BNCS User Guides

Atos IT Solutions and Services

PanelMan

BNCS

Contents

1 Introduction

The Integrated Panel Manager is the single user interface application that hosts all user interface panels. It provides both Title bar and menu bar areas with a client area for application panels.

It manages navigation between control panels and will bring up the appropriate application to provide a context sensitive panel to adjust any device in the system.

High level system alarms are indicated to the user by a permanently visible alarm status indicator button. This button opens the alarm overview panel for this user.

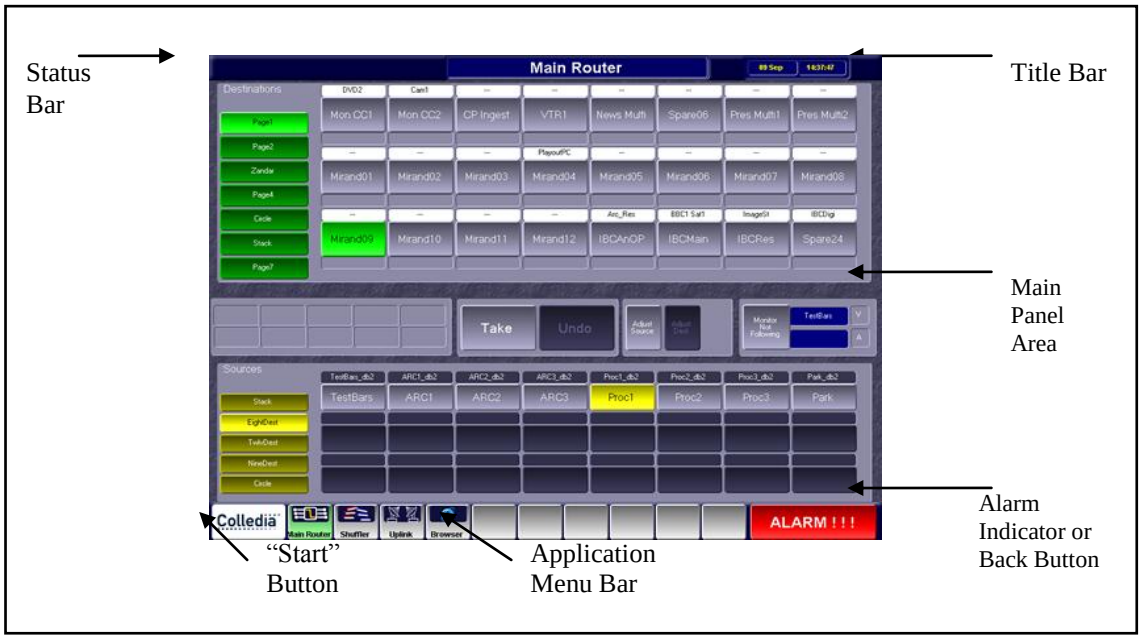


Figure 1. Panelman Overview

1.1 Brief Summary

Each UI workstation in the system will be assigned a user. The same user can be assigned to many different workstations.

Every user has a panelset assigned to them. Again, the same panelset can be assigned to many users.

The panelset itself consists of a sequence of applications that the user can recall by pressing the buttons on the application menu bar along the lower edge of the screen.

An application appears on the Menu bar with an icon and short descriptive caption. When the application is opened it appears in the main panel area of the screen. The full title of the application is displayed above in the Title bar. If the application is adjusting a system device, this name will also appear in the Title bar as a suffix to the main title, this can be an alternate title if specified in instances.xml.

BNCS

2 Panelman Details

2.1 Command Line Switches

The behaviour and appearance of Panelman can be modified by using command line switches. These switches may be combined if required.

Panelman supports the following Command Line Switches:

2.1.1 Size

The default screen size for Panelman is 1024x768

To set the screen size to 1280x1024 add the following switch:

Panelman.exe /size=1280

Versions of panelman before 4.5 supported only the following panel sizes (see section 2.2 for enhanced way this now works):

Command Line Option	Panel Size
1024	1024 x 768
1280	1280 x 1024
1366	1366 x 768
1600	1600 x 1200
1920	1920 x 1080
2048	2048 x 768
3840	3840 x 2160

2.1.2 Load

Panelman loads the pm_blank page on startup

For ease of Panel testing during development any panel can be displayed immediately when Panelman starts by using the following switch:

Panelman.exe /load=examples\rtr

2.1.3 Instance

When using the /load switch this can be used to set the instance (targeting) of the panel. This switch can *only* be used in conjunction with the /load command line parameter.

Panelman.exe /load=examples\rtr /instance=myInstance

2.1.4 Silent

Panelman loads many Panel Applications that are written in C++. It is possible that a coding error may cause a crash. As this DLL has been loaded by Panelman.exe it will crash

when this occurs. The Panelman code includes a C++ exception handler. This means that it will return can error code to the calling application, Workstation Manager.

By default when this occurs Panelman will display a message box.

Figure 2. Panelman Fatal Error

This message box displays the name of the last Panel Application that Panelman loaded. It is likely that this application was the one that crashed. However as there may be many other script DLLs running in the stack it is also possible that the crash happened elsewhere.

If required this message box can be suppressed by using the following switch:

Panelman.exe /silent

2.2 The Panelman Folder and Assets

IMPORTANT NOTE:

Panelman resources before version 4.6 had the size of the dialogs fixed and the resources required to support that size had the size embedded in the filename – e.g. pm_title_1024.bncs_ui

From version 4.6 "size" is merely a name of the folder to look in for the multiple resources that panelman needs and the numeric suffix is no longer required.

To migrate from pre-version 4.6:

- Create a folder of the same name as the command line /size parameter e.g. "1024"
- Copy all the panelman resources with that numeric suffix into the new folder (e.g. pm_title_1024.bncs_ui)
- Remove the numeric suffix from the ".bncs_ui" files (note that you don't need to do this for the image resources) – so "pm_title_1024.bncs_ui" becomes simply "pm_title.bncs_ui".

From this version the /size command line parameter does not have to refer to an actual size or numeric value but can be a name – this then just becomes the name of the folder to find the panelman resources. E.g. this is valid:

```
panelman.exe /size=myPanelSize
```

and panelman will look for it's resources in the folder:

```
%CC_ROOT%\%CC_SYSTEM%\panels\panelman\myPanelSize
```

The Panelman application has a number of bars and areas (see Figure 1. Panelman Overview). Each of these dialogs is external to the Panelman application. The system/panels/panelman directory contains sets of these dialogs – those specific to a particular panelman size are split-out into individual folders.

Panelman resource	Notes
panels\panelman\<folder>\pm_title.bncs_ui	Title bar
panels\panelman\<folder\pm_app.bncs_ui	Application buttons
panels\panelman\<folder>\pm_status.bncs_ui	Status area
panels\panelman\<folder>\pm_blank.bncs_ui	Blank panel –also sets the size/position of the main panel widget stack
panels\panelman\ pm_start.bncs_ui	The Windows-style start menu
panels\panelman\ pm_popup_pin.bncs_ui	Popup dialog for entering engineering menu PIN
panels\panelman\ pm_wait.bncs_ui	Optional dialog that's shown whilst a UI is loading
panels\panelman\pm_confirm_exit.bncs_ui	Option dialog that's shown when panelman is about to exit

The dialogs contain named controls that Panelman expects to receive notifications from. Panelman sets their properties for example to indicate a highlight or to change a caption.

It is possible to change the appearance of Panelman by making modifications to these dialogs in the Visual Editor.

The set of files within the release support the following screen sizes:

- 1024x768 (default)
- 1280x1024

The screen size that Panelman uses is determined by a command line switch – see section

As Panelman needs to set control properties or receive event notifications it expects to find certain controls with pre-defined names. These defined controls appear in this section as **[control_name]** and should not be renamed or removed from the dialog.

2.2.1 pm_app

The Menu bar contains the following controls:

[start] - The "start" menu button takes you to the start menu – see section 5

[1001]- button loads panel 1 in the panelset

[1002]- button loads panel 2 in the panelset

…

[1010]- button loads panel 10 in the panelset (last button on a 1024x768 screen)

[alarm] or **[Back]** – The alarm indicator button or the go back to previous panel

The alarm button is a DLL script that loads **/panels/panelman/pm_alarm.dll**. This script is responsible for the appearance and behaviour of the alarm button – see section 4 (Alarm button).

Note: Larger screen sizes will have more buttons than shown above eg. 1280x1024 has 13 buttons. Panelman obtains the number automatically from the design of the bncs_ui file.

2.2.2 Popup Application Selection

From Panelman version 4.5.13 can have a different app bar mode – where paging is replaced by popup versions. This is configured per panelset:

```
<?xml version="1.0" ?>
      <panelsets>
            <set id="Demo21" level="user" popup="true" >
```

This requires additional resources either:

- Generic app buttons row(s)
- Panelset specific app button row(s)

The generic buttons are (up to) 9 extension (ext) files

e.g.

pm_app_ext1.bncs_ui

This is the first extension for the 1024 sized panelman (which has 10 app buttons). This dialog should have a single row of buttons numerically following on in sequence from the buttons on pm_app.1024.bncs_ui. i.e. buttons 1011+

This theme continues up to pm_app_ext9.bncs_ui which has 9 rows of buttons.

From version 4.5.20 if you include the first row of buttons – duplicating what's already displayed on the app bar then the right-hand "paging" button turns into a "Less" button and can be used to dismiss the popup dialog.

Popup panelsets can also have different versions depending upon the panelset in use.

The panelsets-specific versions are of the format:

pm_app_popup_<panelset>.bncs_ui

e.g.

pm_app_popup_frontDesk.bncs_ui

There are not multiple versions of this file since it's directly related to an individual panelset and the number of buttons displayed does not change dynamically. If the panelset grows and requires more buttons then just add them to the .bncs_ui for that popup panelset.

2.2.3 pm_title

The Title bar contains the following controls:

[title_caption] – The title display for the current panel eg "ARC Control – ARC1"

[title_time_local] [title_date_local] [title_time_utc] [title_date_utc] – A display of the current time and date of the workstation. UTC or local time as indicated.

2.2.4 pm_status

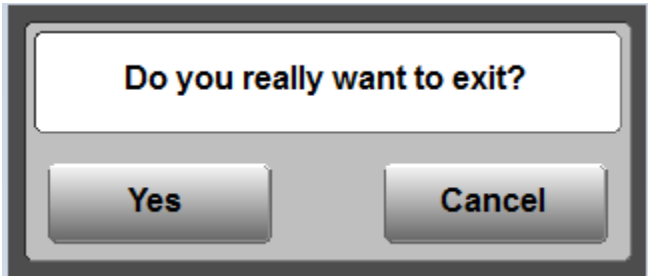
There are no controls on this dialog. It is expected that individual delivery projects might use this as a permanently visible area to display status information. This dialog can be edited as required, above is a typical example. It should only be resized/repositioned as part of a complete set of changes to all the dialogs within Panelman.

2.2.5 pm_start

The "start" menu contains the following controls:

[start] – A button with the BNCS logo that when pressed will open a popup showing the "start" menu. This shows information about the workstation - see section 5 for details

2.2.6 pm_confirm_exit



The "confirm exit" menu allows the user to confirm that they want to exit before the application actually closes. If this panel does not exist the application will automatically exit. The controls that need to be on the bncs_ui are:

[confirm_yes] – A button that when pressed confirms that the user wants to exit panenman.

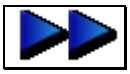
[confirm_cancel] – A button that when pressed does not exit.

2.2.7 pm_blank

The blank dialog contains a single control that says "Colledia Control". It is simply a backdrop that is loaded before any panels are started. This could be modified to suit any project delivery requirements. The size and position of this dialog should be the same as all the individual panels that are in use within this system.

2.2.8 Image Assets

	BNCSlogo.png	Logo for start menu button
	close.png	Icon for the quit button on the start menu
	restart.png	Icon for the restart button on the start menu

	more.png	Icon for the more panels button on the Menu bar
	Icon_grid.png	Icon for the popup menu bar
	icon_less.png	Icon to dismiss the popup menu bar (from 4.5.20)
	default.png	Default icon for panels on the Menu bar to be shown when no custom icon is specified
	pm_appbar_1024.png pm_appbar_1280.png pm_status_1024.png pm_status_1280.png pm_title_1024.png pm_title_1280.png	Background images for each dialog

2.2.9 pm_wait

Simple dialog to show whilst loading V4.5 dialogs – this dialog just confirms that the user is navigating to another panel and should wait. A typical implementation should simply say "Loading…".

3 Panel Applications

3.1 Location of Panel Applications

All panels are located in **%CC_ROOT%/%CC_SYSTEM%/panels/**

Sub-folders below this path are used to store specific types of panel as shown in the tree below.

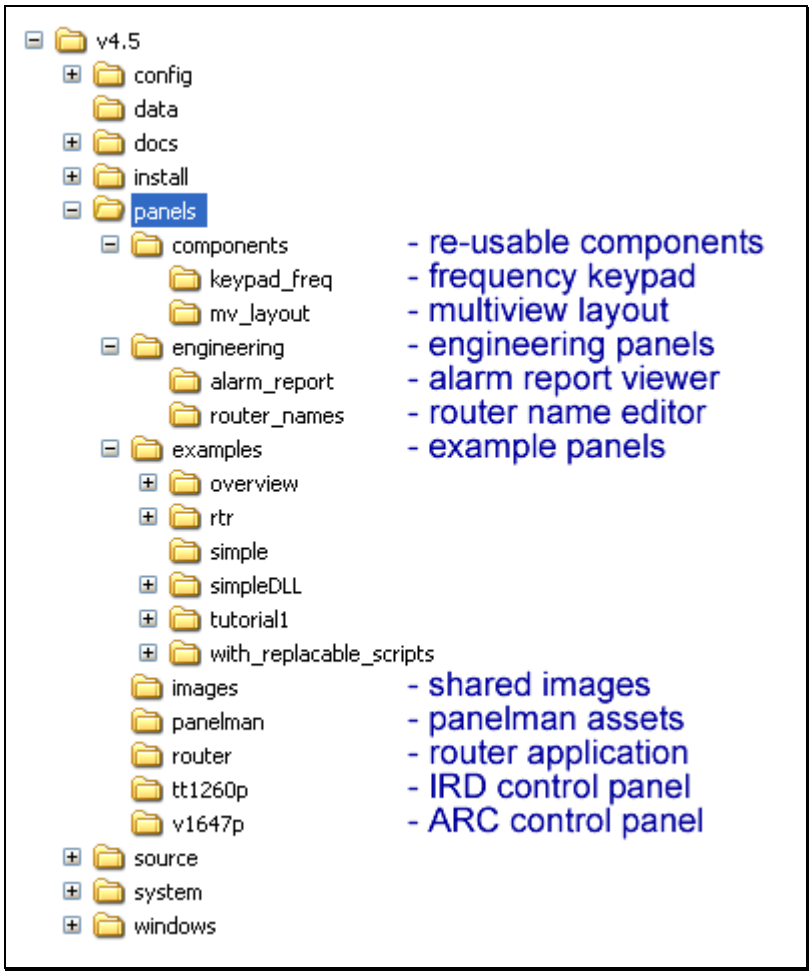


Figure 3. Location of Panel Applications

Each panel has its own folder eg. **/panels/router/** or **/panels/v1647p/**

This folder contains the main panel file for the application:

- For a non-scripted panel the main panel file might be **`v1647p.bncs_ui`**
- For a scripted panel the main panel file might be router.dll

The rule is that if a file **`foldername.dll`** is found then this will be loaded otherwise a **`foldername.bncs_ui`** file will be loaded instead. The dll will then load other bncs_ui files from the folder.

The folder can also contain all of the assets associated with the application:

- The icon to be shown on the app bar in Panelman eg. icon_router.png or indeed in **/images**

- Any png image files that are specific to the this panel – common image files such as **`arcmode1.png`** should be stored in the shared location **/panels/images/**

Panels can be located in sub folders to allow more logical grouping. From the example above the **/panels/engineering/** folder contains the alarm_report and router_name applications

3.2 How to add a panel

Having created the panel using the Visual Editor and Script Wizard the following steps are required to make the panel appear.

3.2.1 Copy the files to the Server

Create a new panel folder on the server under **`%CC_ROOT%/%CC_SYSTEM%/panels/`** or in a sub folder if appropriate. Copy the bncs_ui, dll and png files to the new folder.

3.2.2 Add an item to Applications in the Config Editor

Set the following properties for the application:

Name	The unique identifier of the application	router or for an application in a sub-folder `examples/rtr`
Title	The name that will appear in the Title Bar above the panel	`Main Router`
Caption	The name that will appear on the button in the Menu Bar - keep it short	`Main Router`
Icon	The name of the icon to display on the button in the Menu Bar – this must be the name of the png file in the panel folder but without the extension or the icon can be in the images folder with the appropriate path ie/images/icon_router	`icon_router` or if no icon has been created you can set this to use the default icon default
kill	Sets the period in milliseconds that the application is in the background for before being terminated. Only valid for v4 and external applications.	`60000 for 1 minute`

The application then can be added to the necessary panelsets

4 The "Alarm" Button

The Alarm button is external script pm_alarm.dll that is loaded (if necessary) as part of the Menu Bar. The button should normally show the summary of alarm status that is relevant to the particular User or Workstation. This is normally indicated by a button that displays:

- "Alarms OK" in the stylesheet state **`alarm_ok`**
- or "Alarm !!!" in the stylesheet state **`alarm_alarm`**

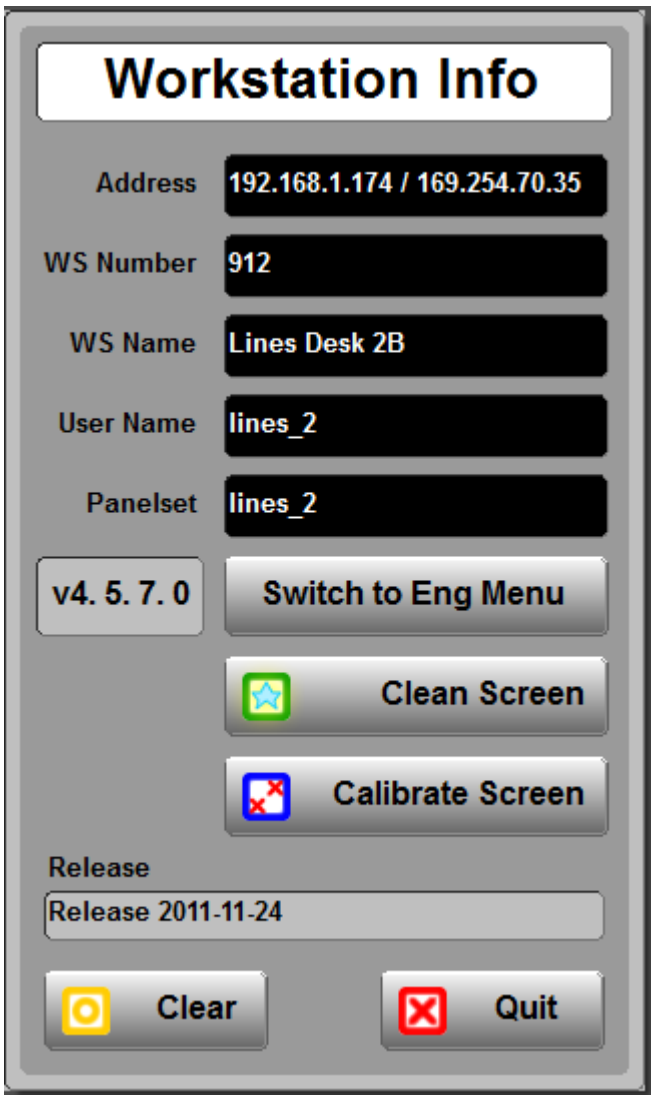
When pressed the button will Navigate to the specific Alarm Panel Application that is again relevant to the particular User or Workstation.

At this stage it is not defined how to determine the Alarm status or Alarm Panel that is relevant to any particular combination of User or Workstation. This business logic is wholly within the domain of each delivery project. For this reason any specific coding is outside the scope of Panelman.

An example `pm_alarm.dll` project is included in the BNCS v4.5 distribution.

5 The "Start" Menu

The "Start" Menu appears when the BNCS logo button is pressed on the Menu bar



5.1 Workstation Information

The popup shows the following information about the Workstation:

- The Workstation IP address
- The Workstation number
- The name of the Workstation
- The current User on the Workstation
- The Panelset that is currently loaded
- The Release number (obtained via Workstation tracker from its device ini)

5.2 Switch to Eng Menu

Each User within the System has two Panelsets defined for their use. The main Panelset is loaded automatically by Panelman as the Workstation starts. It is possible to switch to the engineering menu by pressing the "Switch to Eng Menu" button.

A dialog requesting the engineering PIN number will appear. If entered correctly the main Menu bar will switch to show the engineering Applications.

To return to the main Menu press the same button - it now shows "Switch to Main Menu". No PIN number entry is required to switch back to the main Menu. However the PIN will need to be entered again for each time that the menu is switched to engineering.

From version 4.5.13 the PIN is stored in a file (g_pnlman.ini) and must be generated using the PIN generator utility pinGenerator.exe

5.3 Clean screen

This will display a screen where the touch element is disabled that includes a grill for the setting of touch screen frequency and phase

5.4 Calibrate Screen

This will call the touch screen manufacturers calibration software as defined in workstation_settings.xml see Section 8

5.5 Quit

This button may popup a confirmation box before closing Panelman and will take you back to Workstation Manager.

5.6 Clear

This will cause Panelmanager to unload all its dlls

6 The Adjust Mechanism

6.1PA – navigateAdjust()

Panelman can "Adjust" an Instance within the system.

Eg. `navigateAdjust("ARC1")`

On receiving a navigateAdjust() message Panelman does the following:

- Get the Devicetype of the supplied Instance – **`ARC1`** is of type **`V1647`**
- Get the Typehandler of this Devicetype – **`V1647`** is handled by Application **`V1647P`**
- Check if Application **`V1647P`** is already loaded and load if required
- Switch **`V1647P`** to the top of the Panel stack
- Target the panel to `ARC1` by sending the Instance name – see section 6.5
- Set the title bar to as specified for the application, appending the instance name, or the alternate ID if one is available.

6.2PX – navigateExecute()

Panelman can "Execute" an Application within the system.

Eg. **`navigateExecute("router")`**

- On receiving a navigateExecute() message Panelman does the following:
- Get the title of the supplied Application – **`router`** has a title of **`Router Control`**
- Check if Application **`router`** is already loaded and load if required
- Switch router to the top of the Panel stack

If the Application is not found in **`applications.xml`** then the name supplied will be used as a title

Any parameters that are required to be passed to either external or v4 applications must be separated from the application name by a comma.
`navigateExecute``("v4_panel,arc1,123")` will start v4_panel passing arc1 and 123 to the application.

6.3 PP – panelman Panelset

This command can be used to set an arbitrary panelset on panelmanager.

The command is of the format:

PP <panelset>

If the <panelset> command is omitted then panelmanager reverts to the normal (as at startup) panelset.

6.4Application Paths when using navigateExecute

Specify the path relative to **`%CC_ROOT%/%CC_SYSTEM%/panels/`**

Example:

- To run **`%CC_ROOT%/%CC_SYSTEM%/panels/examples/router`**

- Send **navigateExecute("examples/router")**

6.5Targeting panels

Panelman calls the setCaption method of the panel to send the details of which instance it is being targeted to.

setCaption("instance=ARC1")

The panel container receives this target and passes the information on to all controls and scripts within the panel.

6.6Usage of UDP Multicast Messages

Panelman also listens for Multicast UDP messages. Panelman messages are sent on port 18**xxx** on multicast address **225.x.y.z**

The messages within <cc_packet> that Panelman handles are:

<cc_navAdjust instance="ARC1" workstation="11"/>

This means adjust the Instance ARC1 on Workstation 11

<cc_navExecute application="ARC20P" param="ARC1" workstation="11"/>

This means execute the application ARC20P with an Instance target of ARC1 on Workstation 11.

6.7Usage of InfoDriver 998

Panelman registers with the slot corresponding to this workstation on InfoDriver 998.

If it receives a message from this slot it checks the first two characters to see if it is a PA or PX message.

Eg. **IW 998 'PA ARC1' 33**

This message is interpreted as **navigateAdjust("ARC1")**

7 V4 & External applications

Panelman can run v4 or external application as well as hosting v4.5 panels, how these are treated is set in applications.xml.

When a non v4.5 panel is called the main client area of the panelman is hidden to allow the new application to appear on top and still allow the menu bar to appear on top. Panelman will attempt to bring any existing instance of an application to the front, this will only work if the main window caption has not been changed, if the application is not already running it will be started.

Once panelman has determined that an application is no longer visible, because another application has been brought to the front, if the kill time has been set it will start a timer, once this time has elapsed if the application is not currently the active window panelman will attempt to close it.

8 Touch screen calibration

If there is an entry in "workstation_settings.xml" for this workstation as shown below, then that application will be run.

```
<setting id="touchscreen" value="control monmouse.cpl"/>
```

The entry can be an application or a standard windows control panel application run using “control somename.cpl”

If you have an application named “control” this may get run instead, so in that case replace control with “rundll32.exe shell32.dll,Control_RunDLL”.

The application can also be a normal external application as shown below.

```
<setting id="touchscreen" value="C:\Program Files\Tsharc\hwincal.exe"/>
```

The working directory for the application will default to the directory the application is in, this can be overridden with the following entry.

```
<setting id="touchscreen_work_dir" value="c:\temp"/>
```

If nothing is set in “workstation_settings.xml” to maintain backwards compatibility workstation mangaer will try and launch monmouse.cpl and mtstouch.cpl.

9 Configuration Files

The appearance and behaviour of Panelman is configured by the following system configuration tables:

- Workstations.xml
- Users.xml
- Panelsets.xml
- Applications.xml
- Instances.xml
- Typehandlers.xml

The configuration of Applications.xml and Panelsets.xml are described in section 3.2

For the configuration of the remaining tables see the configuration tools documentation.

10 Configuration Files

The appearance and behaviour of Panelman is configured by the following system configuration tables:

- Workstations.xml
- Users.xml
- Panelsets.xml
- Applications.xml
- Instances.xml
- Typehandlers.xml

The configuration of Applications.xml and Panelsets.xml are described in section 3.2

For the configuration of the remaining tables see the configuration tools documentation.

11 Custom Title Bar / Button Text

Version 4.5.18 of panelman passes various parameters to the panelman element in order that it might make decisions about what text to display and indeed what colour to make itself.

Backwards compatibility is retained by continuing to send the text parameter but also sending many other values too.

The panelman titlebar dialog requires pre-defined resources to be present in the dialog. The only one of importance is a button/label/component called “title_caption”

The values sent are defined below:

Value	Description
text=title bar text	This is the original message sent by panelman. This is required for backwards compatibility
_PANELSET_START=_PANELSET_START	Sent at the start of a panelset configuration change – or at startup
panelset=<panelset name>	
level=<user level>	Value taken directly out of the panelset.xml file
panelsetLocation=<location>	NEW VALUE taken directly out of the panelsets.xml file
_PANELSET_END=_PANELSET_END	Sent at the end of a panelset configuration change (and at startup) This signifies that there are no more parameters following in this block.
_TARGET_START=_TARGET_START	Sent at the start of a block of parameters assembled when a new panel is shown

Value	Description
instance	Targeting information for this panel. Top level panels may not have an instance value so this parameter is null
panelType	Only type "V4.5" is likely to work
altId	Value taken for the given instance taken from instances.xml
instanceLocation	Value taken for the given instance taken from instances.xml
type	Value taken for the given instance taken from instances.xml
application	Application name for the application being started
caption	Value of caption field from applications.xml
title	Value of title field from applications.xml
icon	Value of icon field from applications.xml
applicationLocation	NEW ATTRIBUTE VALUE taken from applications.xml
_TARGET_END=_TARGET_END	This signifies that this is the end of the block of targeting data
__CLEAR=_CLEAR	Sent when the panelset is cleared (note two underscores)

The reason for the _TARGET_END and _PANELSET_END (etc.) tags is so that you don't have to guess if there's more parameters to come – you get a definite "that's it – get on with it!" trigger.

So…on receiving _TARGET_START you should clear your internal variables and wait…you'll get a number of other notifications which if you're interested you can store the values. When you get the _TARGET_END value you can formulate a result and display it.

Note: not all values are sent in all circumstances. If instance is null then you won't get the instance defining parameters (e.g. altId).

11.1Summary of enhancements used in settings files:

applications.xml (location):

```
<app id="mcr/router_xy_sdi" caption="SDI|MCR" title="SDI XY Router"
icon="/images/icon_xy" location="SkyStudios" />
```

panelsets.xml (new caption attribute):

```
<item
id="studio_1/video_ipd_converters" caption="IPD|Converters" title="IPD
Format Converters" icon="/images/icon_cross_converter"/>
```

workstation_settings.xml:

```
<workstation id="14">
      <setting id="location" value="SkyStudios"/>
```