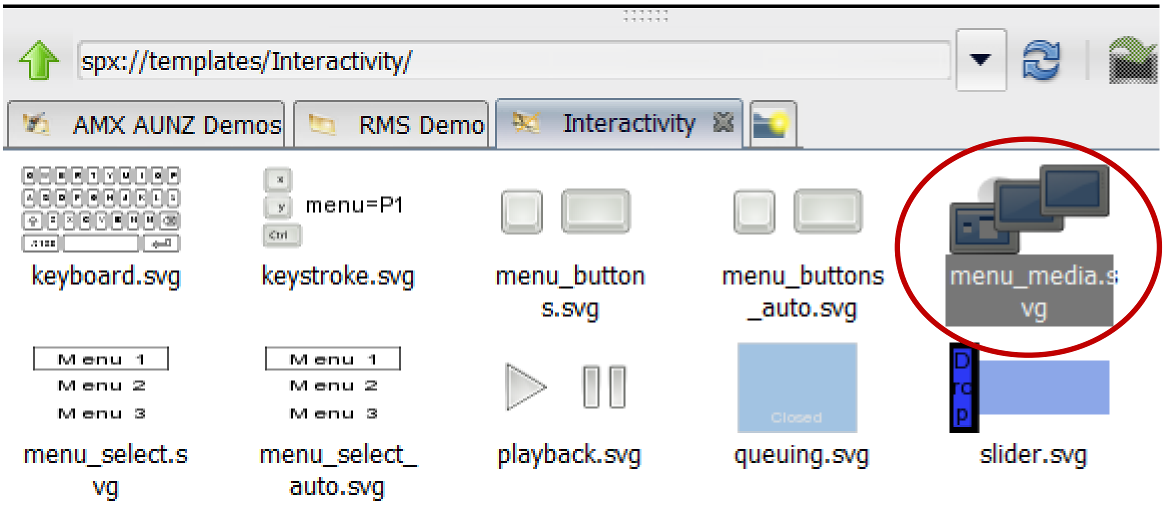
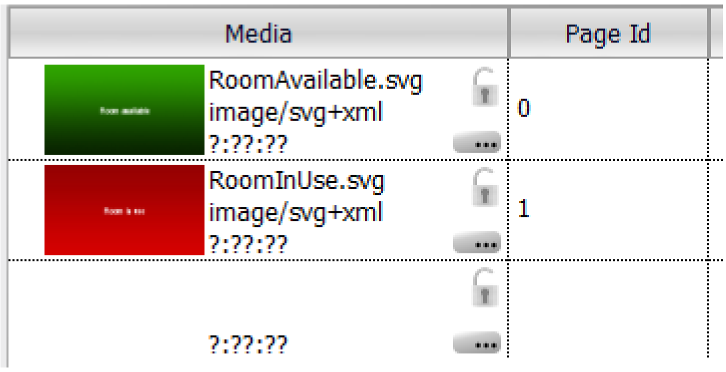
Displaying RMS meeting information on an SPX-1000 player

This guide assumes a knowledge of programming in Netlinx Studio and Inspired Express.

1. In Inspired Express create a project, navigate to "//templates/Interactivity/" and add a **menu_media.svg** to your project.

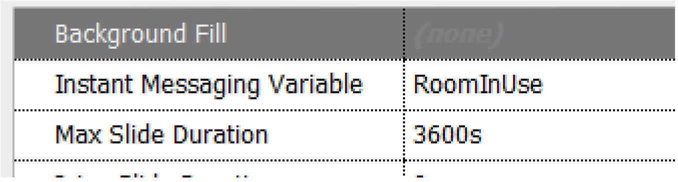


2. Put the **menu_media.svg** into your **main.svg**
3. Create a new svg file, call it **RoomInUse.svg** and add a red background with the text "Room in use"
4. Create a new svg file, call it **RoomAvailable.svg** and add a green background with the text "Room available"



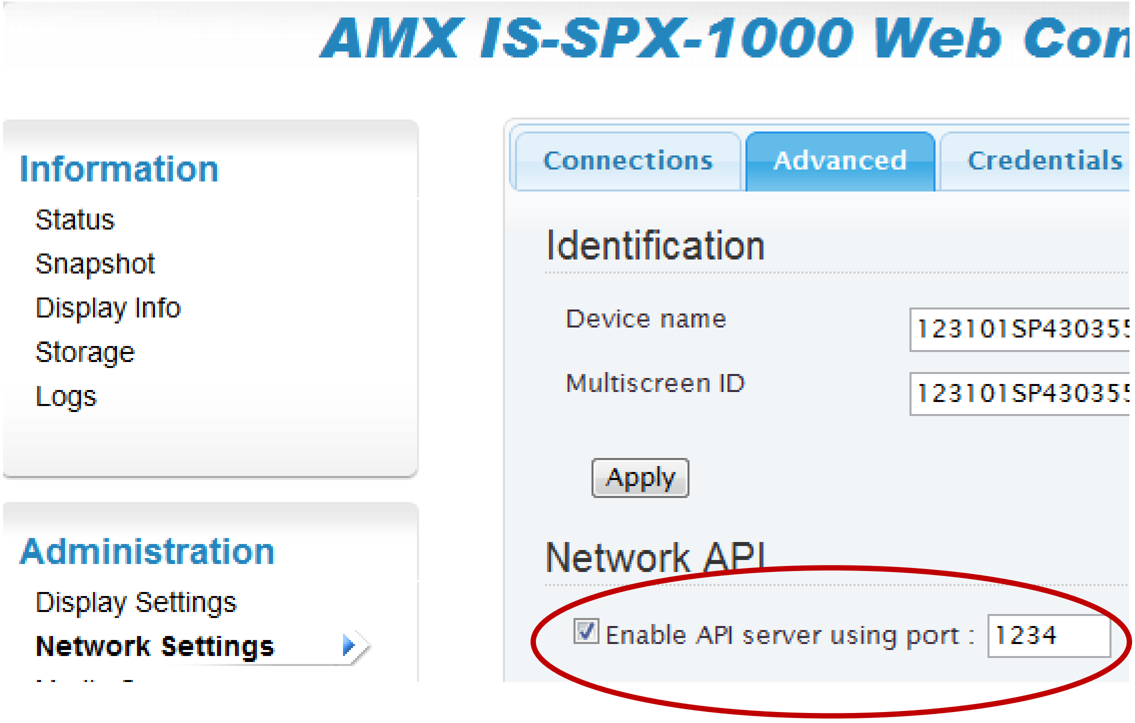
Media	Page Id
RoomAvailable.svg image/svg+xml ?:??:??	0
RoomInUse.svg image/svg+xml ?:??:??	1
?:??:??	

5. Select **menu_media.svg** then the **Playlist** tab, add **RoomAvailable.svg** and give it a **Page Id** as "0"
6. Select **menu_media.svg** then the **Playlist** tab, add **RoomInUse.svg** and give it a **Page Id** as **"1"**



Background Fill	(none)
Instant Messaging Variable	RoomInUse
Max Slide Duration	3600s

7. Select the **Properties** tab of **menu_media.svg** and enter **"RoomInUse"** into the **Instant Messaging Variable** section
8. Upload this to the player
9. Go to the management web page for the player, select **Network Settings** followed by the **Advanced** tab and tick **Enable API server**.



Device name 123101SP43035

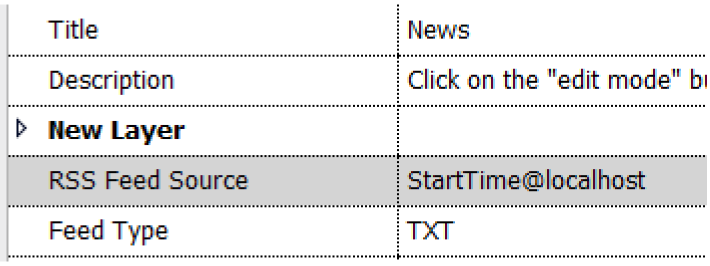
Multiscreen ID 123101SP43035

☑ Enable API server using port : 1234

10. Open your existing AMX program which has an existing RMS touchpanel.
11. Write code so that whenever the "Room In Use" button on the AMX touchpanel turns on or off a command is sent to the player to set the shared variable "RoomInUse" to either 1 or 0.

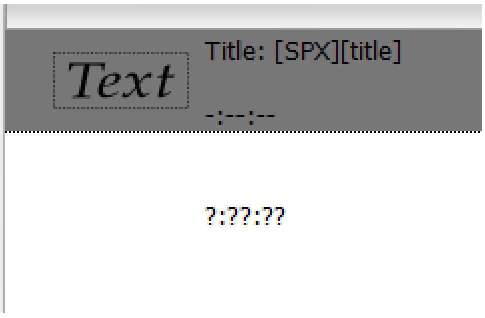
```
DEFINE_DEVICE

dvSignage = 0:4:0 // select next available port for TCP connection

dvTP_RMS = 10001:1:0 // use the existing RMS welcome touchpanel

INCLUDE 'RMSMain.axi' // All RMS code lives in here

DEFINE_VARIABLE

CHAR sSignageIPAddress[] = '192.168.1.1' // Enter the IP address of the player

INTEGER nSignageIPPort = 1234 // Default for players is 1234

DEFINE_START

WAIT 300 // this is only example IP socket, manage socket better than this in real program

        IP_CLIENT_OPEN(dvSignage.Port, sSignageIPAddress, nSignageIPPort, IP_TCP)

DEFINE_EVENT

CHANNEL_EVENT[dvTP_RMS, 252] // button 252 is the button used by RMSUIMod.axs for RoomInUse

{

        ON : SEND_STRING dvSignage, "'UPDATE "RoomInUse" "1"',$0D,$0A" // Set players shared variable to 1

        OFF: SEND_STRING dvSignage, "'UPDATE "RoomInUse" "0"',$0D,$0A" // Set players shared variable to 0

}
```

12. Upload the code, now the player will switch pages to indicate if the room is in use. Next we will display meeting data.
13. Open **RMS (400x800).TP4** and find the subpage **rmsMeetingInfo**, we are going to display the text on each of these buttons on the signage player so take note of the addresses (ie 161 to 168)
14. Open your IS Express project again, add **news.svg** from “//Templates/News” and rename it to **StartTime.svg.**
15. In the **Properties** tab of **StartTime.svg** set **RSS Feed Source** to “StartTime@localhost” and **Feed Type** to “TXT”



Title	News
Description	Click on the "edit mode" bı
▷ **New Layer**	
RSS Feed Source	StartTime@localhost
Feed Type	TXT

16. In the **Layers** tab of **StartTime.svg** delete the description and media objects so that only **Title: [SPX][title**] is left.

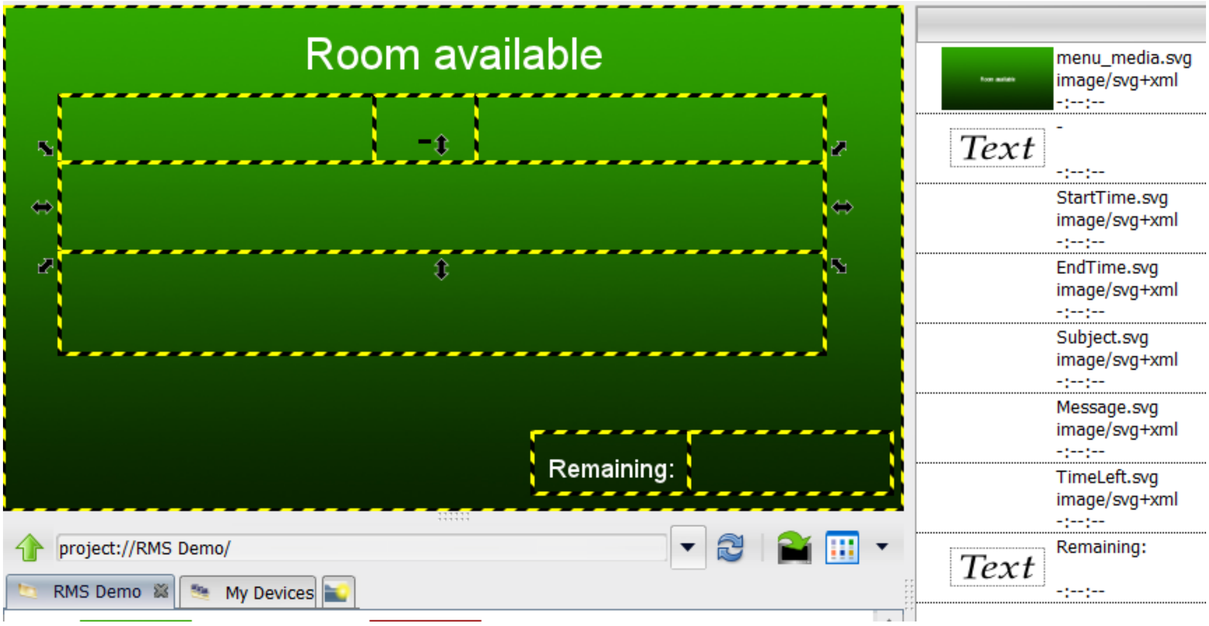


Title: [SPX][title]

-:--:--

?:??:??

17. Edit the text so that it only says “[SPX][title]”, also adjust the size, colour and font to suit how you would like it to appear on the screen.
18. Insert **StartTime.svg** onto your main document in front of **menu_media.svg**.
19. Copy **StartTime.svg** several times and use the copies as templates for all of the other text windows on the **rmsMeetingInfo** subpage (eg EndTime, Subject etc.)
20. For each of the new documents change the **RSS Feed Source** to suit (eg EndTime@localhost) and arrange them on the main page then upload to player.



menu_media.svg
image/svg+xml
-:--:--

-
-:--:--

StartTime.svg
image/svg+xml
-:--:--

EndTime.svg
image/svg+xml
-:--:--

Subject.svg
image/svg+xml
-:--:--

Message.svg
image/svg+xml
-:--:--

TimeLeft.svg
image/svg+xml
-:--:--

Remaining:
-:--:--

21. In Netlinx Studio you are going to need to catch data events that are being sent from RMSUIMod.axs, you cannot catch data events sent to a physical panel so you must redefine the panel as a virtual panel and combine it with the real panel.

```
DEFINE_DEVICE

/*

You cannot catch data events sent to a physical panel so we need to make the RMS welcome panel a virtual panel so we can send the same text strings to the signage player.

Don’t rename the panel or you will break the existing program.

//dvTP_RMS = 10001:1:0 // original D:P:S of RMS welcome touchpanel

*/

dvTP_RMS = 33005:1:0 // virtual RMS welcome panel  (note name not changed).

dvTP_RMS_real = 10001:1:0 // the actual RMS welcome touchpanel

DEFINE_COMBINE(dvUI_01_RMS_Main, dvUI_01_RMS_Main_Real)
```

22. Add the following code to catch strings sent to the RMS touchpanel and send equivalent strings to the signage player.

```
DEFINE_CONSTANT
CHAR cSharedVarName[][15] = // these strings must match the shared variables in the signage player
{
	'StartTime', // Defined as StartTime@localhost in RSS Feed Source of news.svg
	'EndTime', // as above...
	'ScheduledBy',
	'Subject',
	'Attending',
	'Message',
	'TimeLeft',
	'NextAppointment'
}

DEFINE_EVENT
DATA_EVENT[dvTP_RMS] // You can only catch commands sent to the TP if dvTP_RMS is a virtual
device
{
	COMMAND :
	{
		STACK_VAR iTextAddress_

		IF(FIND_STRING(DATA.TEXT, '^TXT-', 1))
		{
			REMOVE_STRING(DATA.TEXT, '^TXT-', 1)
			iTextAddress_ = ATOI(DATA.TEXT)
			IF (iTextAddress_ > 160 && iTextAddress_ < 169)
			{
				REMOVE_STRING(DATA.TEXT, ',', 1)
				REMOVE_STRING (DATA.TEXT, ',', 1)
				SEND_STRING dvSignage, "'UPDATE "',
cSharedVarName[iTextAddress_-160], '" "', DATA.TEXT, '"',$0D,$0A"
			}
		}
	}
}
```

Now you can apply these principles to send any information that is displayed on the RMS touch panel on the signage player including all text, page flips and sub pages.

You do not even need a physical touch panel in the project as long as the RMS welcome touch panel is defined in your program as a virtual panel.