes across multiple states, select the states that you want to edit and they will appear listed in the States tab of the Properties window.*

All States

To make changes that affect all states on a button at once, use the *All States* option in the States tab (located directly above the other listed states for the selected button).

Prev and Next

The *Prev* and *Next* buttons are activated only when more than one button is selected on a page, and the Apply To All button is not in its active state. These buttons allow you to quickly edit the selected buttons individually. Hold down the Shift key and click to select multiple buttons.

NOTE: *When you have multiple buttons selected, only one of them has the edit focus at any given time. Use the Prev and Next buttons to cycle through the selected buttons to view each button's properties in the Properties window. The button represented in the Properties window is the one with edit focus. All modifications are always done on the button with the edit focus.*

- Properties can be dragged and dropped from the Properties window onto the Design View. The selected property or state is automatically applied to all states of the drop target. General and State properties can also be copied and pasted to a Design View using standard copy/paste menu and keyboard mechanisms.
- Another feature of the Properties window (all tabs) is that you can click on any value in the right column, and drag it to another field. When you release the mouse button, the value is copied to the new location. With an item selected, the cursor will change to indicate any fields that cannot accept the selected value, and if the selected value is out of the acceptable range for a target field, TPD5 alerts you with an error dialog, and the original value is left unchanged.

Quick Input

Select **Quick Input** from the Edit menu or Design View context menu to access the *Quick Input* sub-menu.

The Quick Input setting determines how typing directly into a Design View or into the State Manager will be handled:

- **Current Property** - This setting redirects keyboard input to the currently selected property on the currently visible tab of the Properties window (assuming one is selected).

  For example, if you select the Name property (in the General tab of the Properties window), any time you select a button in the Design View, you can just type and press the Enter key to enter a new button name for the selected button. The result of the keystroke will depend on the property selected.

- **Text** - This setting redirects keyboard input to the button Text property for all selected button states (in the States tab of the Properties window).

  For example, if you select several states on a Multi-State button (in the State Manager window), you can type and press the Enter key to enter new button text for the selected states. If no states are selected, the text will be applied to all states of the button selected in the Design View.

- **Disabled** - Disables the Quick Input option.

Searching For Properties

Use the *Find* dialog to search for any button property value either within the currently open Page, or across the entire project. You can specify to search for any General or State button property. For example, you can perform a search based on button type, name, border style, and state count (among many others), or any combination of search criteria.

To search for button properties:

1. Select **Edit > Find** (or click the toolbar button) to open the *Find* dialog (FIG. 347):

FIG. 347 Find dialog

2. In the *Search Criteria* table, select the button properties to use as the search criteria. You can include any General, Programming, State or Action button property as search criteria. Properties are separated into four sections (FIG. 348):

FIG. 348 Find dialog - Search Criteria headings

- Scroll down to view all of the available button properties that can be used as search criteria.

- Click the minus symbol (-) next to the General and States headings to collapse the sections. Click the plus symbol (+) to expand the views.
- To select all button properties under any heading, click in the checkbox next to the heading: (FIG. 349):

FIG. 349 Find dialog - Search Criteria headings (All General and States properties selected)

3. In the *Search Scope* area, select either **Entire Panel** or **Current Page**.
   - If the scope of the search is set to *Entire Panel*, the *Select All* button is disabled, in which case you can use the Find Next button to cycle through the pages that contain buttons whose values match the search criteria.
   - If the scope of the search is set to *Current Page*, use the **Find Next** button to search the current page only, based on the specified criteria and scope.
4. When the first instance of the criteria is found, the *Find In Page* dialog is compressed to only show the buttons that satisfy the search criteria, and the first button found that satisfies the search criteria is selected in the Design View.
   - Select **Find Next** to continue the search.
   - Select **Select All** to close the *Find* dialog and select every button that meets the criteria.
   - The program will inform you if no buttons are found that match the search criteria.

Finding and Replacing Properties

Use the *Find and Replace* dialog to find (and optionally replace) any property value with another value of your choosing. You can specify the scope of the search to either the currently open Page only, or across the entire project.

To search and replace button properties:

1. Select **Edit > Find & Replace** (or click the toolbar button) to open the *Find & Replace* dialog (FIG. 350):
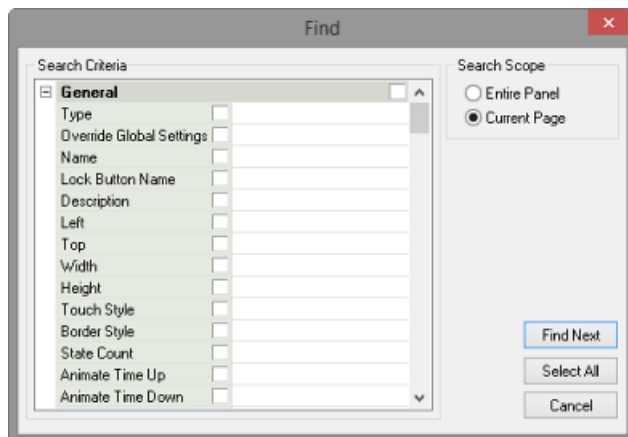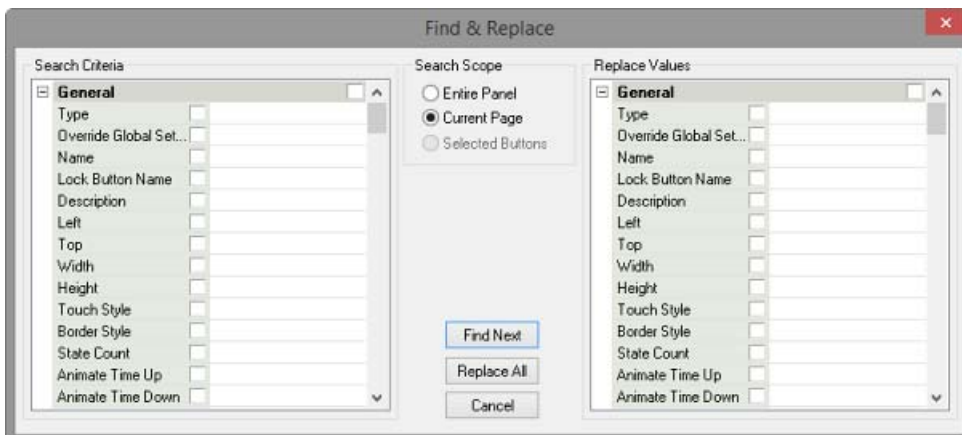

FIG. 350 Find & Replace dialog

2. In the *Search Criteria* table, select the button properties to use as the search criteria. You can include any property as search criteria.
3. In the *Search Scope* area, select either **Entire Panel** or **Current Page**.
4. In the *Replace Values* table, select the button properties to use as the replace values. The Replace Values do not necessarily have to match the Search Criteria (although they may). It is also possible to replace multiple values or establish multiple search criteria.

   For example, to change the Border Style on one or more buttons, select *Border Style* in the Search Criteria table, then select the particular border style to search for from the drop-down list. Then, select Border Style in the Replace Values table, and select the desired replacement border style from the drop-down list. Repeat this process for as many other button properties as needed.
5. When the first instance of the criteria is found, the *Find In Page* dialog is displayed, listing the buttons that satisfy the search criteria, and the first button found that satisfies the search criteria is selected in the Design View.
   - Select **Find Next** to continue the search.
   - Select **Replace All** to close the *Find* dialog select every button that meets the criteria. The program informs you of the number of buttons affected by this change.
   - All replace actions support full Undo / Redo capabilities.
   - The program will inform you if no buttons are found that match the search criteria.

NOTE: *When you select a General as well as a State-oriented search criteria, only buttons that match the General criteria, and within that set, the states that match the State criteria will be candidates for the replace operation. For example, if you set the search criteria to include both the Hard Drop Shadow border style (a General property) and yellow as the Fill Color (a State property), only those buttons with a Border Style of Hard Drop Shadow are candidates, and within that set of candidates, only those states whose Fill Color is set to yellow satisfy the search criteria.*

Cut, Copy and Paste - Properties

TPD5 allows you to cut, copy and paste Properties across TPD5 elements, either within the project, or across Projects.

The Cut, Copy and Paste functions always work on the element which has the current edit focus.

Note that the *last thing selected* (not necessarily a button displayed in the Design View windows) has the edit focus.

For example, if you have selected an item in the Properties window (for example, "Channel Port"), and you perform a "Copy", then you will have copied only the Channel Port entry in the Properties window, and not the associated button (even though it is selected in the Design View window).

You can then paste the Channel Port number anywhere else in the Properties window (where the copied data is considered to be a valid entry).

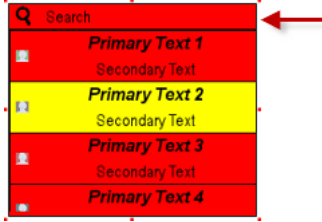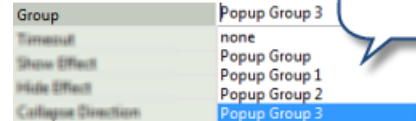If for example you had copied the value "410" from the Left (position) field in the Properties window, you are not allowed to paste it to the Address Port field, since it is not a valid Address Port number - these errors are indicated by an error dialog.
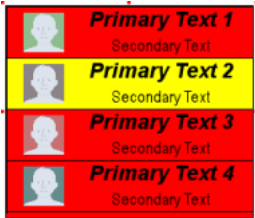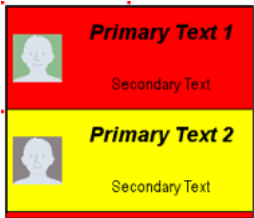
General Properties

The Properties presented in the *General* tab of the Properties window will depend on the current selection in the active Design View window (Page, Popup Page, Sub-Page, Application window or Button). Some properties also only apply to specific panel types.

General Properties	
Allow Dynamic Reordering	This option determines whether the Sub-Pages contained within a Scrolling Region can be re-arranged by an end-user (Yes/No, default = No). Note that with this property enabled, the order in which the user last left the Sub-Pages will be saved to the panel. The new order is maintained across reboots and power cycles. Downloading a new project will reset Sub-Page ordering to the settings in the new project. Drag and Drop Sub Pages When this property is set to Yes, the end-user can press and hold any Sub-Page to make it a floating element, then drag and drop it into another position within the Scrolling Region. Note that Sub Pages cannot be drag and dropped in or out of their Scrolling Region. *Note: This property applies only to Sub-Page View buttons.*
Alphabet Scrollbar	Use this property to enable/disable the alphabet scrollbar feature for Listview buttons. By default, this property is set to no (disabled). To enable this feature, select yes from the drop-down menu. If enabled, an alphabetical index will be rendered on the Listview button. This allows the end user to quickly scroll through a large numbers of list items, with an indication of where the user is in the current view, relative to the alphabetic order of the list items. Note that the scrollbar is not visible in TPDesign5, it is only rendered on the panel (when enabled).
Anchor Position	This property determines how the Sub-Page Sets associated with the selected Sub-Page View button are initially displayed and justified within the Sub-Page View button. *Note: This property applies only to Sub-Page View buttons.* The options in this menu depend on the Orientation setting of the selected Sub-Page View button: For Horizontal orientation: • Left: First sub-page is displayed aligned to the left side of the button. • Middle: The middle sub-page is displayed positioned in the center of the button (default setting). • Right: Last sub-page is displayed aligned to the right side of the button. For Vertical orientation: • Top: First sub-page is displayed aligned to the top of the button. • Center: The middle sub-page is displayed positioned in the center of the button (default setting). • Bottom: Last sub-page is displayed aligned to the bottom of the button.
Animation Time (tenths/sec)	Use this property to specify the length of time (in tenths of a second) that the page flip animation will use to complete the page flip (default = 0).
Animate Time Down	Enter the timer interval used between states as the button animates from the On to the Off state. This value is in 1/100th second increments (default = 2). *Note: This property is available for Multi-State General buttons only.*
Animate Time Up	Enter the time interval used between states as the button animates from the Off to the On state. This value is in 1/100th second increments (default = 2). *Note: This property is available for Multi-State General buttons only.*

General Properties (Cont.)

App Parameters	Some Applications require or support configuration parameters at startup that must be configured. For example, the browser application may take a home page or startup page parameter so that when the browser is launched it opens to the configured page. With an Application window selected, the App Parameters (General) Property is available. *Note: This property is available for Application windows only.* Use this property to enter/edit parameters for specific application types. Click on the field and then click on the Browse button (...) to open the *Edit Parameter List* dialog. See the *Editing Application Parameters* section on page 62 for details.
Auto-Repeat	Select whether to apply auto-repeat to the button. Auto-repeat causes the button to constantly cycle through its states (Yes/No, default = No). *Note: This property is available for Multi-State General buttons only.*
Border Style	To change the Border Style for the selected button, click Border Style, and select the desired style from the drop-down list. There are several types of border styles to choose from, and they all can all be assigned to both popup pages and buttons. *Note: If you don't want a border on the button, select "none" as the border style.*
Collapse Direction	This setting (None, Left, Right, Up, Down) controls the direction a popup page will collapse when it closes. *Note: This property is available for Popup Pages only.*
Collapse Offset	This setting controls the number of seconds before a popup page collapses when it closes. *Note: This property is available for Popup Pages with a set Collapse Direction only.*
Description	Use this text field to enter a general or functional description for this button. Click the browse (...) button to open the *Enter Text* dialog, where you can type the description.
Disable Touch Scrolling	The Disable Touch Scrolling (General property) for Sub-Page View Buttons allows or prevents touch scrolling on a Sub-Page View button. The default is set to "no".
Disabled	Indicates how the selected button will be rendered. If the button is set as Disabled (select Yes from the drop-down), the button will be rendered by the panel in a subdued state (default = No).
Display Type	Click to select the display type to be invoked by this Text Input button (single line or multiple lines). The default is single line. *Note: This button property is available for Text Input buttons only.*
Drag/Drop Type	Sets the selected General or Multi-State General button as either "draggable" or as a "drop target" button. By default, this property is set to "none". See the *Drag and Drop* section on page 167 for details.
Drop Group	Click to associate the selected Draggable button with a specific Drop Group. See the *Drag and Drop* section on page 167 for details.
Dynamic Data Source	Use this property to specify the data source to use as the source for content that will be displayed on the selected Listview button: Click the browse button on the Dynamic Data Source property to open the *Select Resource* dialog. Use the *Select Resource* dialog to specify which components (Primary Text, Secondary Text and/or Image) will be displayed. See the *Assigning a Data Source to a Listview button* section on page 112 for details.
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 (see below). The remaining area of the Listview button will be available for the display of list items:

General Properties (Cont.)

Filter Height	Use this property to specify the height of the filter entry box for a Listview button (in pixels). The minimum allowed value (and the default setting) is 24 pixels.
Group	Use this field to add the selected Popup Page to a specific Popup Page Group. *Note: This property is available for Popup Pages only.* To create a new Popup Page Group, type a group name directly in the Group field. This will create a corresponding folder under the Popup Pages folder in the Workspace window (Pages tab): To place a popup page in a particular popup group, simply drag the desired popup pages into the desired group folder.
Height	Enter the Height size value (in pixels) to specify the vertical dimensions of the selected button. You can edit these fields to apply specific dimension info for the button. Note that if you select the button and manually resize it on the page, these values constantly update to indicate the current dimensions.
Hidden	Indicates whether or not the selected button is displayed on the panel. If the button is set as Hidden (select Yes from the drop-down), the button will disappear when transferred to the panel (default = No).
Hide Effect	This property allows you to apply a transition effect to the popup page, to be invoked when the popup is closed (hidden). *Note: This property is available for Popup Pages only.*
Hide Effect Time	This property allows you to specify the total amount of time it will take to execute the selected Hide effect, measured in 1/10th-second increments. Use this option to synchronize your popup page transition effects. *Note: This property is available for Popup Pages only.*

Hide Effect X/Y Pos	Measured in pixels, the Hide Effect X/Y Pos (position) fields allow you to specify the starting point on the page for the selected Slide Hide Effect.
	Depending on the Slide Hide Effect selected, the Hide Effect X Pos. or Hide Effect Y Pos. option may become available:
	Note: This property is available for Popup Pages only.
	• If you have selected a left or right slide effect (including slide/fade effects), you can set the X position for the start of the hide effect (range = 0 - 9999).
	  The default is 0 (the left edge of the page).
	  In some cases, depending on the page design and graphics, you may decide to start the slide at some other point than the absolute left edge of the page.
	• If you have selected a top or bottom slide effect (including slide/fade effects), you can set the Y position for the start of the hide effect (range = 0 - 9999).
	  The default setting is 0 (top edge of the page).

General Properties (Cont.)

Input Mask	This field allows you to apply a mask to user input on the panel. *Note: This button property is available for Text Input buttons only.* An input mask allows you to force the user to enter the correct type of characters (numbers vs. characters), suggest a proper format with fixed characters, to change or force character case, to create multiple logical fields that act as a single field, to specify a range of characters / numbers for each field, to set the input as required or optional, and/or fill the field from the right or the left. **Input Mask Characters** The following table lists the available input mask characters, and which characters each input mask allows in any given position (for use with the Input Mask property for Text Input buttons): **0** - Digit (0 to 9, plus [+] and minus [-] signs not allowed). **9** - Digit or space (plus and minus signs not allowed). **#** - Digit or space (plus and minus signs allowed). **L** - Letter (A to Z). **?** - Letter (A to Z, or space). **A** - Letter or digit. **a** - Letter or digit (or space). **&** - Any character or a space. **C** - Any character or a space. **H** - Hex digits (0-9, a-f, or A-F) **Input Mask Ranges** Input Mask Ranges provide a method to specify the minimum and maximum numeric values for a given field. Only one range is allowed per field and the use of a range implies numeric entry only. The following table lists the available input mask ranges (for use with the Input Mask state property for Text Input buttons): **[** - Start Range **]** - End Range **|** - Range Separator **Input Mask Operators** Input Mask Operators change the behavior of the field in various ways. The available Input Marks Operators for use with the Input Mask state property for Text Input buttons are: **<** - Causes all characters to be converted to lowercase. **>** - Causes all characters to be converted to uppercase. To define a literal, enter any character other than those shown above, including spaces and symbols. A back-slash ('\') causes the character that follows to be displayed as the literal character. For example, **\A** is displayed as just **A**.
Input Type	This field allows you select the type of text input that may be displayed on a Text Input button. You may select between alpha-numeric (letters and numbers), numeric (numerals only), telephone (numbers and characters such as "-" and "()"), and date/time (month/day/hour/minute). *Note: This button property is available for Text Input buttons only.*
Item Height	Use this property to specify the item height for the selected Listview button (in pixels). Note that all list items are drawn to the height specified here, regardless of the overall size of the Listview button itself. That is, adjusting the size of the button does not affect the size of the list items, only the number of list items that can be displayed within the button at a time. The end user will typically need to scroll vertically through the list to see all list items. Example: Item Height set to 48 Example: Item Height set to 96 The minimum allowed value (and the default setting) is 48 pixels.
Left	Left/Top - Position values. The Left and Top rows indicate the position of the selected button, in pixels, relative to the upper-left corner of the Design View window. • You can edit these fields to apply specific positioning info for the button. • Note that if you select the button and manually move it around on the page, these values constantly update to indicate the button's current position.

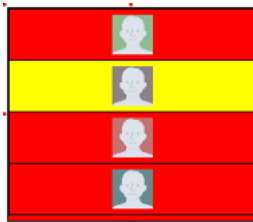| General Properties (Cont.)	
Listview Columns	Use this property to specify the number of columns to display in the selected Listview button. By default, this value is set to 1. This property provides the ability to present a "grid view" on the Listview button, if desired. The number of columns allowed depends on the size of the Listview button. If the number of columns exceeds the display area of the selected Listview button, the program displays an error message indicating that either the number of columns must be reduced, or the width of the button must be increased to accommodate the desired number of columns. The following example shows a Listview button with one column, two columns, and three columns. Note that the example for three columns displays the Image component only, as set via the *Listview Components* (General) property. Listview Columns set to 1 Listview Columns set to 3 (note that this example uses only the Image component) Listview Columns set to 2 (button has been resized to accommodate 2 columns)
Listview Components	Use this property to specify the combination of the three supported Components (Primary Text, Secondary Text and/or Image) that will be displayed for list item content on the selected Listview button. 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. • 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). The *Listview Components* (General) property will indicate **single-line text**: • If **Primary Text** and **Secondary Text** are selected in the *Edit Listview Components* dialog, each list item is represented with two lines of 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 *Listview Components* property will indicate **two-line text**:

General Properties (Cont.)

Listview Components (Cont.)	• If **Primary Text, Secondary Text** and **Image** are selected in the *Edit Listview Components* dialog, each list item is represented with two lines of text and an image on the left side.
	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 *Listview Components* 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 *Listview Components* 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.

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 Size* (States) properties.

The *Listview Components* Property will indicate **single-line text w/ Image**:
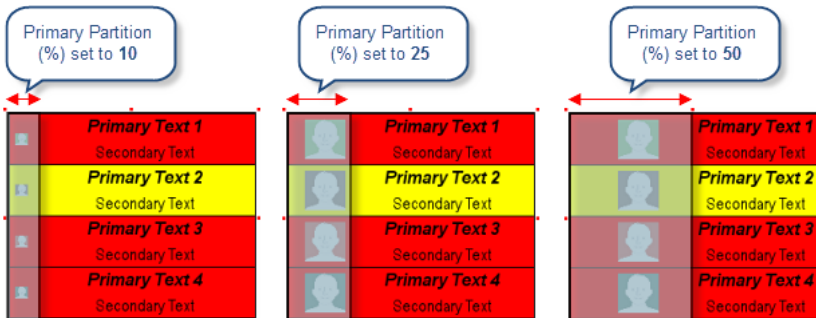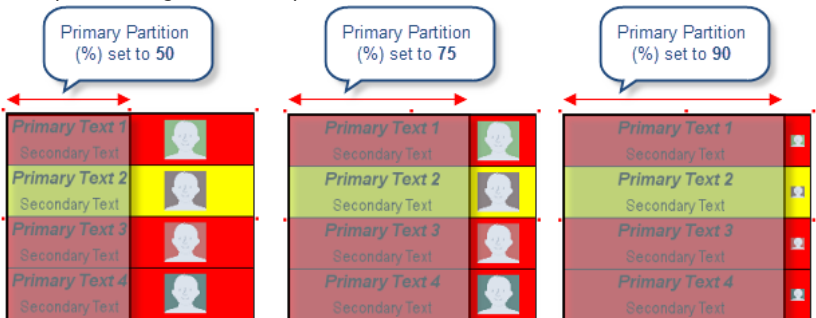
Use this property to specify the layout of the components (Primary Text, Secondary Text and Image) specified to display on the list items in the selected Listview button. Listview components are selected via the *List View Components* (General) property.
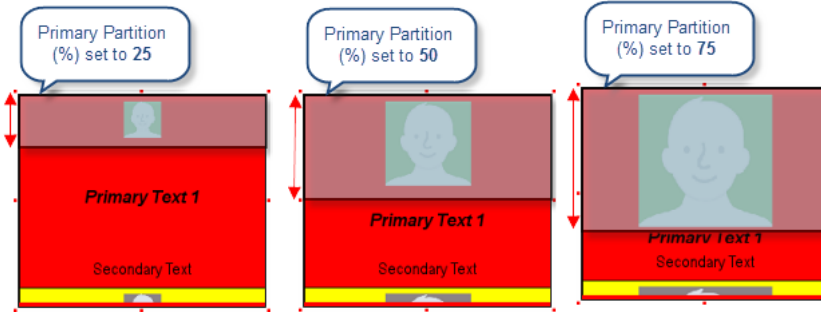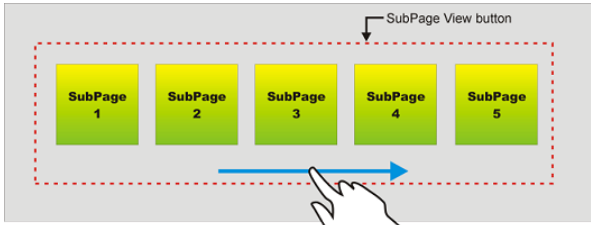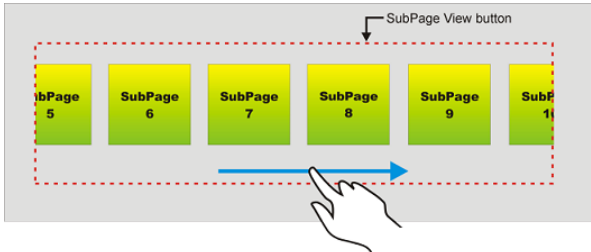
Click in the Listview Item Layout field to select from a drop-down of layout options for list items:

• *horizontal - image left* (default setting): The image (if displayed) will appear to the left of the Primary (and Secondary) Text:

General Properties (Cont.)	
Listview Components (Cont.)	• *horizontal - image right*: The image will appear to the right of the text: • *vertical - image top*: The image will appear centered above the text: *Note: Once the Listview Item Layout has been selected, the placement of the layout components can be adjusted via the Primary Partition (%) and Secondary Partition (%) properties. In these examples, adjustments have been made to both the partition (%) properties and in the case of the vertical layout example, the Item Height (General) property was adjusted to increase the height to allow the display of all three components.*
Listview Item Layout	Use this property to specify the layout of the components (Primary Text, Secondary Text and Image) specified to display on the list items in the selected Listview button. Listview components are selected via the *List View Components* (General) property. Click in the Listview Item Layout field to select from a drop-down of layout options for list items: • *horizontal - image left* (default setting): The image (if displayed) will appear to the left of the Primary (and Secondary) Text: • *horizontal - image right*: The image will appear to the right of the text: • *vertical - image top*: The image will appear centered above the text: *Note: Once the Listview Item Layout has been selected, the placement of the layout components can be adjusted via the Primary Partition (%) and Secondary Partition (%) properties. In these examples, adjustments have been made to both the partition (%) properties and in the case of the vertical layout example, the Item Height (General) property was adjusted to increase the height to allow the display of all three components.*

General Properties (Cont.)

Lock Button Name	This option controls how the name of the selected button is managed by the program (Yes/No, default = *No*). When new buttons are created, by default the buttons are automatically given a sequential two-part name composed of the button number (relative to the number of buttons already created in the project) and button type, separated by a colon (i.e. "Button 1 : general", "Button 2 : multi-state general", etc). See *Generated Button Names* on page 87 for details.
Max Text Length	Use this field to specify the maximum number of characters allowed to be entered via this button. The range is 0-2000 (default = 0). *Note: This button property is available for Text Input buttons only.*
Name	To give the selected element a specific name other than the default Button/Popup 1, Button/Popup 2, Button/Popup 3 etc., click Name in the Properties window (General tab) to activate the text field, where you can type the new name. *Note: You must use a unique name for each button, page and popup page, and you cannot apply the Job name (set in the New Project Wizard) to a page.*
Orientation	Select the orientation for the selected Sub-Page View button (Horizontal/Vertical, default = Horizontal). *Note: This property applies only to Sub-Page View buttons.*
Password Character	Type a single character to be used as the password to access this Text Input button on the panel. *Note: This button property is available for Text Input buttons only.*
Popup Type	This setting (Standard/Sub-Page) sets the Type for the selected Popup Page. • This property is available for Popup Pages and Sub-Pages only. • Sub-Page Popups are used (with Sub-Page Sets and Sub-Page View buttons) to create Scrolling Regions (supported only by Modero X-Series panels).
Primary Partition (%)	Use this property to specify the relative size of the Primary Partition on the list items displayed on the selected Listview button. The allowed range is 5-95%. The portion of the list item that is controlled by the Primary Partition (%) property depends on the Listview Components selected (see page 104), as well as the Listview Item Layout selected (see page 107). Note that for all layout options, if Image is not an included component, then Primary Partition (%) is ignored. The following examples show a Listview button with all Listview Components (Primary Text, Secondary Text and Image) selected. With "horizontal - image left" layout selected: Primary Partition (%) represents the area used by the Image component: With "horizontal - image right" layout selected: Primary Partition (%) represents the area used by the Primary Text component. In this case, Primary Partition (%) sets the position of the separation between the Primary Text and the Image as a percentage of cell width (allowed range = 5%-95%):

General Properties (Cont.)

Primary Partition (%) (Cont.)	With "vertical - image top" selected: Primary Partition (%) represents the area used by the Image. In this case, Primary Partition (%) sets the position of the separation between the Image and the Primary Text as a percentage of cell width (allowed range = 5%-95%):
Reset Pos. On Show	If this option is turned on, the popup page will always appear at the position established during popup page design each time it is displayed. This might be desirable if the popup page contains a button which provides the end user with the ability to move the popup page at will. *Note: This property is available for Popup Pages only.*
Reset View On Show	This property determines whether to reset the positioning of the Sub-Pages displayed within a Scrolling Region the next time the Scrolling Region is displayed (Yes/No, default = No). *Note: This property applies only to Sub-Page View buttons.* For example, imagine a set of 10 Sub-Pages within a Scrolling Region that is large enough to display five Sub-Pages simultaneously. By design, Sub-Pages 1-5 will appear on-screen: • With Reset On Show enabled (the default setting), the Sub-Pages in the Scrolling Popup will be displayed in their last positions the next time the Scrolling Region is displayed. • With Reset View On Show disabled, the Sub-Pages in the Scrolling Region will be displayed in their default positions the next time the Scrolling Region is displayed. In other words, the Sub-Pages will always appear in the position established during design each time it is displayed. This might be desirable if the popup page contains a button which provides the end user with the ability to move the popup page at will. When a user scrolls the Sub-Pages to access Sub-Pages 9 and 10, the overall view within the Scrolling Region will have changed:
ScrollBar	Select whether to display a ScrollBar on this Sub-Page View button (default = no). Modero-X panel firmware supports the ScrollBar for Sub-Page View buttons. This ScrollBar provides a position indicator within the sub-page set, and does not provide any dragging or scrolling functionality. Dragging and scrolling is provided by sub-page view button itself (see *Scrolling Regions* on page 53 for details). The ScrollBar is useful especially for scrolling regions that contain a large number of sub-pages - it provides a simple visual indication of where the current view is, relative to the entire list. *Note: Sub-Page View ScrollBar requires panel firmware v2.103.x or greater.*

General Properties (Cont.)	
ScrollBar Offset	The ScrollBar Offset (General property) for Sub-Page View Buttons sets the offset position of the ScrollBar relative to the area of the button. Modero-X panel firmware supports a ScrollBar for Sub-Page View buttons (see Adding a ScrollBar to a Sub-Page View button). The ScrollBar is a position indicator within the Sub-Page Set, and does not provide dragging or scrolling functionality. ScrollBar Offset is a non-negative integer value (Default = 0). For horizontally-oriented sub-page views, the default value (zero) causes the ScrollBar to appear along the right side of the button drawing area: For vertically-oriented sub-page views, the default value (zero) causes the ScrollBar to appear along the bottom of the button drawing area: Increasing the value will incrementally increase the distance that the ScrollBar is positioned relative to the default position. The button drawing area is defined by the dimensions of the button, together with any assigned border. *Note: If the offset value exceeds the value of the bounding dimension, the ScrollBar will be positioned along the top or left side of the button drawing area, for horizontally- or vertically-oriented sub-page views, respectively.*
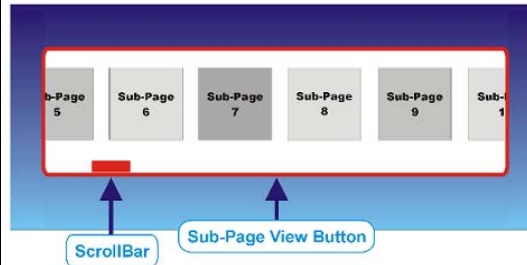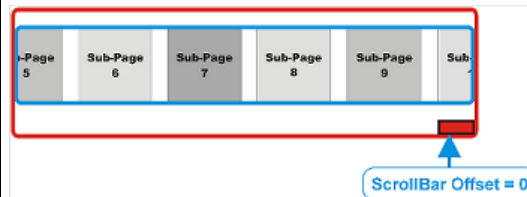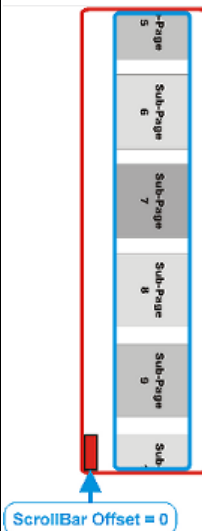
General Properties (Cont.)

Secondary Partition (%)	Use this property to specify the relative size of the Secondary Partition on the list items displayed on the selected Listview button. The allowed range is 5-95%.
	The portion of the list item that is controlled by the Secondary Partition (%) property will depend on the Listview Components selected as well as the Listview Item Layout selected.
	The following examples show a Listview button with all Listview Components (Primary Text, Secondary Text and Image) selected.
	With "**horizontal - image left**" layout selected:
	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%):
	(examples: Secondary Partition (%) set to 10, set to 50, set to 75)
	With "**horizontal - image right**" layout selected:
	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%):
	(examples: Secondary Partition (%) set to 10, set to 50, set to 75)
	With "**vertical - image top**" selected:
	Secondary Partition (%) represents the area 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%):
	(examples: Secondary Partition (%) set to 50, set to 75, set to 90)
Show Effect	This property allows you to apply a transition effect to the popup page, to be invoked when the popup is opened (shown).
	Note: This property is available for Popup Pages only.
Show Effect Time	This property allows you to specify the total amount of time it will take to execute the Fade Hide effect, measured in 1/10th-second increments.
	Use this option to synchronize your popup page transition effects.
	Note: This property is available for Popup Pages only.
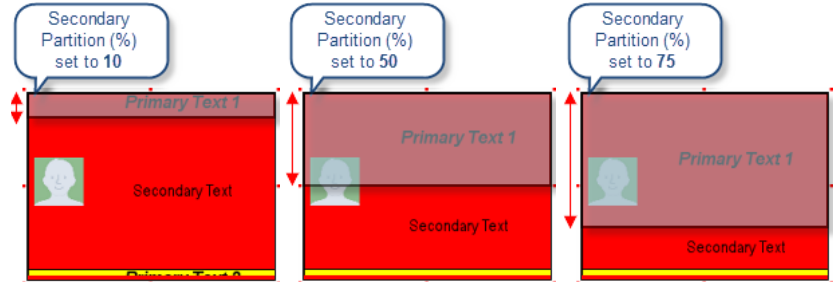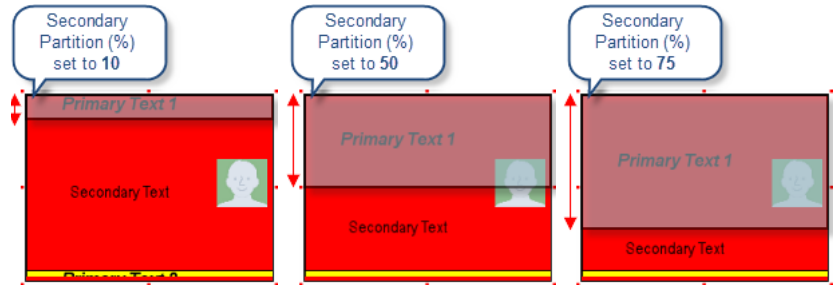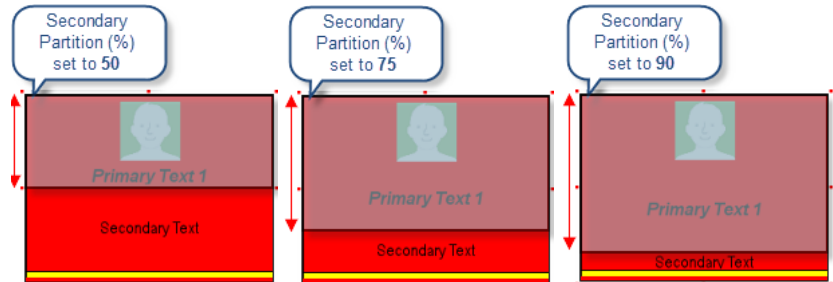
General Properties (Cont.)

Show Effect X/Y Pos	Measured in pixels, the Show Effect X/Y Pos (position) fields allow you to specify the starting point on the page for the selected Show Effect.
	Depending on the Slide Show Effect selected, either the Show Effect X Pos. or Show Effect Y Pos. option may become available:
	Note: This property is available for Popup Pages only.
	• If you have selected a left or right slide effect, you can set the X position for the start of the slide transition effect (range = 0 - 9999).
	  The default setting is 0 (the left edge of the page).
	  In some cases, depending on the page design and graphics, you may decide to start the slide at some other point than the absolute left edge of the page.
	• If you have selected a top or bottom slide effect, you can set the Y position for the start of the slide transition effect (range = 0 - 9999).
	  The default setting is 0 (top edge of the page).

Show Open	This setting determines whether a Collapsible Popup Page will initially be shown as open. *Note: This property is available for Popup Pages only, and only if Collapse Direction (see page 236) has been set to anything other than None.*
Show Sub-Pages	Select Yes or No from the drop-down menu (default = Yes). • Yes - Sub-Pages will be initially displayed. • No - The Sub-Page View button will initially be displayed without sub-pages. *Note: This property applies only to Sub-Page View buttons.*
Slider Color	Select a color to apply to the Bargraph slider. Click the browse button (...) to open the Colors dialog. *Note: This property is available for Bargraph buttons only.*
Slider Name	Select the desired visual style for the Bargraph slider from the list of Slider types. There are several slider types available for use with Bargraph buttons: *Note: This property is available for Bargraph buttons only. Slider types are not available for Multi-State Bargraph buttons.*
Spacing (%)	Enter an Integer (percentage) value to specify the amount of spacing between Sub-Pages when they are displayed within a Sub-Page View button (0-100, default = 0). *Note: This property applies only to Sub-Page View buttons.* This value represents the percentage of the Sub-Page Popups width (for Horizontal Sub-Page View buttons) or height (for Vertical Sub-Page View buttons) defined by the first Sub-Page in the Sub-Page Set associated with this Sub-Page View button. For example, 0 (the default setting) will result in no spacing between the Sub-Pages displayed within a Scrolling Region. A value of 100 will insert a space that is equal to either the horizontal or vertical dimension (depending on whether the Scrolling Region is set to Horizontal or Vertical orientation) of the first Sub-Page in the Sub-Page Set.
State Count	This field indicates the number of states currently associated with the selected button. *Note: This property is available for Multi-State (General and Bargraph) buttons only.* To change the state count for the selected button, click inside the text field and enter the desired number. • If the state count is increased, new states are added to the end of the set as a duplicate of the last existing state. If the count is decreased, states are removed from the end of the set. • This feature allows the state count to be changed via Edit > Find & Replace and with the Paint Properties tool.
Sub-Page Set	Click the down arrow to select from a listing of all Sub-Page Sets that have been defined via the *Edit Sub-Page Sets* dialog (default = None).
Timeout	This property allows you to specify the Popup Page Timeout, in 1/10th second increments. Popup Page Timeout specifies how long a popup page will remain open and active without a button press (default = 0). *Note: This property is available for Popup Pages only.*
Top	Left/Top - Position values. The Left and Top rows indicate the position of the selected button, in pixels, relative to the upper-left corner of the Design View window. • You can edit these fields to apply specific positioning info for the button. • Note that if you select the button and manually move it around on the page, these values constantly update to indicate the button's current position.
Touch Map	Click the browse button (...) to select an image to use as the Touch Map image, via the Select Resource dialog. *Note: This button property is available for Multi-State Bargraph buttons only, and only if the Value Direction is set to Touch Map).*

General Properties (Cont.)

Touch Style	This selection drop-down allows you to set a "touch style" for the selected button(s). Touch style describes the way buttons behave when pressed, in terms of the shape and border style used. For example, by using transparencies you could create a button that appears to be round (although the actual shape of the button is rectangular), in which case you may not want the button to respond if the user presses outside of the circular border. • Active touch: This touch style limits the active touch area to the visible area of the button. Areas of the button that are totally transparent will not respond to a press. For example, if you created a totally transparent button with no border and a bitmap, only the bitmap would respond to a press. Similarly, if a transparent button has a visible border but no bitmap, only the border will respond to a press. Touching the transparent areas of the button does not active the button. *Note: Active Touch requires total transparency on the button in order to work. To make a button totally transparent, set the Overall Opacity (state) setting to 0. If Overall Opacity is set to any other value (for partial transparency), Active Touch will not work.* • Bounding box: This touch style forces the panel to respond to a press anywhere within the rectangular boundaries of the button (regardless of transparencies or border styles). • Pass through: This style allows the user to press "through" one button to press another button underneath. If there is no other button underneath the pass through button, the user simply presses the page (with no resulting action).
Type	The Type (button type) defaults to the button type that was set when the button was created. To change a selected button's type, click Type in the Properties window (General tab) to activate the button type drop-down menu, containing a list of all available button types.
Value Direction	Click the down arrow to select the orientation of the Bargraph: *Note: This property is available for Bargraph and Multi-State Bargraph buttons only.* • For Bargraph buttons, the options are Vertical or Horizontal. • For Multi State Bargraph buttons, the options are Vertical, Horizontal , or Touch Map.
Width	Enter the Width size value (in pixels) to specify the vertical dimensions of the selected button. • You can edit this field to apply specific dimension info for the button. • Note that if you select the button and manually resize it on the page, these values constantly update to indicate the current dimensions.
window Type	Specifies the type of window displayed for Application windows. Select one of the following: • Floating, Resizeable, Moveable • Floating, Fixed Size, Moveable • Floating, Fixed Size, Non-moveable • Docked Left, Fixed Size, Non-moveable • Docked Right, Fixed Size, Non-moveable • Docked Top, Fixed Size, Non-moveable • Docked Bottom, Fixed Size, Non-moveable *Note: This property is available for Application windows only.*
Z-Order	This read-only field indicates the current Z-order setting of the selected button. Note that Z-Order is managed vis the *Bitmaps* dialog (see *Assigning Bitmaps to a Page, Popup Page or Button* on page 34).

Programming Properties

The Properties presented in the *Programming* tab of the Properties window will depend on the current selection in the active Design View window (Page, Popup Page, Sub-Page or Button). Some properties also only apply to specific panel types.

Programming Properties

Feedback	Feedback - Select the type of feedback to associate with this button (channel, inverted channel, always on, momentary, or none). *Note: This property is only available for General and Multi-State General Buttons.* • none - the button will always display the Off state (and will not indicate a Push/Release) • channel - the button will change states (Off to On) on a Push/Release to indicate a channel event • inverted channel - the button will change states (On to Off) on a Push/Release to indicate a channel event • always on - the button will always display the On state (and will not indicate a Push/Release) • momentary - the button will change states, only while the button is being pressed.
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. See Address Codes (Basic and Advanced). *Notes:* • *The combination of Address Port and Address Code must be unique.* • *Address Port and Address Code assignments for Sub-Pages and Sub-Page View Buttons are provided only for use in SEND-COMMANDS (not to trigger actions).*

Programming Properties (Cont.)

Address Code	Select or enter the address code sent to the master 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:

- 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:
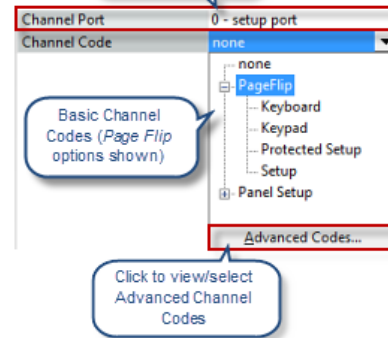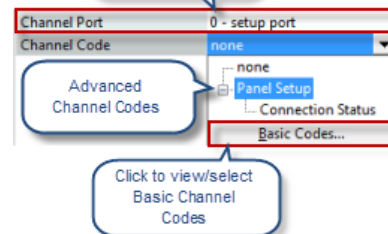
Click on *Date Display* to select from a list of date display formats.

Click on *Time Display* to select from a list of time display formats.

Click **Advanced Codes** to view the Advanced Channel Code options:

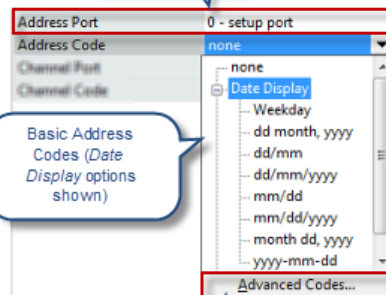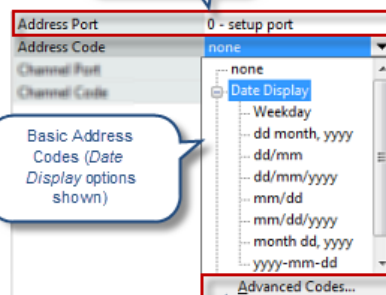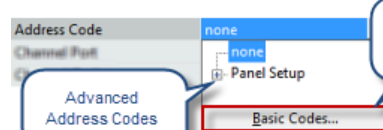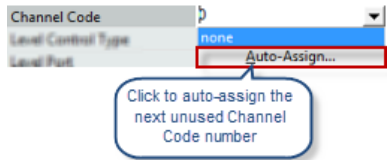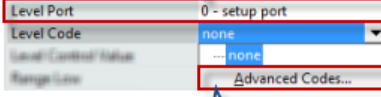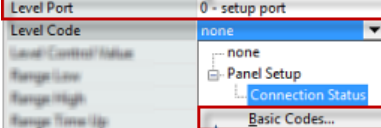| Programming Properties (Cont.)	
Address Code (Cont.)	Click on *None* to leave the Address Code unspecified.
	Click on *Panel Setup* to select **Connection Status**. This option will display the panel's current connection status on the selected element.
	Notes
	• The combination of Address Port and Address Code must be unique.
	The Address Port and Code assignments for Sub-Pages and Sub-Page View Buttons are provided only for use in SEND-COMMANDS (not to trigger actions).
Channel Port	Select or enter the port to which the selected button's Channel Code will be applied.
	The options are "1" (the default setting) and "0-setup port":
	• If **1** is selected as the Channel Port, then the options for the Channel Code property are None and Auto-Assign.
	• If **0-Setup Port** is selected as the Channel Port, then the options for Channel Code are Advanced Codes or Basic Codes. By default, the Basic Channel Codes are displayed. See Channel Codes (Basic and Advanced).
	Notes:
	• The combination of Channel Port and Channel Code must be unique.
	• Channel Port and Channel Code assignments for Sub-Pages and Sub-Page View Buttons are provided only for use in SEND-COMMANDS (not to trigger actions).
Channel Code	Select or enter the channel code sent to the master on the selected port. The options available to the Channel Code property depend on the Channel Port selection:
	• If **1** is selected as the Channel Port, then the options for Channel Code are *None* and Auto-Assign.
	Select **None** to leave the Channel Code unspecified.
	Select **Auto-Assign** to automatically assign the next available Channel Code to the selected TPD5 element.
	If **0-Setup Port** is selected as the Channel Port, then the options for Channel Code are *Advanced Codes* or *Basic Codes*. By default, the Basic Channel Codes are displayed:
	Click on *Page Flip* to select from a list of special page flip options.
	Under *Panel Setup*, click *Popup Drag* to enable the ability for users to drag popup pages around on the panel.
	Click **Advanced Codes** to view the Advanced Channel Code options:
	Click on *None* to leave the Channel Code unspecified.
	Click on *Panel Setup* to select Connection Status. This option will display the panel's current connection status on the selected element.

Programming Properties (Cont.)

Level Control Type	These options allow General and Multi-State General Buttons to directly control a level without the need for NetLinx code. Select a level control type for the selected button (Absolute, Relative or None). • Absolute: The button acts like a preset and sets the level to the desired value. • Relative: The button increments or decrements the current level value by a fixed amount. This option requires that the Level Control Repeat Time property is specified. If either *Absolute* or *Relative* or selected, the following additional Level-related Properties are presented in the Properties window (Programming tab): • Level Port • Level Code • Level Control Value • Level Control Repeat Time (only applicable if relative is selected as the Level Control Type.
Level Port	Select or enter the port to which the selected element's Level Code will be associated. The options are "1" (the default setting) and "0-setup port": • If 1 is selected as the Level Port, then the options for the Level Code property are None and Auto-Assign. • If 0-Setup Port is selected as the Level Port, then the options for Level Code are Advanced Codes or Basic Codes. By default, the Basic Level Codes are displayed, however there are no basic Level Codes at this time. See Level Codes (Basic and Advanced).

Level Code	Select or enter the level code sent to the Master on the selected port (none, 1, or auto-assign).
	
	The options available to the Level Code property depend on the Level Port selection:
	• If **1** is selected as the Level Port, then the options for Level Code are None and Auto-Assign.
	   Select **None** to leave the Level Code unspecified.
	   Select **Auto-Assign** to automatically assign the next available Level Code to the selected TPD5 element.
	• If **0-Setup Port** is selected as the Level Port, then the options for Level Code are Advanced Codes or Basic Codes. By default, the Basic Level Codes are displayed:
	
	Note there no Basic Level Codes to select at this time.
	Click **Advanced Codes** to view the Advanced Level 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.
	Notes
	• The combination of Level Port and Address Code must be unique.
	• The Level Port and Code assignments for Sub-Pages and Sub-Page View Buttons are provided only for use in SEND-COMMANDS (not to trigger actions).

Programming Properties (Cont.)	
Level Function	Select the desired level function for the selected Bargraph or Multi-State Bargraph button.
	Note: This option is available for Bargraph and, Multi-State Bargraph buttons only .
	• Display Only: This option creates a Bargraph button that only displays level information (but cannot be used to control levels). This selection adds the following properties to the Properties window (Programming tab):
	Range Low
	Range High
	Range Inverted
	• Active: This option creates a Bargraph button for controlling levels according the other button parameters set here. This selection adds the following properties to the Properties window (Programming tab):
	Range Low
	Range High
	Range Inverted
	Range Time Up
	Range Time Down
Level Control Value	• For Absolute level control, this value determines the level value that will be recalled by this button.
	• For Relative level control, this value determines the amount of adjustment relative to the current level resulting from this button. For relative level control, this value can be entered as a positive integer (to raise the relative level setting) or a negative integer (to lower the relative level setting).
Level Control Repeat Time	This value determines the repeat time (in 1/10th-second increments) for relative level control.
	Note: This option is only available if the Level Control Type is set to Relative.
	For example, if this value is set to 10, the user can hold the button down for one second to cause the level to raise or lower repeatedly (according to the specified Level Control Value) without having to press the button each time.
Range Low	Enter the bottom of the level range (0-255).
Range High	Enter the top of the level range (0-255).
Range Inverted	If set to Yes, the range is inverted, so that the top of the level range is 0 and the bottom of the range is 255 on both the X and Y axis (default = No).
	Note: This button property is available for Bargraph and Multi-State Bargraph buttons only.
Range Time Up	Specify the amount of time (in 1/10th seconds) it will take for the Bargraph to go from the bottom to the top of the specified range (default = 2).
	This button property is available for General buttons, if a Level Control Type (other than "none") has been selected.
	Note: It is not applicable if the Level Function property is set to "display only".
Range Time Down	Specify the amount of time (in 1/10th seconds) it will take for the Bargraph to go from the top to the bottom of the specified range (default = 2).
	This button property is available for General buttons, if a Level Control Type (other than "none") has been selected.
	Note: It is not applicable if the Level Function property is set to "display only".
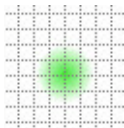
State Properties
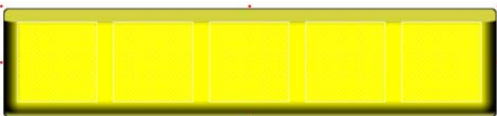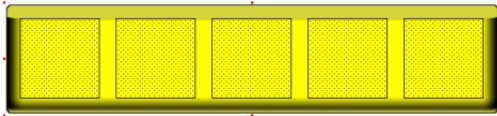
The Properties presented in the *States* tab of the Properties window will depend on the TPD5 Element selected in the active Design View window (Page, Popup Page, Sub-Page, or Button). Some properties also only apply to specific panel types.

Using the All States Option

Use the **All States** option (in the States tab of the Properties window) to apply any changes you make to all states on the selected button. Note that if you have multiple buttons selected (Shift+click to select multiple buttons a page), the All States option only affects states for the button that has Edit Focus. The button with edit focus would be the last one selected, and is indicated by having red-colored square handles (as opposed to the black squares that indicate that a button is selected, but does not currently have edit focus).

State Properties	
Bitmaps	To apply image files to the selected state(s), click the browse button (...) to open the *Bitmaps* dialog, where you can select an image file from among those imported into the project. In TPD5, you can assign up to five bitmap image files to Pages, Popup pages, Sub-Pages, and Buttons, as a state property. Use the options in the Bitmaps dialog to specify each with their own independent justification and placement properties.
	Click the browse (...) button on the Bitmap property (States tab of the Properties window) to open the Bitmaps dialog.
	*Note: If pairs of image resources exist that end in *off/*on, * f/*n, *0/*1, *1/*2 (case insensitive), and the first in the pair is applied to the Off state of a General button, the second will be automatically applied to the On state to make it easier to set up images on a General button.*
Border Color	To change the border color for the selected state(s), click the browse button (...) to open the *Colors* dialog.

State Properties (Cont.)	
Border Name	To change the Border Name for the selected button, click Border Name, and select the desired border from the drop-down list. • If a Border Style was specified (in the General tab), then the borders listed here are limited to those contained in the selected Border Style. • If no Border Style was specified (none), then all border names are available in the provided list.
Chameleon Image	This field allows you to apply a Chameleon Image to the selected state(s). *Note: This property is only available only if the Border Name has been set to None.* • Chameleon Images can be used to create special effects such as drop-shadows. • Click the browse button (...) to open the *Select Resource* dialog, where you can select an image to apply as a Chameleon Image.
Fill Type	To change the fill type for the selected state, click the down arrow to select from a listing of supported fill types: • Solid • Radial • Sweep • Left to Right • Top-Left to Bottom-Right • Top to Bottom • Top-Right to Bottom-Left • Right to Left • Bottom-Right to Top-Left • Bottom to Top • Bottom-Left to Top-Right If *Solid* is selected as the Fill Type, then the fill color is set via the *Fill Color* (State) property. All fill types other than Solid represent various *gradient* fills.
Fill Color	To change the fill color for the selected state, click the browse button (...) to open the *Colors* dialog. For Bargraph buttons, the preview image in the State Manager works differently than for the other button types. For Bargraph buttons, the on and off states are used to indicate a level setting rather than a push/release. As a result, the button image in the Design View window will indicate the Bargraph button as it will appear on the touch panel, but the thumbnails in the State Manager window indicate each state as a separate preview image. For example, the Bargraph button shown below uses yellow as the On state fill color, and green as the Off state fill color. In the State Manager window you would see the On state (yellow) and the Off state (green) as individual thumbnails. The Button Preview window works differently for Bargraph buttons than for the other button types. Rather than using the Push button to view the different states, click and drag on the slider with your mouse cursor (in the Button Preview window) to preview the feedback.
Fill Gradient Colors	If you choose any Fill Type other than *Solid*, the *Fill Color State* property is replaced with the *Fill Gradient Colors* property. This property requires that you choose between two colors in order to generate a gradient across the page. To change the two fill colors for the selected state, click the browse button (...) to open the *Fill Colors* dialog. Click on a particular color to open the Colors dialog for further options. With the exception of "radial", all Fill Type properties that use gradient colors have only the Fill Gradient Colors property. The "radial" Fill Type also controls three separate properties: *Gradient Radius*, *Gradient Center X%*, and *Gradient Center Y%*. When selected, the center of the radial gradient is displayed in the Workspace window. The first color selected in the *Fill Colors* dialog always appears in the center, with the subsequent colors radiating outward in order. When selecting the "radial" Fill Type, the three remaining properties control the appearance of the gradient: • **Gradient Radius** controls the radius, in pixels, in which the gradient will occur. If the Gradient Radius is larger than the element being selected, the gradient will be clipped by the element's dimensions. • **Gradient Center X%** controls the center position of the gradient based on its horizontal position on the selected element, on a scale of 0 to 100. This is determined by the percentage of pixels on the page: if "0" or "100" are selected, the gradient will start at the far left or far right of the selected element, respectively. • **Gradient Center Y%** controls the center position of the gradient based on its vertical position on the selected element, on a scale of 0 to 100. This is determined by the percentage of pixels on the page: if "0" or "100" are selected, the gradient will start at the top or the bottom of the selected element, respectively. To confirm a gradient property, enter the number in the property field and then select any other property or State. The radial gradient will appear in the Workspace window.
Font	To change the font used for text on the selected state(s), click the browse button (...) to open the *Font* dialog, where you can select a Font for the text on the selected state(s). Note that for Listview buttons, the Font property affects the *Primary Text* component.

State Properties (Cont.)	
Font Size	To change the font size used for text on the selected state(s), enter the desired font size in the field. Note that for Listview buttons, the Font Size property affects the *Primary Text* component.
Gradient Center X%	This property is available only if *Radial* has been selected as the *Fill Type*. The *Gradient Center X%* and *Gradient Center Y%* State properties specify the center position of the radial pattern in terms of a percentage of the x/y coordinate. They are integer values ranging from 0 to 100 (percent) where 0 (X), 0 (Y) is the upper-left corner of the element and 100 (X), 100 (Y) is the bottom-right corner. *Note: Values of 50 (X) and 50 (Y) will position the radial pattern in the center of the element.*
Gradient Center Y%	This property is available only if Radial has been selected as the Fill Type. The *Gradient Center X%* and *Gradient Center Y%* State properties specify the center position of the radial pattern in terms of a percentage of the x/y coordinate. They are integer values ranging from 0 to 100 (percent) where 0 (X), 0 (Y) is the upper-left corner of the element and 100 (X), 100 (Y) is the bottom-right corner. *Note: Values of 50 (X) and 50 (Y) will position the radial pattern in the center of the element.*
Gradient Radius	This property is available only if *Radial* has been selected as the *Fill Type*. Use this property to specify the size of the radius (in pixels) where blending will occur. The last color shall fill in the rest of the element from the end of the radius out to the edges of the element when the radius size fits within the element's dimensions; otherwise the pattern will be clipped.
Overall Opacity	Use this field to specify the level of opacity for the selected button (0 - 255, where 0 is totally transparent, and 255 is totally opaque). The default is 255.
Secondary Font	Use this property to specify the font used for the Secondary Font component (the second text line) of a Listview item.
Secondary Font Size	Use this property to specify the font size used for the Secondary Font component (the second text line) of a Listview item. Note that the *Secondary Font* and *Secondary Font Size* State properties are available even if the selected Listview button only uses a single line of text. In this case, if the List View Type is changed to either two-line text or two-line text with icon, the second line of text will use these settings.
Sound	To change or apply a new sound file to the selected state(s), click the browse button (...) to open the *Select Resource* dialog, where you can select a sound file (WAV or MP3) from among those imported into the project. *Note: Only Buttons support the Sound property.*
Streaming Source	Enter the URL or IP Address of the server that will provide the video stream. *Note: This property is available only if Streaming Video is selected as the Video Fill).*
Sub-Page Layout Color	This property is available only for Sub-Page View buttons. It provides the ability to change the color of the Sub-Page placeholders for Sub-Page View buttons in the Design View. Use this feature in situations where the Sub-Page View button uses a fill color or bitmap that provides too little contrast to make the Sub-Page placeholders readily visible on-screen. For example, the Sub-Page placeholders are difficult to see in white on a Sub-Page View button with a light yellow fill: Using this property, the Sub-Page Layout color could be changed to a darker color to provide enough contrast: *Note: This color setting does not affect the color settings of the Sub-Page View button, or any other element of the Scrolling Region when it is displayed on the touch panel. It only affects the Design View in TPDesign5.*

State Properties (Cont.)	
Text	To change or enter the text to be displayed on the selected state(s), click the browse button (...) to open the *Enter Text* dialog, where you can type the new button text. Use the Preview Using Font option to view the text as it will appear in the selected font, style and size (on by default).
	• Unicode characters must be entered via the Enter Text dialog only (not through in-place editing in the States tab of the Properties window).
	• When Unicode text is input, the name of the button will not match it's Off state text.
	• TPD5 (v1.0 or higher) supports complex script languages (to the extent that the True Type font currently selected for that state supports the language in question). These languages include (but are not limited to) Arabic, Hebrew, Thai and Devangari.
	Note: Due to an OS limitation, Hindi (as well as some other languages) will not display properly when typed directly into the Enter Text dialog. This limitation has to do with keyboard support for certain languages (namely that Microsoft does not implement "code-pages" for Hindi and some other languages).
	See this Microsoft FAQ topic for the locales that do not have code-pages:
	http://msdn.microsoft.com/en-gb/goglobal/bb688174.aspx#ques12.
	The result of this limitation is that Hindi (as well as some other languages) cannot be entered via the keyboard. In these cases, the text must be pasted into the Enter Text dialog from the clipboard.
	Note: Formatting codes can be used in the state text for Bargraph and Multi-State Bargraph buttons.
Text Color	To change the text color for the selected state, click the browse button (...) to open the *Colors* dialog.
Text Effect	Text effects are graphic effects that can applied to button/page/popup text. Each text effect is available in several variations ((i.e. Small, Medium, Large or XtraLarge).
	To apply a text effect to the button text, click the browse button (...) to access the *Text Effect* sub-menu. This sub-menu presents all available text effects, sorted by type.
	• Click the + symbol next to any effect type in the sub-menu to see all of the variations on that effect.
	• Once you have selected a text effect, use the Text Effect Color field to specify a color for the effect.
Text Effect Color	If you have selected to apply a text effect, use this field to specify the color of the selected effect. Click the browse button (...) to open the *Colors* dialog.
Text Justification	To set or reset the justification setting for the button text (on the selected state(s)), click the down-arrow and select an option from the list.
Text X Offset	To apply an X and/or Y offset to the text (on the selected state(s)), enter the value for the desired offset (in pixels) in these text fields.
	Alternatively, you can click the browse button (...) to open the Image/Text Positioning dialog where you can make several alignment adjustments to the bitmap, icon and text elements of the button.
	Note: This property is available only if the Text Justification State property has been set to Absolute).
Text Y Offset	To apply an X and/or Y offset to the text (on the selected state(s)), enter the value for the desired offset (in pixels) in these text fields. Alternatively, you can click the browse button (...) to open the Image/Text Positioning dialog where you can make several alignment adjustments to the bitmap and text elements of the button.
	Note: This property is available only if the Text Justification State property has been set to Absolute).
Video Fill	Click the down arrow to select the source of the video to be used as a fill for the selected TPD5 element. G5 panels support streaming video or the MXA-MPL (Modero X® Series Multi Preview Live) as the source for streaming video.
Word Wrap	Use this option to enable the wrapping of text strings that are too long to be displayed across the page on one line.
	Note: Wrapping takes place only at a space. It will not wrap in the middle of continuous text.
	Click the down-arrow and select Yes or No from the drop-down list (default = No).

True Type Font Support

Since G5 panels have the ability to decode and display windows True Type Font files (TTF), TPD5 directly utilizes TTF files. Fonts are presented in the Properties window (States tab), as well as the Button Selection/Draw toolbar and the Add Page and Add Popup Page dialogs. The TTF files listed represent those TTF files installed in windows with their available point sizes.

Formatting Codes

Formatting codes can be used in the Text for Bargraph and Multi-State Bargraph buttons. The following formatting codes will be replaced with the identified values:

- **$P** - level percentage
- **$V** - raw level value
- **$L** - range low
- **$H** - range high
- **$A** - adjusted level value (raw level value - range low)
- **$R** - range (range high - range low)
- **$$** - $ character

Complex Script Support

For page and button state text properties, TPDesign5 supports complex script languages (to the extent that the True Type font currently selected for that state supports the language in question). These languages include (but are not limited to) Arabic, Hebrew, Thai and Devanagari. Complex-script rendering is supported on Modero X Series panels.

NOTE: *Most languages can be entered into the state property field via the windows language bar. However, some languages (notably Hindi and Tamil) are not supported by code-pages. These languages will display ??? for characters entered via the language bar, even if the selected font supports the language. However, text in these languages can still be pasted via the clipboard or via the Alt-<Scan Code> method.*

Assigning Borders to TPD5 Elements

Borders can be assigned to Standard Popup Pages, Sub-Pages and Buttons, via the *Border Name* and *Border Color* State Properties. Use the **Border Style** (General) property to limit the *Border Names* available in the States tab to those that belong to the selected Border Style.

1. In a Design View window, select the Popup Page (Standard or Sub-View) or Button to which you want to add or change the border (with the Selection tool).
2. In the *States* tab of the Properties window, click on the **Border Name** property. Click the down arrow to open a drop-down menu of available Border Styles (FIG. 351):
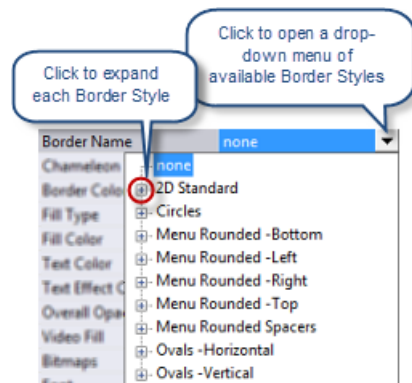
FIG. 351 Border Name (State) Property - Border Styles drop-down menu

> **NOTE:** *If None is selected in the Border Style (General) property, then all border names are included in this list. However, if a Border Style was selected, then only that Border will be listed here (along with the "none" option).*

3. Select a *Border* to apply it to the selected element.
4. Click on the **Border Color** (State) property to select a color for the border, via the *Colors* dialog.
5. Click **OK** to close the *Colors* dialog.

NOTE: *Alternatively, drag and drop the Border Name property from the States tab directly onto a Popup Page or Button to apply the indicated border. This technique can also be used to apply Border Colors.*

Assigning Fills (Fill Type and Color) to TPD5 Elements

Color Fills can be assigned to Pages, Standard Popup Pages, Sub-Pages and Buttons, via the Fill Type and Fill Gradient Colors State Properties.

1. In a Design View window, select the Page, Popup Page (Standard or Sub-View) or Button to which you want to add or change the fill (with the Selection tool).
2. In the States tab of the Properties window, click on the **Fill Type** field. Click the down arrow to open a drop-down menu of available Fill Types (FIG. 352):

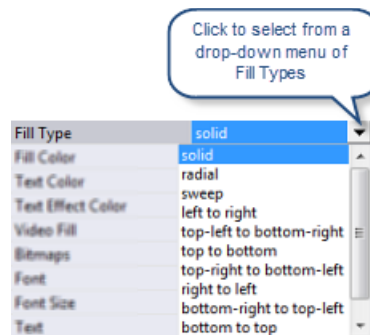**FIG. 352** Border Name (State) Property - Fill Type drop-down menu

3. Select a *Fill Type* to apply it to the selected element.
4. Click on the **Fill Color** property, then click the browse (...) button to select a color via the *Colors* dialog.

5. Select a color (or multiple colors if a gradient Fill Type is selected). Note that if you have selected **Solid** as the *Fill Type*, use the **Fill Color** property to select the color to use (via the *Colors* dialog). However, all of the other Fill Types represent various *gradient* fills. In these cases, use the **Fill Gradient Colors** property to select at least two colors to use for the gradient fill (via the Fill *Colors* dialog).

6. Click **OK** to close the *Colors* or *Fill Colors* dialog.

NOTE: *Alternatively, drag and drop the Fill Type property from the States tab directly onto a Popup Page or Button to apply the indicated Fill Type. This technique can also be used to apply Fill Colors.*

Assigning Video Fills to TPD5 Elements

Video Fills can be assigned to Pages, Standard Popup Pages, Sub-Pages and Buttons, via the Video Fill State Property. G5 Panels can use either *Streaming Video* or an *MXA-MPL* as the source for the video displayed (FIG. 346):
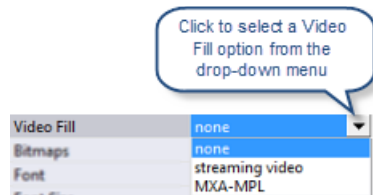

FIG. 353 Video Fill State Property

1. In a Design View window, select the Page, Popup Page (Standard or Sub-View) or Button to which you want to add or change the video fill (with the Selection tool).

2. In the Properties window - States Tab, click **Video Fill** to select a video source (*None*, *Streaming Video* or *MXA-MPL*). Since this is a state-oriented setting, be sure to consider all of the button states when applying the video fill. To apply the video fill across all states, use the All States option in the Properties window. Alternatively, use Ctrl+A to select all states in the State Manager window.

   ● Select **Streaming Video** to add Streaming Source to the list of State properties. Enter the URL or IP Address of the server that will provide the video stream in the Streaming Source field:

   ● Select **MXA-MPL** if you will use an MXA-MPL (Modero X® and Modero S Series Multi Preview Live) to provide the video stream.

Assigning Text to TPD5 Elements

Text can be assigned to Pages, Standard Popup Pages, Sub-Pages and Buttons, via the Text and Font-related State Properties. These include:

   ● Text Color (see page 254)
   ● Text Effect Color (see page 254)
   ● Font (see page 252)
   ● Font Size (see page 253)
   ● Text (see page 254)
   ● Text Justification (see page 254)
   ● Text Effect (see page 254)
   ● Word Wrap (see page 254)

Assigning Text to a Page, Popup Page Sub-Page or Button

1. In a Design View window, select the Page, Popup Page, Sub-Page or Button to which you want to add or change the text (with the Selection tool).

2. In the *States* tab of the Properties window, click on the **Text** property. Click the browse (...) button to open the *Enter Text* dialog (FIG. 354).
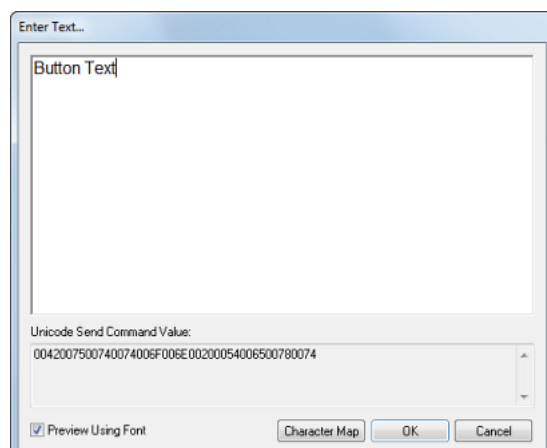

FIG. 354 Enter Text dialog

> **NOTE:** *Alternatively, type directly in the Text property field in the Properties window.*

3. Type the text that should appear on the selected element and click **OK** to close the dialog.
    Note that the entered text is indicated in the *Text* (State) property field (FIG. 355):

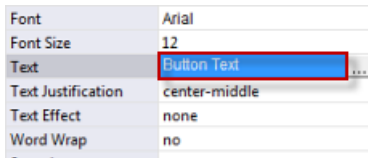| Font	Arial
Font Size	12
Text	Button Text
Text Justification	center-middle
Text Effect	none
Word Wrap	no

FIG. 355 Text (State) Property indicating text entered via the Enter Text dialog.

4. Click on the **Text Color** property. Click the browse (...) button to select a color via the *Colors* dialog, and click **OK** to close the dialog. Note that the selected text color is indicated in the Text Color property field (in the States tab).
5. Use the **Font** and **Font Size** properties to specify a font and size for the entered text.
6. Use the **Text Justification** property to select a justification setting for the entered text from the drop-down menu.
7. Optionally, click on **Text Effect** to apply a Text Effect to the entered text. If a Text Effect is applied, then use the **Text Effect Color** property to specify a color for the selected effect.
8. Click on the **Word Wrap** property, and select either *Yes* or *No* from the drop-down menu to specify whether to automatically wrap words to fit the area of the element.

> **NOTE:** *Alternatively, drag and drop the Text property from the States tab directly onto a Page, Popup Page or Button to apply the indicated text. This technique can also be used to apply Text Colors, Font/Size, Text Justification, Text Effect/Color and Word Wrap.*

Event Properties

The Properties presented in the *Events* tab of the Properties window are supported only by Pages, and General and Multi-State General Buttons. Click the browse (...) button to open the *Edit Event Actions* dialog, where you can specify an Event Action to be triggered by a button press (FIG. 356).

FIG. 356 Edit Event Actions dialog

Events include:
*   Page Flips
*   Launch Actions
*   Actions (Command or String)

See the *Events* section on page 266 for details.

Event Properties	
Button Press	Use this property to associate one or more Event Actions when a button press is performed on the panel.
Button Release	Use this property to associate one or more Event Actions when a button release is performed on the panel.
Show Page	Use this property to associate one or more Event Actions when the selected page is shown.
Hide Page	Use this property to associate one or more Event Actions when the selected page is hidden.
Gesture Any	Use this property to associate one or more Event Actions when any gesture is performed on the panel.
Gesture Up	Use this property to associate one or more Event Actions when a Gesture Up gesture is performed on the panel.
Gesture Down	Use this property to associate one or more Event Actions when a Gesture Down gesture is performed on the panel.

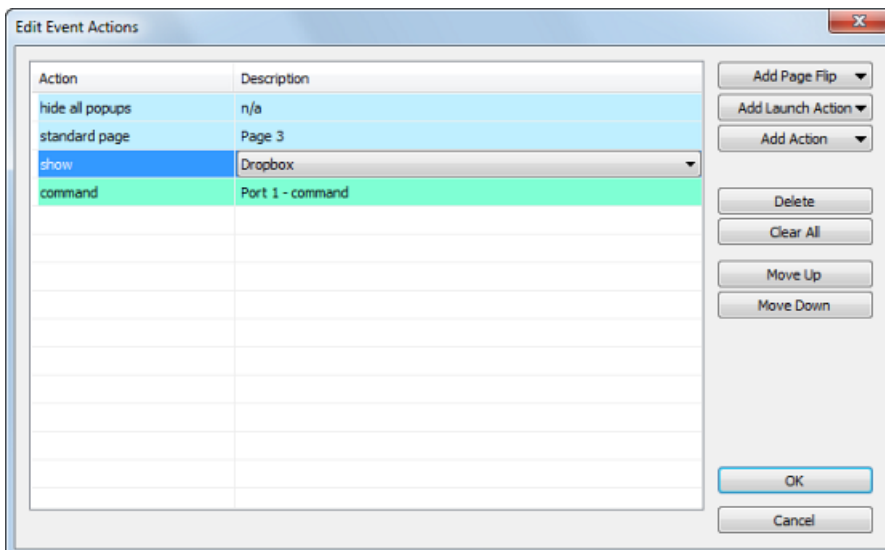| Event Properties (Cont.)	
Gesture Right	Use this property to associate one or more Event Actions when a Gesture Right gesture is performed on the panel.
Gesture Left	Use this property to associate one or more Event Actions when a Gesture Left gesture is performed on the panel.
Gesture Dbl Tap	Use this property to associate one or more Event Actions when a Double-Tap gesture is performed on the panel.
Gesture 2-Finger Up	Use this property to associate one or more Event Actions when a Two-Finger Gesture Up gesture is performed on the panel.
Gesture 2-Finger Dn	Use this property to associate one or more Event Actions when a Two-Finger Gesture Down gesture is performed on the panel.
Gesture 2-Finger Rt	Use this property to associate one or more Event Actions when a Two-Finger Gesture Right gesture is performed on the panel.
Gesture 2-Finger Lt	Use this property to associate one or more Event Actions when a Two-Finger Gesture Left gesture is performed on the panel.
Item Selected	Use this property to specify an event to occur when a list item is selected in a Listview button. Click the browse (...) button to open the *Edit Event Actions* dialog, where you can specify an Event Action to be triggered by Item Selected. Events include Page Flips, Launch Actions and Actions (Command or String).
Scrollbar Begin	Use this property to specify an event to occur when the beginning (top) of the scrollbar is reached on a Listview button. Click the browse (...) button to open the *Edit Event Actions* dialog, where you can specify an Event Action to be triggered by Scrollbar Begin. Events include Page Flips, Launch Actions and Actions (Command or String).
Scrollbar End	Use this property to specify an event to occur when the end (bottom) of the scrollbar is reached on a Listview button. Click the browse (...) button to open the Edit Event Actions dialog, where you can specify an Event Action to be triggered by Scrollbar End. Events include Page Flips, Launch Actions and Actions (Command or String).

Grab Properties and Paint Properties Tools

The *Grab Properties* and *Paint Properties* tools work together with the Property Painter dialog to allow you to grab (copy) the properties of a selected Button, Page or Popup Page, and paint (copy) them onto another Button, Page or Popup Page (FIG. 357):
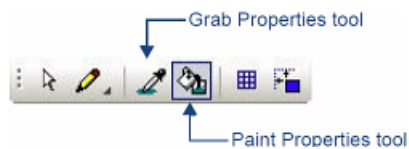
FIG. 357 Grab Properties and Paint Properties Tools

Grabbing Properties (via the Grab Properties Tool)

Use the Grab Properties function to copy a specific set of properties from a selected design element (Button, Page or Popup Page). Once a property set has been Grabbed, it can be applied (Painted) to another design element in the project. This technique can save time as well as promote consistency in the TPDesign5 project.

1. Select **Grab Properties Tool** from the Edit menu, the Design View context menu, or click the toolbar button to activate the Grab Properties tool. Note that the cursor changes to reflect this tool selection (FIG. 358):

FIG. 358 Grab Properties tool

2. Click on a design element in a Design View window to grab the properties and settings of the selected design element. This action invokes the Property Painter dialog (also accessible via the View menu).

   The following example shows the Property Painter dialog invoked as a result of selecting a Multi-State General button with 12 states (FIG. 359):

FIG. 359 Property Painter

3. Select the Properties of the selected design element that you want to grab:
   - Select one of the pre-defined Property Sets from the Property Set drop-down menu:
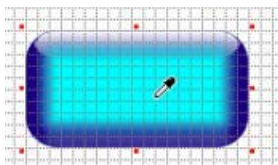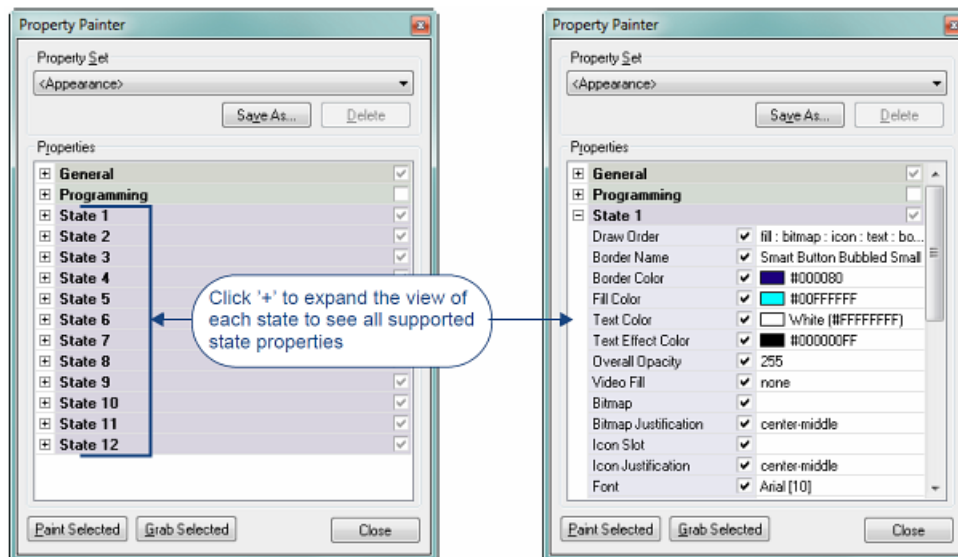   - Select <**Appearance>** to grab all appearance-oriented properties.
     - Select <**Border, Fill and Text Colors>** to grab only the color settings for border, fill and text colors.
   - Manually click to select or de-select properties. Note that custom Property Sets can be saved via the Save As button.
4. With a set of properties selected (and with the design element selected), click **Grab Selected**.

Once a Property Set has been grabbed, it can be applied (Painted) to another design element, via the Paint Properties tool.

Painting Properties (via the Paint Properties Tool)

1. With a Property Set selected in the *Property Painter* dialog, select **Paint Properties Tool** from the Edit menu, the Design View context menu, or click the toolbar button to activate the Paint Properties tool. Note that the cursor icon changes to reflect this tool selection (FIG. 360):
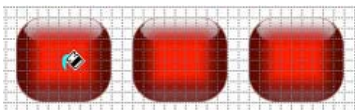
FIG. 360 Paint Properties tool

2. Click on each design element that you want to Paint using the active Property Set in the *Property Painter* dialog (FIG. 361):
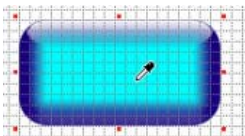
FIG. 361 Grab Properties tool

The figure below shows three buttons being Painted with the Border, Fill and Text Colors from the button shown above (FIG. 362):
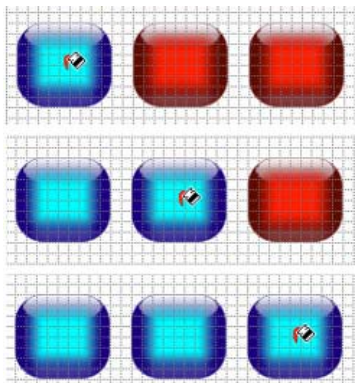
FIG. 362 Paint Properties tool - button example

Saving a Properties Set

Use the **Save As** option under *Property Set* in the Property Painter dialog to save a set of Properties (but not their values) that can be recalled later.

States

Overview

All G5 panel entities (Pages, Popup Pages and Buttons) have at least one state.

- Pages, Standard Popup Pages and Sub-Pages have only one state.
- General, Bargraph and Text Input buttons have only two states (on/off).
- Sub-Page View buttons only use one state (Off).
- Multi-State General and Multi-State Bargraph buttons can have up to 256 states.
- States start at 1.

The State Manager window allows you to view and modify individual states. The State Manager window supports full Cut, Copy, Delete, Insert, Replace and Paste as well as drag and drop capabilities. The State Manager context window (open via right mouse click on any thumbnail in the State Manager) support allows the user to Add single or multiple states, Replace states, Insert single or multiple states and Remove states.

- For Multi-State General buttons the different states (up to 256) are used to animate a button from Off to On (Range Time Up) and back again to Off (Range Time Down). When the button is turned on it will display all the assigned states from first to last with a specified time interval between each state's display. When the button is turned back off, the states will be displayed in reverse order. The interstate time intervals are user definable in 1/10th second increments. A zero entry will automatically advance / retreat to the ending / beginning state without displaying any intervening states.
- For Multi-State Bargraph buttons, the level will directly reflect the displayed state. You can set an allowable range within a Bargraph that has states. Anything outside of that range will not be represented by a state.
- For buttons with multiple states, Send Commands can set the state number, provided it is not a level type button.

Setting State Properties

The ability to set the State Properties (including border name, border color, fill color, text color, video fill, bitmap, bitmap justification, font, text, text justification, word wrap preference and sound) is provided at the state level, via the Properties window - *States* Tab.

The State Manager interacts with the States tab of the Properties window to allow the visual aspects of a button, page, or popup page to be set.

- If the State Manager is not visible, or if no states have been selected, the State Properties will show a list of all of the states for the selected button, page, or popup page. The individual properties for a state can be shown or hidden by clicking either on the "State n" category item or by clicking the +/- tree control for that state.
- If multiple states are selected in the State Manager, the State Properties will represent the intersection of the selected states, reflected by the text of the title item. Setting a property value will propagate that value across all selected states.

State Manager window

The *State Manager* is typically located along the bottom edge of the screen (although it is a dockable window and you may move it anywhere you like) and is used to view/edit the various States of a selected button. Each state of the selected button is displayed as a thumbnail image in this window.

To display the State Manager window, select **View > State Manager** (FIG. 363):

FIG. 363 State Manager window

The State Manager interacts with the Properties window to allow the visual aspects of a page, popup page or button to be set. Select a Page, Popup page or button to view the state or states associated with it.

Double-click on any thumbnail in the State Manager window to view/edit the properties for the selected state, in the Properties window (States tab).

The State Manager window allows the viewing and modification of individual states, and supports full Cut, Copy, Delete, Insert, Replace and Paste as well as drag and drop capabilities.

Right mouse click on any thumbnail to open the State Manager context menu, which includes options to Add single or multiple states, Replace states, Insert single or multiple states and Remove states.

NOTE: *Because the thumbnails displayed in the State Manager window are scaled versions of the button images, some visual distortion may occur. This is only a result of the scaling, and does not represent distortion on the button itself.*

Adding States To a Multi-State Button

There are several ways in which new states can be added to a multi-state button:

Add States

The *Add States* option provides a method of adding states to a multi-state button by duplicating an existing state a specified number of times:

To add States to a Multi-State button, via the *Add States* dialog:

1. Select a Multi-State Button in the Design View.

2. Select a State in the State Manager window.

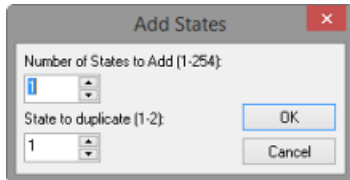3. Select B**utton > Add States** to open the *Add States* dialog (FIG. 364):

FIG. 364 Add States dialog

4. In the **Number of States to Add (1-254)** field, enter the number of states that you want to add to this button (max = 254). Alternatively, you can use the up and down arrows to change the number.

5. In the **State to Duplicate (1-<#>),** enter the number of the state that you want to duplicate.
   - By default, the selected state is set as the state to be duplicated.
   - The state specified here will be used for all new states added via this dialog until this value is changed.

6. Click **OK** to close the *Add States* dialog. The new States are added after the last state of the button (in the State Manager window).

Insert States

The *Insert States* option provides a method of inserting states into a multi-state button by duplicating an existing state a specified number of times:

To insert States into a Multi-State button, via the *Insert States* dialog:

1. Select a Multi-State Button in the Design View.

2. Select a State in the State Manager window.

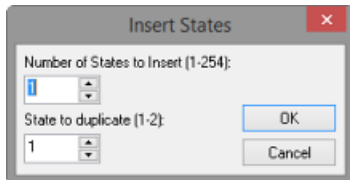3. Select **Button > Insert States** to open the *Insert States* dialog (FIG. 365):

FIG. 365 Insert States dialog

4. In the **Number of States to Insert (1-254)** field, enter the number of states that you want to insert into this button (max = 254). Alternatively, you can use the up and down arrows to change the number.

5. In the **State to Duplicate (1-<#>)**, enter the number of the state that you want to duplicate.
   - By default, the selected state is set as the state to be duplicated.
   - The state specified here will be used for all states inserted via this dialog until this value is changed.

6. Click **OK** to close the *Insert States* dialog. The new States are inserted after the selected state (in the State Manager window).

Adding States via Drag-and-Drop

States can be added to a multi-state button via drag-and-drop in the State Manager window:

1. Select a multi-state button in the Design View.

2. In the State Manager window, right-click on the State that you want to duplicate.

3. Holding the right mouse button, drag the selected State to the desired position. This will highlight a second (target) State in the State Manager window.

4. Release the right-mouse button to invoke the *State Manager Drag-and-Drop* Menu.

5. Select **Insert copy before State <#>** .

State Manager Drag-and-Drop Menu

To access the *State Manager Drag-and-Drop* menu, select a button state (thumbnail view), and hold the left mouse button down while dragging the selected state to another location in the State Manager window (FIG. 366):
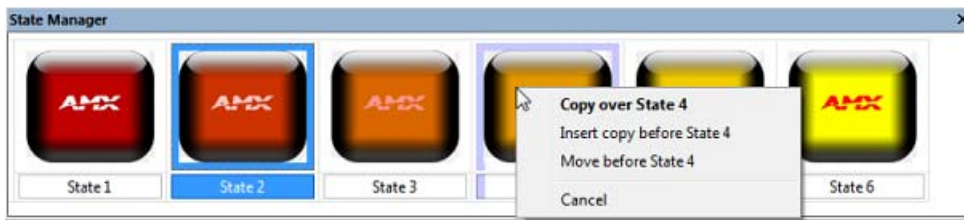
FIG. 366 State Manager Drag-and-Drop Menu

The options in the *State Manager Drag-and-Drop* menu are described below:

- **Copy over State <#>** - Select to replace the target (highlighted) State with the source (selected) State. This option is only presented if the Source (selected) State is dragged directly above the target (highlighted) State (FIG. 367):

FIG. 367 Drag-and-Drop replace icon

The remaining options are available if the cursor is placed either directly above the target (highlighted) State, or between two States (FIG. 368):

FIG. 368 Drag-and-Drop between icon

- **Insert copy before State <#>** - Select to insert a copy of the Source (selected) State directly before the target (highlighted) State. In this case, the original Source State is left in place.
- **Move before State <#>** - Select to move the Source (selected) State directly before the target (highlighted) State.
- **Cancel** - Select to cancel the drag-and-drop operation.

Adding States from the Clipboard

States may be cut/copied and pasted via clipboard memory:

1. Select the button from which the states will be copied.
2. In the State Manager window, select one or more states.
   - Hold down the Ctrl key while left-clicking to add states to the selection.
   - Left-click + Shift to select a range of states.
3. Press Ctrl-C to copy the selected states to the clipboard (or select **Edit > Copy** from the main menu or click on the Copy button from the main toolbar, or right click and select Copy from the context menu).
4. Select a multi-state button as a target for the paste operation.
5. In the State Manager, select the state prior to which the new states will be inserted.

   NOTE: *To add the copied states to the end of the series, ensure that no states are currently selected in the State Manager (or select the last state in the series).*

6. Press Ctrl-V to paste the states from the clipboard (or select **Edit > Paste** from the main menu or click on the **Paste** button from the main toolbar, or right click and select **Paste** from the State Manager context menu).

Replacing States

There are two ways to replace states in a multi-state button:

Replacing States From the Clipboard

1. Select a multi-state button in the Design View.
2. In the State Manager window, select the source state(s).
   - Hold down the Ctrl key while left-clicking to add states to the selection.
   - Hold down the Shift key while left-clicking to add a range of states to the selection.
3. Select **Copy** from the Edit menu or the State Manager Context Menu, or press Ctrl-C.
4. Select the multi-state button whose states will be replaced.

5. Select the destination states.
6. Select Paste from the Edit menu or the State Manager Context Menu, or press Ctrl-V.
   - If the number of destination states is equal to the number of source states, the destination states are replaced one-for-one from the clipboard.
   - If the number of destination states is less than the number of source states, the destination states are replaced one-for-one until all have been replaced, with remaining source states being unused.
   - If the number of destination states is greater than the number of source states, the destination states are replace one-for-one until all source states have been used, at which point replacing will start again at the beginning of the source states until all destination states have been replaced.

Replacing States via Drag-and-Drop

1. Select a multi-state button in the Design View.
2. In the State Manager window, select the source state(s).
3. Press and hold the right mouse button, and drag the mouse over the first state to be replaced.

Beginning at the state the drop occurred over, states will be replaced one-for-one until either the number of source states have been used or the end of the states collection is reached.

NOTE: *The same operation can be performed with a left mouse button drag-and-drop, selecting Copy over State <#> from the State Manager Drag-and-Drop Menu.*

Setting the Maximum Active State For a Button

You can set the maximum active state on a multi-state button by selecting the last state in a sequence (in the State Manager window) and selecting the **States > Set As Max Active State** option. The state tagged as the max active state will be the last one included in the multi-state sequence.

All states beyond the max active state are ignored when the button is pushed.

Note that the states that occur after the max active state in the sequence are displayed with crosshatching across the labels on the thumbnails in the State Manager window, to indicate which states will not be included in the multi-state sequence.

Removing States From A Button

States can be removed from a Multi-State General or Multi-State Bargraph button (the number of states is fixed for the other button types) by either deleting them from the collection, or by cutting them to the clipboard.

Note: Multi-General or Multi-Bargraph buttons must have at least two states. Actions that would cause the number of states to drop below two are not allowed.

Deleting States

1. Select the states to be deleted. Hold down the Ctrl key while left-clicking to add states to the selection. Hold down the Shift key while left-clicking to add a range of states to the selection.
2. Delete the selected states by selecting the **Edit > Delete**, **State Manager Context Menu > Delete**, or the **Del** key.

Cutting States To the Clipboard

1. Select the states to be cut. Hold down the Ctrl key while left-clicking to add states to the selection. Hold down the Shift key while left-clicking to add a range of states to the selection.
2. Cut the selected states to the clipboard by selecting **Edit > Cut**, **State Manager Context Menu > Cut**, or **Ctrl-X** .

Reordering States On a Button

Changing the order of states in a Multi-State General or Multi-State Bargraph button can be accomplished either through the clipboard or by drag-and-drop:

Reordering States Via the Clipboard

1. In the State Manager window, select the states to be moved.
   - Ctrl + click to select multiple states individually.
   - Shift + click to select a range of states.
2. Cut the selected states to the clipboard (Ctrl-X).
3. If the states are to be moved to the end of the collection, ensure that no states are currently selected in the State Manager (click anywhere outside of a state thumbnail, or press ESC). Then, paste the states from the clipboard by selecting **Edit > Paste**, **State Manager Context Menu > Paste**, or **Ctrl-V**.
4. If the states are to be moved elsewhere in the collection, first left-click to select the state prior to which the new states will be inserted. Then insert the states from the clipboard by selecting **Edit > Insert**, **State Manager Context Menu > Insert**, or **Ctrl-V**.

Reordering States Via Drag-and-Drop

A simplified alternative to using the clipboard to reorder states is to use drag-and-drop.

1. Select the states to be moved.
   - Ctrl + click to select multiple states individually.
   - Shift + click to select a range of states.
2. Press and hold the left mouse button while over one of the selected states. While continuing to press the left mouse button, move the mouse to the location where the states will be moved. If the states are to be moved to the end of the collection, drag the states beyond the last state. If they are to be moved elsewhere in the collection, drag the states over the space between state thumbnails where they will be moved.
3. Release the left mouse button (the same operation can also be performed with a right mouse button drag-and-drop, selecting "**Move...**" from the drag-and-drop menu).

Events

Overview

In TPD5, *Events* are used to define the behavior of Pages and (General and Multi-State General) Buttons.

- For Pages, Events can be triggered by Show/Hide Page or by any Gesture.
- For General and Multi-State General Buttons, Events can be triggered by a Button Press/Release or by any Gesture.
- Refer to page 273 for details on Gestures.

Each Event represents an Action List - an ordered list containing one or more event actions. Any of the supported event actions can be added to an action list in any order. Events are assigned to Pages and (General/Multi-State General) Buttons via the *Edit Event Actions* dialog (FIG. 369):
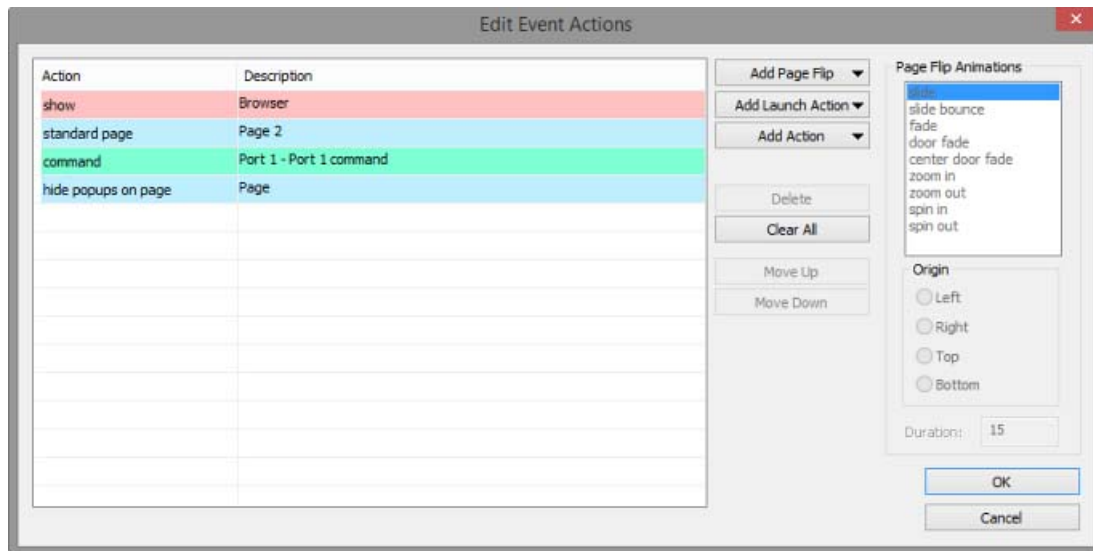

FIG. 369 Edit Event Actions dialog

Event actions include:

- **Page Flips** - *Page Flips* provide the ability to "flip" to a different page on the panel, based on either a button press or release, a gesture, or when a specific Page is either opened or closed. Page flip events replace the *page-flip* G4 button property in TPDesign4. See *Page Flips* on page 268 for details.
- **Launch Actions** - *Launch Actions* provide the ability to open an Application window on the panel, based on either a button press or release, a gesture, or when a specific Page is either opened or closed. See *Launch Actions* on page 271 for details.
- **Actions** - *Actions* provide the ability to trigger a NetLinx Command or send a String based on either a button press or release, a gesture, or when a specific Page is either opened or closed. See *Actions* on page 272 for details.

Assigning Events to Pages or Buttons

Events can be assigned to Pages or (General or Multi-State General) Buttons:

1. Select a Page or Button (General or Multi-State General) in the Workspace window or Design View window.
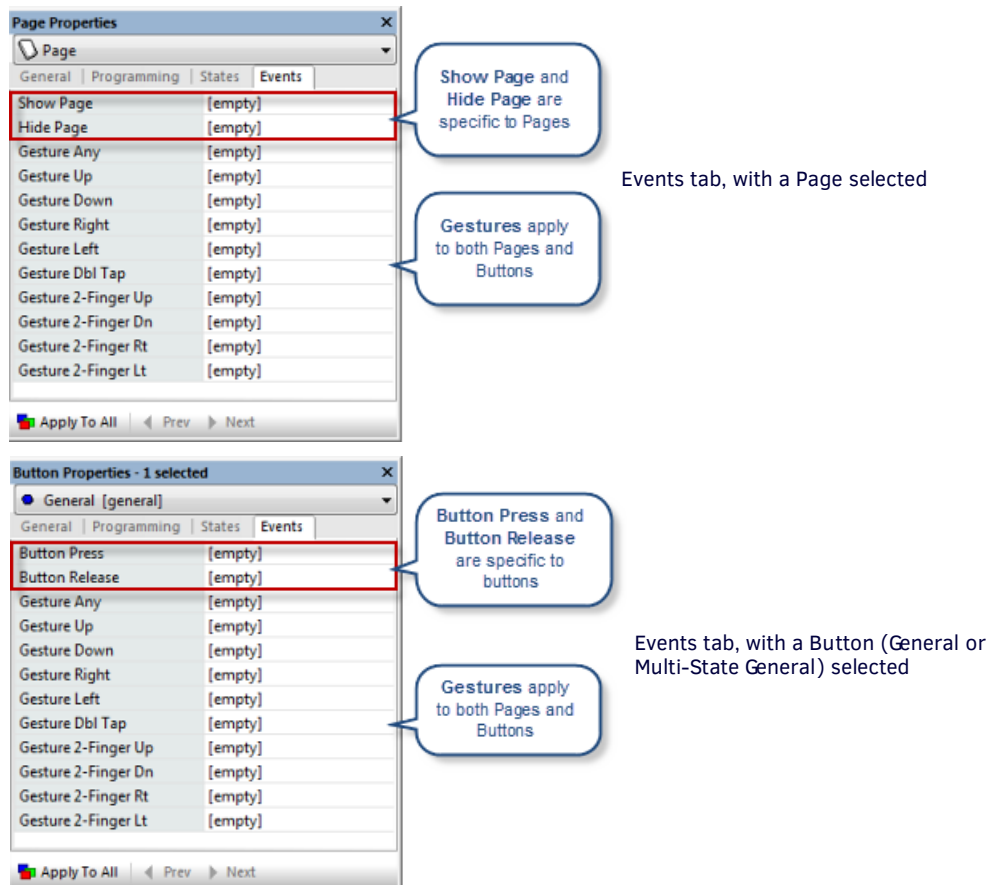2. In the Properties window, open the *Events* tab (FIG. 370):

FIG. 370 Properties window - Events tab

If a *Page* is selected, then the following Page-specific Events are indicated in this tab:

- **Show Page:** The event will occur when the specified Page is shown.
- **Hide Page**: The event will occur when the specified Page is hidden.
- **Gestures** (Any, Gesture Up/Down, Gesture Right/Left, Dbl tap, 2-Finger Up/Dn, Gesture 2-Finger Rt/Lt): The event will be triggered by the selected Gesture.

If a General or Multi-State General *Button* is selected, then the following Button-specific Events are indicated in this tab:

- **Button Press**: The event will occur when the specified Button is pressed.
- **Button Release**: The event will occur when the specified Button is released.
- **Gestures** (Any, Gesture Up/Down, Gesture Right/Left, Dbl tap, 2-Finger Up/Dn, Gesture 2-Finger Rt/Lt): The event will be triggered by the selected Gesture when performed on the selected Page or Button.

3. Click to select any Event in this tab, and click the browse (...) button to edit the Event Actions for the selection, via the *Edit Event Actions* dialog (FIG. 371):

FIG. 371 Accessing the Edit Event Actions dialog

4. In the *Edit Event Actions* dialog, add and edit the event actions included in this Event. Event actions include Page Flips, Launch Actions or Actions. Note that each Event is essentially an Action List - an ordered list containing one or more of the supported actions.

    Any of the supported actions can be added to an action list in any order.

Re-Ordering Event Actions

Event actions are triggered in the order in which they appear in the E*dit Event Actions* dialog. By default, event actions are listed in the order in which they were added.

To re-order the event actions, select an event action and use the **Move Up** and **Move Down** button to adjust it's position in the ordered list.

Deleting Event Actions

Select an event action and click **Delete** to delete that event action from the list.

Clearing All Event Actions from an Event

Click **Clear All** to remove all event actions from the list.

5.   Click **OK** to save changes and close the *Edit Event Actions* dialog.

Page Flips

Page Flips provide the ability to "flip" to a different page on the panel, based on either a button press or release, a gesture, or when a specific Page is either opened or closed.

In TPD5, page flips are Events that can be assigned to Pages or Buttons. Events are defined via the *Events* tab of the Properties window.

Page flip actions replace the G4 button *page-flip* property in TPDesign4.

NOTE: *In addition to Page Flips, Launch Actions and/or Actions (NetLinx commands and strings) can be assigned as Events to Pages and Buttons. See Working With Events for details.*

Page Flip Types

Click the **Add Page Flip** command button to access the *Page Flip Types* drop-down list, which allows you to select from a list of page flip types (see FIG. 380 on page 270):

Standard Page

This selection displays a drop-down menu in the *Description* column listing the standard pages in your project - select a target page for this page flip (FIG. 372):

Action	Description
standard page	Page 1 ∨
	Page 1
	Page 2
	Page 3
	Page 4
	Page 5

FIG. 372 Standard Page Flip - Pages list

Previous Page

This selection sets the page flip to go to the previous page (relative to the order of existing page flips) when this page flip is triggered.

Show Popup

This selection populates the *Description* column with a drop-down list of popup pages in your project - select a target popup page to show when this page flip is triggered (FIG. 373).

Action	Description
show popup	Popup Page 1 ∨
	Popup Page 1
	Popup Page 2
	Popup Page 3
	Popup Page 4
	Popup Page 5
	Popup Page 6
	Popup Page 7

FIG. 373 Show Popup Page Flip - Popup Pages list

Hide Popup

This selection populates the *Description* column with a drop-down list of popup pages in your project - select a target popup page to hide when this page flip is triggered (FIG. 374):

Action	Description
hide popup	Popup Page 1 ∨
	Popup Page 1
	Popup Page 2
	Popup Page 3
	Popup Page 4
	Popup Page 5
	Popup Page 6
	Popup Page 7

FIG. 374 Hide Popup Page Flip - Popup Pages list

Toggle Popup

This selection populates the *Description* column with a drop-down list of popup pages in your project. Select a target popup page to toggle hide/show when this page flip is triggered (FIG. 375):

FIG. 375 Toggle Popup Page Flip - Popup Pages list

Hide Popup Group

This selection populates the *Description* column with a drop-down list of popup page groups in your project. Select a target popup page group to hide when the event is triggered (FIG. 376):

FIG. 376 Hide Popup Group Page Flip - Popup Groups list

Hide Popups On Page

This selection populates the *Description* column with a drop-down list of standard pages in your project. Select the page that you want to hide the Popups on when the event is triggered (FIG. 377):

FIG. 377 Hide Popups On Page Flip - Popup Pages list

Hide All Popups

This selection sets the page flip to clear all popup pages when the event is triggered.

NOTE: *Multiple Popup actions like Toggle popup, Show Popup and Hide Popup for the same Popup Page are allowed on the same button.*

Standard Animated

This selection populates the *Description* column with a drop-down list of standard pages in your project - select a target page for this page flip (FIG. 378):

Page Flip Animation options only apply to specific animation types (i.e. Slide, Door Fade).

FIG. 378 Hide Popups On Page Flip - Popup Pages list

Depending on the animation type selected, the *Page Flip Animations* options may be enabled. Use these options to set the Origin and/or Duration for this page flip.

Previous Animated

This selection sets the page flip to go to the previous page (relative to the order of existing page flips) when this page flip is triggered, and displays the *Action* drop-down menu - select a page flip animation to use for this page flip.

Password-Protected

This option provides the option of requiring the end-user to provide a valid password in order to flip to a specified page. This selection populates the *Description* column with a drop-down list of standard pages in your project (FIG. 379):
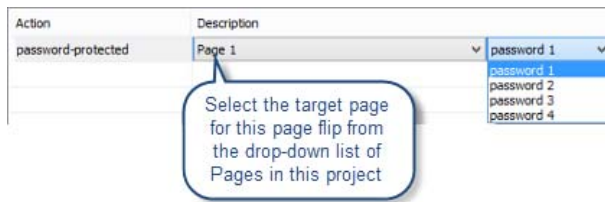
FIG. 379 Hide Popups On Page Flip - Popup Pages list

This selection adds a second drop-down menu to select which of the four panels passwords will be required to flip to the specified target page.

NOTE: *Passwords 1-4 are set on the touch panel via the Settings > G5 Settings page. Refer to the* X Series G5 Touch Panels Configuration and Programming Instruction Manual *for details.*

Adding a Page Flip to a Button

In TPD5, Page Flips are managed as Events that can be assigned to General and Multi-State General buttons. Events are defined via the Events tab of the Properties window:

1. Select the Button to which you will assign the Page Flip action in the Design View.

2. In the *Events* tab of the Properties window, select the Event to which you will assign the Page Flip. Page Flips can be assigned to any Event, but in this example it will be triggered by a *Button Press*.

3. Click the browse (...) button to open the *Edit Event Actions* dialog.

4. Click **Add Page Flip** and select a page flip type from the *Add Page Flip* drop-down menu (FIG. 380):
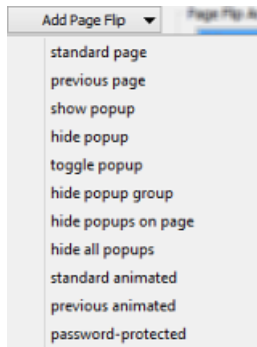

FIG. 380 Add Page Flip drop-down menu

Note that the options in the *Add Page Flip* menu are enabled/disabled based on the active Project. For example, the "popup" and "popup group" related options are disabled if the Project does not have any Popup Pages or Popup Groups.

5. The selected type of *Page Flip* is added to the *Action* column - in this example, a *Standard* page flip (FIG. 381):
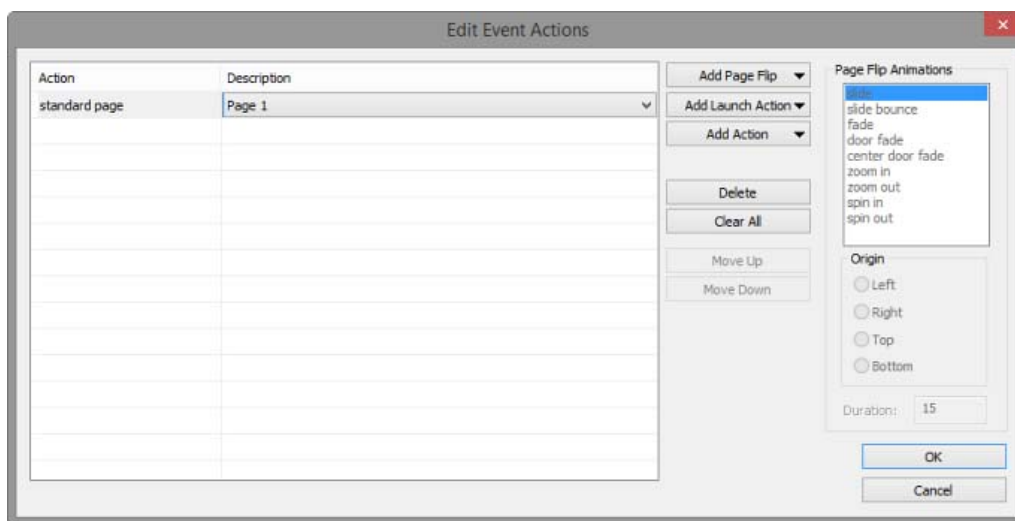

FIG. 381 Edit Event Actions dialog indicating a Standard page flip

6. Under *Description*, select the target for this Page Flip from a drop-down menu of all pages in this project (FIG. 382):
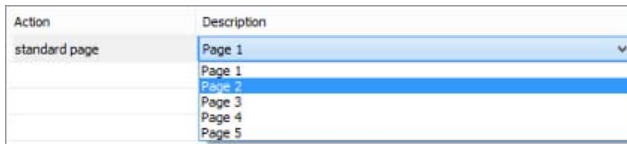
FIG. 382 Standard Page Flip drop-down menu

7.   Click **OK** to save changes and close the *Edit Event Actions* dialog.

Launch Actions

Launch Actions provide the ability to open an Application window on the panel, based on either a button press or release, a gesture, or when a specific Page is either opened or closed. In TPD5, Launch Actions are Events that can be assigned to Pages or Buttons. Events are defined via the Events tab of the Properties window.

NOTE: *In addition to Launch Actions, Page Flips and/or Actions (NetLinx commands and strings) can be assigned as Events to Pages and Buttons. See Working With Events for details.*

Launch Action Types

With a Page or Button selected in a Design View window, select an event in the Properties window (Events tab) to open the *Edit Event Actions* dialog. Click the **Add Launch Action** command button to access the *Launch Action Types* drop-down list, which allows you to select from a list of launch actions (FIG. 383):
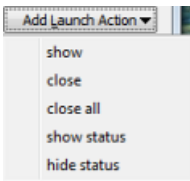

FIG. 383 Launch Action Types drop-down list

Launch Actions allow you to show/hide Application windows based on a Page or Button Event.

- **Show** - This selection displays a drop-down menu listing the Application windows in your project. Select a target Application window for the launch action to open.
- **Close** - This selection displays a drop-down menu listing the Application windows in your project. Select a target Application window for the launch action to close.
- **Close All** - This selection closes all open Application windows.
- **Show Status** - This selection displays Application Status information on the panel.
- **Hide Status** - This selection hides Application Status information on the panel.

Adding a Launch Action to a Page or Button

Launch Actions allow you to launch an Application window based on an Event associated with a Page or Button (General and Multi-State General only). In TPD5, Launch Actions are managed as Events that can be assigned to Pages or Buttons. Events are defined via the Events tab of the Properties window:

1.   Select the Page or Button to which you will assign the Launch Action in the Workspace window or Design View.

2.   In the *Events* tab of the Properties window, select the Event to which you will assign the Launch Action.

3.   Click the browse (...) button to open the *Edit Event Actions* dialog.

4.   Click **Add Launch Action** and select a launch action type from the drop-down menu (FIG. 384):
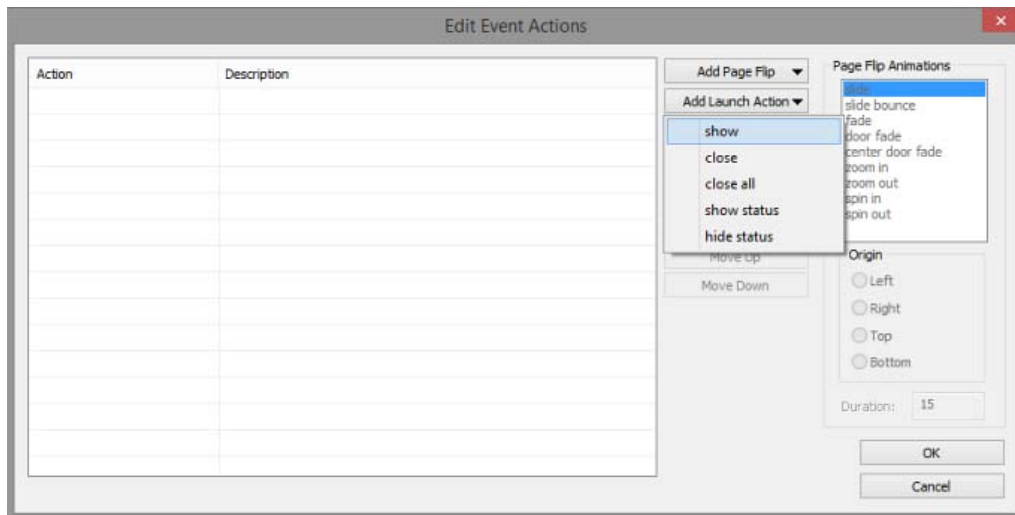
FIG. 384 Edit Event Actions dialog - Add Launch Action drop-down menu

This adds a Launch Action to the Actions list.

5. Under **Description**, select an Application for this Launch Action from a drop-down menu of all Application windows in this project (FIG. 385):
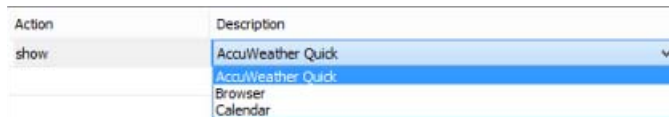

FIG. 385 Edit Event Actions dialog - Application added

6. Click **OK** to save changes and close the *Edit Event Actions* dialog.

Actions

Actions provide the ability to trigger a NetLinx Command or send a String based on either a button press or release, a gesture, or when a specific Page is either opened or closed.

Adding a Command (Action) to a Page or Button

In TPD5, *Command* actions are managed as Events that can be assigned to Pages or Buttons (General and Multi-State General only). Events are defined via the *Events* tab of the Properties window:

1. Select the Page or Button to which you will assign the Command event action in the Workspace window or Design View.

2. In the *Events* tab of the Properties window, select the Event to which you will assign the Command event action.

3. Click the browse (...) button to open the *Edit Event Actions* dialog.

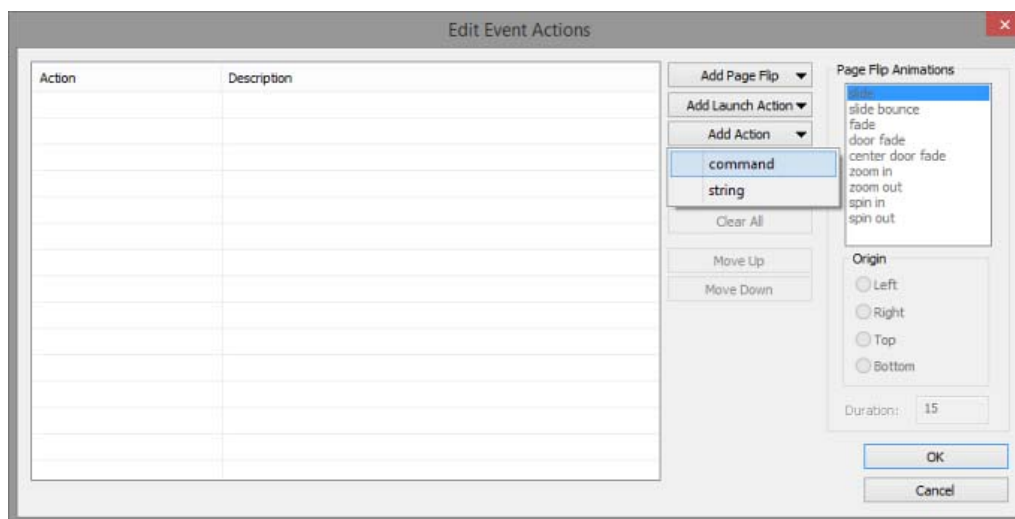4. Click **Add Action** and select *command* from the drop-down menu (FIG. 386):

FIG. 386 Edit Event Actions dialog - Add Action drop-down menu

This adds a **command** to the *Actions* list (FIG. 387):

FIG. 387 Edit Event Actions dialog - command added

5. Under **Description**, specify a command port and output string for this command:
   - Select the command port from a drop-down menu of available ports.
   - Enter the command output in the text field to the right of the Port menu.

   NOTE: *Maximum command, string and text length = 4096 characters.*

6. Click **OK** to save changes and close the *Edit Event Actions* dialog.

Adding a String (Action) to a Page or Button

In TPD5, *String* actions are managed as Events that can be assigned to Pages or Buttons (General and Multi-State General only). Events are defined via the Events tab of the Properties window:

1. Select the Page or Button to which you will assign the String event action in the Workspace window or Design View.
2. In the *Events* tab of the Properties window, select the Event to which you will assign the String event action.
3. Click the browse (...) button to open the *Edit Event Actions* dialog.
4. Click **Add Action** and select **string** from the drop-down menu (FIG. 388):
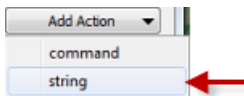

FIG. 388 Add Action - string

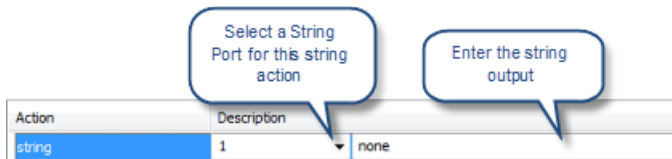This adds a **string** to the *Actions* list: (FIG. 389):

FIG. 389 Edit Event Actions dialog - string added

5. Under **Description**, specify a string port and output for this string:
   - Select the string port from a drop-down menu of available ports.
   - Enter the string output in the text field to the right of the Port menu.

   NOTE: *Maximum command, string and text length = 4096 characters.*

6. Click **OK** to save changes and close the *Edit Event Actions* dialog.

Gestures

G5 Series panels support *Gestures* for on-screen navigation. Gestures can be applied to Pages and General and Multi-State General Buttons, and provide a method of triggering Event Actions, including Page Flips, Launch Actions and (Command or String) Actions. Gestures are presented in the *Events* tab of the Properties window (FIG. 390):
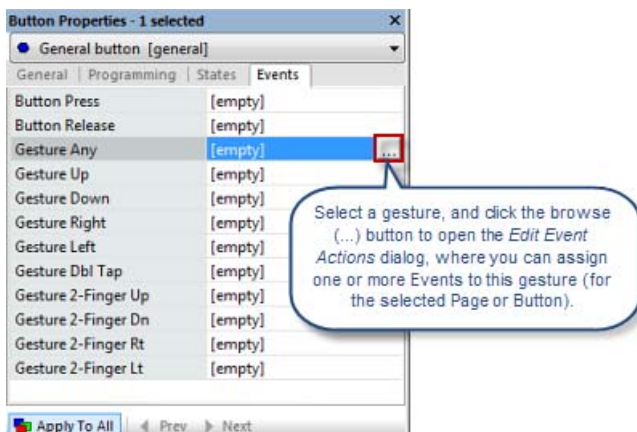

FIG. 390 Properties window - Events tab

TPD5 supports a set of Single-Finger gestures, and a set of 2-Finger gestures. Using the *Edit Event Actions* dialog, multiple event actions can be assigned to a gesture - see Assigning Events to Pages or Buttons.

Copying/Converting Gestures Between Panels

TPDesign5 supports copying and converting the properties for Gesture Controls: When copying a page between panels, or when converting between panel types that both support Gestures, the configuration of the source panel's Gesture Controls are copied to the destination panel's Gesture Controls.

Single-Finger Gestures

G5 Touch Panels support Gestures for on-screen navigation. G5 touch panels support both single- and two-finger gestures. Gestures are presented in TPDesign5 as a set of events in the Properties window (*Events* tab), and can be assigned to Pages and Buttons (General and Multi-State General only) through the *Edit Event Actions* dialog.

Supported *Single-Finger* Gestures include:

Single-Finger Gestures	
Gesture Left	A swipe across the touch panel in the left direction.
Gesture Right	A swipe across the touch panel in the right direction.
Gesture Up	An upward swipe across the touch panel.
Gesture Down	A downward swipe across the touch panel.
Double-Tap	A double-tap on the touch panel.

Using Single-Finger Gestures:

1. Place one finger on the panel.
2. Swipe (slide) the finger left or right for horizontal navigation (Left, Right Swipes), up or down for vertical navigation (Two-finger Upward, Two-Finger Downward Swipes), in a circular motion (Clockwise and Counterclockwise).
3. The Gesture ends when the finger is lifted from panel.

Two-Finger Gestures

G5 Touch Panels support Gestures for on-screen navigation. G5 touch panels support both single- and two-finger gestures. Gestures are presented in TPDesign5 as a set of events in the Properties window (Events tab), and can be assigned to Pages and Buttons (General and Multi-State General only) through the Edit Event Actions window.

Supported *Two-Finger* Gestures include:

Single-Finger Gestures	
2-Finger Gesture Left	A two-finger swipe across the touch panel in the left direction.
2-Finger Gesture Right	A two-finger swipe across the touch panel in the right direction.

Single-Finger Gestures (Cont.)	
2-Finger Gesture Up	A two-finger upward swipe across the touch panel.
2-Finger Gesture Down	A two-finger downward swipe across the touch panel.

Using Two-Finger Gestures:

1. Place two fingers slightly separated on the panel.
2. Swipe (slide) both fingers in unison left or right for horizontal navigation (Two-finger Left, Two-Finger Right Swipes), or up or down for vertical navigation (Two-finger Upward, Two-Finger Downward Swipes).
3. The Gesture ends when both fingers are lifted from panel.

Function Codes

Overview

TPD5 uses Function Codes to specify how TPD5 elements interact with and control devices on the control system. In TPD5, the term Function Codes refers to all three of the code types that can be assigned to buttons:

- **Channel Codes**: Displayed in the upper-left corner of the button, channel codes indicate the port number and the channel code associated with the button. The channel codes represent communication out of the panel to the master controller.
- **Address Codes**: Displayed in the lower-right corner of the button, address codes represent communication from the master controller to the panel, causing the panel to do something (i.e. indicate feedback, display a text string, etc).
- **Level Codes**: Displayed in the lower-left corner of the button, level codes represent bi-directional communication between the panel and the master controller (i.e. the panel can cause a change in a level setting, and a changed level setting generates feedback on the panel).

Function Codes are assigned via the Programming tab of the Properties window.

NOTE: *The easiest way to handle function codes is to create and finalize your touch panel pages and buttons (with function codes) before generating the supporting NetLinx code. That way, in case you have to change any aspect of the project (i.e. add/remove controlled equipment, test strings, graphics, etc), you can update the function code assignments in TPD5, rather than having to re-write your code.*

Power Assign

One of the big time-eating factors of creating a touch panel file is correctly setting up the channel, address and level function codes for any given button, and other properties that depend on button type. Use *Power Assign* (**Button > Power Assign**) to streamline this process.
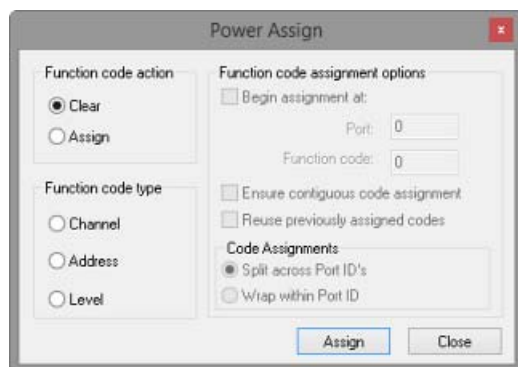

FIG. 391 Power Assign dialog

The Power Assign feature can operate on a single button or on a group of selected buttons, but does not affect Pages, Popup Pages, Sub-Pages or Application windows.

One valuable use of this feature is to establish a contiguous range of channel / address codes on a set of buttons. To accomplish this, left-click on the button in the set that should receive the first channel / address code, then by holding down the CTRL key, select each of the remaining buttons in the set in the order in which you want the channel / address codes to be assigned.

By following the instructions below and utilizing the Begin Assignment At and Ensure Contiguous Code Assignment options (available in the Power Assign dialog), one can assign channel / address codes to every button in the set with significantly fewer mouse clicks and keystrokes than would otherwise be possible.

There are two basic steps to using Power Assign (select a help topic):

1. Step One - Clear Channels
2. Step Two - Assign Codes

Function Code Assignment Options

To use take full advantage of Power Assign, you should understand the following Function Code Assignment options (in the *Power Assign* dialog):

- **Begin Assignment At:** When Begin Assignment At is selected you can specify the starting value of the Port and Function (Channel, Address, or Level) code.

  NOTE: *If you don' t check the Begin Assignment At option, the assign operation begins at the first available channel, just like standard Auto Assign.*

  The *Begin Assignment At* function is particularly useful for setting up things like numeric keypads, where you need the channel codes to begin at a specific value, since you might be using offset math in your code to process the button pushes.

- **Ensure Contiguous Code Assignment**: Select to assign all codes in unbroken numerical order.

  This is used when the channel codes have to be in order, with no breaks between them, as in a numeric keypad. In these cases, since the order is important, use CTRL-select to select the buttons in the order you want to assign the channels.

If you don' t care what channels are assigned, uncheck this option, and TPD5 will find the next available free channels, skipping used channels and continuing on until all available channels have been assigned.

In this case the values will be subject to whether or not you've set the Begin Assignment At checkbox.

● **Wrap Within Port ID**: Select to assign all codes within a single port.

Since touch panels support more than one port, you're not limited to 256 channel and address codes.

However, things like SYSTEM_CALLS are based upon all the channels coming from a single device (port). If the channels are split across ports, the feedback part of the System Call will not work.

Another thing that comes into play with being able to use multiple ports is that you may need to confine certain ranges of channel codes to a single port to take advantage of the DEFINE_MUTUALLY_EXCLUSIVE channel grouping in the code. Since MUTUALLY EXCLUSIVE works on a particular port, one must ensure that the group of buttons is entirely within a single port.

In either of these cases, check the Wrap Within Port ID option, which ensures that all channel codes assigned fall within a single port.

Limitations

The following limitations apply to Function Codes:

- The maximum number of channel codes per port is 4000.
- The maximum number of address codes per port is 4000.
- The maximum number of level codes per port is 600.
- The maximum number of ports in TPD5 is 100.

Step One - Clear Channels

It is good practice to clear the function codes before any Assign Codes operations. If you don't clear the codes first, and any of the buttons in the group you are assigning already have codes assigned, they will show up as used when Power Assign does its' checks. In some cases, this may be what you want to have happen, but usually you will want to clear them before assigning.

One powerful feature of Power Assign is the ability to clear the various function codes from the buttons:

1. In a Design View window, select the buttons that you want to include in this Power Assign (FIG. 392):

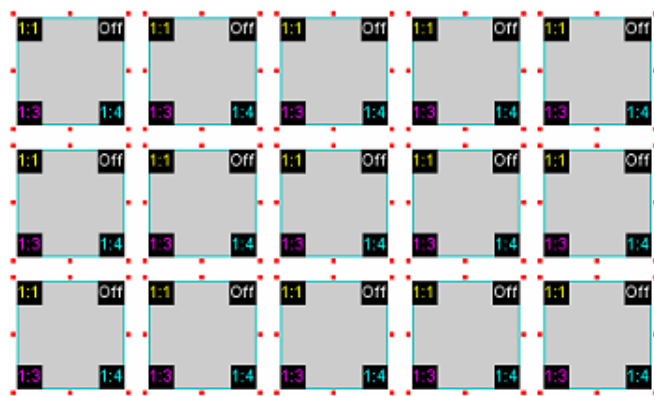**FIG. 392** Buttons selected for Power Assign

2. Select **Button > Power Assign** (or press **F8**) to open the *Power Assign* dialog.
3. Under *Function Code Action*, select **Clear**.
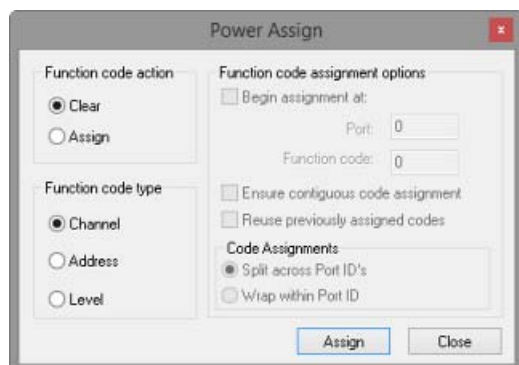4. Under *Function Code Type*, select **Channel** (FIG. 393):

FIG. 393 Power Assign - Clear Channels

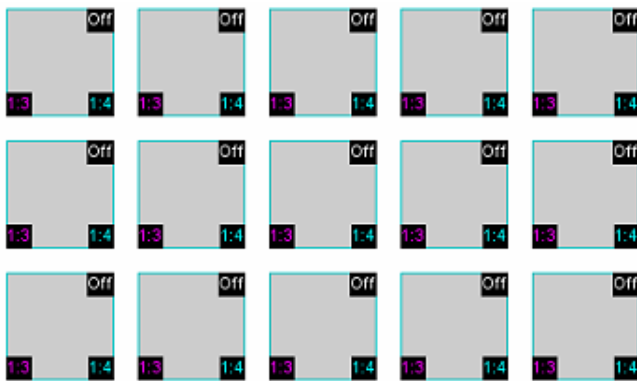5. Click **Assign**. The figure below shows the same group of buttons, with their Channel Codes cleared (FIG. 394):

FIG. 394 Buttons selected for Power Assign - Channel Codes cleared

6. Repeat this process for *Address* and *Level* codes.
   - Note that the *Power Assign* dialog is a floating dialog - you can leave it open after clicking **Assign** to do the next action: Simply select *Address* (under Function code type) and **Assign**. Then select *Level* and click **Assign**.
   - The figure below shows the same group of buttons, with their Channel, Address and Level codes cleared (FIG. 395):

FIG. 395 Buttons selected for Power Assign - Channel, Address and Level codes cleared

Step Two - Assign Codes

Next assign new function codes:

1. Select one or buttons in the Design View window.
2. In the *Power Assign* dialog, under *Function Code Action*, select **Assign**.
3. Under *Function Code Type*, select **Channel**.
4. Select one or more *Function Code Assignment* options. These options are described in the *Function Code Assignment Options* section on page 276.
5. Click **Assign**.
6. Repeat this process for *Address* and *Level* codes.
   - Since the selection is retained after any operation, if you make a mistake, it's easy to select *Clear* and click the Assign button to clear the codes. One can also simply undo the action if desired.
   - Power Assign processes the function code assignments based on the order in the selection. Therefore, if the order is important, make sure to create the selection using CTRL-Select.

Address Codes (Basic and Advanced)

When a TPD5 element's Address Port (Programming) Property is set to **0 - setup port,** the following Address Code options are available. Address Code options are separated into two categories: Basic and Advanced.

By default, when the Address Code options are displayed, the *Basic* Address Codes are listed (FIG. 396):
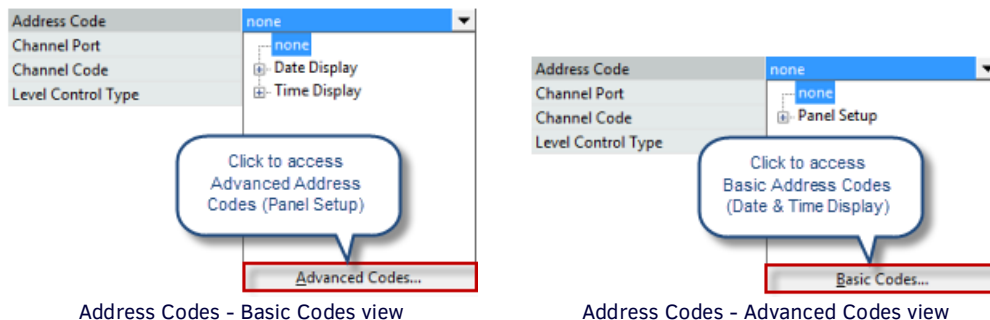
Address Codes - Basic Codes view   Address Codes - Advanced Codes view

FIG. 396 Address Codes - Basic and Advanced Views

Basic Address Codes (Date and Time Display)

Use these options to create buttons that display the current calendar date and time on the selected button (FIG. 397):
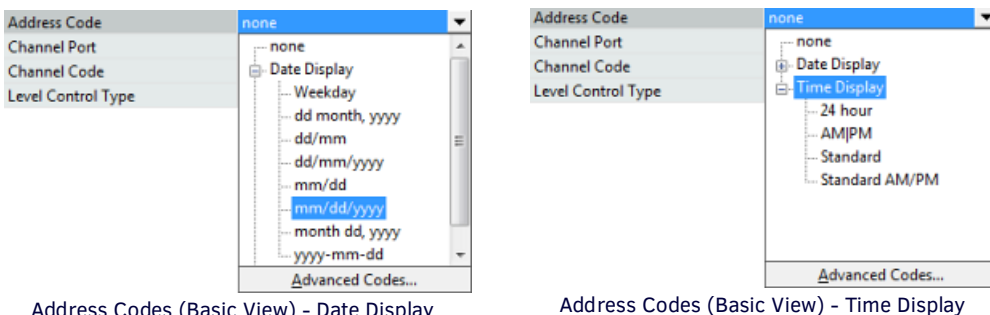
Address Codes (Basic View) - Date Display   Address Codes (Basic View) - Time Display

FIG. 397 Address Codes (Basic View) - Date and Time Display options

Date Display

Select a format to use for the date display:

Basic Address Codes - Date Display Formats	
Weekday	Displays the current date. Example: "Monday".
dd month, yyyy	Displays the current date. Example: "29 March, 2005"
dd/mm	Displays the current date. Example: "29/04"
dd/mm/yyyy	Displays the current date. Example: "29/05/2005"
mm/dd	Displays the current date. Example: "05/29"
mm/dd/yyyy	Displays the current date. Example: "05/29/2005"
month dd, yyyy	Displays the current date. Example: "March 29, 2005"
yyyy-mm-dd	Displays the current date. Example: "2005-05-29"

Time Display

Select a format to use for the time display:

Basic Address Codes	
24-hour	Displays the current time. Example: "13:30"
AM/PM	Displays either AM or PM. Example: "AM"
Standard	Displays the current time. Example: "1:30"
Standard AM/PM	Displays the current time. Example: "1:30 PM"

Advanced Address Codes (Panel Setup)

To switch the view to Advanced Address Codes, click on Advanced Codes at the bottom of the code list window. For G5 panels, the only Advanced Address Code is *Panel Setup* (FIG. 398):
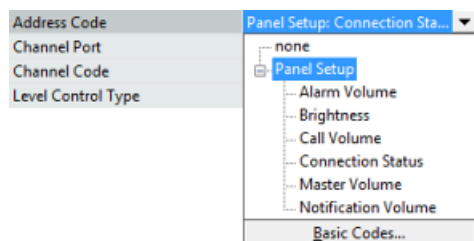
FIG. 398 Advanced Address Code - Panel Setup

Advanced Address Codes	
Alarm Volume	Displays the current volume setting for the panel's alarm sound.
Brightness	Displays the current brightness setting for the panel.
Call Volume	Displays the current call volume setting for the panel.
Connection Status	Displays the panel's current connection status.
Master Volume	Displays the current volume setting for the panel.
Notification Volume	Displays the current volume setting for the panel notifications.

Channel Codes (Basic and Advanced)

When a TPD5 element's Channel Port (Programming) Property is set to **0 - setup port,** the following Channel Code options are available. Channel Code options are separated into two categories: Basic and Advanced. By default, when the Channel Code options are displayed, the Basic Channel Codes are listed (FIG. 399):
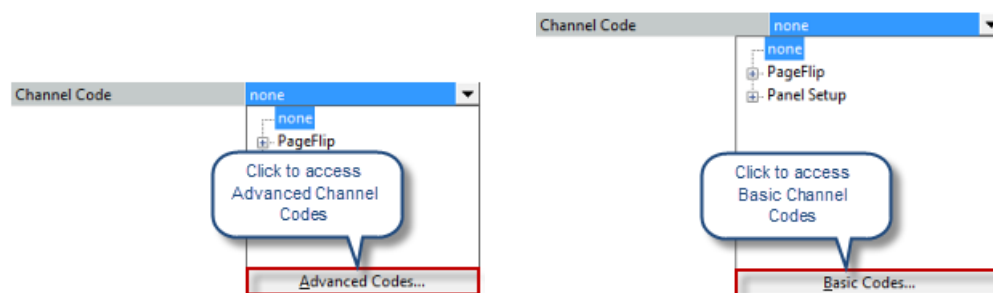

Channel Codes - Basic Codes view   Channel Codes - Advanced Codes view

FIG. 399 Channel Codes - Basic and Advanced Views

Basic Channel Codes (PageFlip and Panel Setup)

Use these options to create buttons that provide special Page Flips and Popup Drag functionality (FIG. 400):
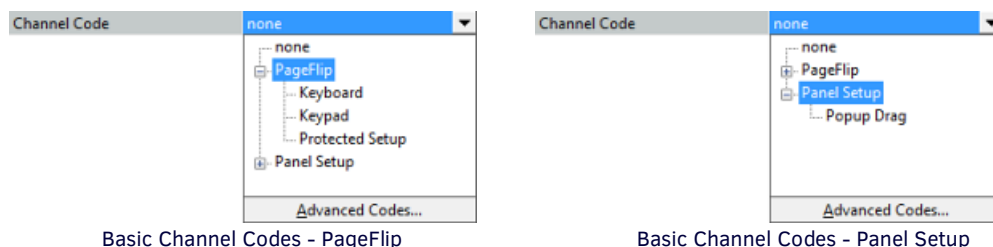
Basic Channel Codes - PageFlip   Basic Channel Codes - Panel Setup

FIG. 400 Basic Channel Codes - PageFlip and Panel Setup

PageFlip

Select a target page for a Page Flip from the list of special Pages:

Basic Channel Codes - Page Flip options	
Keyboard	Invokes the on-screen keyboard.
Keypad	Invokes the on-screen numeric keypad.
Protected Setup	Creates a page flip to the Protected Setup page.

Panel Setup

Under *Panel Setup*, click **Popup Drag** to enable the ability for users to drag popup pages around on the panel.

Advanced Channel Codes (PageFlip and Panel Setup)

The Advanced Channel Codes options are *PageFlip* and *Panel Setup* (FIG. 401):

Advanced Channel Codes - PageFlip   Advanced Channel Codes - Panel Setup
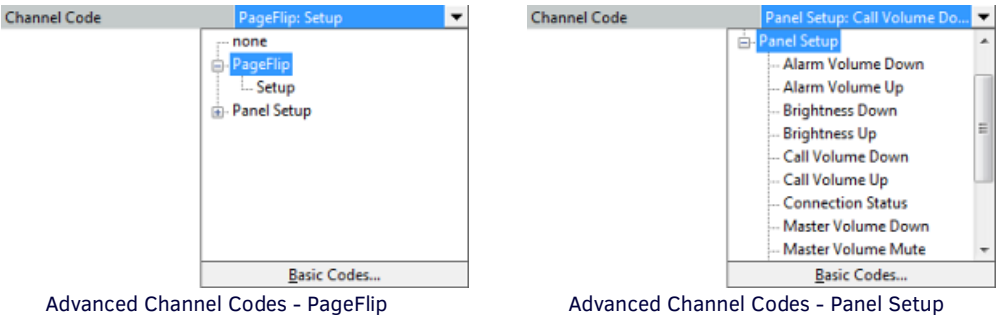
FIG. 401 Advanced Channel Codes (PageFlip and Panel Setup)

PageFlip

Under *PageFlip*, select **Setup** to create a page flip to the *Setup* page.

Panel Setup

Select a target page for a page flip from an extended list of special pages.

Level Control Type

There are three options available for Level Control Type: *none*, *absolute* and *relative* (FIG. 402):
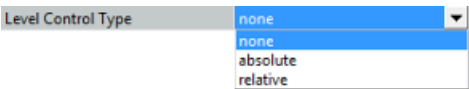
FIG. 402 Level Control Type options

Depending on the Level Control Type selected (*absolute* or *relative)*, other type-specific level control options are presented:

Level Control Options (Absolute or Relative)

With **absolute** or **relative** selected as the *Level Control Type*, additional level control options are presented (FIG. 403):

FIG. 403 Absolute Level Control Options

When a TPD5 element's Level Port (Programming) Property is set to **0 - setup port,** the following *Level Code* options are available. Level Code options are separated into two categories: *Basic* and *Advanced*.

Note that are no *Basic* Level Code options (FIG. 404):

FIG. 404 Level Code options - Basic and Advanced Views

Advanced Level Codes (Panel Setup)

The Advanced Channel Codes option are contained in the *Panel Setup* folder (see FIG. 404):

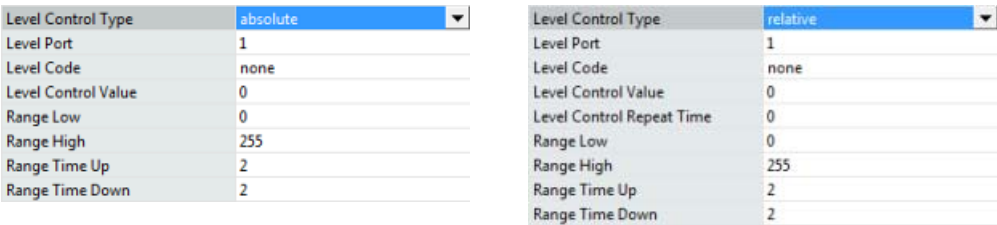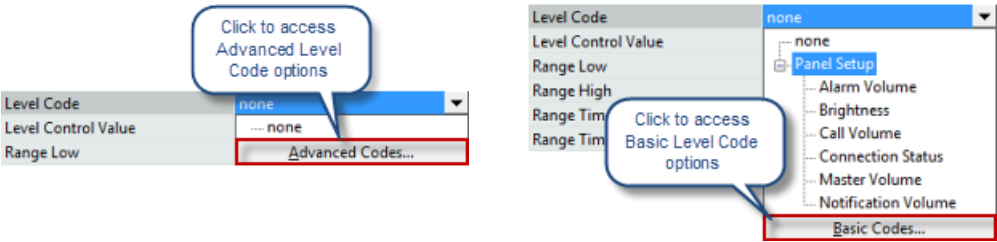| Advanced Level Codes	
Alarm Volume	With this item selected, use the *Level Control Value*, *Range Low/High*, and *Range Time Up/Down* (Programming) properties to configure the Alarm Volume level function.
Brightness	With this item selected, use the *Level Control Value*, *Range Low/High*, and *Range Time Up/Down* (Programming) properties to configure the Brightness level function.
Call Volume	With this item selected, use the *Level Control Value*, *Range Low/High*, and *Range Time Up/Down* (Programming) properties to configure the Call Volume level function.
Connection Status	With this item selected, use the *Level Control Value*, *Range Low/High*, and *Range Time Up/Down* (Programming) properties to configure the Connection Status level function.
Master Volume	With this item selected, use the *Level Control Value*, *Range Low/High*, and *Range Time Up/Down* (Programming) properties to configure the Master Volume level function.
Notification Volume	With this item selected, use the *Level Control Value*, *Range Low/High*, and *Range Time Up/Down* (Programming) properties to configure the Notification Volume level function.

Show/Hide Function Codes & State Overlay

To display Function Codes, as well as the current display state of buttons in the Design View window, select **View > Display Function & State Overlay** (or the toolbar button, or press F7).

The function codes and current display state assigned to each button are displayed in the Design View window, as shown below (FIG. 405):

FIG. 405 Show/Hide Function Codes & State Overlay

Each function code is a two-part number separated by a colon:

Port Number:Channel/Address/Level Number

NOTE: *If Display Function Codes & State Overlay is enabled, they will also be included in printed output.*

Date and Time Display Buttons

NOTE: *Time and Date buttons do not display any text when viewed in TPD5. The time or date is only visible once the project is loaded on a touch panel.*

Creating a Date Display Button

A Date Display button is a button that displays the current date on the panel. It is a read-only (non-interactive) button.

To create a date button:

1. Create a new button. The button can be of any type, but normally you would use the General button type.
2. In the *Programming* tab of the Properties window, click the **Address Port** field to enable the drop-down menu.
3. Select **0-setup port** from the drop-down menu (FIG. 406):

FIG. 406 Address Port: 0 - Setup Port

4. Click the **Address Code** field to enable the drop-down menu (*none*, *Date Display*, *Time Display*).
5. Click the plus (+) symbol next to **Date Display** to expose a drop-down menu of time display formats to choose from (FIG. 407):
6. Select the desired display format.

The following table provide visual representations of each date display style, as it appears on the touch panel:

FIG. 407 Address Code: Date Display options

Date Display Options	
Address Code Setting	**Example (as the button appears on the panel)**
Weekday	
dd month, yyyy	
dd/mm	
dd/mm/yyyy	
mm/dd	
mm/dd/yyyy	
month dd, yyyy	
yyyy-mm-dd	

Creating a Time Display Button

A Time Display button is a button that displays the current time on the panel. It is a read-only (non-interactive) button.

To create a time button:

1. Create a new button. The button can be of any type, but normally you would use the General button type.
2. In the Programming tab of the Properties window, click the **Address Port** field to enable the drop-down menu.
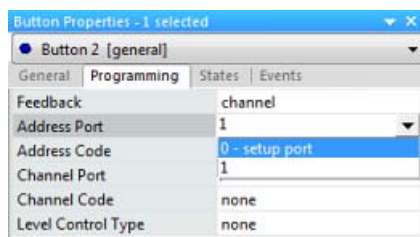3. Select **0-setup port** from the drop-down menu (FIG. 408):
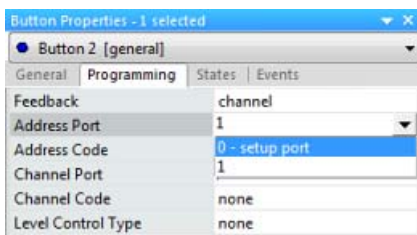
FIG. 408 Address Port: 0 - Setup Port

4. Click the **Address Code** field to enable the drop-down menu (*none*, *Date Display*, *Time Display*).

5. Click the plus (+) symbol next to **Time Display** to expose a drop-down menu of time display formats to choose from (FIG. 409):
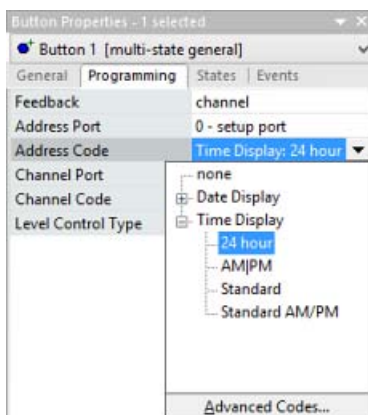
FIG. 409 Address Code: Time Display options

6. Select the desired display format.

The following table provide visual representations of each time display style, as it appears on the touch panel:
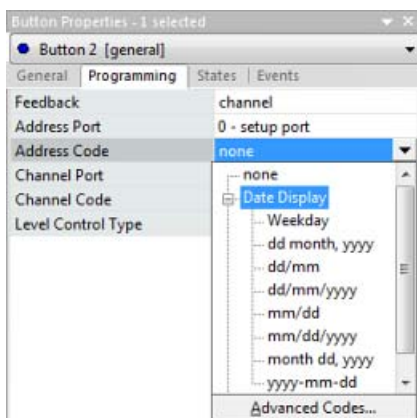
Time Display Options	
Address Code Setting	**Example (as the button appears on the panel)**
24 hour	
AM|PM	
Standard	
Standard AM/PM	

File Transfer Operations

Overview

In TPDesign5, all file transfer operations are routed through the NetLinx Master to which the target/source touch panels are connected (either via TCP/IP or Serial connection). While all file transfer operations to touch panels are managed by the Master, the files themselves are routed to the panels, where they will reside (touch panel files never reside on the Master).

There are three types of file transfer operations in TPD5 (accessible via the **Transfer** menu):

- **Send To Panel**: Sends the currently open project (*.TP5) file to a specified Master.
- **Send File To Panel**: Sends a selected project file to a specified Master, without opening the file in TPD5.
- **Receive From Panel**: Receives a project file from a Master.

Creating and Saving Connection Settings

Connection information is maintained separate from the transfer itself, so it does not need to be established/dropped each time a transfer is performed.

- Use the *Connection Settings* dialog to define and save one or more connection settings (Serial or TCP/IP).
- Use the *Connect* dialog to select a pre-defined connection setting.

Once a connection setting is specified it can be saved under a friendly name, and re-used later. You can create as many connection setting as desired.

- To recall a saved communication setting, select **Transfer > Connect** to open the *Connect* dialog, and select the desired setting from the Connection drop-down list.
- To delete a saved communication setting, select a setting (in the *Connect* dialog) and click the **Delete** command button.

Configuring a New TCP/IP Connection

1. Select **Transfer > Connect** (or click the toolbar button) to open the *Connect* dialog (FIG. 410):
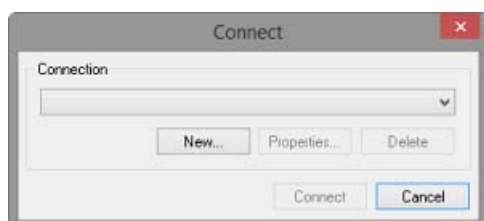
FIG. 410 Connect dialog

2. Click **New** to open the *Connection Settings* dialog (FIG. 411):

FIG. 411 Connection Settings dialog (TCP/IP Transport selected)

3. In the **Name** field, enter a unique name for this connection setting.
4. Select **TCP/IP** from the *Transport* drop-down list.
5. Enter the IP Address of the target NetLinx Master in the **IP Address/DNS Name** field.

   NOTE: *The IP Port should always be set to 1319 (default). Do not change this number.*

6. If the target NetLinx Master has authentication enabled, click the **Connection requires authentication** checkbox to enable the *User* and *Password* text fields. Enter the User Name/Password combination to save them as part of this setting.

   NOTE: *Refer to the NetLinx Studio online help for details on enabling authentication on NetLinx Masters.*

7. Click **OK** to save these settings and return to the *Connect* dialog (FIG. 412):

FIG. 412 Connection Settings dialog

Once this setting has been saved, it can be selected from the *Connection* drop-down menu (press **Connect** to establish the connection).

Configuring a New Serial Connection

1. Select **Transfer > Connect** (or click the toolbar button) to open the *Connect* dialog (see FIG. 410 on page 285).

2. Click **New** to open the *Connection Settings* dialog (FIG. 413):
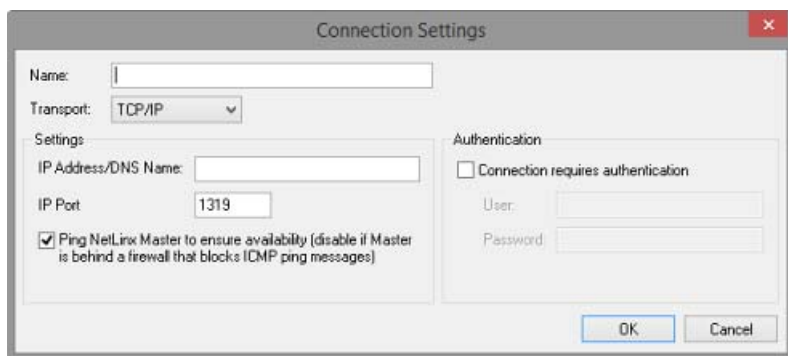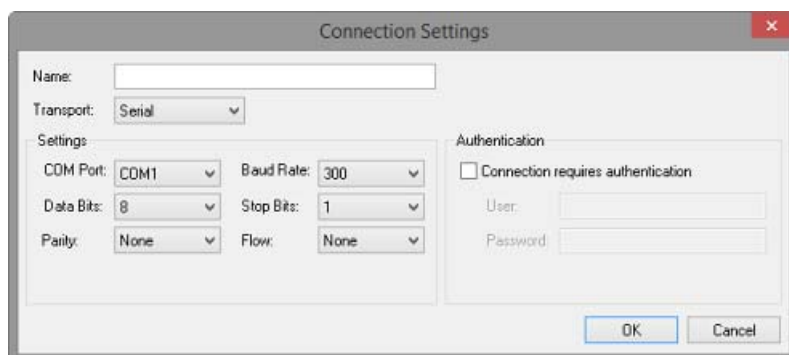

FIG. 413 Connection Settings dialog (Serial Transport selected)

3. In the **Name** field, enter a unique name for this connection setting.

4. Select **Serial** from the *Transport* drop-down list.

5. Use the Settings fields to configure the serial communication parameters.

6. If the target NetLinx Master has authentication enabled, click the **Connection requires authentication** checkbox to enable the *User* and *Password* text fields. Enter the User Name/Password combination to save them as part of this setting.

   NOTE: *Refer to the NetLinx Studio online help for details on enabling authentication on NetLinx Masters.*

7. Click **OK** to save these settings and return to the *Connect* dialog (FIG. 412):
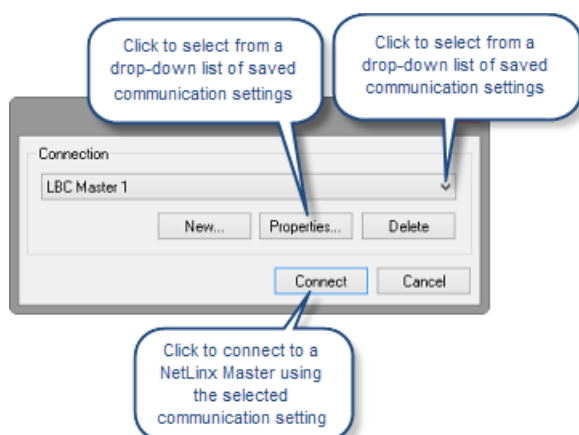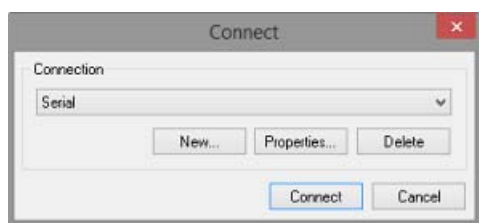

FIG. 414 Connection Settings dialog

Once this setting has been saved, it can be selected from the *Connection* drop-down list (press **Connect** to establish the connection).

Editing Settings on an Existing Connection Setting

1. Select **Transfer > Connect** to open the *Connect* dialog.

2. Select the setting that you want to edit from the **Connection** drop-down list (FIG. 415):
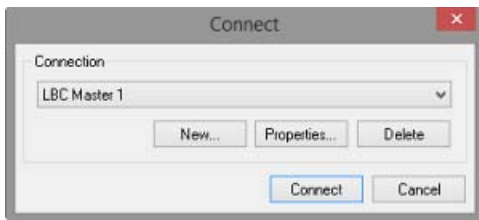
FIG. 415 Connect dialog - with a Connection Setting selected

3.   Click the **Properties** button to invoke the *Connection Settings* dialog (FIG. 416):
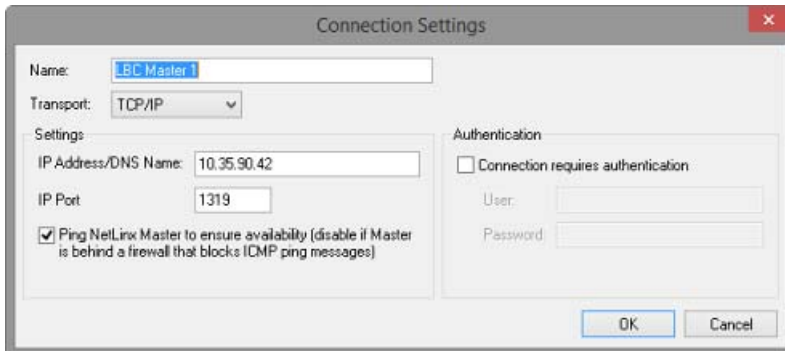
FIG. 416 Connect Settings dialog (editing an existing setting)

4.   Edit the settings as needed, and click **OK** to save your changes and return to the *Connect* dialog.

Connecting to a NetLinx Master

Once a TCP/IP or Serial connection configuration has been defined and saved, the process of actually connecting to the Master is simple:

1.   Select **Transfer > Connect** to open the *Connect* dialog.

2.   Select the appropriate connection configuration from the **Connection** drop-down menu.

3.   Click **Connect**.

The status bar reflect the status of the connection as follows (FIG. 417):

 not connected

 connected to an unsecured master

 connected to an secured but unencrypted master

 connected to an secured and encrypted master
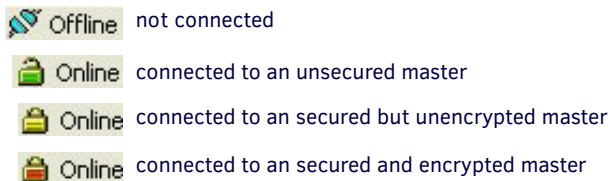
FIG. 417 Connection Status Icons

Sending a Panel File To a NetLinx Master

1.   Select **Transfer > Send To Panel**.

   NOTE: *Use the Send File To Panel option to send a project file without having to open it in TPD5.*

2.   If you are not already connected to the Master, the *Connect* dialog is invoked.
   a.   Select the appropriate connection configuration from the *Connection* drop-down list
   b.   Click **Connect** to establish the connection.

3.   Once communication is established, select **Transfer > Send to Panel**. This opens the *Send to Panel* dialog (FIG. 418):

FIG. 418 Send To Panel dialog

4. Select one or more target panels to include in the transfer.

5. Under *Options*, select transfer options as desired:
   - **Smart transfer (updated panel files only):** The *Smart Transfer* feature reduces the transfer time by only replacing those panel files that have been updated (relative to the files already loaded in the panel). Any bitmaps, sound files and fonts that all already resident on the target panel, or in your panel file on your PC (for uploads) are not included in the transfer. By default, *Smart Transfer* is enabled.
   - **Normal transfer (all panel files):** This option sends all panel files.
   - **Clear from status queue when complete**: This option clears this transfer from the Transfer Status window when the transfer is complete. By default, this feature is enabled.

6. Click **Send**.

The status of the transfer is indicated in the *Transfer Status* window.

Receiving a Panel File From a NetLinx Master

Use the **Transfer > Receive From Panel** option (or click the toolbar button) to connect to a Master and upload a panel file from a compatible G5 touch panel on that Master's bus.

1. Select **Transfer > Receive From Panel.**

   If you are not already connected to the Master, the *Connect* dialog is invoked.
   a. Select the appropriate connection configuration from the **Connection** drop-down list and
   b. Click **Connect** to establish the connection.

2. Once communication is established, select **Transfer > Receive from Panel**. This opens the *Receive from Panel* dialog (FIG. 419):

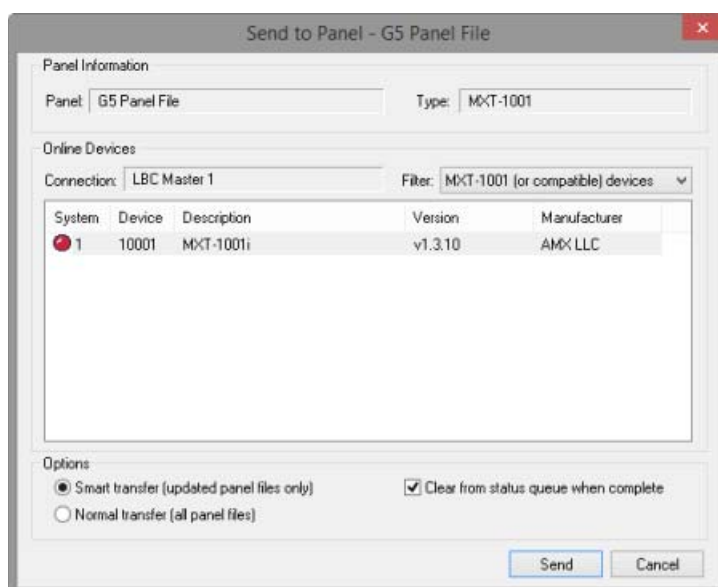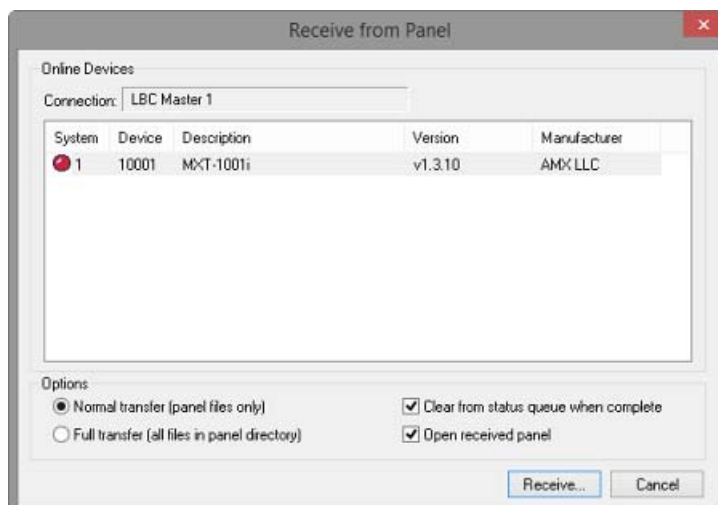**FIG. 419** Receive From Panel dialog

3. Select one or more target panels to include in the transfer.
4. Under *Options*, select transfer options as desired:
    - **Normal transfer (panel files only)**: This option receives only panel files from the source panel.
    - **Full transfer (all files in panel directory)**: Select this option to receive all project files from the source panel.
    - **Clear from status queue when complete**: This option (enabled by default) clears each transfer from the Transfer Status window when complete.
    - **Open received panel**: Select this option to automatically open the panel file once it is received.
5. Click **Receive**.
6. Select a target directory for the received files in the *Receive From...* dialog.
7. Click **Save** to start the transfer.

The status of the Transfer is indicated in the *Transfer Status* window.

Working With Colors and Palettes

Working With Colors

A key feature of TPD5 is it's ability to utilize the full 32-bit RGB color palette, which allows you to specify RGB (Red, Blue and Green) values, plus Hue, Saturation, Brightness and Opacity. The RGB palette offers millions of possible colors that can be applied to fills (pages, popup pages, and buttons), transparencies (popup pages and buttons), and text (pages, popup pages and buttons).

TPD5 also allows you to save or load custom palettes. Every color element that is not assigned either directly via an RGBA selection or the named color table will reference this palette and be affected by any changes made to it. Palettes are saved as part of the panel file. Additionally, TPD5 supports the importing of either a JASC® formatted palette file, a Microsoft® formatted palette file, or a custom palette file previously saved from within the application.

Color assignments are made through the *Colors* dialog. There are several ways to open the Colors dialog:

- When setting new button parameters, click on the *Border Color*, *Fill Color* or *Text Color* toolbar icons to open the base palette, then click **More Colors** (FIG. 420):

FIG. 420 Border Color, Fill Color or Text Color toolbar icons - More Colors option

- Click to select an existing page, popup page or button, and click **any color-oriented State property (Border Color, Fill Color, Text Color, etc.)** in the States tab of the Properties window.

The *Colors* dialog supports three methods for selecting colors (FIG. 421):

FIG. 421 Colors dialog

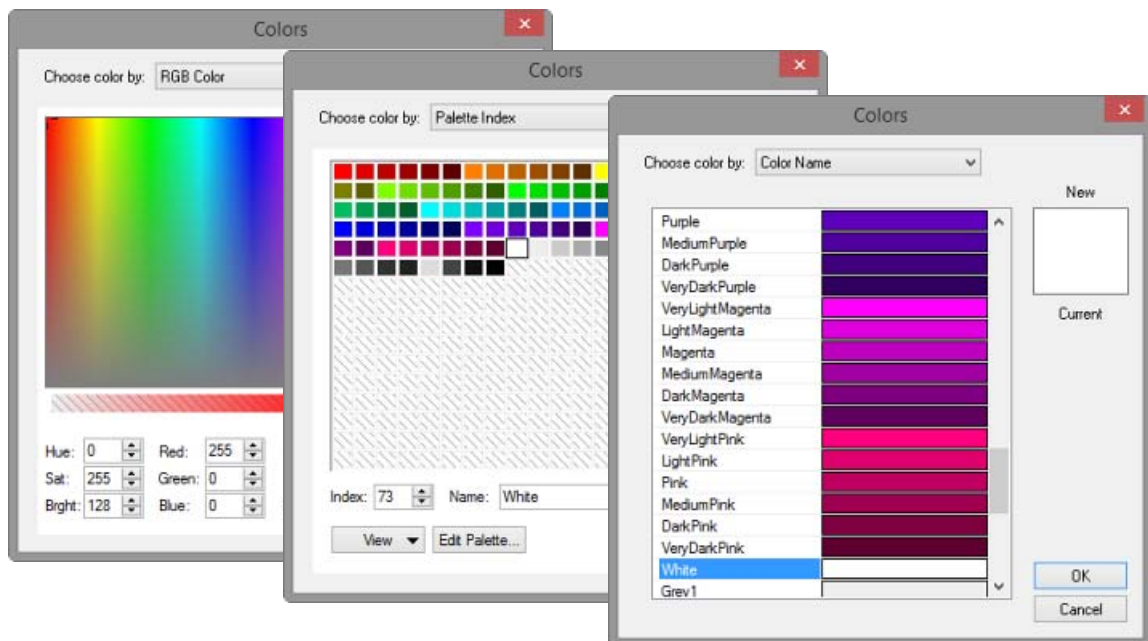1. **RGB Color**: a full-feature RGB palette that allows you specify RGB (plus Hue, Saturation, Brightness and Opacity) values numerically, or by dragging the cursor around the palette. The RGB palette offers millions of possible colors.

2. **Palette Index**: a default palette that provides the Base 88 colors (which the can be modified if desired). The Palette Index offers a maximum of 255 colors.

3. **Color Name**: a named color selection dialog based on the Base 88 color scheme. These Base 88 colors are identical to those provided in previous versions of TPDesign, and include the transparent color in position 255).

NOTE: *Neither JASC nor Microsoft palette files support transparency in the same manner that TPD5 utilizes transparency, so once imported, custom palettes cannot be reopened in another graphics package.*

NOTE: *Because the RGB Color palette supports more colors than the Palette Index, you might see a slight variation in some colors if you switch from the RGB Color palette to Palette Index.*

Gradient Fills

Gradient Fills allow you assign gradient color fills using up to 10 colors to Pages, Popups, Sub-pages, and Buttons. Gradient Fills are managed via States properties.

Gradient fills utilize a minimum of two colors to create a multi-color graded fill effect. Gradient fills can use up to ten colors.

When any *Fill Type* other than **Solid** is chosen, the colors used for the gradient are selected via the Fill Gradient Colors (State) property.

NOTE: *The transparency mask (alpha channel) color used for Pages is not supported as a gradient fill color. All other elements support the transparency mask.*

Gradient Fill Types

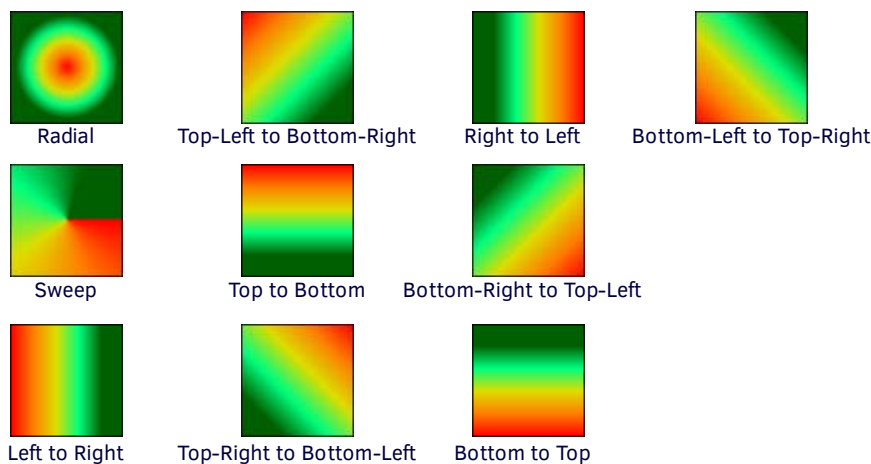TPD5 provides the following types of Gradient Fills, selected via the *Fill Type* (State) Property (FIG. 422).

| Radial | Top-Left to Bottom-Right | Right to Left | Bottom-Left to Top-Right |

| Sweep | Top to Bottom | Bottom-Right to Top-Left |

| Left to Right | Top-Right to Bottom-Left | Bottom to Top |

FIG. 422 Gradient Fill Types

Radial Fills

Radial is a radial gradient fill pattern starting at the center of a specified point blending in circular fashion out to the edges of the element. There are specific (State) properties associated with Radial gradient fills.

Note that If **Radial** is selected as the *Fill Type*, the following additional State Properties are provided:

- State Properties - Gradient Radius (see page 253)
- State Properties - Gradient Center X% (see page 253)
- State Properties - Gradient Center Y% (see page 253)

Sweep Fills

Sweep is a gradient fill pattern blending colors counter-clockwise in radial sweep fashion around the center of the element.

The starting point of the sweep is on the center-right-half of the element.

NOTE: *In order to create a complete blending of colors (i.e. without a hard transition on the right) the start and end colors must be the same.*

Selecting Colors for a Gradient Fill

1. With any gradient Fill Type selected, click on the **Fill Gradient Colors** (State) Property and click the browse (...) button to open the *Fill Colors* dialog. Use this dialog to include up to 10 colors in the gradient fill for the selected element.

2. In the *Fill Colors* dialog, click **Add** to select the first color, via the *Colors* dialog. With a color selected, click **OK** to close the *Colors* dialog and return to the *Fill Colors* dialog.

3. Click **Add** again, and select a second color via the *Colors* dialog. Click **OK** to close the *Colors* dialog and add the second color to the list in the *Fill Colors* dialog.

4. Repeat this process to add up to 10 colors to this gradient fill.

The example below indicates a gradient fill with five colors assigned (FIG. 423):.
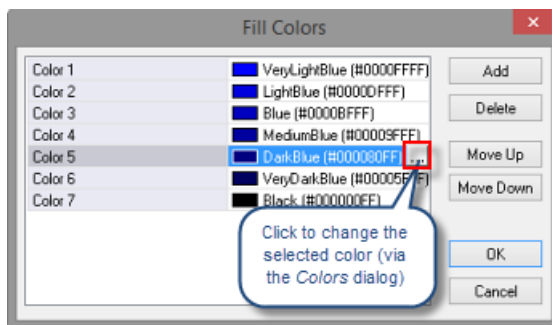
FIG. 423 Fill Colors dialog

- To change a color in this list, select the color entry and click the browse (...) button to open the *Colors* dialog, to select a different color.
- To delete a color from this gradient fill, select a color and click **Delete**.
- Use the **Move Up** and **Move Down** buttons to arrange to order of the colors as desired.

NOTE: *Regardless of type, gradient blending starts with the first color defined in the Fill Gradient Colors list and ends with the last color defined.*

5.   Click **OK** to save changes and close this dialog.

Working With Palettes

TPD5 supports creating and saving multiple custom color palettes within a project. Use the options in the Edit Palettes dialog to create and save custom palettes. Custom Palettes can be saved as *.PAL files, which can then be imported/exported for use in other TPD5 projects.

NOTE: *When you copy a button from one project into another project that is using a different palette, the pasted button will use the palette that is currently active in the project that the button is copied into (as opposed to the palette that was used to create the button). Depending on the differences between the palette in the button's source project and the palette in the target project, this can cause color shifting on the button.*

Working With Multiple Color Palettes

TPDesign supports multiple color palettes to allow easy switching between color schemes, and named palette entries. Use the options in the Edit Palettes dialog to create custom palettes and save them as *.PAL files that can then be imported/exported for use in other projects.

Creating New Palette Entries

1.   Select **Panel > Edit Palettes** (or click the toolbar button) to open the *Edit Palettes* dialog (FIG. 424).

FIG. 424 Edit Palettes dialog

By default, the palette that is currently being used for the active project is selected. Note that it is tagged as (active) in the palette selection drop-down list (in the upper-left corner of this dialog).

- Each color that is listed in the palette is considered to be a palette entry, and each palette entry is represented by an index number (1-256).
- The index numbers correlate to the slot in the palette that this color occupies.

2. Select the palette that you want to add a new palette entry (color) to, in the palette selection drop-down list.

3. Select a palette entry:
   - To add a new palette entry to the palette, select a slot with no color assignment.
   - To edit an existing palette entry, select an existing color.

4. Use the cursor in the *Color Value* chart, in conjunction with the *Hue/Sat/Bright, Red/Blue/Green*, *opacity* (and/or *Hex value*) to specify the color that you want to add to the palette.

5. Enter a description of the new palette entry in the **Name** text field. This is optional, but highly recommended since this is the name that will appear with the color in the palette when the view option is set to details.

6. Click on the **Commit** button to add the selected color and color name (if applicable) to the selected slot in the palette.

Creating Custom Palettes

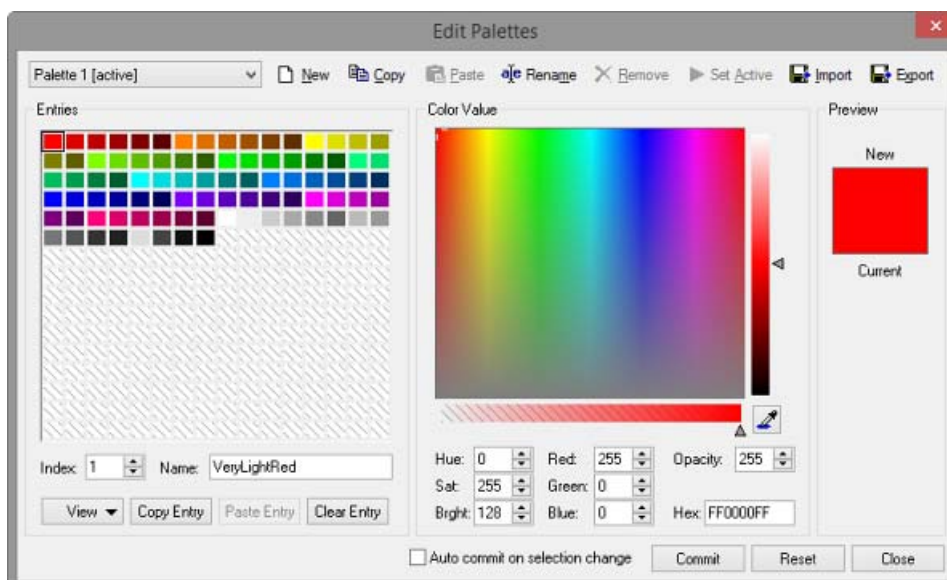1. Select **Panel > Edit Palettes** (or click the toolbar button) to open the *Edit Palettes* dialog. By default, the palette that is currently being used for the active project is selected. Note that it is tagged as (active) in the drop-down list.

2. Click the **New** button to clear the palette index of all entries. By default, the new palette is titled *Unnamed*, as indicated in the palette selection drop-down list in the upper-left corner of this dialog.

3. To add palette entries (colors) to this palette, use the cursor in the *Color Value* chart, in conjunction with the *Hue/Sat/Bright*, *Red/Blue/Green*, *opacity* (and/or *Hex value*) to specify the color that you want to add to the palette.

4. Enter a description of the new palette entry in the **Name** text field. This is optional, but highly recommended since this is the name that will appear with the color in the palette when the view option is set to details.

5. Click on the **Commit** button to add the selected color (and color name if applicable) to the selected slot in the palette.

6. Repeat steps 3 - 5 to add as many additional colors to this palette as needed.

Renaming Palettes

1. Select **Panel > Edit Palettes** (or click the toolbar button) to open the *Edit Palettes* dialog. By default, the palette that is currently being used for the active project is selected. Note that it is tagged as (active) in the drop-down list.

2. Select the palette that you want to rename from the palette selection drop-down list, in the upper-left corner of this dialog.

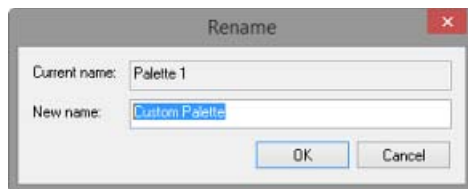3. Click the **Rename** button to open the *Rename* dialog (FIG. 425):.

FIG. 425 Rename dialog

4. Enter the new name for this palette in the text field and select **OK**.

5. The new name of the palette is indicated in the palette selection drop-down list. Note that the new name overwrites the previous name (removing the previous name from the list).

Changing the Active Palette

1. To select a different palette to use, select **Panel > Edit Palettes** (or click the toolbar button) to access the *Edit Palettes* dialog, and select from the listing of available palettes in the palette selection drop-down list, in the upper-left corner of the dialog.

2. Click **Select Active**.

3. Click **Commit**.

4. Click **Close**.

Importing Palette Files

You can import palette (*.PAL) files for use in your project via the Import option in the *Edit Palettes* dialog:

1. Select **Panel > Edit Palettes** to open the *Edit Palettes* dialog.

2. Click the **Import** button to access the *Open* dialog. Use this dialog to locate and select the desired ***.PAL** file.

3. Click **Open** to import the selected palette file and close the *Open* dialog.

4. If you desire to make the imported palette the active palette, you must select the **Set Active** button at the top of the dialog.

Exporting Palette Files

You can export palette (*.PAL) files for use in other projects via the *Export* option in the *Edit Palettes* dialog. Use this feature to save and distribute custom palettes that can be imported back into TPD5 via the Import option:

1. Select **Panel > Edit Palettes** to open the *Edit Palettes* dialog.

2. Click the **Export** button to access the *Save As* dialog. Use this dialog to save the palette to a specified directory, as a ***.PAL** file.

Copying/Pasting Palettes

Use the Copy and Paste buttons at the top of the Edit Palettes dialog to copy and paste entire palettes:

1. Select **Panel > Edit Palettes** (or click the toolbar button) to open the *Edit Palettes* dialog.
2. Select the palette that you want to copy from the palette selection drop-down list (in the upper-left corner of the dialog).
3. Click the **Copy** button.
4. Click the **Paste** button to paste the contents of the source palette into the new (target) palette.

Alternatively, you could open an existing palette and paste over the existing palette entries.

Copying Palette Entries

Use the *Copy Entry* and *Paste Entry* buttons at the bottom of the *Edit Palettes* dialog to copy and paste individual palette entries (colors):

1. Select **Panel > Edit Palettes** (or click the toolbar button) to open the *Edit Palettes* dialog.
2. Select the palette that contains the color(s) that you want to copy from the palette selection drop-down list (in the upper-left corner of the dialog).
3. Click to select the palette entry that you want to copy.
4. Click the **Copy Entry** button.
5. Click the **New** button to open a new (empty palette) or select an existing one from the drop-down list.
6. Select the slot that you want to paste the copied palette entry into and click **Paste Entry.**

If you select a slot that already has a palette entry, the copied color will overwrite the original.

Program Preferences

Setting Program Preferences

Select **Edit > Preferences** to open the *Preferences* dialog, where you can set general program preferences for TPD5.

NOTE: *Use the Customize dialog (View > Customize) to customize the TPD5 GUI. Refer to the TPD5 online help for details.*
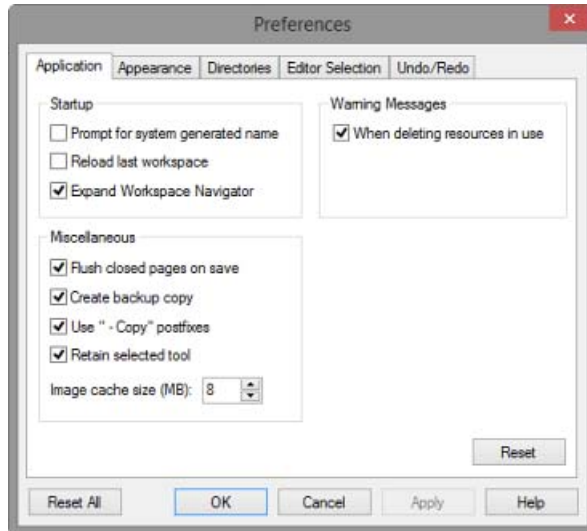
Preferences Dialog - Application tab

FIG. 426 Preferences Dialog - Application tab

The items in the *Application* tab include:

Preferences Dialog - Application tab Options	
Startup	
• Prompt for system generated name:	This option will default the checkbox for system generated filenames (in the New Project Wizard) to the checked position. With this option unchecked, system generated filenames are not generated, although the data that makes them up is still saved.
• Reload last workspace:	This option will reopen the last panel file and the pages (including popup pages) that were open when the application was closed (assuming that a panel file was open when the application was closed). *Note: If the System Page Template was opened as part of the Workspace, the Reload last workspace option will cause the application to attempt to open the (password-protected) System Page Template as part of the last opened workspace.* In this case you will be presented with the Enter Access Password dialog. Since there is not a password to unlock the file, by design the only option is to open the System Page Template as a Read-Only file.
• Expand Workspace Navigator:	This option will expand the tree structure (in the Workspace Navigator - Pages tab) on application startup. Note that this option does not take effect immediately on pressing Apply, but will take effect the next time you open a project file.
Miscellaneous	
• Flush Closed Pages on Save:	When enabled, this option flushes system memory of any pages that were previously opened but now closed, when the project is saved successfully (default = enabled).
• Create backup copy:	This option saves a backup copy of the panel file to the backup folder every time you perform a save operation.
• Use "Copy of" prefixes:	This option automatically adds the prefix "Copy of" to any pasted pages and popup pages, if a name conflict occurs. With this option unchecked, the user will be asked to resolve the name conflict.
• Retain selected tool:	This option locks the selected tool (Selection Tool or Button Draw Tool). With this option unchecked, the tool is reverted to the Selection Tool at the completion of every button draw operation.
• Image Cache Size (MB):	Use the up and down arrows to adjust the size of the image cache (default = 8 MB). The image cache size value specifies the amount of memory allocated for images used in displaying buttons (in the Design View, State Manager, and Button Preview). If an image is not found in the in-memory cache, it must be loaded again from disc, which is a much slower operation. Increase the cache size to keep more images in memory (potentially, depending on their size) to speed up loading and displaying those images.

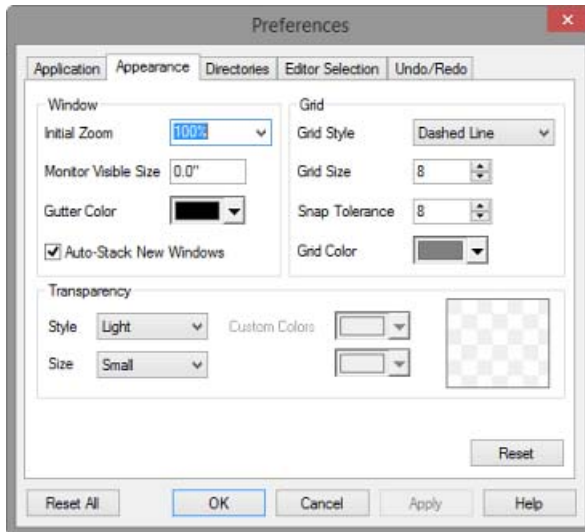| Preferences Dialog - Application tab Options (Cont.)	
Warning Messages	
• When deleting resources in use:	Click to receive a warning from TPD5 when you attempt to delete resources that are currently being used by the open project.
Reset	Click to reset all options in this tab to their default settings.
Reset All	Click to reset all options in all tabs to their default settings.

Preferences Dialog - Appearance tab

FIG. 427 Preferences Dialog - Appearance tab

Preferences Dialog - Appearance tab Options	
Window	
• Initial Zoom:	Click the down arrow to open a drop-down list of the zoom settings that can be applied as the default initial zoom setting for all new Design View windows.
	This setting does not affect the zoom setting for pages that are already open. You can also manually set the zoom factor on any page or popup page by selecting the page and using the Zoom Toolbar, where you can either use the zoom in and out icons to zoom in/out (in 25% increments), or select the zoom setting (including Fit Page, Fit Width, Fit Height) from the drop-down list.
	Note: One of the Initial Zoom options is called Actual Size. Because there are variations in dots per inch measurements among panels and among computer monitors, this option allows you to see how large a page will appear when actually displayed on the panel. To use this option, you must first specify their monitor's visible size (in the Monitor Size field - see below). This number should represent the diagonal measurement of the visible portion of the monitor or of the space occupied by a full-screen image, as opposed to the monitor's overall diagonal size. For example, most 21" monitors typically have between a 19.5" and 20" viewable area.
• Monitor Size:	Use this field to specify the size of the visible portion of the monitor that your pages will actually be displayed on (see Initial Zoom and Note above).
• Gutter Color:	The Gutter is the area around the outer edge of the Design View windows. Click the down arrow to open a palette used to set the default color for the gutter on all new Design View windows.
• Auto Stack New windows:	This setting causes all page windows to stack directly on top of the last active (selected) page window. Use this option if you are working on a resolution or zoom setting that takes up most or all of your screen area, and you want to work with multiple pages without cascading them (which could result in part of the page window being outside of the viewable area). Default = enabled.
	Note: The Auto Stack and Initial Zoom settings are retained the next time you launch the program.
Grid	
• Grid Style:	Select from the drop-down list of styles that the grid can be displayed in (Line, Dashed Line or Dots).
• Grid Size:	Use the up/down arrows to set the default grid size (measured in pixels). The range is 4 - 255, the default setting is 8.
• Snap Tolerance:	Use the up/down arrows to specify the snap tolerance for the grid. The snap tolerance represents the number of pixels within which objects in the Design View window will "snap" to the nearest grid line (applicable only when the Snap To Grid option is enabled, via the Button Selection/Drawing Tools toolbar).
• Grid Color:	Click the down arrow to open a palette used to set the default color for the grid (if enabled) on all new Design View windows.

| Preferences Dialog - Appearance tab Options (Cont.)	
Transparency	
• Style:	Click the down arrow to open a drop-down list of available checkerboard styles that can be used to represent transparency. The options are Light, Medium, Dark and Custom. If Custom is selected, the Custom Colors option is enabled (see below).
• Size:	Click the down arrow to open a drop-down list of available checkerboard sizes. The options are Tiny, Small, Medium and Large.
• Custom Colors:	If Custom was selected as the Style (see above), then you can specify a custom color combination for the checkerboard by selecting from these two drop-down lists.
Reset	Click to reset all options in this tab to their default settings.
Reset All	Click to reset all options in all tabs to their default settings.

Preferences Dialog - Directories tab
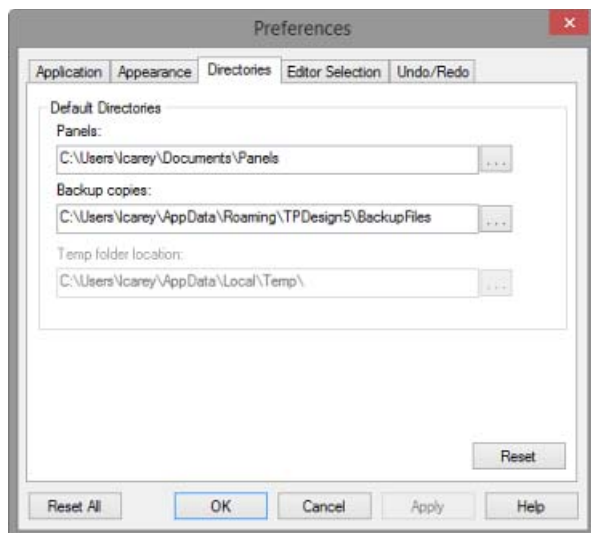
FIG. 428 Preferences Dialog - Directories tab

Preferences Dialog - Directories tab Options	
Default Directories	
• Panels:	This text box displays the directory that is currently set as the default directory for all Panel (or project) files (*.TP5). Click the browse button (...) to open the Browse For Folder dialog, where you can navigate to an alternative folder.
• Backup copies:	This text box displays the directory that is currently set as the default directory for all *.TP5 Backup files. Click the browse button (...) to open the Browse For Folder dialog, where you can navigate to an alternative folder.
• Temp folder location:	Use this field to set the location of the folder TPD5 uses for any temp files. This feature accommodates the potential for large projects which need an especially large amount of temporary disk space.
Note: If you either have a panel open or a transfer in progress the Temp Folder Location field is disabled. In these situations the Temp folder cannot be changed since it is being actively used. This field is re-enabled once all panels are closed and transfers are completed.	
Reset	Click to reset all options in this tab to their default settings.
Reset All	Click to reset all options in all tabs to their default settings.

Preferences Dialog - Editor Selection tab

The options in the *Editor Selection* tab allow you to associate external programs of your choice with image and sound files, to accommodate in-place editing of the images and sounds used in your project. Once you have associated an external program to image and/or sound editor, you can edit image and files by selecting the file in either the *Images* or *Sounds* tab of the Resource Manager and clicking the **Edit** button.
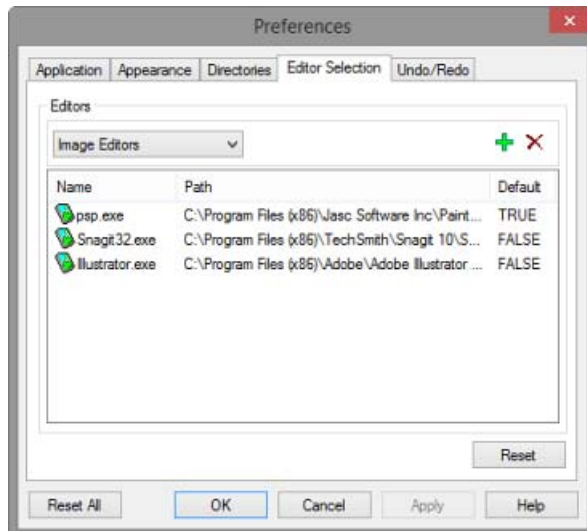

FIG. 429 Preferences Dialog - Editor Selection tab

Preferences Dialog - Editor Selection tab Options	
• Editor Type menu	Click the down arrow to select either Image Editors or Sound Editors from the drop-down list.
• Add Editor button	Click to add either an image or a sound editing program to the Editors list (de-pending on the Editor Type selected).
• Remove Editor button	With an editor selected (in the Editors list), click this button to remove it from the list, and disassociate it from image or sound files in TPD5.
• Editors table	Lists all external image or sound editing programs (depending on the Editor Type selected) that have been added, by filename (*.EXE and file path. The Default column indicates which of the programs is currently set as the default editor.
Reset	Click to reset all options in this tab to their default settings.
Reset All	Click to reset all options in all tabs to their default settings.

Adding an External Image Editing Program

Use the options in the Editor Selection tab of the Preferences dialog to associate one or more image editing programs with image files in TPD5 projects. Note that you can associate multiple editor programs with image files, but one is specified as the default image editor:

1. Select **Edit > Preferences** to open the *Preferences* dialog, and open the **Editor Selection** tab. Note that *Image Editors* is already selected in the *Editor Type* drop-down menu.

2. Click the **Add Editor (+)** button to access the *Choose Editor* dialog.

3. Click the Browse button (...) to locate and select the desired program's executable (.EXE) file, in the *Open* dialog (FIG. 430):
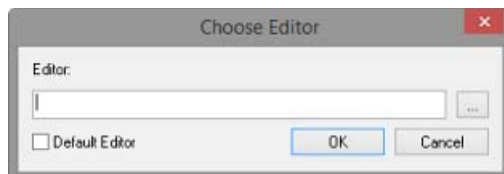

FIG. 430 Choose Editor dialog

> **NOTE:** *The first program added to the Editors list is automatically designated as the default image editor. If you add additional programs to the list, you have the option (in the Choose Editor dialog) to set the default image editor.*

4. Click **OK** in the *Choose Editor* dialog to add the selected program to the *Editors* list (FIG. 431):
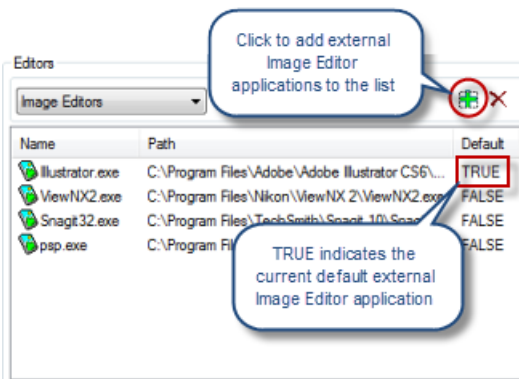
FIG. 431 Preferences Dialog (Editor Selection tab) - Image Editors list

Changing the Default External Image Editor Program

1. In the *Preferences* dialog (*Editor Selection* tab), double-click the Image Editor application that you want to set as the new default program. This opens the *Choose Editor* dialog.

2. Click in the **Default Editor** checkbox and click **OK** to save changes and close the dialog.

3. The application now indicates **TRUE** in the *Default* column in the *Editor Selection* tab.

Adding an External Sound Editing Program

Use the options in the Editor Selection tab of the Preferences dialog to associate one or more sound editing programs with sound files in TPD5 projects. Note that you can associate multiple editor programs with sound files, but one is specified as the default image editor:

1. Select **Edit > Preferences** to open the *Preferences* dialog, and open the *Editor Selection* tab.

2. Click the down arrow and select **Sound Editors** from the *Editor Type* drop-down menu.

3. Click the **Add Editor (+)** button to access the *Choose Editor* dialog.

4. Click the Browse button to locate and select the desired program's executable (.EXE) file.

   NOTE: *The first program added to the Editors list is automatically designated as the default sound editor. If you add additional programs to the list, you have the option (in the Choose Editor dialog) to set the default sound editor.*

5. Click **OK** in the *Choose Editor* dialog to add the selected program to the Editors list (FIG. 432):
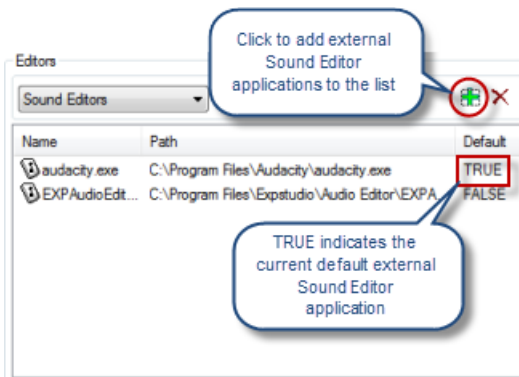

FIG. 432 Preferences Dialog (Editor Selection tab) - Sound Editors list

Changing the Default External Sound Editor Program

1. In the *Preferences* dialog (*Editor Selection* tab), double-click the Sound Editor application that you want to set as the new default program. This opens the *Choose Editor* dialog.

2. Click in the **Default Editor** checkbox and click **OK** to save changes and close the dialog.

3. The application now indicates **TRUE** in the *Default* column in the *Editor Selection* tab.
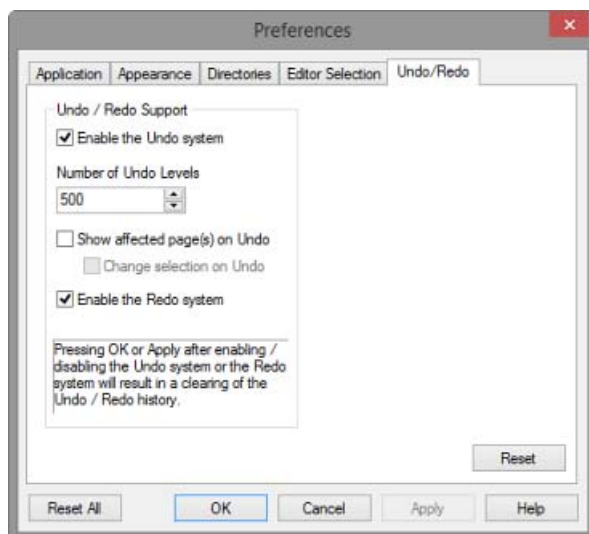
Preferences Dialog - Undo/Redo tab

FIG. 433 Preferences Dialog - Undo/Redo tab

Preferences Dialog - Undo/Redo tab Options	
Undo / Redo Support	
• Enable the Undo system:	This option enables/disables the ability to undo actions. By default, this option is selected. *Note: Pressing OK or Accept after enabling/disabling the undo system will result in a clearing of the undo/redo history.*
• Number of Undo Levels:	Use the up/down arrows to change the number of undo actions allowed (default = 500).
• Show affected pages on Undo:	With this option enabled, the program will always display the page(s) affect-ed by an undo operation, even they have been closed since that action was performed. For example, if you changed a button property on a button that exists on a page that has been closed, that page will automatically be reopened to show the button that was affected at the point that the Undo is performed.
• Change selection on Undo:	When enabled, this option will automatically change the selection so that the item(s) affected by the Undo / Redo action are selected.
• Enable the Redo system:	This option enables/disables the ability to redo actions. By default, this option is selected.
Reset	Click to reset all options in this tab to their default settings.
Reset All	Click to reset all options in all tabs to their default settings.

G4Utility (TPD4-to-TPD5 Conversion)

Overview

TPDesign5 is not backward-compatible with TPDesign4 - TPD4 project files must be converted in order to be compatible with TPDesign5 and G5 touch panels. The conversion of TPD4 projects to TPD5 projects is accomplished via the *G4Utility*, available in the TPDesign5 Tools menu.

- G4 Projects are limited to Modero X and S series panel-types (see *Supported Panel Types* below).
- Button page-flips will be migrated to the Release event on the button
- String outputs will be migrated to the Release event on the button
- Command outputs will be migrated to the Release event on the button
- Any utilized slots on states will be converted to a second bitmap
- Unsupported button-types (TakeNote, Computer Control, Joystick, List-Box) will be removed
- Unsupported borders will be removed
- Text Effects: Outline effects are not supported in G5
- Text Effects: All Drop Shadow w/ Outline effects will be converted to their standard counterparts

Supported Panel Types

TPD4-to-TPD5 conversion is limited to the following panel types:

Panel Types Supported for Conversion via the G4Utility	
G4 Panel	**G5 Panel**
MST-701, MXT-700	MXT-701
MSD-701, MXD-700	MXD-701
MST-1001, MXT-1000	MXT-1001
MSD-1001, MXD-1000	MXD-1001
MXT-1900L-PAN	MXT-1901-PAN
MXD-1900L-PAN	MXD-1901-PAN
MXT-2000XL-PAN	MXT-2001-PAN
MXD-2000XL-PAN	MXD-2001-PAN

TP5 Project File Size

Because of implementation differences between the G5 project format and G4 project format, you will likely notice a significant variation in size between your TP4 and TP5 projects. This is mainly a result of compression algorithms no longer being utilized within G5 projects due to project-format changes. There should be abundant disk space on Modero-X G5 panels to contain the panel project, but be aware that TP5 projects will be significantly larger on disk than G4 projects were.

Font Replacement

- TPDesign5 will check for projects which may be using an older version of AMX Bold (amxbold.ttf) and replace it with the updated version (amxbold_.ttf).
- The discrepancy between the two versions would cause some projects using the older version to render incorrectly either in TPDesign5 or on the touch panel, or both.
- TPDesign5 will notify the user when the project is opened that the substitution has taken place and that they should save their projects afterward.
- G4Utility v1.1 will perform the substitution as part of the conversion process.

Converting a TPD4 Project to a TPD5 Project

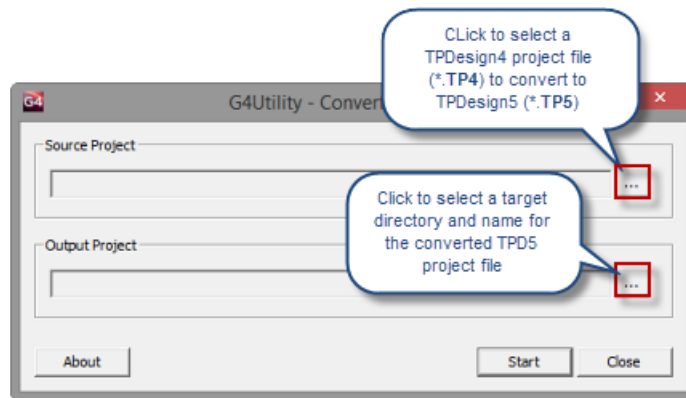1. Select **Tools > G4 Utility** to launch the *G4Utility* dialog:

FIG. 434 G4Utility

> **NOTE:** *The G4 Utility can also be launched as a stand-alone application by selecting Programs > AMX Control Disc > TPDesign5 > G4 Utility.*

2. Under **Source Project**, click the browse (...) button to locate and select the source (TPD4) project file that will be converted to TPD5, via the *Open* dialog. Select the file and click **Open** to add a TPD4 project file.

3. Under **Output Project**, click the browse (...) button to open the *Select Output File* dialog. Use this dialog to select a target directory and enter a filename for the converted project. Note that the program automatically adds the .TPD5 file extension to the filename.

4. Click **Start**.

5. Once the files have been read by the program, the *Convert G4 Project to G5* dialog is displayed. Note that this dialog indicates the G5 panel type that is targeted for this conversion, based on the selected Source Project (G4) file (FIG. 435):
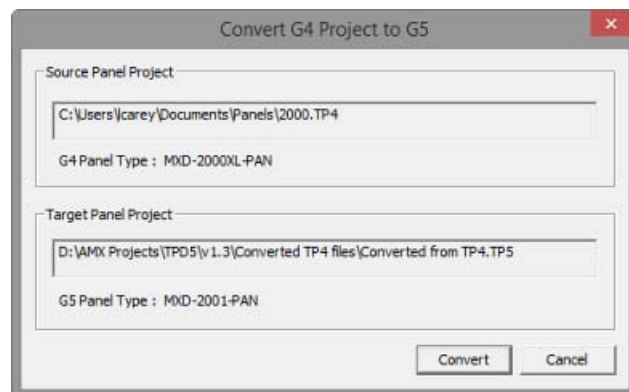

FIG. 435 G4Utility, with Source and Target Panel Projects specified

> **NOTE:** *The Conversion Tool will check the locked status of the source panel project and offer a password challenge to the user if the project is password-protected.*

6. Click **Convert** to start the conversion process.

7. The program will indicate when finished - click **OK** to close the notification dialog (FIG. 436):
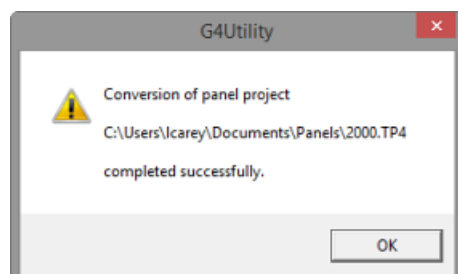

FIG. 436 G4Utility - Conversion completed successfully

Notes on TPD4-to-TPD5 Project Conversion

Bitmaps and Icons

With existing bitmap and icon properties in the source G4 Panel project, the Conversion Utility will:

- Convert existing state bitmap and bitmap justification properties to the new G5 multiple-bitmap format
- Convert any utilized state icon and icon justification properties into an additional bitmap in the G5 multiple-bitmap format.

Page Flip Conversion

G4-style page-flips will be converted from a Button property to individual page-flip actions. These actions will be added to the Button Release event on the corresponding button.

Animated Page-Flips

Animated Page-Flips will be converted to their Standard counterparts for G5 panels.

String Output Conversion

G4-style string outputs will be converted from a Button property to individual string actions. These new string actions will be added to the Button Release event on the corresponding button. The string output port value from G4 will be assigned as a property of ea. of the new string output actions.

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'"`

Command Output Conversion

G4-style command outputs will be converted from a Button property to individual command actions. These new command actions will be added to the Button Release event on the corresponding button. The command port value from G4 will be assigned as a property of ea. of the new command output actions.

G4 Properties

The Conversion Utility will remove the following deprecated G4 properties from the output project:

Deprecated G4 Properties	
States	• Marquee • Marquee Repeat • Draw Order
Buttons	• Password Protect • Above Popups • Wrap Sub-Pages • Dynamic Reordering • Feedback (blink)
Popup Pages	• Display Modal

G4 Button Types

The Conversion Tool will remove the following deprecated button-types:

- Joystick
- List-Box
- Computer-Control
- TakeNote

External Buttons

The proxy pages by which external buttons are implemented in G4 will not be carried over to the target G5 panel project.

3000 RESEARCH DRIVE, RICHARDSON, TX 75082 AMX.com | 800.222.0193 | 469.624.8000 | +1.469.624.7400 | fax 469.624.7153
AMX (UK) LTD, AMX by HARMAN - Unit C, Auster Road, Clifton Moor, York, YO30 4GD United Kingdom • +44 1904-343-100 • www.amx.com/eu/