FACILITY MONITORING & CONTROL

iControl WidgetLibrary
User guide

21 July 2015

Grass Valley, a Belden Brand
3499 Douglas-B.-Floreani
St-Laurent, Québec, Canada H4S 1Y6
Tel. 514-333-1772
Fax. 514-333-9828
www.grassvalley.com

Table of Contents

1. Getting Started

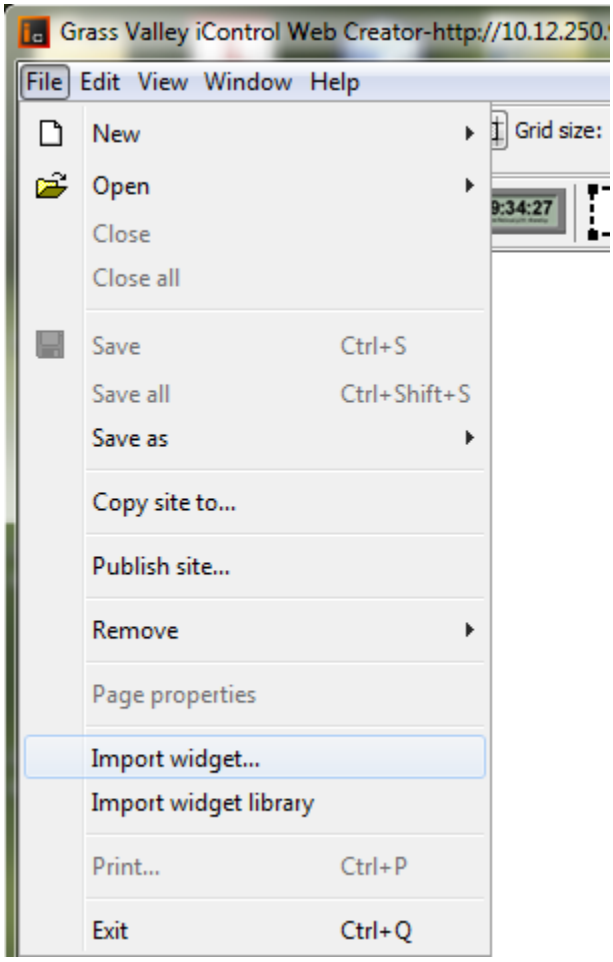
Widgets are graphical elements that are used to represent devices, alarm panels, sources, routers as well as other miscellaneous elements that are used in the iControl system in order to build signal paths and even entire sites.

The widgets reside on the server in a special site called “WidgetLibrary”. From this site the user has the possibility to view the entire widget library and to decide which ones are needed to build the new site. For convenience, the library is divided in folders that group the widgets by type, thus making it easier for the user to find what he needs. It is not recommended to modify or add pages to the “WidgetLibrary”. This site should only be used to view and import the widgets.

To include widgets in a site, the following steps must be performed (it is assumed that the user has already browsed the library and knows what he needs to import):

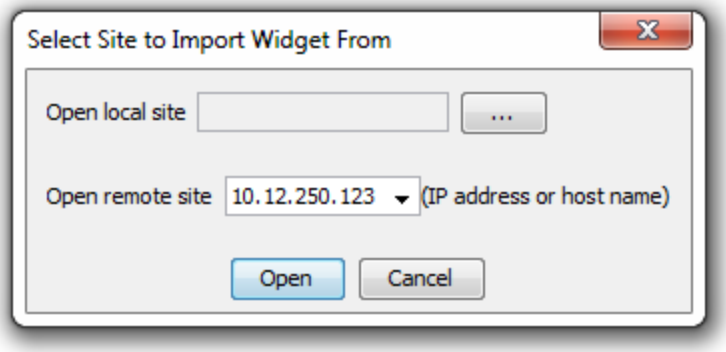
1. Create a new site on the server, if not already created, and open it. Note that the user can import widgets in a site at any given time.

2. Select *Import Widget …* from the *File* menu.



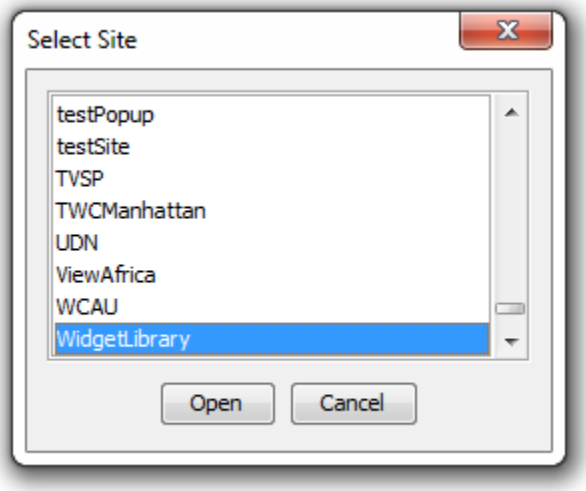
A pop-up will be displayed prompting the user to select the site from which the widgets should be imported.

3. Enter the IP address of the site where the WidgetLibrary resides and press *Open.* In this example the widgets are on 10.12.250.123.



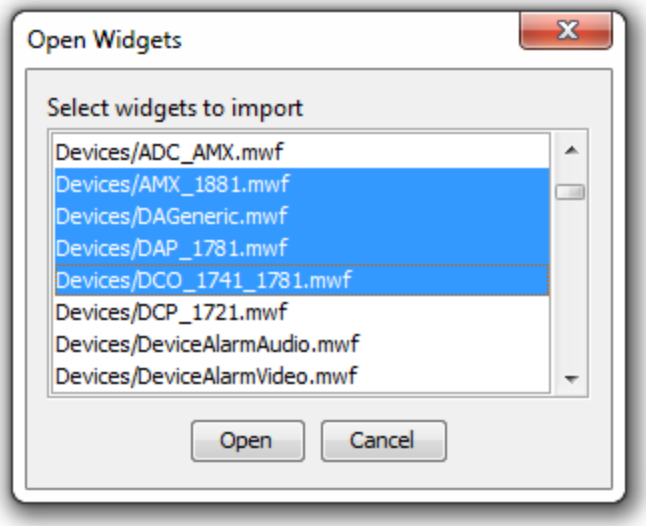
A pop-up will be displayed prompting the user to select a site from the list of sites available on the selected server.

4. Scroll down to the *WidgetLibrary* site and press *Open*.



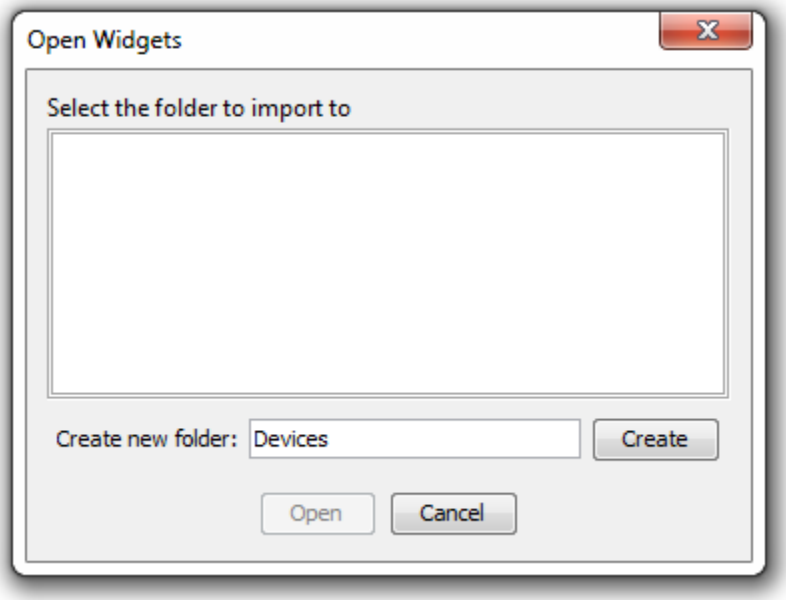
A pop-up will be displayed listing all the available widgets within the selected site.

5. Scroll down in the list and select the widgets that need to be imported. Note that several widgets can be selected (and thus imported) at one time, by pressing down the CTRL key and the widget (for random selections) and the SHIFT key and the first and last widgets (for a sequential selection). After making the selection, press *Open*. In this example the alarm panels of type 1 and 2 were sequentially selected.

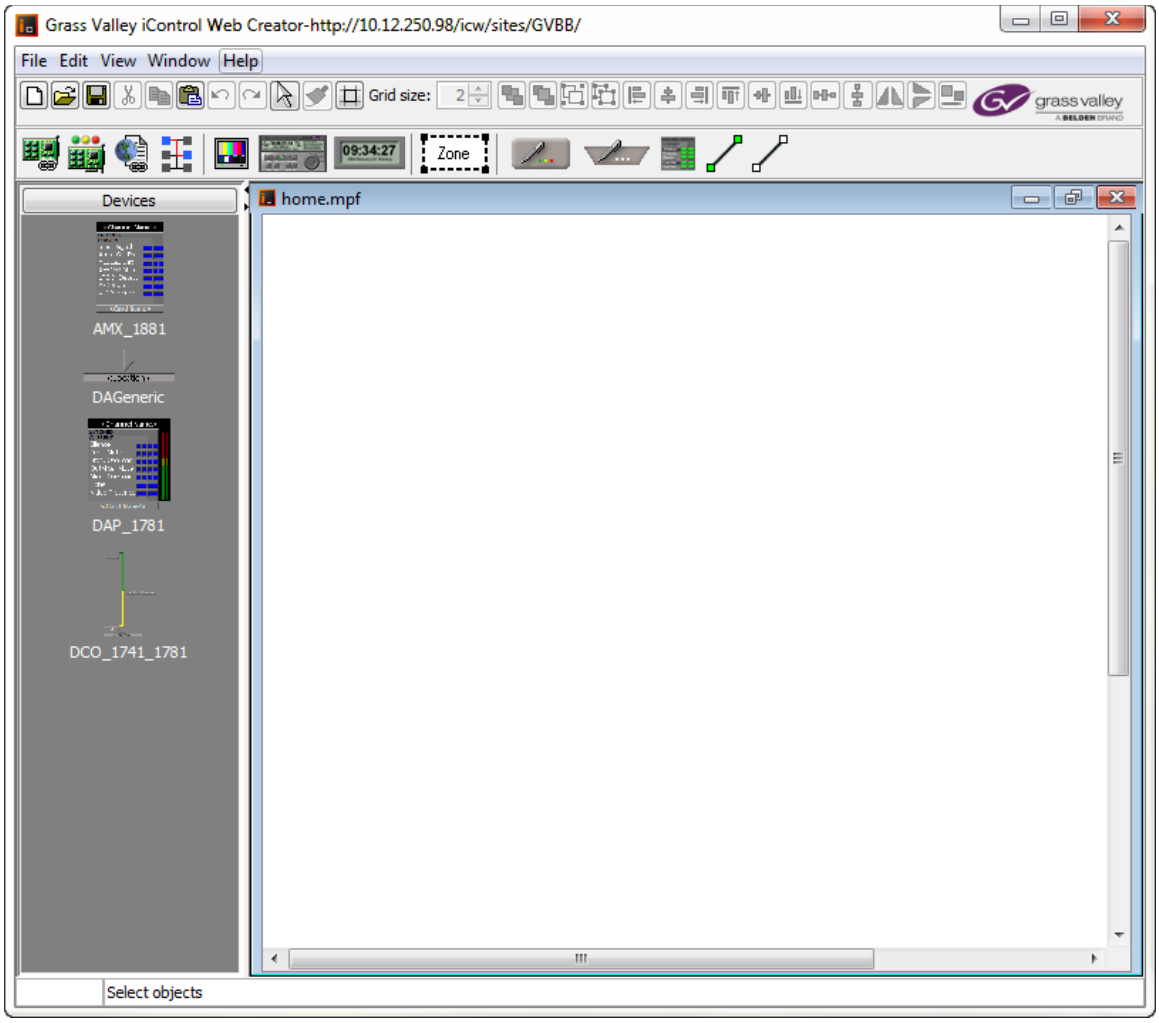


A pop-up is displayed prompting the operator to select a folder to which the widgets should be imported.

6. Create a new folder to import the widgets to and select *Create*. In this example a folder with the same name as the one the widgets were selected from is created. Note that when a widget is imported the folder is not automatically created. A folder with the same name or a different one must be created first.



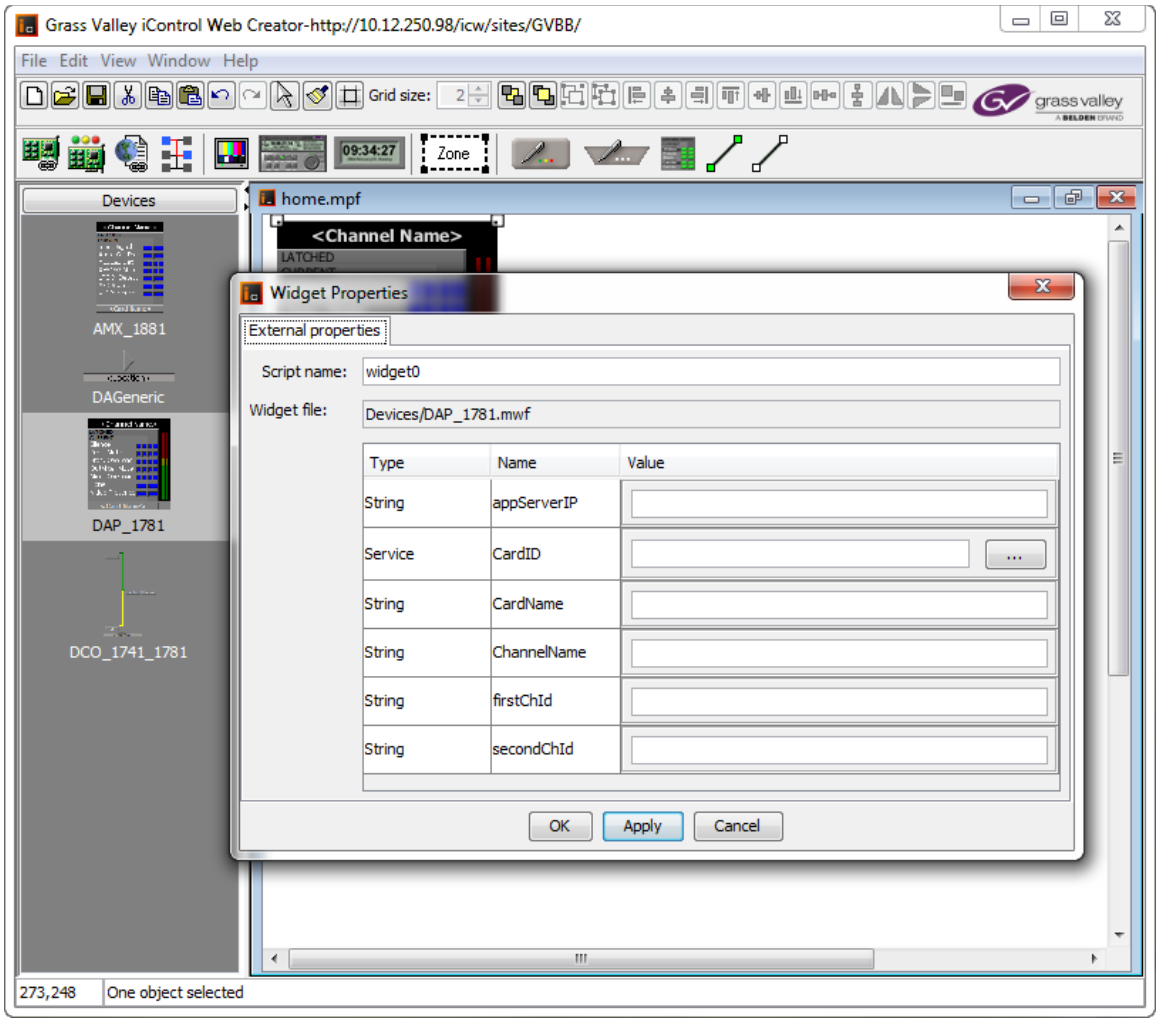
Once the *Create* or *Open* buttons were selected the widgets will be imported in the project. A sidebar will be automatically created in the site (like in the case of the WidgetLibrary site) where the widgets will be placed under the selected folder.



Note that, when widgets are imported for the first time, the “folders to import to” box is empty so a new folder must be created. If later in the project, more widgets need to be imported in the same folder, the same procedure must be used. This time the already created folder should be selected from the list prior to selecting *Open* (in this case).

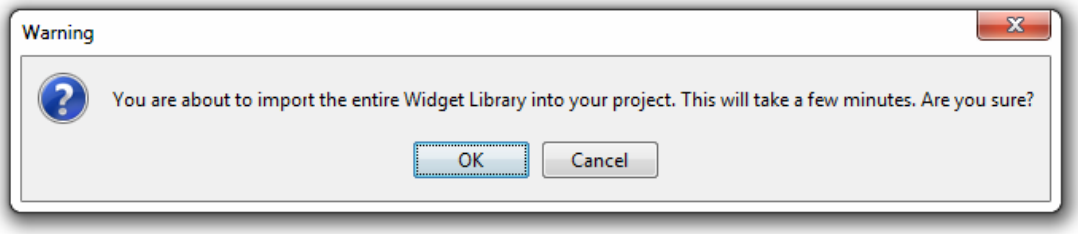
Now the widgets are ready to be used! Pages can be created using widgets following the same procedures as the ones detailed in the iControl User Manual for creating a page.

To place a widget on a page, the widget must first be selected from the side bar (using the left mouse button) and then clicking (the left mouse button again) on the page to make the widget appear at a desired location. Double clicking on the widget after it was placed on the page will display the properties panel of the widget, where the information pertaining to the widget should be filled in.



The following section details all the widgets currently in the library, by indicating their general use and the parameters that must be filled in for every widget.

Note that selecting the option *Import widget library* from the *File* menu will import the entire widget library from the current server into the current site. A pop message is displayed prior to importing the library to warn the user that all the widgets from the library will now be part of the current site. Selecting Cancel on the popup will cancel the widget library import.

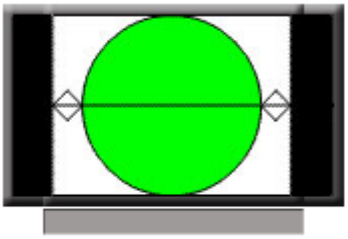


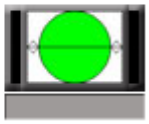
2. Widget Library

This section details all the widgets currently available for use in the widget library.

2.1 AFD

2.1.1 AFDButton & AFDButtonSmall





This widget should be used to set the AFD output for an AMX-188, XVP-1801 or XVP-811i card to any one of the formats detailed in the table below. Note that in the case of the XVP, the AFD mode must also be specified. If the AFD mode parameter is left blank, it is assumed that the card to be set is an AMX and not an XVP!

Parameter Name	Parameter Type	Expected Value
AFDMode	String	" "; 4:3; 16:9; AUTO; FORCED
AFDValue	String	16:9_11; 16:9_13; 16:9_14; 16:9_15; 16:9_8; 16:9_9; 4:3_10; 4:3_11; 4:3_13; 4:3_14; 4:3_2; 4:3_3; 4:3_4; 4:3_8
CardID	Service	Card Id selected from the browse menu
Card_Name	String	Name to be displayed on the label, along with the AFD format

2.2 Components

2.2.1 AudibleAlarm & AudibleAlarm_Large

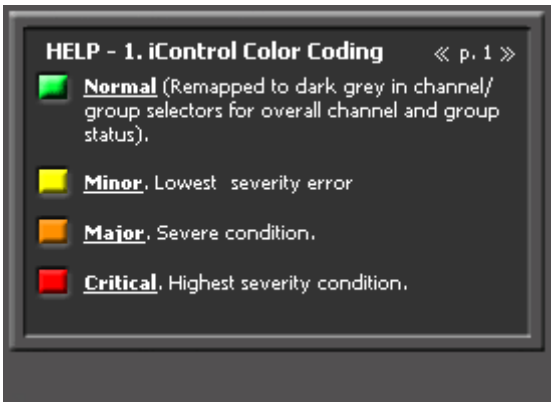
These widgets are used to acknowledge and silence a given set of alarms. To acknowledge the alarms, the left (bottom for the large widget) icon should be pressed. This can be done each time a new alarm(s) is generated in the system. The icon will also show the overall status of the alarm(s) that is monitoring. The graphic will always be the same. To silence the alarms the right (top for the large widget) icon should be pressed. This will display the icon with a red X over it. To make the alarms audible again, simply reselect the icon. Sounds will be played again once an alarm will be detected and the icon will revert back to the original graphic.

Parameter Name	Parameter Type	Expected Value
Alarm	Alarm URI	Virtual alarm composed of all the alarms that need to be monitored by this widget.
appServerIP	String	The IP address of the server where the site resides.
CriticalSound* **	String	Full path of the sound wave file associated with a critical alarm.
MajorSoundAlarm* **	String	Full path of the sound wave file associated with a major alarm.
MinorSoundAlarm* **	String	Full path of the sound wave file associated with a minor alarm.
StopAudibleAlarmWhenCleared	Boolean	When set to **true** the audible alarm is stopped once it was cleared. Set to **false** by default.

* Path should be of format: “http:// appserver ip / site name / sound folder / file name”
If for example appserver ip is 10.10.10.10, site name is GVBB, sound folder is AudioClips and file is AudibleAlarm_Critical.wav, your path will be:
http://10.10.10.10/GVBB/AudioClips/AudibleAlarm_Critical.wav

**Sound file can be of type: WAV, AIFF, AU

2.2.2 Help



iControlColorCoding panel. Selelcting the top right arrows at runtime, will let the user browse through the help pages.

This widget has no parameters.

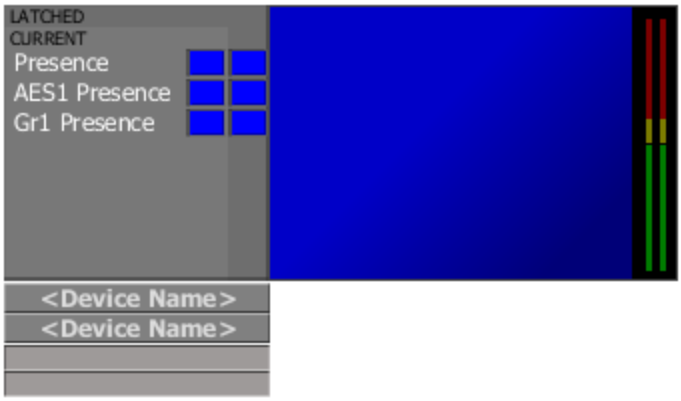
2.2.3 UserPreset

This widget should be used to set the user presets for a card directly from the page rather than going through the control panel of the card.

Parameter Name	Parameter Type	Expected Value
CardID	Service	Card Id selected from the browse menu
buttonText	String	The text to be displayed on the button
confirmation	String	Confirmation string to be displayed after the button is selected and before the command is actually send to the card. Defaulted to **Are you sure?**
index	String	Index of the preset -1 (e.g. if USER 1 preset is used, then the value entered should be 0)

2.3 Devices

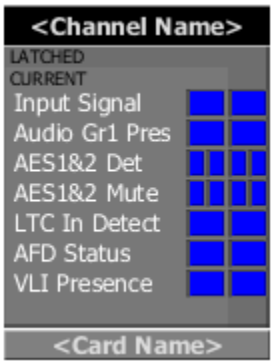
2.3.1 ADC_AMX



This widget is used for the ADC/AMX card combination to display basic alarms and stream with audio. The buttons display the overall status of the cards and when selected, they display the card panel. The labels below the buttons are used to display the location of the cards.

Parameter Name	Parameter Type	Expected Value
ADCDeviceName	String	ADC-1101
ADCDeviceOverallAlarm	Alarm URI	URI of the overall alarm of the ADC card.
ADCID	Service	Long id of the ADC card.
ADCLabel	String	Location of the ADC card.
AMXDeviceName	String	AMX-1101
AMXDeviceOverallAlarm	Alarm URI	URI of the overall alarm of the AMX card.
AMXID	Service	Long id of the AMX card.
AMXLabel	String	Location of the AMX card.
appserverId	String	The ip address of the server where the cards are located.

2.3.2 AMX_1881



This widget is dedicated to the AMX_1881 card. The button will show the overall status of the card. Selecting this button will display the AMX control panel in a pop-up window.

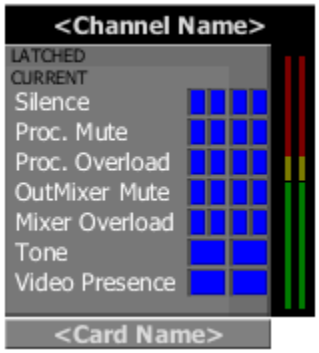
Parameter Name	Parameter Type	Expected Value
CardID	Service	The AMX card to be selected from the browse menu
CardName	String	Name of the card that is currently monitored
ChannelName	String	Name of the channel that is currently monitored

2.3.3 DAGeneric

This widget is used for the DA cards. The widget can show the overall status of the card as well as its location. When the button is selected, the widget will display the DA card panel.

Parameter Name	Parameter Type	Expected Value
deviceId	Service	Long id of the card.
deviceOverallAlarm	Alarm URI	URI of the overall alarm of the card.
label	String	Location of the card.

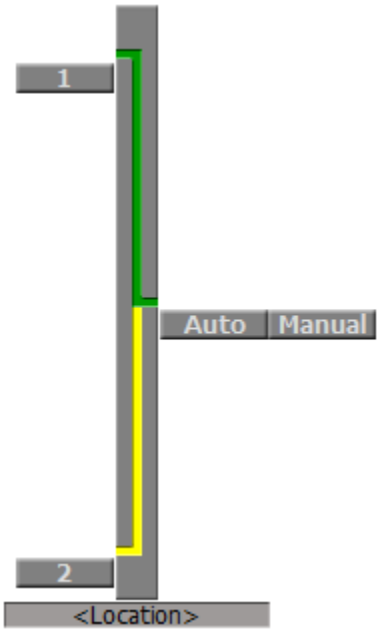
2.3.4 DAP_1781



This widget is dedicated to the DAP_1781 card. The button will show the overall status of the card. Selecting this button will display the DAP control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	The IP address of the server where the DAP card is located
CardID	Service	The DAP card to be selected from the browse menu
CardName	String	Name of the card that is currently monitored
ChannelName	String	Name of the channel that is currently monitored
firstChId	String	First audio channel
secondChId	String	Second audio channel

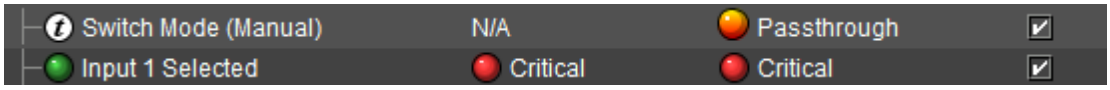
2.3.5 DCO_1741_1781



This widget is dedicated to the DCO_1741 and DCO_1781 cards. The long DCO button will show the overall status of the card as well as the currently selected connection. When the AUTO option is selected, the 1 and 2 input buttons are made unavailable, until the user selects the MANUAL option. Selecting the DCO button will display the card's control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
ChannelName	string	Name of the channel that is currently monitored
DCOID	Service	The DCO card to be selected from the browse menu
Name_IP1	String	The name of the inut 1 to be displayed on the button (e.g. MAIN …), defaulted to1
Name_IP2	String	The name of the inut 2 to be displayed on the button (e.g. MAIN …), defaulted to1
appServerIP	String	The ip address of the server where the DCO card is located
hideDCOControls	Boolean	Defaulted to true.

Note: The following two alarms need to be enabled on the card as shown in order for the widget to work as expected:



Switch Mode - Passthrough
Input 1 Selected – Critical

2.3.6 DCP_1721

This widget is dedicated to the DCP_1721 card. The button will show the overall status of the card. Selecting this button will display the DCP control panel in a pop-up window.

Note that this widget has a two page alarm panel. By default the panel is shown on page1.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	The IP address of the server where the DCP card is located
CardID	Service	The DCP card to be selected from the browse menu
CardName	String	Name of the card that is currently monitored
ChannelName	String	Name of the channel that is currently monitored

2.3.7 DSK_3901

This widget is used for the DSK3901 card to display basic alarms. The widget can show the overall status of the card as well as its location. When the button is selected, the widget will display the DSK card panel.

Parameter Name	Parameter Type	Expected Value
appserverId	String	Appserver where the device is located.
deviceName	String	**DSK-3901**
deviceOverallAlarm	AlarmURI	URI of the overall alarm of the card.
DSKID	Service	Long id of the card.
label	String	Location of the device.

Note: The following two alarms need to be enabled on the card as shown in order for the widget to work as expected:

2.3.8 DeviceAlarmAudio

This widget can be used if a single alarm is needed on the page or on a button.

Parameter Name	Parameter Type	Expected Value
alarm	URI	The uri of the alarm to be monitored

2.3.9 DeviceAlarmVideo

This widget can be used if a single alarm is needed on the page or on a button.

Parameter Name	Parameter Type	Expected Value
alarm	URI	The uri of the alarm to be monitored

2.3.10 DeviceGeneric_1

This device can be used by itself or in combination with an alarm panel in a path. Selecting the button will display the control panel of the device in a pop-up window. The overall alarm of the card will be shown on the button.

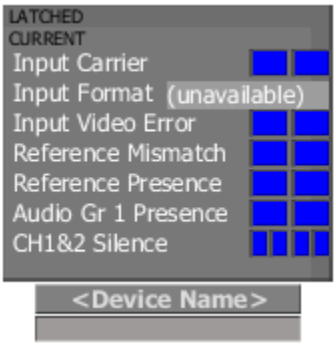
Parameter Name	Parameter Type	Expected Value
DeviceID	Service	Device Id selected from the browse menu
DeviceLabel	String	Label to be displayed on the button
DeviceLocation	String	Physical location of the device

2.3.11 DeviceGeneric_2

This device should be used by itself in a path. The button displays the status of the alarm predefined by the user.

Parameter Name	Parameter Type	Expected Value
DeviceLabel	String	Label to be displayed on the button
DeviceLocation	String	Physical location of the device
alarm	URI	URI of the alarm who's status will be displayed on the button

2.3.12 EAP-3901



This widget is dedicated to the EAP_3901 card. The button will show the overall status of the card. Selecting this button will display the EAP control panel in a pop-up window.

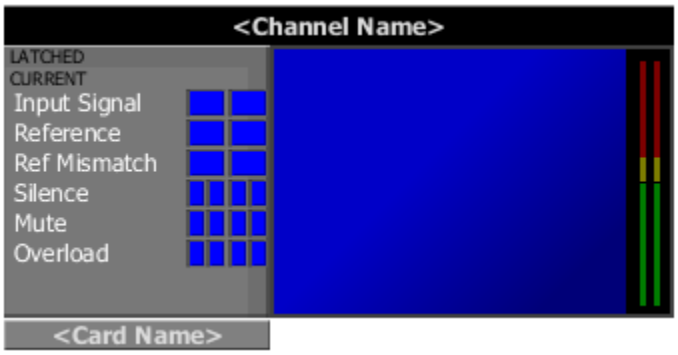
Parameter Name	Parameter Type	Expected Value
deviceName	String	**EAP-3901**
deviceOverallAlarm	AlarmURI	URI of the overall alarm of the card.
EAPID	Service	Long id of the card.
label	String	Location of the device.

2.3.13 EdgeVision Video Player

This widget is dedicated to the EdgeVision device and functions as a video player.
The widget's properties are described as follows:

Parameter Name	Parameter Type	Expected Value
EdgeVision_ServiceID	string	The long ID of the EdgeVision service, of the form {IP}_EdgeVision. The service browser can be used. Will be validated at runtime that it is indeed an EdgeVision service.
EdgeVisionPortNumber	Integer	Port number, range [1-4]. Will be validated at runtime if within bounds. If not, it will default to 1.
InitialMuteState	String	Initially mute audio if true, monitor audio if false. (default = true)
OperatorAllowAlarmOverlay	String	Allow operator to control visibility of overlay alarm panel. (default = true)
OperatorAllowEdgeVisionBrowse	String	Allow operator to switch to another EdgeVision unit. (default = true)
OperatorAllowPortSwitch	String	Allow operator to switch stream input. (default = true)
OperatorAllowRemoteControl	String	Allow operator to popup STB remote control.
OverlayAlarmsShowOnError	String	Only show overlay alarm panel if overall is in error (non-normal). (default = true)
OverlayAlarmsVisible	String	Initial state of overlay alarm panel. (default = true)
OverlayXPosition	String	Horizontal position of the overlay alarm panel. Possible values are: • left: left edge of player (not including border alarm, if visible) • center: centered in player (not including ALM) • right: right edge of player (not including ALM) • {positive integer}: offset from left edge • {negative integer}: offset from right edge (default = left)
OverlayYPosition	String	Vertical position of the overlay alarm panel. Possible values are: • top: top edge of player (not including border alarm, if visible) • center: centered in player • bottom: bottom edge of player • {positive integer}: offset from top edge • {negative integer}: offset from bottom edge (default = -5)
UseBorderAlarm	String	Use border alarm instead of UMD as input overall alarm. (default = false)
UseHighResolutionStream	String	Use the high proxy port instead of low. (default = false)
UseOverallAlarm	String	Enables probing. Set to false if widget is just used for streaming and remote control. (default = true)

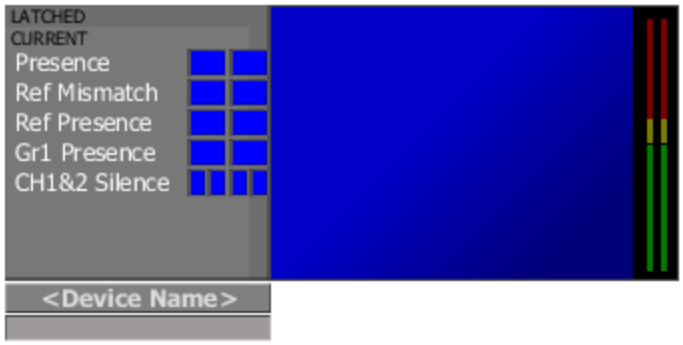
2.3.14 FRS_1101



This widget is dedicated to the FRS_1101 card. The button will show the overall status of the card. Selecting this button will display the FRS control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	The IP address of the server where the FRS card is located
CardID	Service	The FRS card to be selected from the browse menu
CardName	String	Name of the card that is currently monitored
ChannelName	String	Name of the channel that is currently monitored

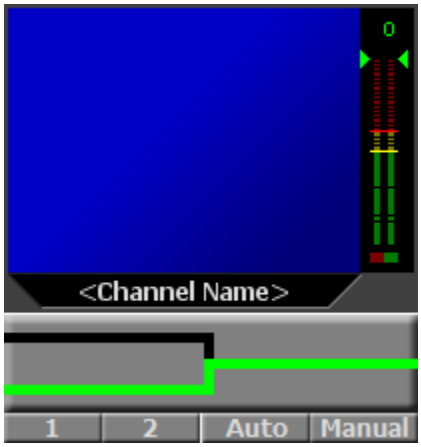
2.3.15 FRS_1801



This widget is dedicated to the FRS_1801 card. The button will show the overall status of the card. Selecting this button will display the FRS control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	The IP address of the server where the FRS card is located
deviceName	String	**FRS-1801**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the FRS card button.
FRSID	Service	The FRS card to be selected from the browse menu
label	String	Location of the card.

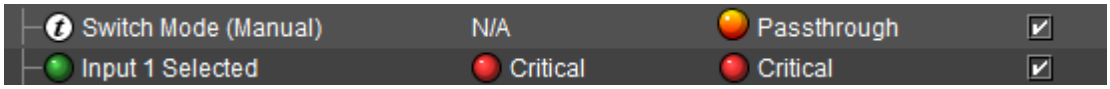
2.3.16 HCO_1822_1



This widget is dedicated to the HCO-1822 card (can also be used for the HCO_1821 card). The button will show the overall status of the card as well as the currently selected connection. When the AUTO option is selected, the 1 and 2 input buttons are made unavailable, until the user selects the MANUAL option. Selecting the HCO button will display the HCO control panel in a pop-up window.

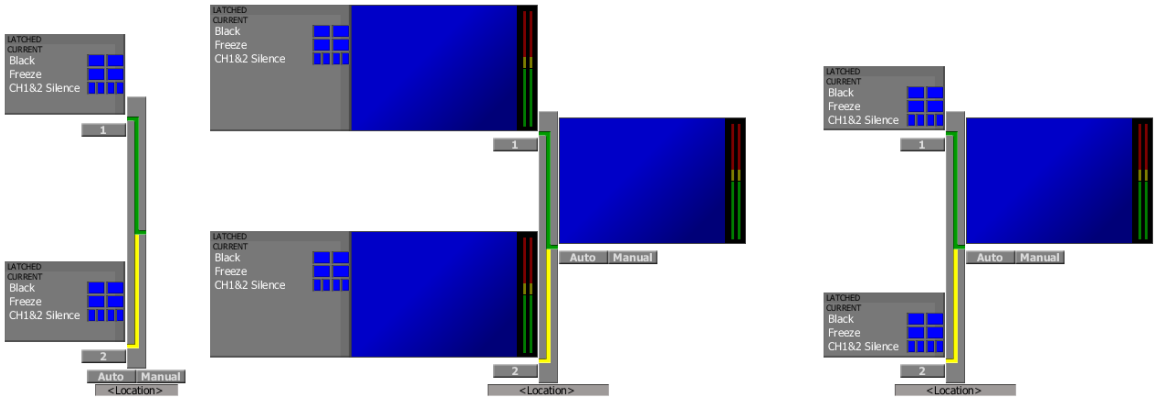
Parameter Name	Parameter Type	Expected Value
ChannelName	string	Name of the channel that is currently monitored
HCOID	Service	The HCO card to be selected from the browse menu
Name_IP1	String	The name of the inut 1 to be displayed on the button (e.g. MAIN …), defaulted to1
Name_IP2	String	The name of the inut 2 to be displayed on the button (e.g. MAIN …), defaulted to1
appServerIP	String	The ip address of the server where the HCO card is located
hideHCOControls	Boolean	Set to true by default (controls will only become available if the user changes the value to false).

Note: The following two alarms need to be enabled on the card as shown in order for the widget to work as expected:



Switch Mode - Passthrough
Input 1 Selected – Critical

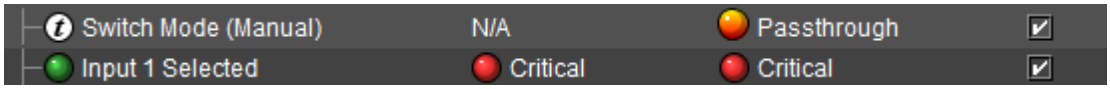
2.3.17 HCO_1822_2, HCO_1822_3, HCO_1822_4



This widget is dedicated to the HCO-1822 card (can also be used for the HCO_1821 card). The button will show the overall status of the card as well as the currently selected connection. When the AUTO option is selected, the 1 and 2 input buttons are made unavailable, until the user selects the MANUAL option. Selecting the HCO button will display the HCO control panel in a pop-up window.

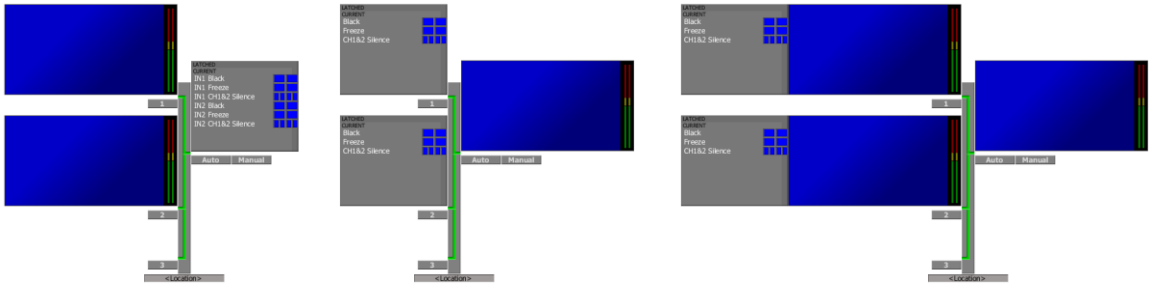
Parameter Name	Parameter Type	Expected Value
appServerIP	String	The ip address of the server where the HCO card is located
HCOID	Service	The HCO card to be selected from the browse menu
hideHCOControls	Boolean	Set to true by default (controls will only become available if the user changes the value to false).
label	String	Location of the card.
Name_IP1	String	The name of the inut 1 to be displayed on the button (e.g. MAIN …), defaulted to1
Name_IP2	String	The name of the inut 2 to be displayed on the button (e.g. MAIN …), defaulted to1

Note: The following two alarms need to be enabled on the card as shown in order for the widget to work as expected:



Switch Mode - Passthrough
Input 1 Selected – Critical

2.3.18 HCO_3901_1, HCO_3901_2, HCO_3901_3



This widget is dedicated to the HCO-3901 card. The button will show the overall status of the card as well as the currently selected connection. When the AUTO option is selected, the 1, 2 and 3 input buttons are made unavailable, until the user selects the MANUAL option. Selecting the HCO button will display the HCO control panel in a pop-up window.

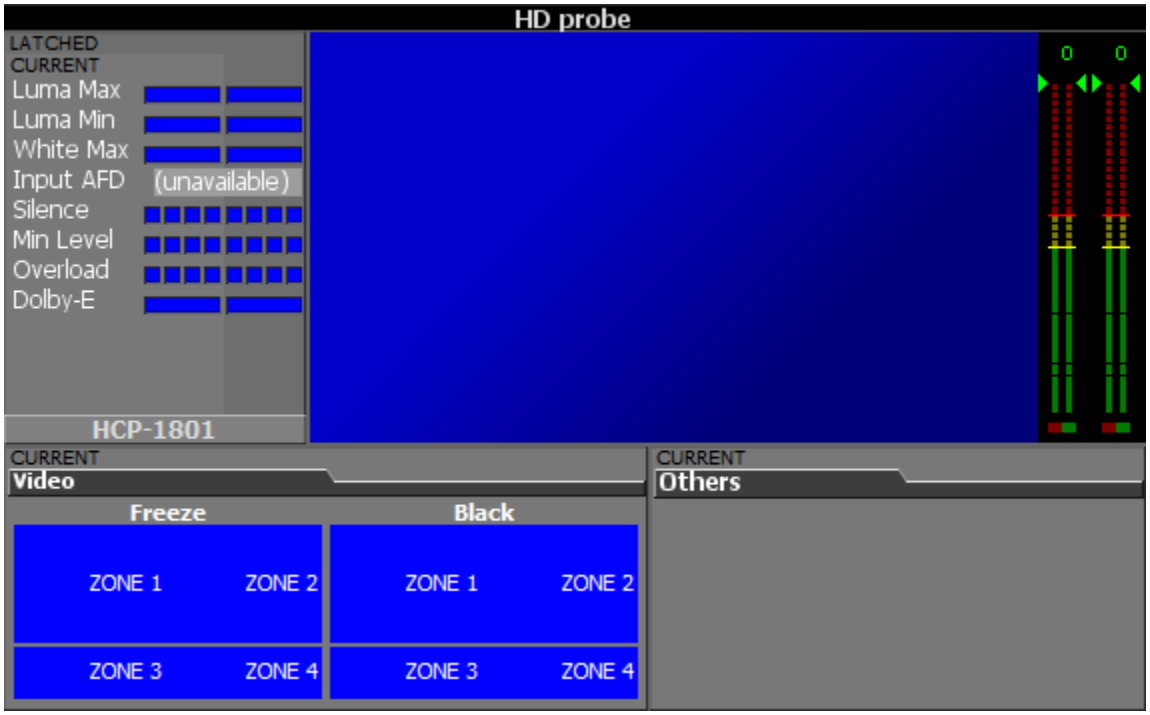
Note that the firmware version on this card needs to be 1.2.1 (released firmware with this service 1.2.0) or higher in order for the widget to work properly.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	The ip address of the server where the HCO card is located
HCOID	Service	The HCO card to be selected from the browse menu
hideHCOControls	Boolean	Set to true by default (controls will only become available if the user changes the value to false).
label	String	Location of the card.
Name_IP1	String	The name of the input 1 to be displayed on the button (e.g. MAIN …), defaulted to1
Name_IP2	String	The name of the input 2 to be displayed on the button (e.g. MAIN …), defaulted to2
Name_IP3	String	The name of the input 3 to be displayed on the button (e.g. MAIN …), defaulted to3

Note: The following two alarms need to be enabled on the card as shown in order for the widget to work as expected:

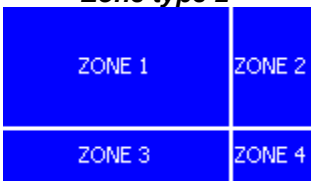
Output - Passthrough
Switch Mode - Passthrough

2.3.19 HCP_1801_1



This widget is dedicated to the HCP-1801 card. It depicts the basic alarms, the HD thumbnail with two ALMs and the two freeze and black zones predefined for this card. An extra alarm panel is provided for up to 8 additional alarms. In the case of composite alarms, *alarm1* is paired with *alarm9*, *alarm2* with *alarm10*, *alarm3* with *alarm11*, *alarm4* with *alarm12*, *alarm5* with *alarm13*, *alarm6* with *alarm14*, *alarm7* with *alarm15*, and *alarm8* with *alarm16*. If the type of the alarms is other than composite, the alarms from *alarm9* to *alarm16* will be ignored. The HCP-1801 button shows the overall status of the card. Moreover, selecting this button will display the control panel of the card in a popup window. The two predefined freeze and black zones are shown below:

Zone type 1

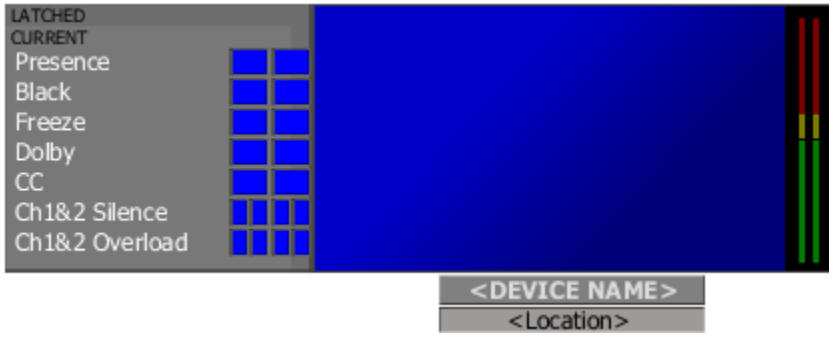


Zone type 2

Parameter Name	Parameter Type	Expected Value
alarm1 to alarm16	AlarmURI	Alarm to be displayed.
alarmText1 to alarmText8	AlarmURI	Text to be displayed in the panel for each alarm.
alarmType1 to alarmType8	AlarmURI	Type of first alarm. Empty for status alarms, ***text*** for text alarms and ***multiple*** for composite alarms.
BackZoneType	String	**1** or **2** (see above)
FreezeZoneType	String	**1** or **2** (see above)

HCPALM1	Audio stream	First ALM stream to be monitored
HCPALM2	Audio stream	Second ALM stream to be monitored
HCPID	service	HCP card id
HCPVideo	Video stream	Video stream to be monitored

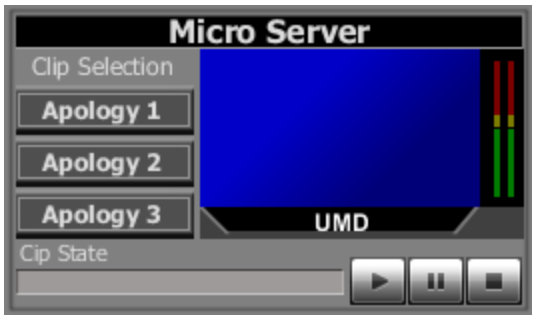
2.3.20 HCP_1801_2



This widget is dedicated to the HCP_1801 card. The button will show the overall status of the card. Selecting this button will display the HCP control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	The IP address of the server where the HCP card is located
deviceName	String	**HCP-1801**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the HCP card button.
HCPID	Service	The HCP card to be selected from the browse menu
label	String	Location of the card.

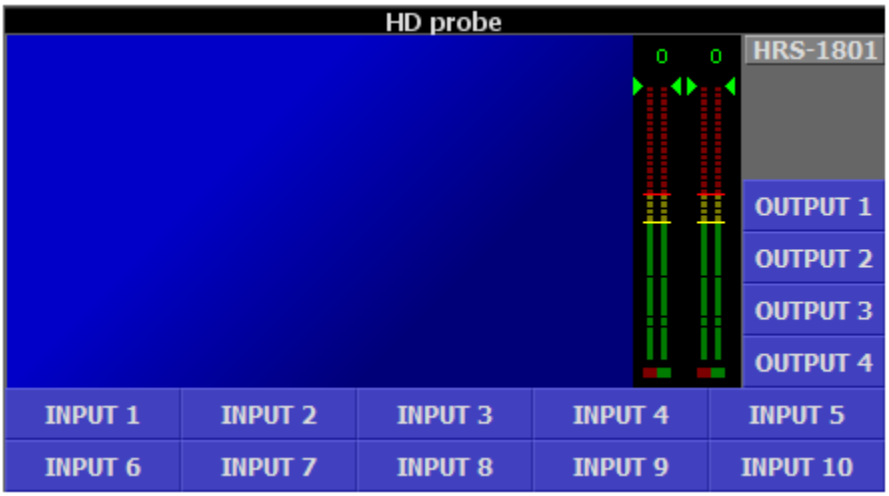
2.3.21 HMP_1801_Simple



This widget is dedicated to the HMP card. It depicts a simpler version of the card, allowing the user to choose between three clips selected from among the ones previously loaded on the card. The state of the clip is also displayed along with buttons to play, pause and stop a currently cued clip. Note that when a clip is selected, the previous clip is automatically stopped and the selected clip is cued instead, ready to be played.

Parameter Name	Parameter Type	Expected Value
clip1Id	String	Id of the first clip
clip1Label	String	Name of the first clip
clip2Id	String	Id of the second clip
clip2Label	String	Name of second clip
clip3Id	String	Id of the third clip
clip3Label	String	Name of the third clip
hmpId	Service	The HMP card to be selected from the browse menu
ipAddress	String	The IP address of the server where the card is located
label	String	The labels to be displayed at the top of the widget
umd	String	The text to be displayed in the player's umd

2.3.22 HRS_1801



This widget is dedicated to the HRS-1801 card. It lets the user monitor the video and audio feeds as well as control the input and output selections. The HRS-1801 button shown in the top right corner depicts the overall status of the card. Selecting this button will display the control panel of the card in a popup window.

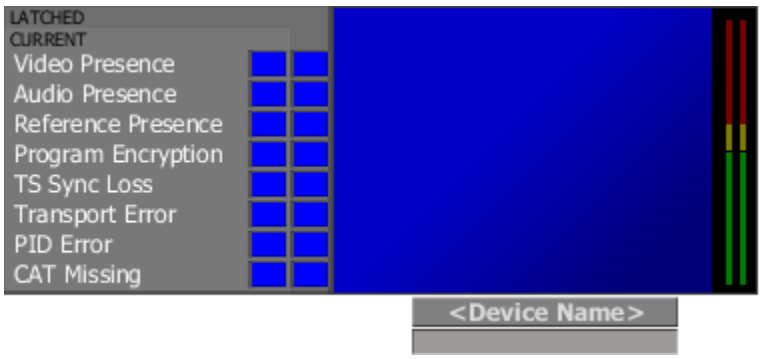
Parameter Name	Parameter Type	Expected Value
HRSID	Service	HRS card to be monitored
appServerIP	String	The ip address of the server where the HRS card is located
level	String	The level of the router to be controlled

2.3.23 IRD3802

This widget is dedicated to the IRD3802 card (can also be used for the IRD3811) and it lets the user monitor the currently selected stream. By default, the audio is muted but the user can un-mute it by selecting the audio button. The widget selects the service to be monitored and sets the port number and the multicast address for a predefined card.

Parameter Name	Parameter Type	Expected Value
appserverIP	String	The IP address of the server where the HRS card is located
mirandaIRD	String	The long ID of the IRD card to be monitored.
programNumber	String	The id of the service to be monitored.
streamIP	String	The multicast address.
streamPort	String	The port number.

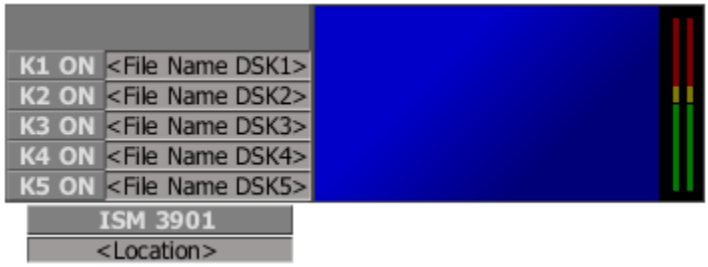
2.3.24 IRD_3802_3811



This widget is dedicated to the IRD_3802 and IRD_3811 cards. The button will show the overall status of the card. Selecting this button will display the IRD control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	The IP address of the server where the IRD card is located
deviceName	String	**IRD-1801**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the IRD card button.
IRDID	Service	The IRD card to be selected from the browse menu
label	String	Location of the card.

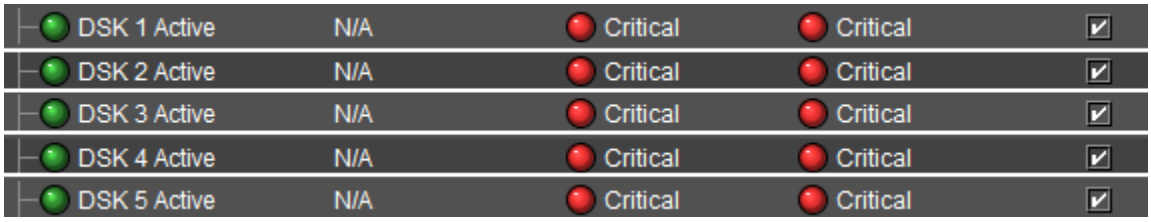
2.3.25 ISM_3901



This widget is dedicated to the ISM_3901 card. The five keyers can be set ON or OFF from the panel. In addition, the image files for each keyer that is ON will be displayed next to its corresponding button. The button will show the overall status of the card. Selecting this button will display the ISM control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	The IP address of the server where the ISM card is located
deviceName	String	**ISM-3901**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the ISM card button.
label	String	Location of the card.
ISMID	Service	The ISM card to be selected from the browse menu

Note: The following five alarms need to be enabled on the card as shown in order for the widget to work as expected:

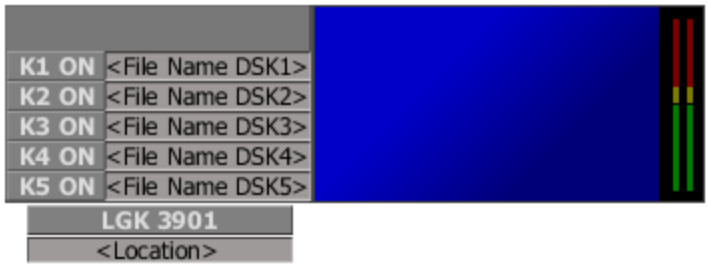


2.3.26 ImageStore

This widget is dedicated to the ImageStore.

Parameter Name	Parameter Type	Expected Value
ImagestoreID	Service	Imagestore id to be selected from the browse menu
Location	String	Physical location of the imagestore
Title	String	Identification for the imagestore

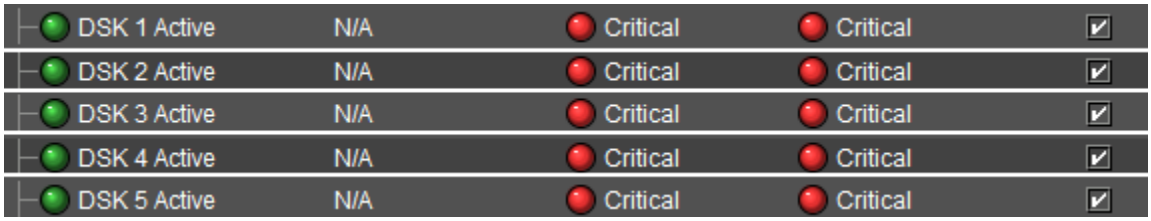
2.3.27 LGK_3901



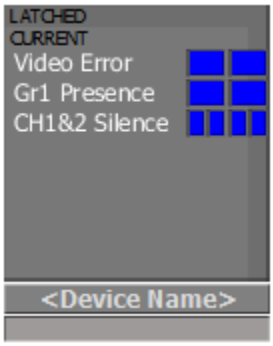
This widget is dedicated to the LGK_3901 card. The five keyers can be set ON or OFF from the panel. In addition, the image files for each keyer that is ON will be displayed next to its corresponding button. The button will show the overall status of the card. Selecting this button will display the LGK control panel in a pop-up window.

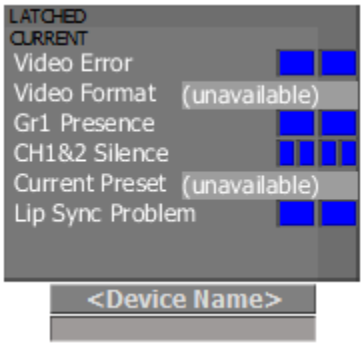
Parameter Name	Parameter Type	Expected Value
appServerIP	String	The IP address of the server where the LGK card is located
deviceName	String	**LGK-3901**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the LGK card button.
label	String	Location of the card.
LGKID	Service	The LGK card to be selected from the browse menu

Note: The following five alarms need to be enabled on the card as shown in order for the widget to work as expected:



2.3.28 MAP_3901_1, MAP_3901_2

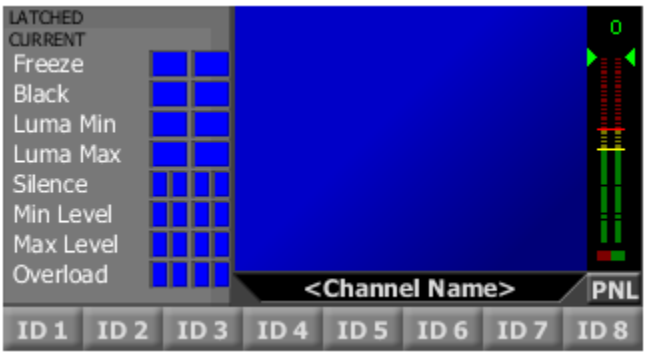




Those widgets are dedicated to the MAP_3901 card. The button will show the overall status of the card. Selecting this button will display the MAP control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
deviceName	String	**MAP-3901**
label	String	Location of the card.
MAPID	Service	The MAP card to be selected from the browse menu

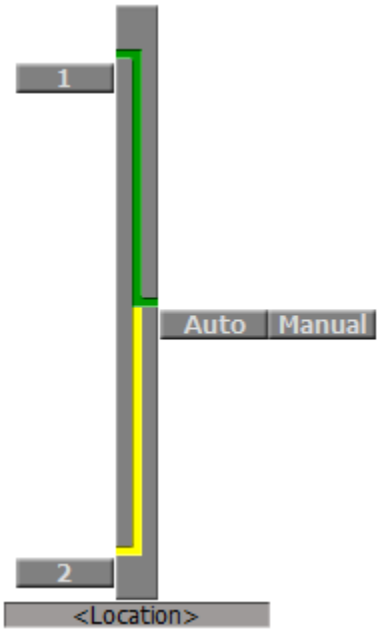
2.3.29 MulticardVideoAlarms



This widget can monitor up to eight cards at a time. The card types that can be used are: DEC1002, DEC1003 and SCP1121. Selecting a button in the bottom of the widget will update the alarms and the thumbnails to a specific predefined card. The button will also depict the overall status of that card. The PNL button, when selected, will display the control panel of the currently selected card in a popup window.

Parameter Name	Parameter Type	Expected Value
Card1_ID	Service	Id of the first card
Card1_UMD	String	Channel name to be displayed
Card1_Value	String	Name displayed on the button
Card1_ID	Service	Id of the second card
Card1_UMD	String	Channel name to be displayed
Card1_Value	String	Name displayed on the button
Card1_ID	Service	Id of the third card
Card1_UMD	String	Channel name to be displayed
Card1_Value	String	Name displayed on the button
Card1_ID	Service	Id of the fourth card
Card1_UMD	String	Channel name to be displayed
Card1_Value	String	Name displayed on the button
Card1_ID	Service	Id of the fifth card
Card1_UMD	String	Channel name to be displayed
Card1_Value	String	Name displayed on the button
Card1_ID	Service	Id of the sixth card
Card1_UMD	String	Channel name to be displayed
Card1_Value	String	Name displayed on the button
Card1_ID	Service	Id of the seventh card
Card1_UMD	String	Channel name to be displayed
Card1_Value	String	Name displayed on the button
Card1_ID	Service	Id of the eight card
Card1_UMD	String	Channel name to be displayed
Card1_Value	String	Name displayed on the button
appServerIP	String	Ip address of the server where the cards are located

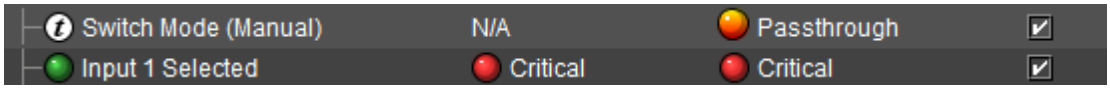
2.3.30 SCO_1421



This widget is dedicated to the SCO-1421 card. The button will show the overall status of the card as well as the currently selected connection. When the AUTO option is selected, the 1 and 2 input buttons are made unavailable, until the user selects the MANUAL option. Selecting the SCO button will display the SCO control panel in a pop-up window.

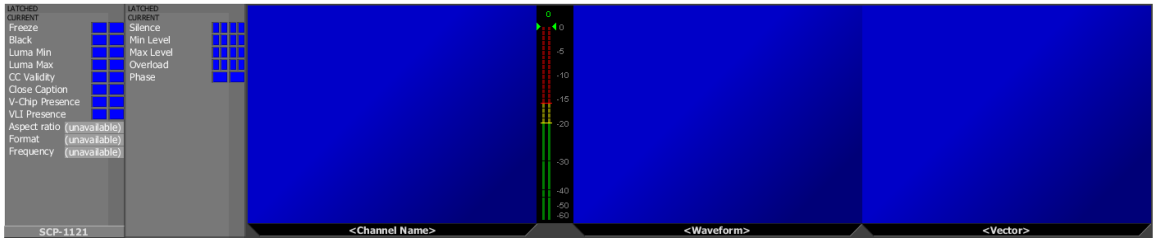
Parameter Name	Parameter Type	Expected Value
appServerIP	String	The ip address of the server where the SCO card is located
SCOID	Service	The SCO card to be selected from the browse menu
hideSCOControls	Boolean	Set to true by default (controls will only become available if the user changes the value to false).
label	String	Location of the card.
Name_IP1	String	The name of the inut 1 to be displayed on the button (e.g. MAIN …), defaulted to1
Name_IP2	String	The name of the inut 2 to be displayed on the button (e.g. MAIN …), defaulted to1

Note: The following two alarms need to be enabled on the card as shown in order for the widget to work as expected:



Switch Mode - Passthrough
Input 1 Selected – Critical

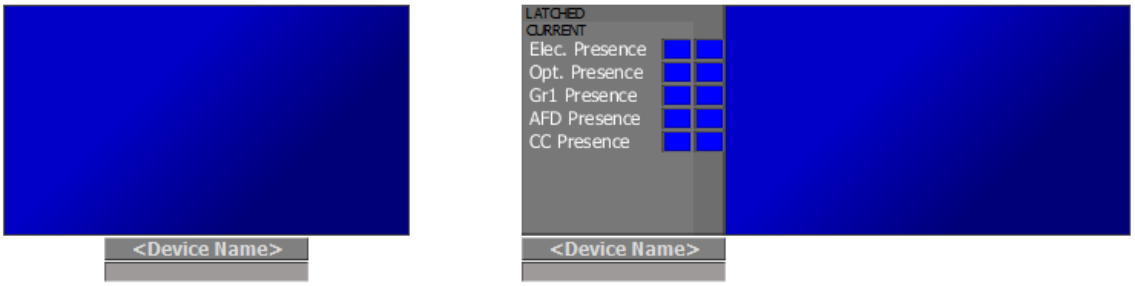
2.3.31 SCP_1121



This widget is dedicated to the SCP-1121 card with the scope option enabled.

Parameter Name	Parameter Type	Expected Value
CardID	Service	SCP card id to be monitored
ChannelName	String	Channel name to be displayed in the umd below the card's thumbnail
Vector	String	Name to be displayed in the umd below the vector thumbnail
Waveform	String	Name to be displayed in the umd below the waveform thumbnail
appserverIP	String	Address of the server where the card is located

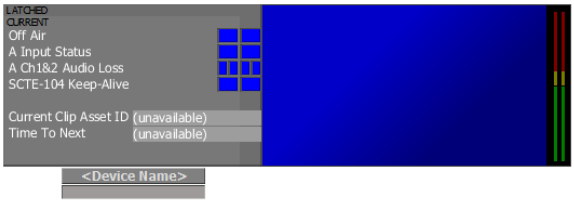
2.3.32 SME_1901_1, SME_1901_2



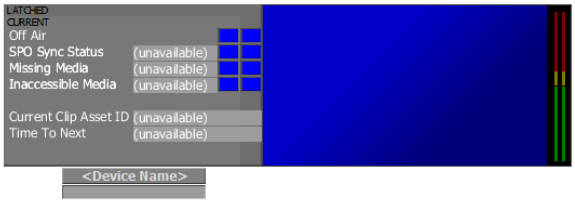
Those widgets are dedicated to the SME-1901 card. They can also be used for the SME-1911 card. The button will show the overall status of the card. Selecting this button will display the SME control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
deviceName	String	**SME-1901**
label	String	Location of the card.
SMEID	Service	The SME card to be selected from the browse menu
streamId	String	Id of the multicast stream

2.3.33 SSP_3801_1, SSP_3801_2



SCTE-104 Regionalization (Manual Control) [SSP_3801_1]



Schedule Playout (SPO) [SSP_3801_2]

Those widgets are dedicated to the SSP-3801 card. The difference in the two widgets consists in the alarms depicted in the alarm panel at the left which differ slightly depending on the use case (see a brief description of the alarms below). In addition to the alarm panel, the widget will also display a thumbnail with ALMs at the right. The button below the alarm panel will show the overall status of the card; selecting this button will display the SSP control panel in a pop-up window.

SCTE-104 Regionalization use case [Manual Control] (use SSP_3801_1 widget)

- Status Alarms:
  - Off Air – the card is currently in a off-air state (i.e. not playout out)
  - A Input Status – input timing not stable with respect to reference
  - A Ch1&2 Audio Loss – embedded audio cannot be detected on channels 1 & 2 pf the A input
  - SCTE-104 Keep Alive – status of the presence of SCTE-104 keep-alive packets
- Status List:
  - Current Clip Asset Id – asset ID of the clip that is currently on air as specified by SCTE-104
  - Time to Next – the time remaining until the next primary event

Schedule Playout use case [SPO] (use SSP_3801_2 widget)

- Status Alarms:
  - Off Air – the card is currently in a off-air state (i.e. not playout out)
  - SPO Sync Status – card no longer in sync with GV STRATUS playout
  - Missing Media – the media file is missing from the card's cache and the event is within the configured search window
  - Inaccessible Media – the specified source media does not exist at the specified location(s) and the event is within the configured inaccessible media search window
- Status List:
  - Current Clip Asset Id - asset id of the clip that is currently on air as specified by the scheduling system
  - Time to Next – the time remaining until the next primary event

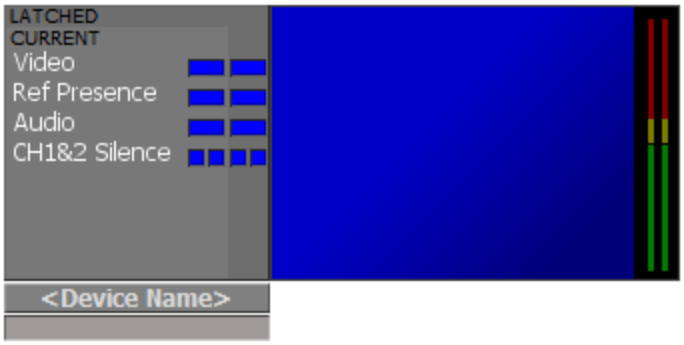
Parameter Name	Parameter Type	Expected Value
appserverId	String	Address of the server where the card is located (note that this is not the IP address of the card itself but rather the IP address of the server to which the frame for this card was added)
deviceName	String	**SSP-3801**
label	String	Location of the card.
SSPID	Service	The SSP card to be selected from the browse menu

XVP_1801_1

This widget is dedicated to the XVP_1801 card. It allows the user to choose one of the two inputs of the card while displaying the resulting output video and audio in the player at the top of the widget. The type of the card and overall alarm are displayed on the button below the player. When selected, the control panel of the card will be displayed as a pop-up.

Parameter Name	Parameter Type	Expected Value
appServerIP	String	Address of the server where the card is located
deviceType	String	Type of the device
label	String	Label of the device (location)
XVPID	Service	The XVP card to be selected from the browse menu

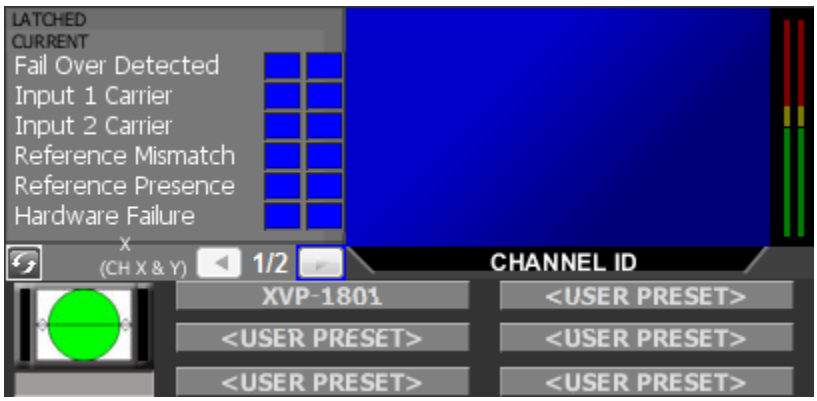
2.3.34 XVP_1801_2



This widget is dedicated to the XVP_1801 card. The button will show the overall status of the card. Selecting this button will display the XVP control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
audioStream	Audio Stream	Audio stream to be displayed
deviceName	String	**XVP-1801**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the ISM card button.
label	String	Location of the card.
videoStream	Video Stream	Video stream to be displayed
XVPID	Service	The XVP card to be selected from the browse menu

2.3.35 XVP_1801_AFD_presets

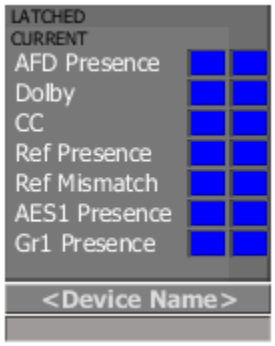


This widget is an enhanced version of the XVP widget. It allows the user to monitor key alarms for the card as well as the audio and video outputs. The user is also able to select between a maximum of 5 presets and to set the AFD to a predefined value. The XVP button depicts the overall status of the card and displays the control panel when selected.

Note that the alarm panel for this widget has two pages. The panel is defaulted to page1.

Parameter Name	Parameter Type	Expected Value
AESGroupID	String	1 to 4
AFDMode	String	" "; 4:3; 16:9; AUTO; FORCED
AFDValue	String	16:9_11; 16:9_13; 16:9_14; 16:9_15; 16:9_8; 16:9_9; 4:3_10; 4:3_11; 4:3_13; 4:3_14; 4:3_2; 4:3_3; 4:3_4; 4:3_8
appServerIP	String	Address of the server where the card is located
audioGroupID	String	1 to 4
cardID	Service	The XVP card to be selected from the browse menu
cardName	String	Type of the card
channelName	String	Name of the channel to be displayed in the player's umd
firstChID	String	First audio channel (1 to 16)
label_userPresets1	String	Name of the first user presets to be displayed on the button. If left empty, the button will not be shown
label_userPresets2	String	Same for second user presets
label_userPresets3	String	Same for third user presets
label_userPresets4	String	Same for fourth user presets
label_userPresets5	String	Same for fifth user presets
secondChID	String	Second audio channel (1 to 16)

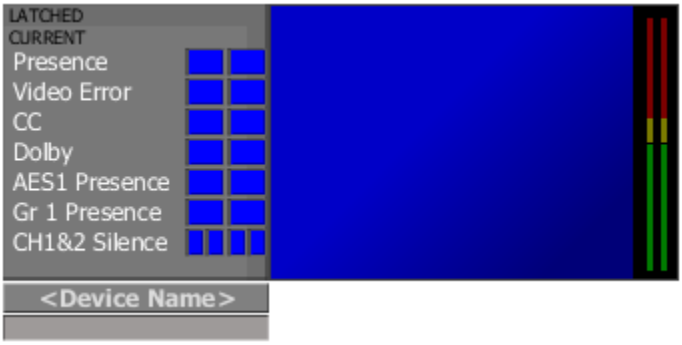
2.3.36 XVP_1801_NoThumbnail



This widget is dedicated to the XVP_1801 card. The button will show the overall status of the card. Selecting this button will display the XVP control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
deviceName	String	**XVP-1801**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the XVP card button.
label	String	Location of the card.
XVPID	Service	The XVP card to be selected from the browse menu

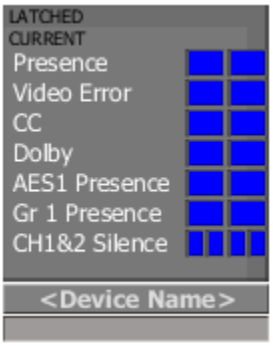
2.3.37 XVP_3901



This widget is dedicated to the XVP_3901 card. The button will show the overall status of the card. Selecting this button will display the XVP control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
audioStream	Audio Stream	Audio stream to be displayed
deviceName	String	**XVP-3901**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the ISM card button.
label	String	Location of the card.
videoStream	Video Stream	Video stream to be displayed
XVPID	Service	The XVP card to be selected from the browse menu

2.3.38 XVP_3901_NoThumbnail



This widget is dedicated to the XVP_3901 card. The button will show the overall status of the card. Selecting this button will display the XVP control panel in a pop-up window.

Parameter Name	Parameter Type	Expected Value
deviceName	String	**XVP-3901**
deviceOverallAlarm	AlarmURI	Status of the overall alarm to be shown on the XVP card button.
label	String	Location of the card.
XVPID	Service	The XVP card to be selected from the browse menu

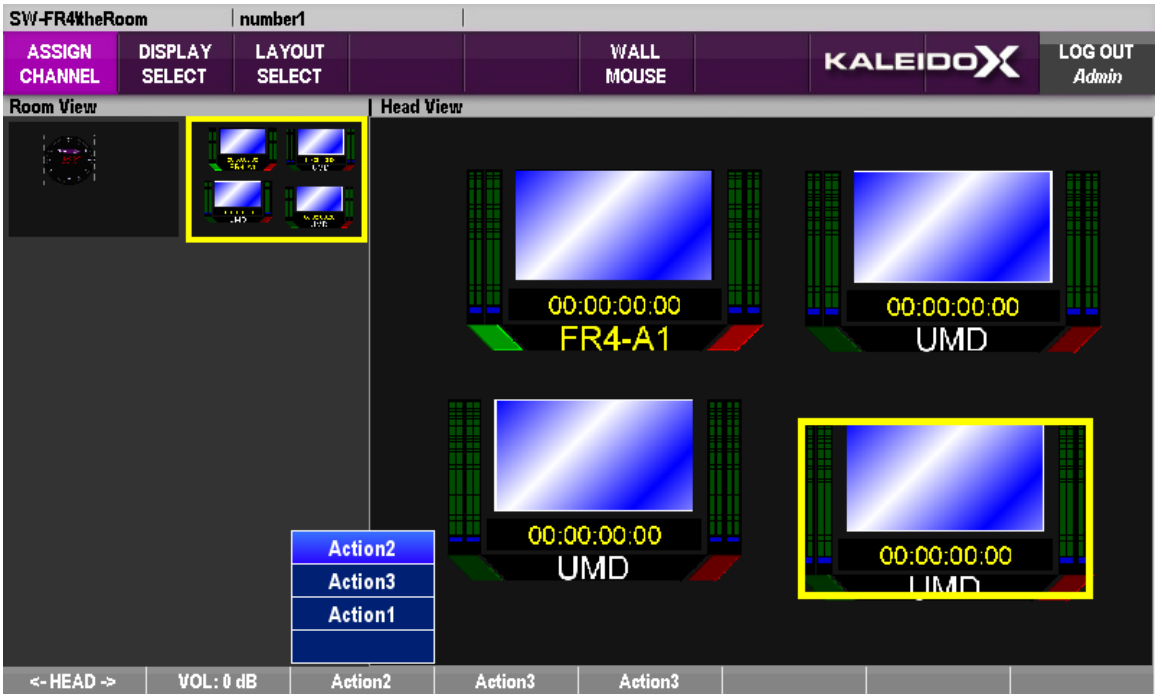
2.4 Kaleido

2.4.1 KXLayouts

This widget is used to select a KX layout from an iControl page. When selected, a list with all the available layouts is displayed. Selecting one layout from that list will trigger a layout change on the KX.

Parameter Name	Parameter Type	Expected Value
KX_IPaddress	String	The KX Ip
KX_ROOM	String	The room for which all the layouts should be displayed. If left empty, all the rooms with all the layouts will be shown.

2.4.2 KX Control Panel



This widget is used to control KX wall from an iControl page. Features include:

- Ability to control the KX wall with the mouse and keyboard
- Loading of layouts with image preview
- Control of audio monitoring functions
- Firing of background actions with programmable buttons

KX wall control:
Mouse control of the KX wall can be enabled by pressing the “WALL MOUSE” button. Mouse control is transferred from the local GUI to the KX wall. Since local control of the mouse is lost, exiting this mode requires typing ALT+X. This mode will also exit when application focus is lost, for example by pressing ALT+TAB in Windows to switch applications.

When in this mode, in lieu of mouse clicks the 8 bottom buttons can be virtually clicked by typing SHIFT+F1 … SHIFT+F8.

Loading layouts:
KX wall layouts can be loaded by clicking the “LAYOUT SELECT” button. A selection screen is shown with which allows the previewing of the layout to be loaded on the wall. Clicking the “TAKE” button will load the layout on the wall and update the local GUI.

Audio monitoring:
The second bottom button can be used to control audio monitoring functions for the current user (recall that KX audio monitoring is specific to each user). Clicking on that button will toggle between the different modes: NORMAL, MUTE, and -20dB. The monitoring output level can be controlled using the mouse scroll wheel.

Programmable buttons:
The programmable buttons allow the firing of either background actions (as defined in XEdit) or KX wall functions. To program the button, click on the ▼ button to display the selection list. Double click the desired background action or KX wall function.

Scripting:
A callback function can be specified that will be called whenever a KX head or monitor button is selected. The provided function, `buttonSelectionHandler`, will be called with a single JavaScript object as a parameter that will specify attributes about the selected button. The following tables describes what attributes will be set, depending on the type of button clicked.

MONITOR attribute	Description
buttonType	Will be set to “MONITOR”
mouseEvent	The associated W3C DOM level 2 MouseEvent. Can be used to determine which mouse button was pressed or location of the click, for example. See the JavaScript scripting API help documents for more info.
destinationIndex	The associated router destination index
systemName	The system name of the room Kaleido
frameReference	The system name of the Kaleido, if part of a cluster
logicalRouterName	The logical router name for this monitor destination
logicalRouterLevel	The logical router level for this monitor destination

HEAD attribute	Description
buttonType	Will be set to “HEAD”
mouseEvent	The associated W3C DOM level 2 MouseEvent. Can be used to determine which mouse button was pressed or location of the click, for example. See the JavaScript scripting API help documents for more info.
headID	The physical ID of the head
logicalID	The logical ID of the head
frameName	The system name of the Kaleido

Example script:

```
var kxWidget = getElementById("kxControlPanel0");
kxWidget.buttonSelectionHandler = function(clickInfo) {
      var message = "";
      if (clickInfo.buttonType == "HEAD") {
            message += "Head " + clickInfo.logicalID + " clicked";
      } else if (clickInfo.buttonType == "MONITOR") {
            message += "Monitor button clicked";
            message += "\nDest index: " + clickInfo.destinationIndex;
            message += "\nRouter ID: " + clickInfo.logicalRouterName;
      }
      var mouseEvent = clickInfo.mouseEvent;
      message += "\nClick location: (" + mouseEvent.clientX + ", " +
                  mouseEvent.clientY + ")";
      window.alert(message);
}
```

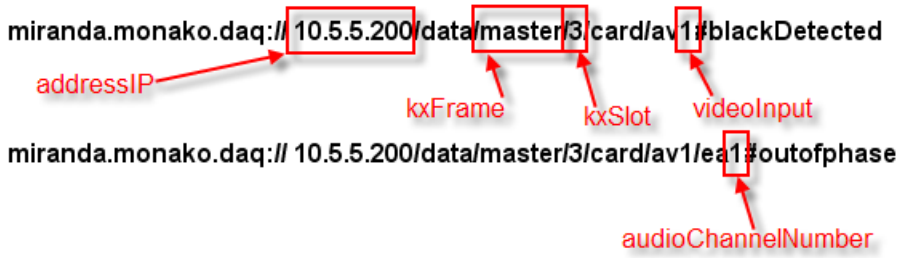
Parameter Name	Parameter Type	Expected Value
kxIP	String	The KX IP address. If left empty, a list of rooms discovered via network multicast will be displayed in a selection list.
roomName	String	The desired room to control. If left empty, the list of rooms from the specified KX will be displayed in a selection list.
userID	String	The user ID to enable auto-login. If left empty, the list of registered users will be displayed in a selection list, along with a numeric keypad to enter the password
userPW	String	The user password for the specified userID parameter.

2.4.3 KXInGSMEmbedAlm

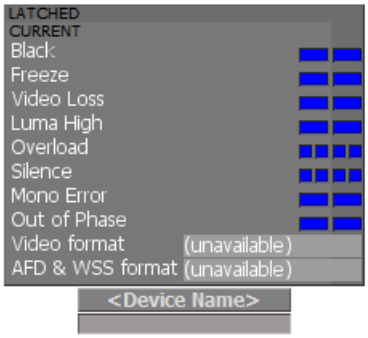
This widget is used to display the alarms for a particular KX input. Note that these alarms are displayed using traps.

Parameter Name	Parameter Type	Expected Value
addressIP	String	The KX Ip
audioChannelNumber	String	The index of the audio input (1,2, …)
inputAlarm	Alarm URI	Overall alarm of the input
kxFrame	String	**master** for main frame, **salve** for expansion frame
kxInputId	String	Name to be displayed on the button
kxSlot	string	Integer; starts at 3 for main frame and at 19 for expansion frame
label	String	Location of the KX or any other information pertinent to the KX or the input alarm
page	Page selector	Page that should be displayed when the KX button is selected. If left empty, the KX admin page will be displayed
popupWindowName	String	**applBrowserWindow.** If left empty nothing will be displayed.
videoInput	String	The index of the video input (1,2, …)

Sample URI with assignments:



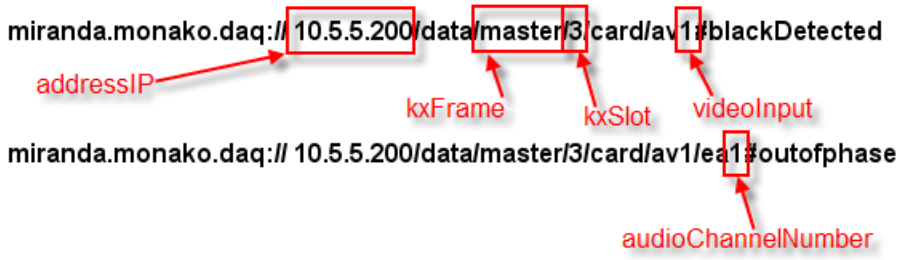
2.4.4 KXInGSMEmbedAlmExtended



This widget is used to display the alarms for a particular KX input. Note that these alarms are displayed using traps.

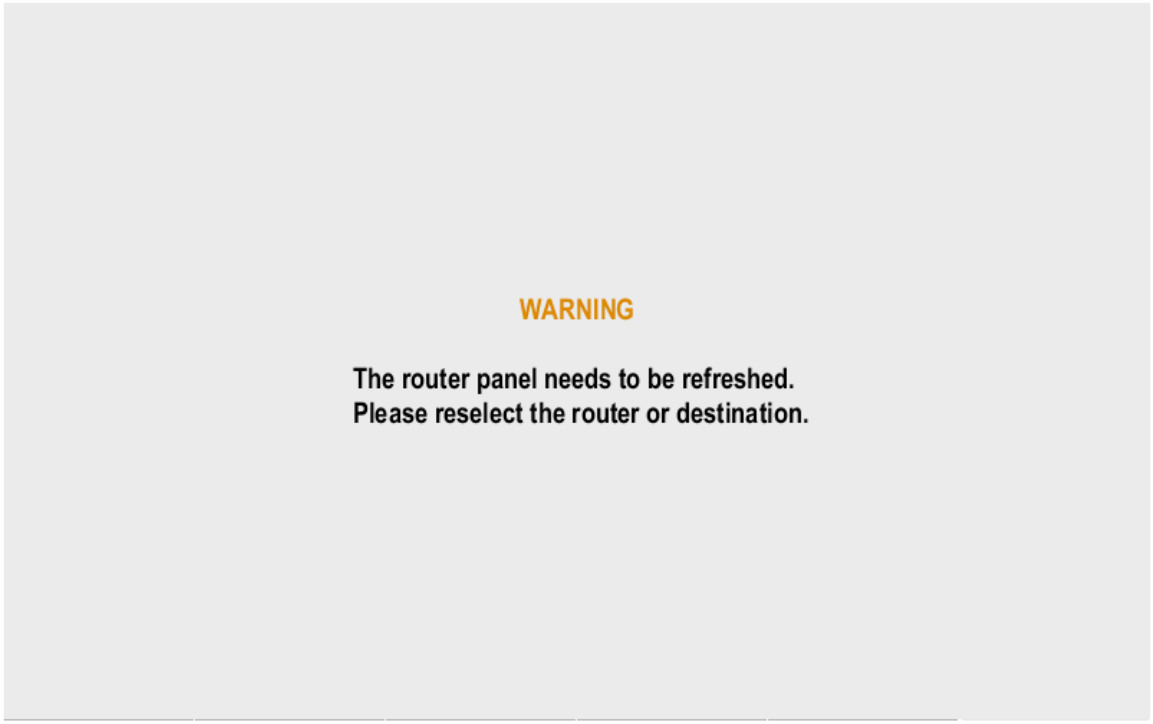
Parameter Name	Parameter Type	Expected Value
addressIP	String	The KX Ip
audioChannelNumber	String	The index of the audio input (1,2, …)
inputAlarm	Alarm URI	Overall alarm of the input
kxFrame	String	**master** for main frame, **salve** for expansion frame
kxInputId	String	Name to be displayed on the button
kxSlot	string	Integer; starts at 3 for main frame and at 19 for expansion frame
label	String	Location of the KX or any other information pertinent to the KX or the input alarm
page	Page selector	Page that should be displayed when the KX button is selected. If left empty, the KX admin page will be displayed
popupWindowName	String	**applBrowserWindow.** If left empty nothing will be displayed.
videoInput	String	The index of the video input (1,2, …)

Sample URI with assignments:



2.5 RCP-200

2.5.1 routerWidgetCatIndex

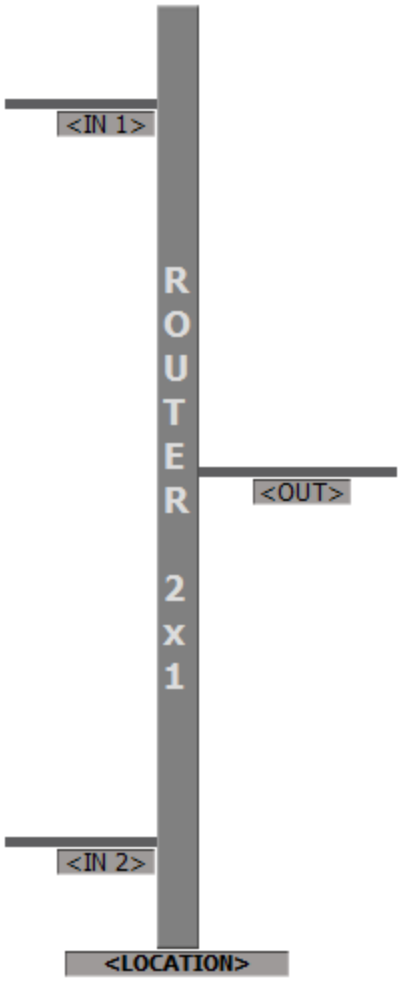


This widget enables the operator to control the router by selecting the input to be routed to a specific destination. It is similar to the cat dialer widget found on the RCP-200.

Parameter Name	Parameter Type	Expected Value
iDst	String	Destination to which a source would be routed.
iLevel	String	Router level.
iRouterID	String	ID of the router to be controlled.

2.6 Routers

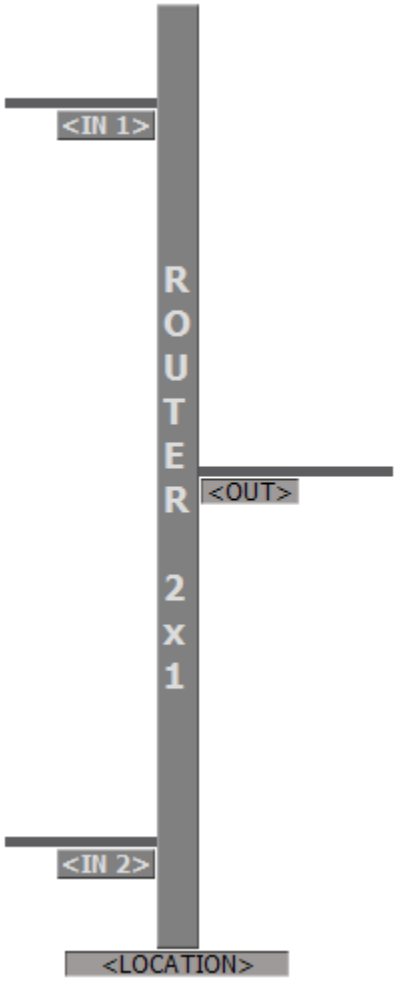
2.6.1 Router_2x1



This widget should be used in a signal path, when the user wants to monitor a 2x1 router.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

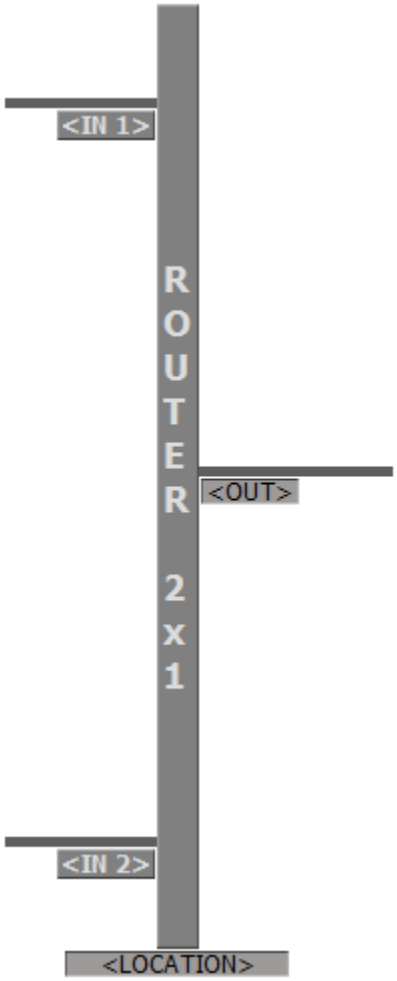
2.6.2 RouterControl_2x1



This widget should be used in a signal path, when the user wants to monitor and control a 2x1 router.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

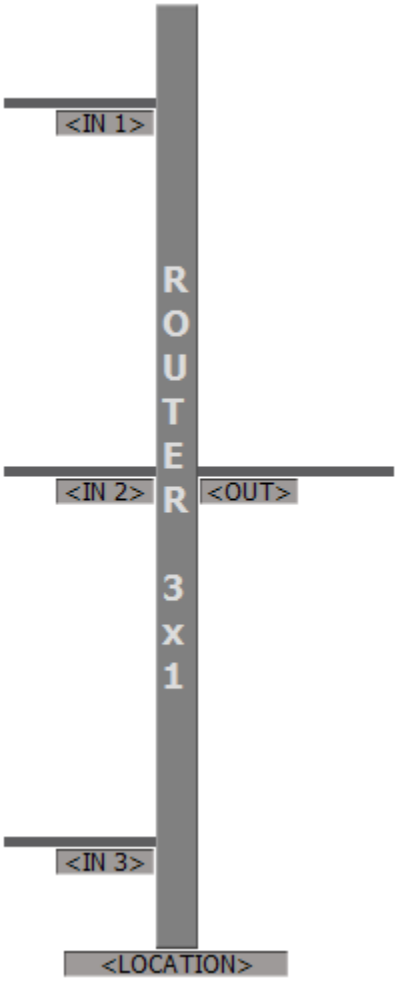
2.6.3 RouterControlStatus_2x1



This widget should be used in a signal path, when the user wants to monitor and control a 2x1 router. In addition, it will also display the status of the inputs.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

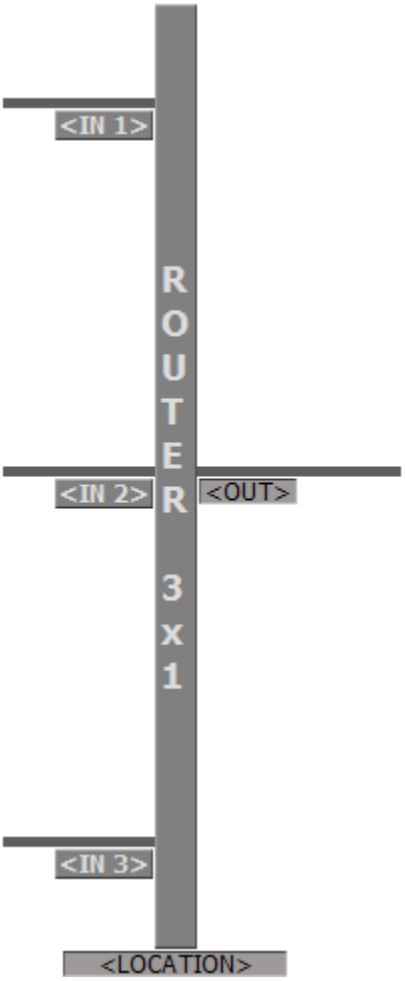
2.6.4 Router_3x1



This widget should be used in a signal path, when the user wants to monitor a 3x1 router.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
IN_3	String	Third input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

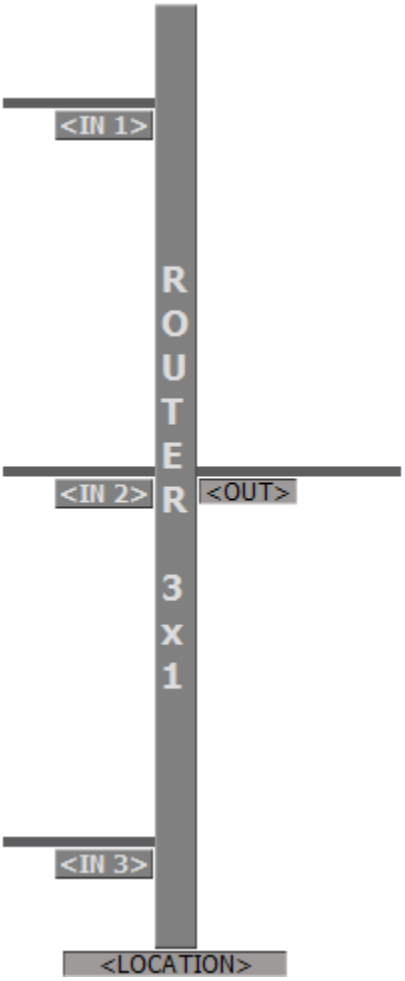
2.6.5 RouterControl_3x1



This widget should be used in a signal path, when the user wants to monitor and control a 3x1 router.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
IN_3	String	Third input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

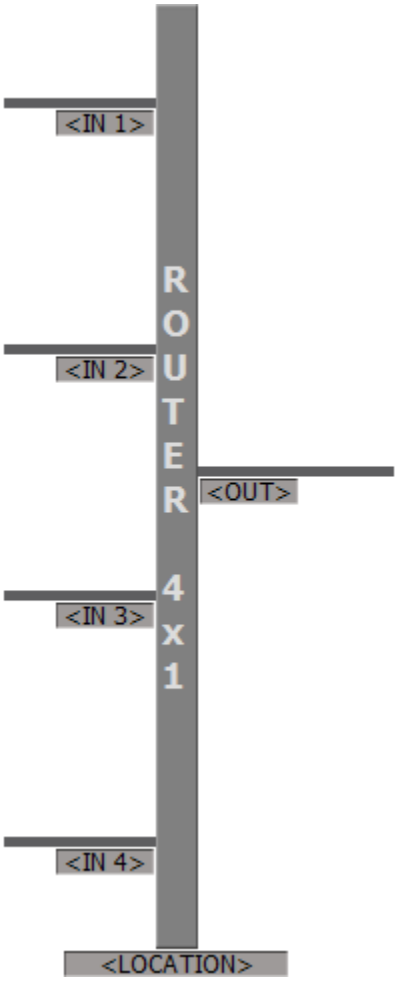
2.6.6 RouterControlStatus_3x1



This widget should be used in a signal path, when the user wants to monitor and control a 3x1 router. In addition, it will also display the status of the inputs.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
IN_3	String	Third input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

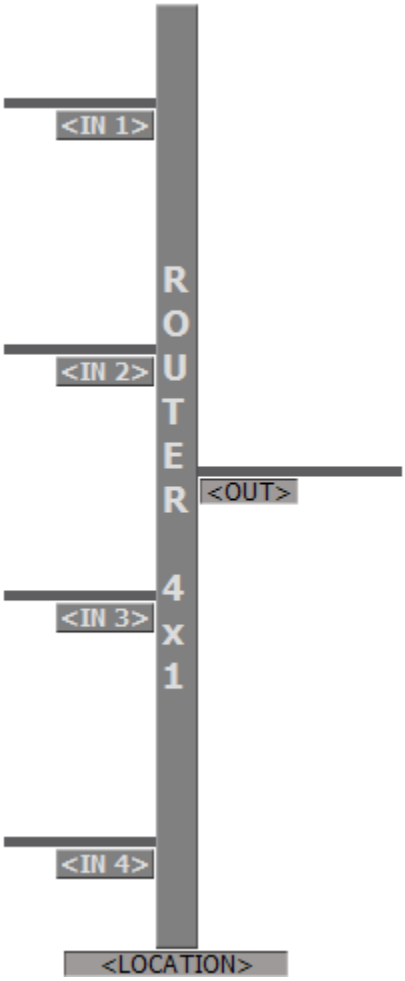
2.6.7 Router_4x1



This widget should be used in a signal path, when the user wants to monitor a 4x1 router.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
IN_3	String	Third input id
IN_4	String	Fourth input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

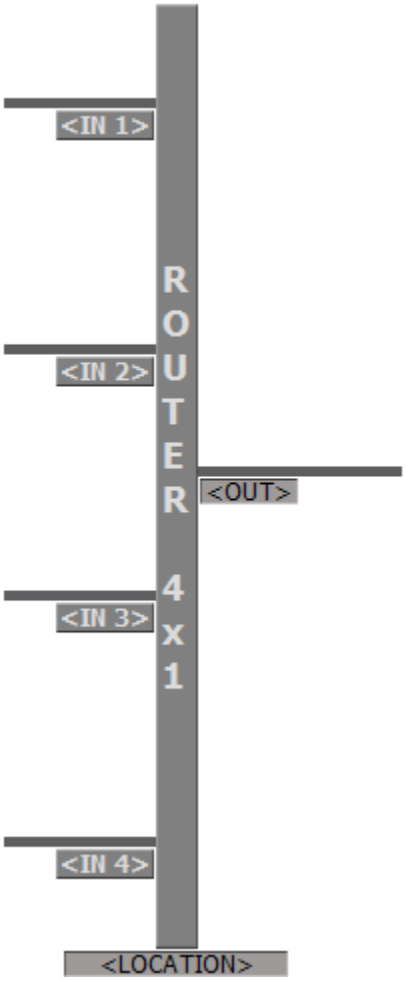
2.6.8 RouterControl_4x1



This widget should be used in a signal path, when the user wants to monitor and control a 4x1 router.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
IN_3	String	Third input id
IN_4	String	Fourth input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

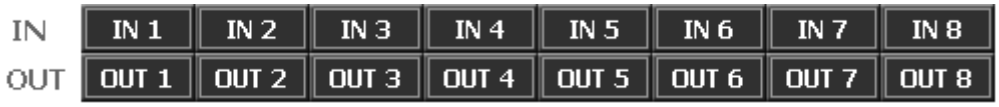
2.6.9 RouterControlStatus_4x1



This widget should be used in a signal path, when the user wants to monitor and control a 4x1 router. In addition, it will also display the status of the inputs.

Parameter Name	Parameter Type	Expected Value
IN_1	String	First input id
IN_2	String	Second input id
IN_3	String	Third input id
IN_4	String	Fourth input id
destination	String	Destination id
level	String	Level id
location	String	Physical location of the router
routerId	Router identification	The router's id to be selected from the browse menu

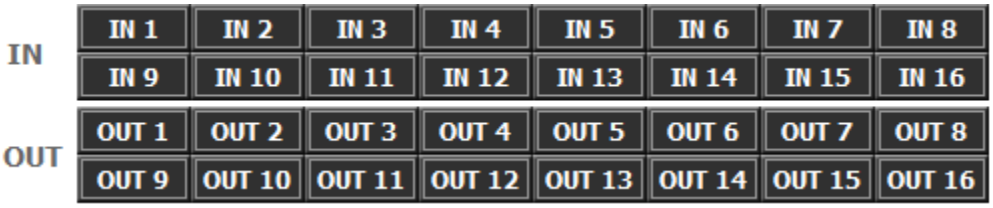
2.6.10 Router_8x8



This widget should be used for an 8x8 router. It depicts the inputs and outputs of a router. When an output is selected, the corresponding input is shown. The user is able to change the input for a selected output.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.11 Router_16x16



This widget should be used for a 16x16 router. It depicts the inputs and outputs of a router. When an output is selected, the corresponding input is shown. The user is able to change the input for a selected output.

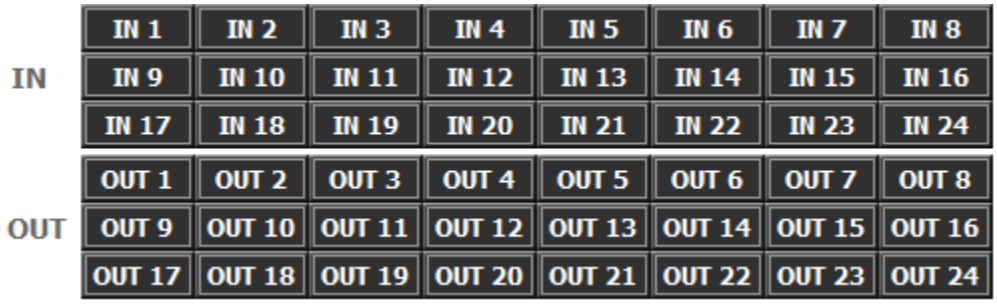
Parameter Name	Parameter Type	Expected Value
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.12 Router_16x2

This widget should be used for a 16x2 router. It depicts the inputs and outputs of a router. When an output is selected, the corresponding input is shown. The user is able to change the input for a selected output.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

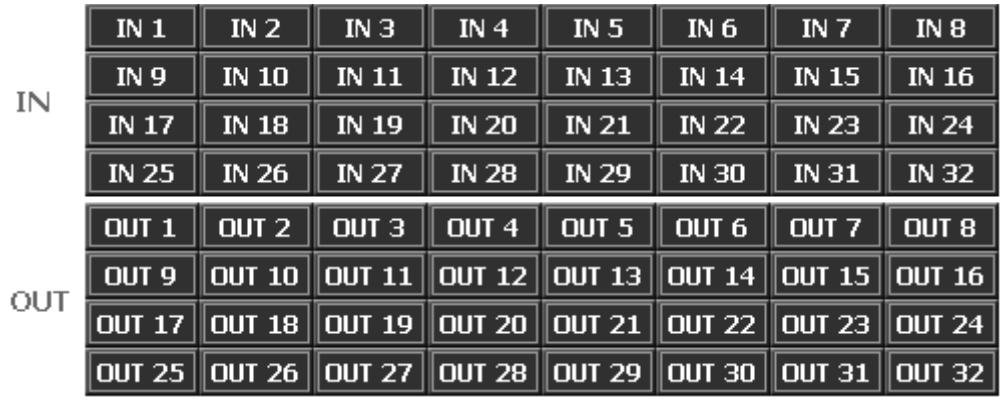
2.6.13 Router_24x24



This widget should be used for a 24x24 router. It depicts the inputs and output of a router. When an output is selected, the corresponding input is shown. The user is able to change the input for a selected output.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

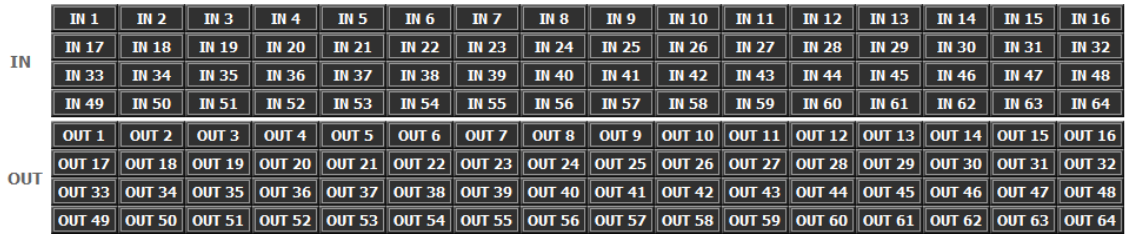
2.6.14 Router_32x32



This widget should be used for a 32x32 router. It depicts the inputs and output of a router. When an output is selected, the corresponding input is shown. The user is able to change the input for a selected output.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.16 Router_64x64



This widget should be used for a 64x64 router. It depicts the inputs and output of a router. When an output is selected, the corresponding input is shown. The user is able to change the input for a selected output.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.17 RouterSinglebus_8

This widget should be used in the source selector area for an 8x1 single bus router. The label of each input will be shown on the buttons as well as the current selection.

Parameter Name	Parameter Type	Expected Value
destination	String	Destination id
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.18 RouterSinglebus_16

This widget should be used in the source selector area for a 16x1 single bus router. The label of each input will be shown on the buttons as well as the current selection.

Parameter Name	Parameter Type	Expected Value
destination	String	Destination id
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.19 RouterSinglebus_24

This widget should be used in the source selector area for a 24x1 single bus router. The label of each input will be shown on the buttons as well as the current selection.

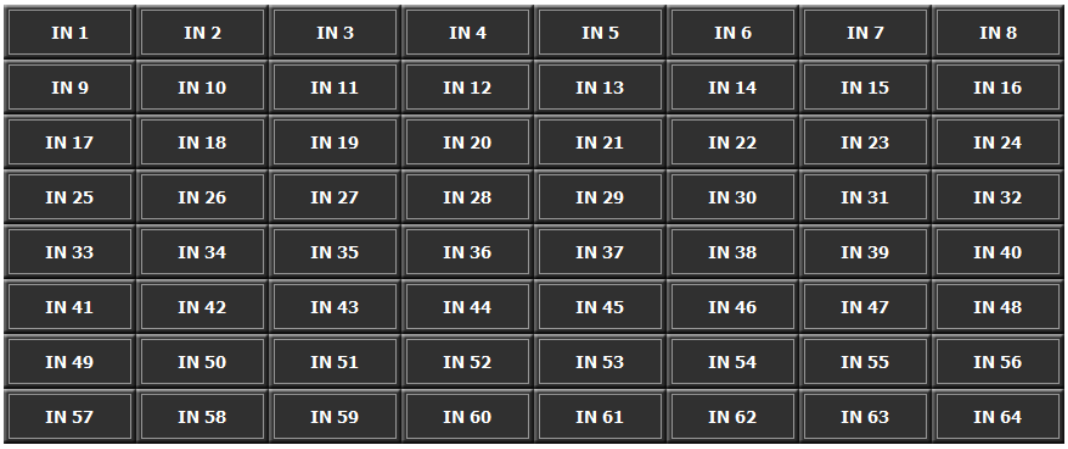
Parameter Name	Parameter Type	Expected Value
destination	String	Destination id
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.20 RouterSinglebus_32

This widget should be used in the source selector area for a 32x1 single bus router. The label of each input will be shown on the buttons as well as the current selection.

Parameter Name	Parameter Type	Expected Value
destination	String	Destination id
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.21 RouterSinglebus_64



This widget should be used in the source selector area for a 64x1 single bus router. The label of each input will be shown on the buttons as well as the current selection.

Parameter Name	Parameter Type	Expected Value
destination	String	Destination id
level	String	Level id
routerId	Router identification	The router's id to be selected from the browse menu

2.6.22 RouterSinglebusSelectDestination_8

This widget should be used in the source selector area for an 8x1 single bus router. The label of each input will be shown on the buttons as well as the current selection. The output button will show the currently selected destination.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
maxOutputs	String	Outputs to be shown in the menu; max is 8; menu is defaulted to 8.
routerId	Router identification	The router's id to be selected from the browse menu

2.6.23 RouterSinglebusSelectDestination_16

This widget should be used in the source selector area for a 16x1 single bus router. The label of each input will be shown on the buttons as well as the current selection. The output button will show the currently selected destination.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
maxOutputs	String	Outputs to be shown in the menu; max is 16; menu is defaulted to 16.
routerId	Router identification	The router's id to be selected from the browse menu

2.6.24 RouterSinglebusSelectDestination _24

This widget should be used in the source selector area for a 24x1 single bus router. The label of each input will be shown on the buttons as well as the current selection. The output button will show the currently selected destination.

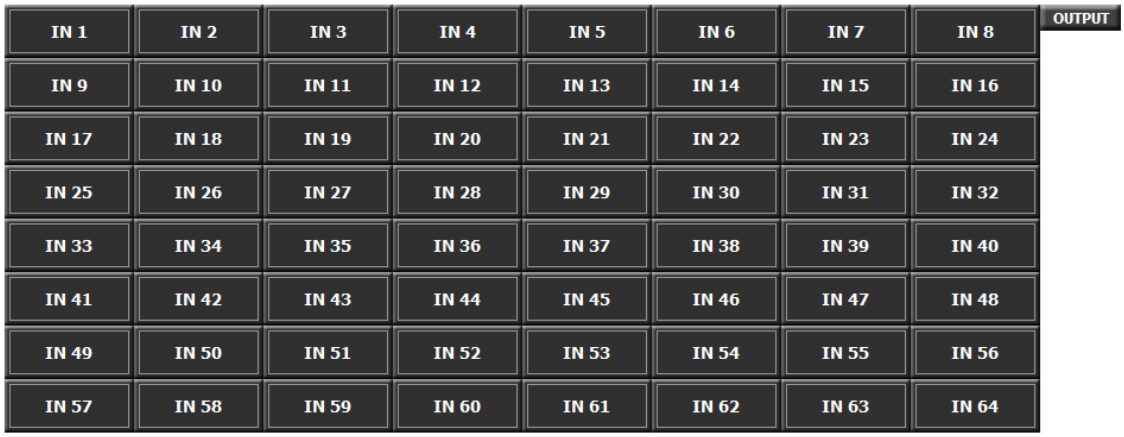
Parameter Name	Parameter Type	Expected Value
level	String	Level id
maxOutputs	String	Outputs to be shown in the menu; max is 24; menu is defaulted to 24.
routerId	Router identification	The router's id to be selected from the browse menu

2.6.25 RouterSinglebusSelectDestination_32

This widget should be used in the source selector area for a 32x1 single bus router. The label each input will be shown on the buttons as well as the current selection. The output button will show the currently selected destination.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
maxOutputs	String	Outputs to be shown in the menu; max is 32; menu is defaulted to 32.
routerId	Router identification	The router's id to be selected from the browse menu

2.6.26 RouterSinglebusSelectDestination_64

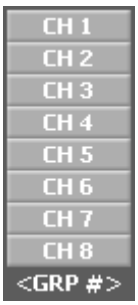


This widget should be used in the source selector area for a 64x1 single bus router. The label each input will be shown on the buttons as well as the current selection. The output button will show the currently selected destination.

Parameter Name	Parameter Type	Expected Value
level	String	Level id
maxOutputs	String	Outputs to be shown in the menu; max is 64; menu is defaulted to 64.
routerId	Router identification	The router's id to be selected from the browse menu

2.7 Sources

2.7.1 SourcesGroup_8



This widget should be used in the source selector area, for sources that need to be shown in groups of eight. The user can decide the number of channels to be shown in the group by entering the first and last channel id, to a maximum of eight. The numbers are sequential, but they do not have to start at 1 (e.g. 9 to 14 is a valid entry). Each button corresponds to a page that will be displayed in a predefined zone. The overall status of each page will be shown on the button. The currently selected page will also be shown. If the zone name is left empty, selecting a button will have no action. The user can also change the name of the buttons. If the names are left empty then the channels are defaulted to numbers.

Parameter Name	Parameter Type	Expected Value
firstChannelId	String	Id of the first source (defaulted to 1)
groupName	String	Name of the group (defaulted to GR 1)
lastChannelId	String	Id of the last source (defaulted to 8)
pageCh1 to pageCh8	Page selector	Page for the 1st source to 8th source
pageNameCh1 to pageNameCh8	String	Name for 1st source to 8th source
pageZoneName	Zone selector	Zone in which the pages are displayed

2.7.2 SourcesGroup_12

This widget should be used in the source selector area, for sources that need to be shown in groups of 12. The user can decide the number of channels to be shown in the group by entering the first and last channel id, to a maximum of eight. The numbers are sequential, but they do not have to start at 1 (e.g. 24 to 30 is a valid entry). Each button corresponds to a page that will be displayed in a predefined zone. The overall status of each page will be shown on the button. The currently selected page will also be shown. If the zone name is left empty, selecting a button will have no action. The user can also change the name of the buttons. If the names are left empty then the channels are defaulted to numbers.

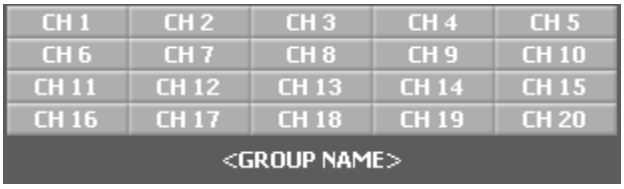
Parameter Name	Parameter Type	Expected Value
firstChannelId	String	Id of the first source (defaulted to 1)
groupName	String	Name of the group (defaulted to GROUP 1)
lastChannelId	String	Id of the last source (defaulted to 12)
pageCh1 to pageCh12	Page selector	Page for the 1st source to 12th source
pageNameCh1 to pageNameCh12	String	Name for 1st source to 12th source
pageZoneName	Zone selector	Zone in which the pages are displayed

2.7.3 SourcesGroup _16

This widget should be used in the source selector area, for sources that need to be shown in groups of 16. The user can decide the number of channels to be shown in the group by entering the first and last channel id, to a maximum of eight. The numbers are sequential, but they do not have to start at 1 (e.g. 32 to 34 is a valid entry). Each button corresponds to a page that will be displayed in a predefined zone. The overall status of each page will be shown on the button. The currently selected page will also be shown. If the zone name is left empty, selecting a button will have no action. The user can also change the name of the buttons. If the names are left empty then the channels are defaulted to numbers.

Parameter Name	Parameter Type	Expected Value
firstChannelId	String	Id of the first source (defaulted to 1)
groupName	String	Name of the group (defaulted to GROUP 1)
lastChannelId	String	Id of the last source (defaulted to 16)
pageCh1 to pageCh16	Page selector	Page for the 1st source to 16th source
pageNameCh1 to pageNameCh16	String	Name for the 1st source to 16th source
pageZoneName	Zone selector	Zone in which the pages are displayed

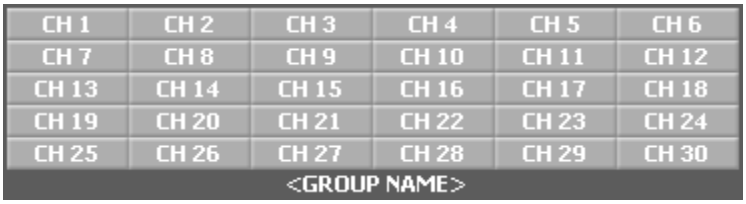
2.7.4 SourcesGroup _20



This widget should be used in the source selector area, for sources that need to be shown in groups of 20. The user can decide the number of channels to be shown in the group by entering the first and last channel id, to a maximum of eight. The numbers are sequential, but they do not have to start at 1 (e.g. 40 to 46 is a valid entry). Each button corresponds to a page that will be displayed in a predefined zone. The overall status of each page will be shown on the button. The currently selected page will also be shown. If the zone name is left empty, selecting a button will have no action. The user can also change the name of the buttons. If the names are left empty then the channels are defaulted to numbers.

Parameter Name	Parameter Type	Expected Value
firstChannelId	String	Id of the first source (defaulted to 1)
groupName	String	Name of the group (defaulted to GROUP 1)
lastChannelId	String	Id of the last source (defaulted to 20)
pageCh1 to pageCh20	Page selector	Page for the 1st source to 20th source
pageNameCh1 to pageNameCh20	String	Name for the 1st source to 20th source
pageZoneName	Zone selector	Zone in which the pages are displayed

2.7.5 SourcesGroup _30



This widget should be used in the source selector area, for sources that need to be shown in groups of 30. The user can decide the number of channels to be shown in the group by entering the first and last channel id, to a maximum of eight. The numbers are sequential, but they do not have to start at 1 (e.g. 90 to 107 is a valid entry). Each button corresponds to a page that will be displayed in a predefined zone. The overall status of each page will be shown on the button. The currently selected page will also be shown. If the zone name is left empty, selecting a button will have no action. The user can also change the name of the buttons. If the names are left empty then the channels are defaulted to numbers.

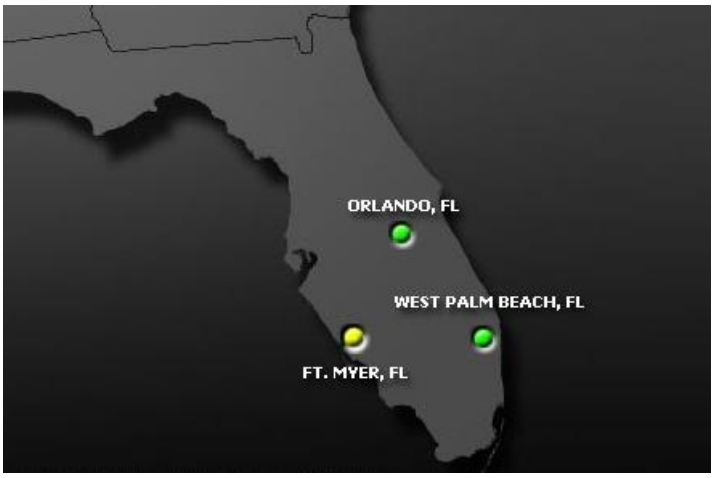
Parameter Name	Parameter Type	Expected Value
firstChannelId	String	Id of the first source (defaulted to 1)
groupName	String	Name of the group (defaulted to GROUP 1)
lastChannelId	String	Id of the last source (defaulted to 30)
pageCh1 to pageCh30	Page selector	Page for the 1st source to 30th source
pageNameCh1 to pageNameCh30	String	Name for the 1st source to 30th souce
pageZoneName	Zone selector	Zone in which the pages are displayed

3. Creating a new widget using iControl Creator

The following sections will provide two examples of how to create new widgets using iControl Creator. It is assumed that the user is knowledgeable of iControl Creator and JavaScript. The first example is the "status_indicator" widget, a status icon to be placed on maps at a specific location to indicate the overall status of that location. In the second example, the "card_panel_message" widget will show how to interface with a densite card and display the value of a specific parameter on a page.

3.1 Creating the "status_indicator" widget

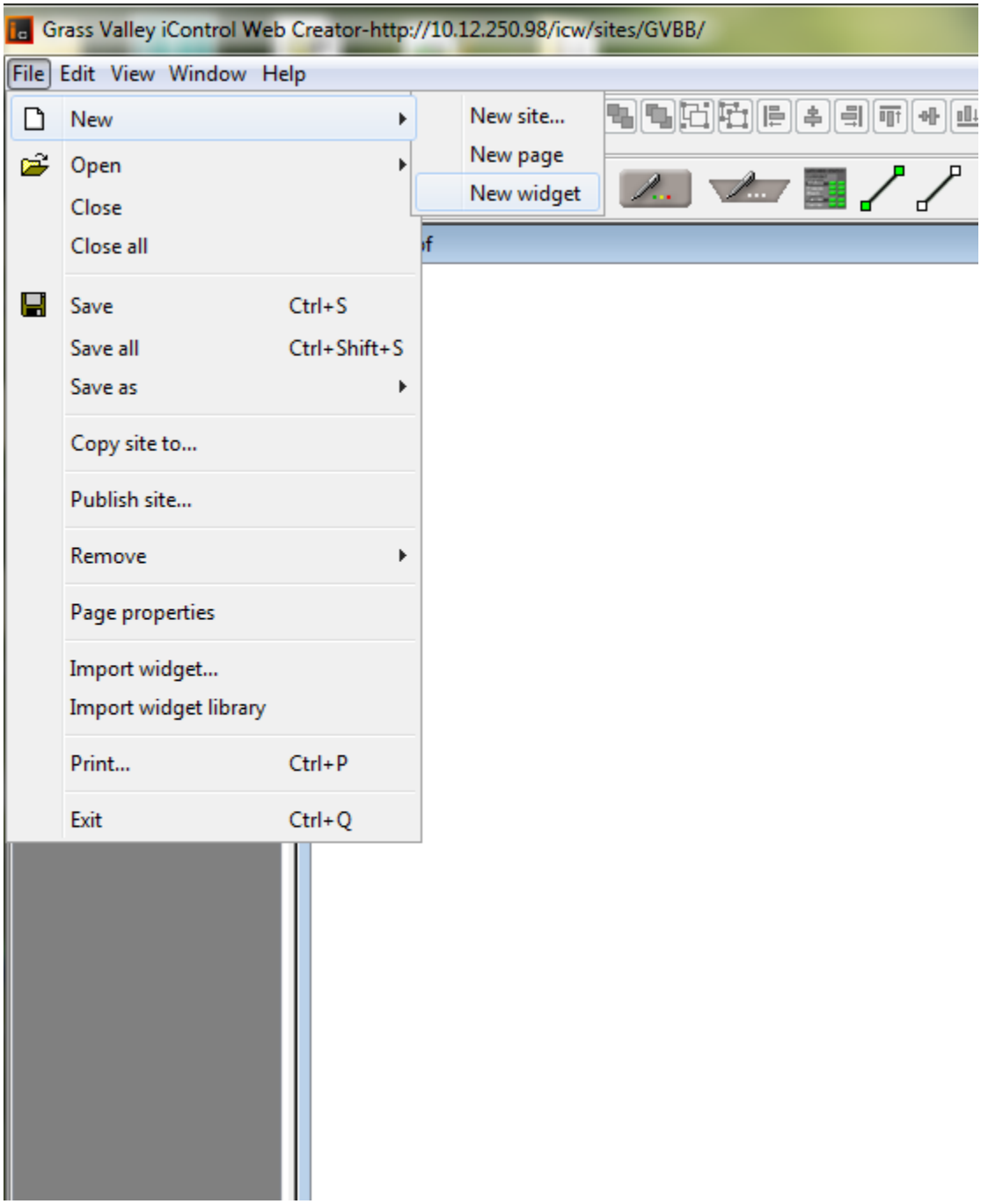
The "status_indicator" widget will be placed on a map to show the overall status at a specific location (an example is shown below). In addition, when the "status_indicator" widget is selected it will display a page that will present a path used for monitoring and controlling devices at the selected location.



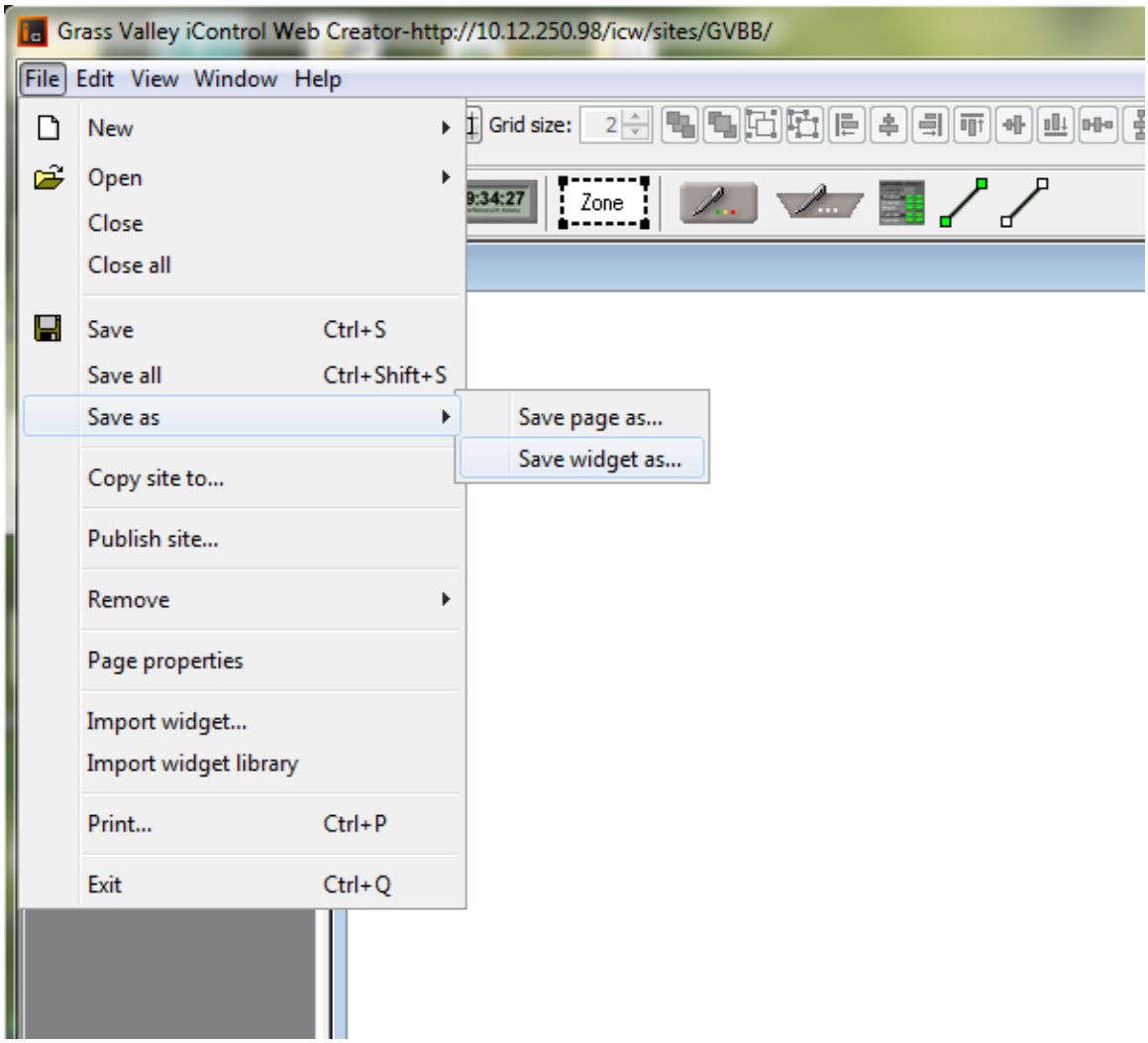
In iControl Creator, the widget will look as shown below:

To create this widget, the following steps must be performed:

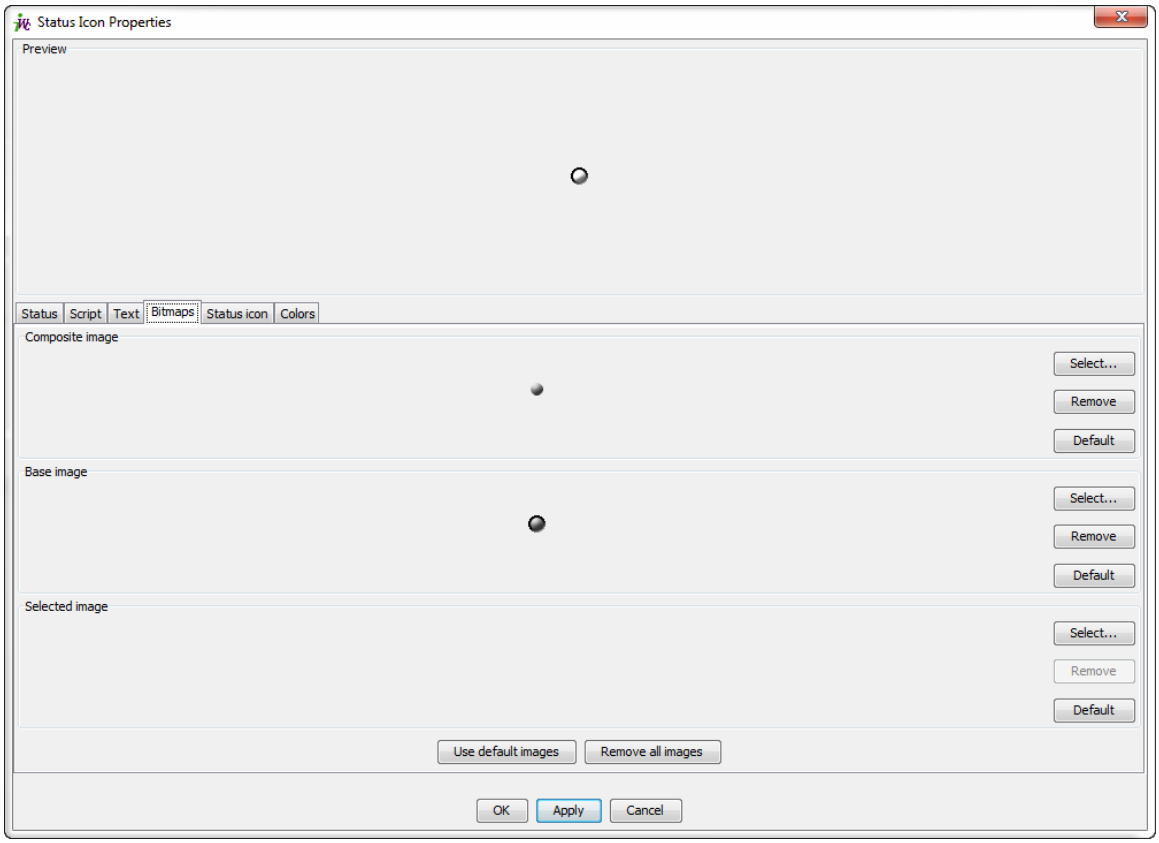
1. Select *New widget* from the *File* menu.



2. Save the newly opened widget as "*status-indicator.mpf*" using the *Save widget as…* option from the *Save as* menu.

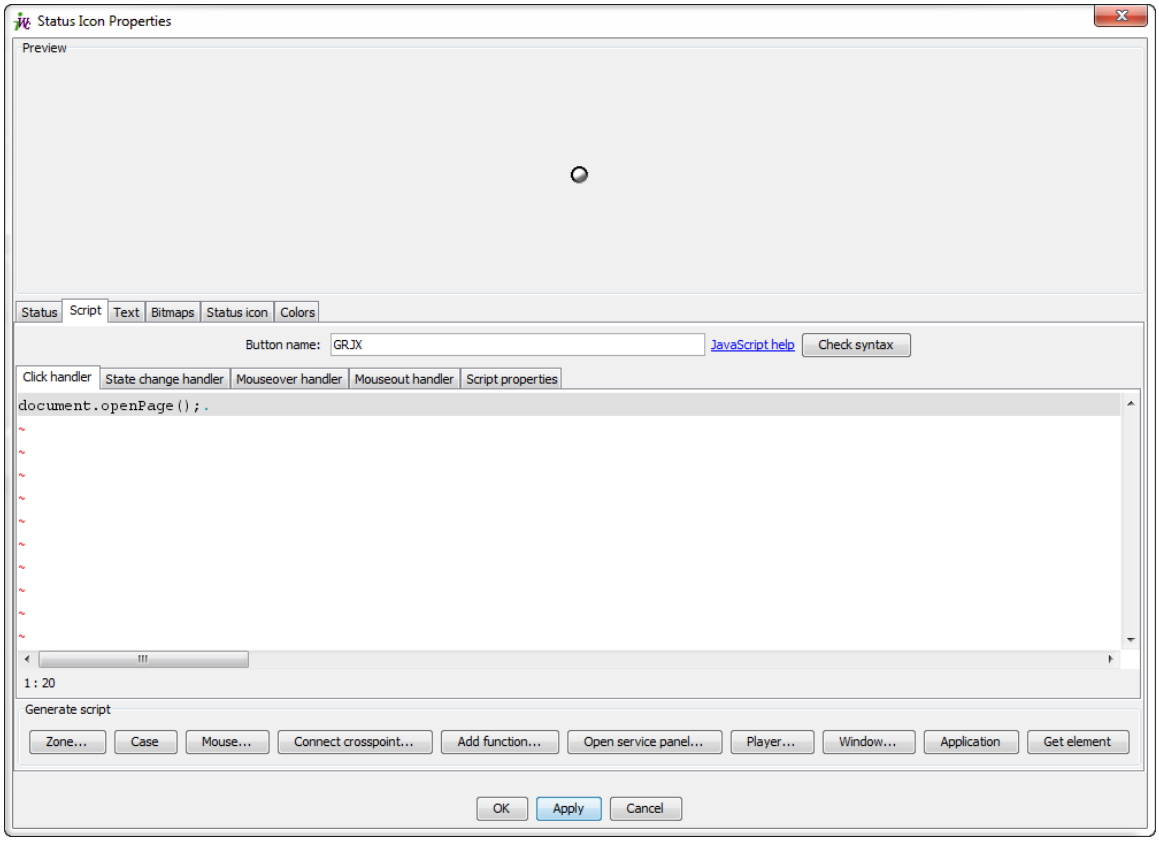

Grass Valley iControl Web Creator-http://10.12.250.98/icw/sites/GVBB/
File Edit View Window Help
New
Open
Close
Close all
Save Ctrl+S
Save all Ctrl+Shift+S
Save as
Copy site to...
Publish site...
Remove
Page properties
Import widget...
Import widget library
Print... Ctrl+P
Exit Ctrl+Q
Save page as...
Save widget as...
Grid size:
Zone

On the newly created widget add a button from the iControl Control toolbar and add the image for the location from the image library as shown below.

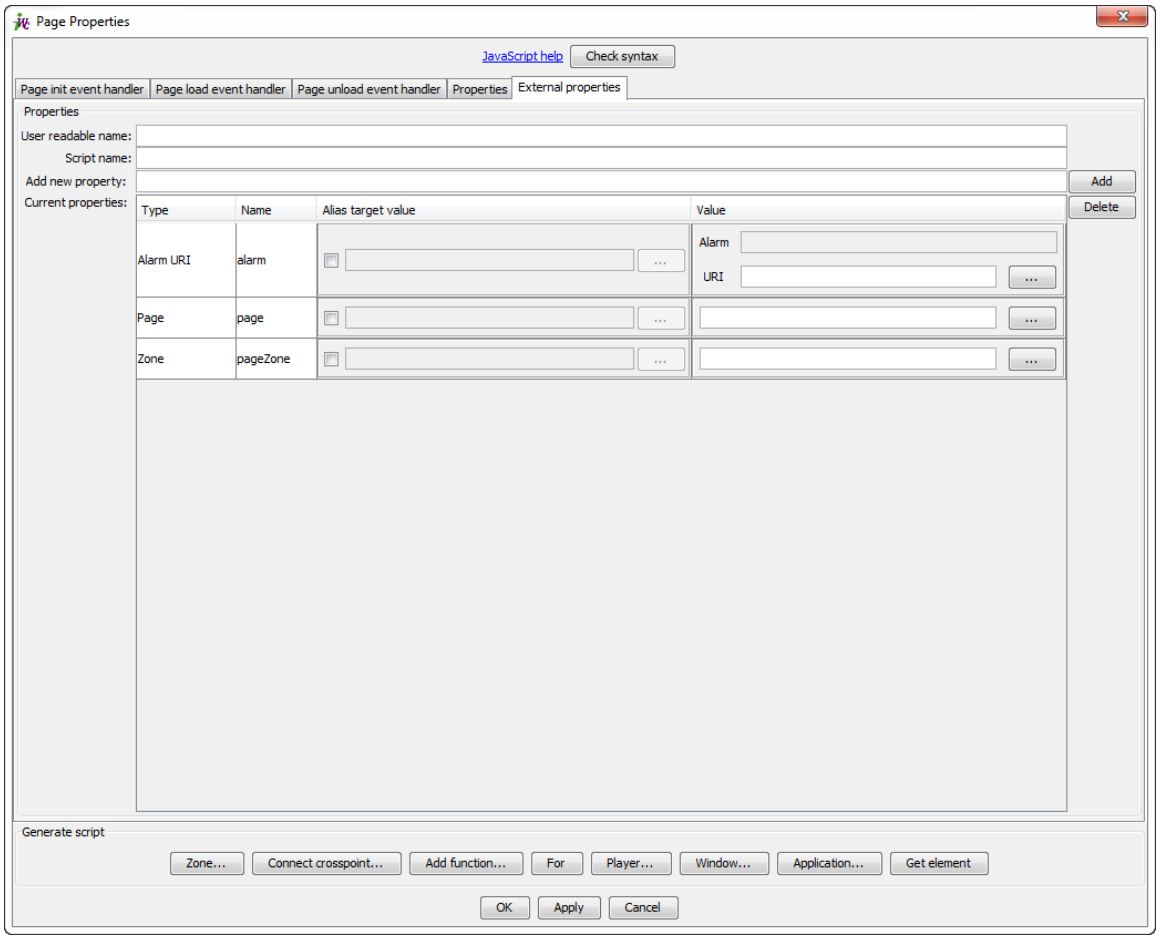


Since this button is supposed to show status, an image for the status was added in the *Composite image* section.

3. In the same popup as before, under the *Scripts* tab, name the button *location* as shown. Also, in the *Click handler* section add a new function called *openPage()*(this function will be detailed in the following steps).



4. Select the *Page properties* option from the *File* menu. Next, from the *Page properties* window select the *External properties* tab. This widget will be used to indicate a status alarm and to open a page when selected. Thus, the alarm URI, page name and zone in which the page should be displayed must be entered as external parameters. To enter these parameters, on the *Add new property:* line, enter the parameters one by one as indicated below. After typing the name of each parameter select the *Add* button. This will make the name of the parameter appear in the list area of the window (to remove a property already entered in the list, select the property line first then select the *Remove* button). By default, each property is entered as a *String* type. To change the type of a property, select its type name and then from the drop down menu select the new required type. After all the properties are entered and the types modified (at this point the window should look as shown below), select the *Apply* button to save those changes and to keep the window open for the next steps.



5. With the *Page properties* window still open select the *Page load event handler* tab and enter the following:

   ```
   getElementById("location").alarms.status = alarm;
   ```

   This will associate the alarm specified in the parameter with the *location* button. After entering the code, select the *Apply* button to save the changes and keep the *Page properties* window open.

6. With the *Page properties* window still open select the *Page init event handler* tab and enter the following:

   ```
   var zone = null;

   var homeDoc = navigator.windows[0].document;

   this.openPage = function() {
       if (pageZone == "") {
         return;
       }
       else {
         if (zone == undefined ) {
           temp = pageZone.indexOf(".") + 1;
           zone = homeDoc.getElementById(pageZone.substring(temp));
         }

         try{
           zone.loadDocument(page);
         }
         catch (ex) {
           window.alert("The page name you entered for is not valid or null.");
         }
       }
   }
   ```

   This will open the page specified in the *page* parameter, in the *pageZone* also specified in the parameters. Moreover, if nothing is entered for the zone, no page will be displayed. This will allow the use of this widget as a status indicator only as well, for location where no pages are required. After entering the code, select the *Save* button to save and close the *Page properties* window.

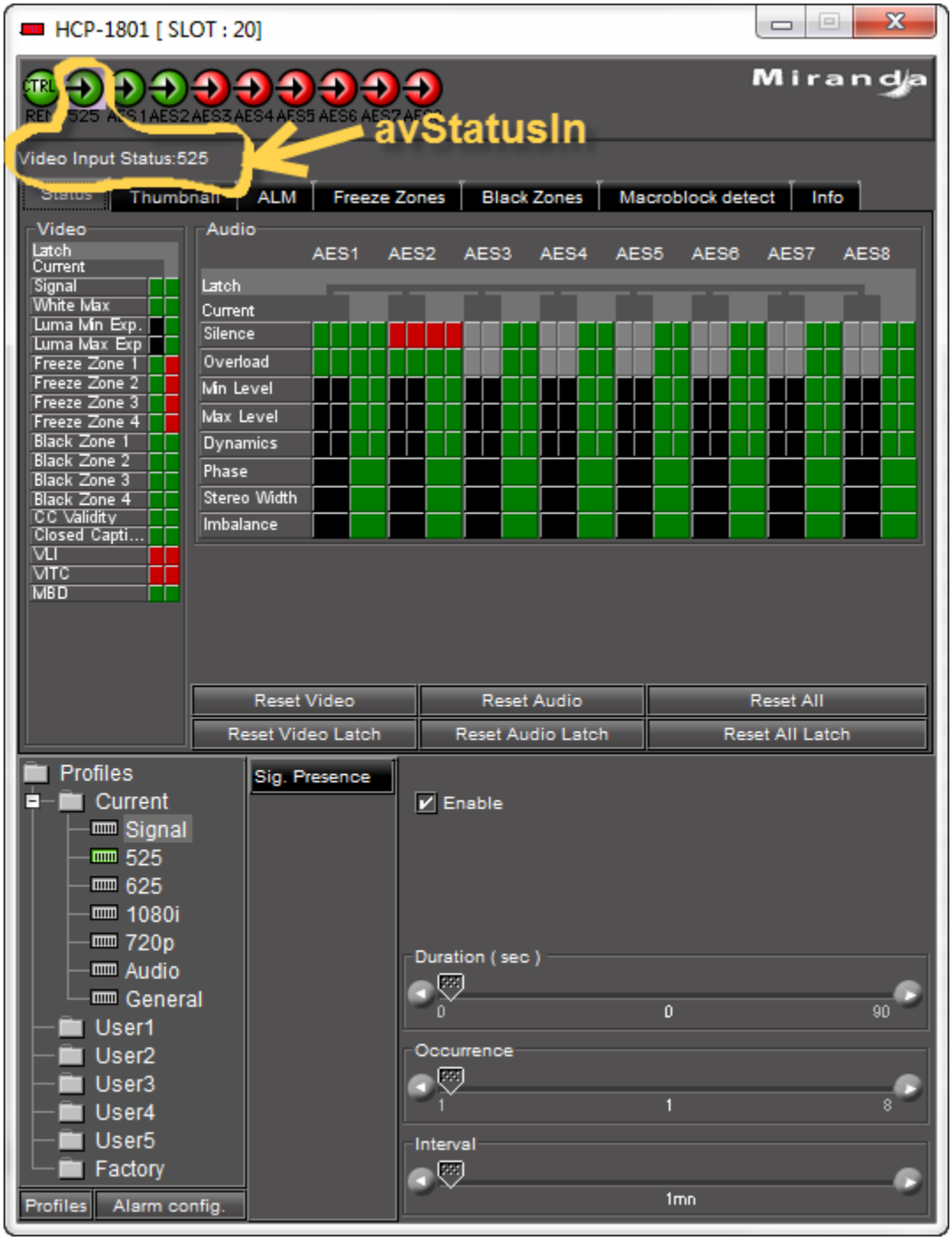
7. Save the widget by selecting the *Save* option in the *File* menu or by pressing Ctrl+S.

The widget is now ready to be used on the pages. Following is a summary of the parameters of the widget:

Parameter Name	Parameter Type	Expected Value
alarm	Alarm URI	The URI address of the alarm to be monitored.
page	Page	The page to be displayed when the button is selected.
pageZone	Zone	Zone in which the page should be displayed. If left empty, no page will be displayed upon selection.

3.2 Creating the “card_panel_message” widget

The “card_panel_message” widget can be used to display directly on the page, card messages that are shown in the top of a card’s panel. Below, the *Video Input Status* is depicted for the HCP-1801 card; the parameter name is also shown (*avStatusIn*). By placing this parameter on the page, the operator can know at all times what the video input of the card is without having to open the card panel each time.



In iControl Creator the widget will look as shown below:

To create this widget, the same steps as in section 3.1 should be followed.

1. Create an empty widget file and name it “card_panel_message.mwf”.
2. Add a label on the page long enough to hold the entire output string and name it *label*. Given the fact that this widget will only display text but no status or perform an action when selected, there is no need of a composite image or of a function in the *Click handler* section. Save and close the *Status icon properties* window.
3. Open the *Page properties* window and enter the external parameters shown below with the correct types. Apply the changes without closing the window.
4. In the *Page properties* window open the *Page load event handler* tab and enter the following:

```
this.getCardPanelMsg = function(){
        try{
                var proxy = navigator.getService(cardId, 100000);
                var param = proxy.getParameter(paramName + "_INFO");
                var info = param.getValue();
                var val = proxy.getParameter(paramName);
                var txt = info.status[val.getValue()].message;
                getElementById("label").value = txt;
        }
        catch(ex){
                //do nothing
        }
}

navigator.windows[0].setInterval(getCardPanelMsg,5000);
```

For a specified HCP-1801 card, this will get the value of the parameter entered in the properties and display it in the label. Once finished, save and close the *Page properties* window and save the “card_panel_message” widget.

The widget is now ready to be used on the pages. Following is a summary of the parameters of the widget:

Parameter Name	**Parameter Type**	**Expected Value**
cardId	Service	The long Id of the HCP-1801 card.
paramName	String	**avStatusIn**