



Creating & Editing Dynamic Layouts (Rev D)

Contents

Introduction.....	3
Who Should Use This Guide	3
What Will You Need	3
Where Can You Find The Dynamic Layout Files	3
The Basic Structure.....	5
A Basic Dynamic Layout XML File	6
Empty Dynamic Layout Template Example.....	8
Explaining The Header Section	9
Explaining The Screens Section	10
Explaining The Containers Section	10
Dynamic Layout Tile Types	12
MediaWall Tile.....	12
Favourites Tile	13
Library Tile	14
Log Tile.....	15
Dashboard Tile.....	16
Media Players	17
Mini Media Players.....	18
SmartInfo Tile	19
Clock Tile.....	20
Pad Tile	23
Media Browser Tile.....	24
Media Item Information Tile	25
QuickRecorder Tile	26
Segue Editor Tile	27
SmartDisplay Tile	28
Simple Mixer View Tile (Mic Live for SRM / Webstation / Airlite / Capitol IP / Forum IP).....	30
Audio Monitor Tile (Line In / Broadcast Radio Audio Monitor / D&R Webstation / D&R Airlite).....	33
Real World Example	37
Grid View	37
Adding The First Container	38
Container Position & Size	39
Adding In The Tiles	39
Reviewing Your Layouts In Myriad Payout.....	42
Downloading More Dynamic Layouts From Our Website.....	43

Introduction

One of the most interesting features of Myriad 5 Payout is the ability to design your own interface and switch between interfaces using the Dynamic Layout systems.

Dynamic Layouts are used to define what tools and elements Myriad 5 will display on screen, how they are displayed and what options to use.

The Dynamic Layouts systems in Myriad 5 is enhanced further by the ability for individual users customise certain options which are stored in their profile and used whenever a specific Dynamic Layout is used. An example of this may be that you have a Dynamic Layout that has a Media Wall Tile on it. By default it might display 10 Media Items (depends on size) but User A might like to increase this to 15 items and User B might prefer 8 items. In each case, their preference will be saved and used next time they are logged in and load that specific Dynamic Layout.

Dynamic Layouts can be used to dramatically alter the way Myriad 5 Payout looks and the tools on offer. This means that systems used for different roles in your station (on air studio, production etc) can be optimised for their primary use.

Who Should Use This Guide

Dynamic Layouts offer a powerful way to optimise Myriad 5 Payout to suit the style of your station or to perform specific tasks. Dynamic Layouts are currently both created and edited using an XML file to define the 'tiles' that should be visible and the options within those tiles.

XML files are commonly used by software to define and transfer data and the structure will be familiar to many users. If you are not familiar with XML files and how they are commonly used then you may find some elements of this documentation challenging.

The eventual aim is to create a user interface to allow standard users to edit Dynamic Layouts but for now, editing Dynamic Layouts will involve understanding and working with XML files.

What Will You Need

You can edit the XML files used to define the Dynamic Layouts using a simple text editor such as Notepad. By default, Windows will open them in your web browser but this is no use as you cannot edit them. Instead you will need to right click on the XML files and select 'Edit' or 'Open With' and then select your editor.


Whilst Notepad allows you to edit the files, we would recommend using a specialist application that will help you to check that your XML structure is correct. We use a product called **Notepad++** which you can download for free from this website:

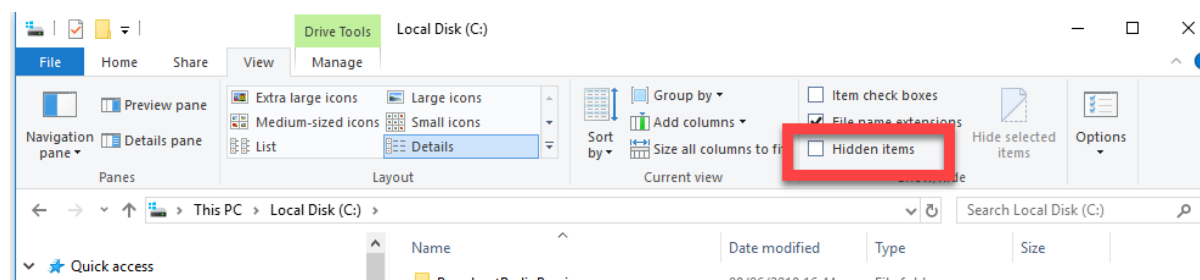
<https://notepad-plus-plus.org/download/>

Simply download and install Notepad++ and you will be ready to start looking at the Dynamic Layout files.

Where Can You Find The Dynamic Layout Files





The XML files used to define the Dynamic Layouts are hidden by default so you will need to change a couple of things in order to provide easy access.

Open **Windows File Explorer**  and go to the **View** tab (you may need to click on the menu option to expand the tab). Locate the **Hidden Items** option and make sure it is ticked.



This will show folders and files the Windows usually hides from you.

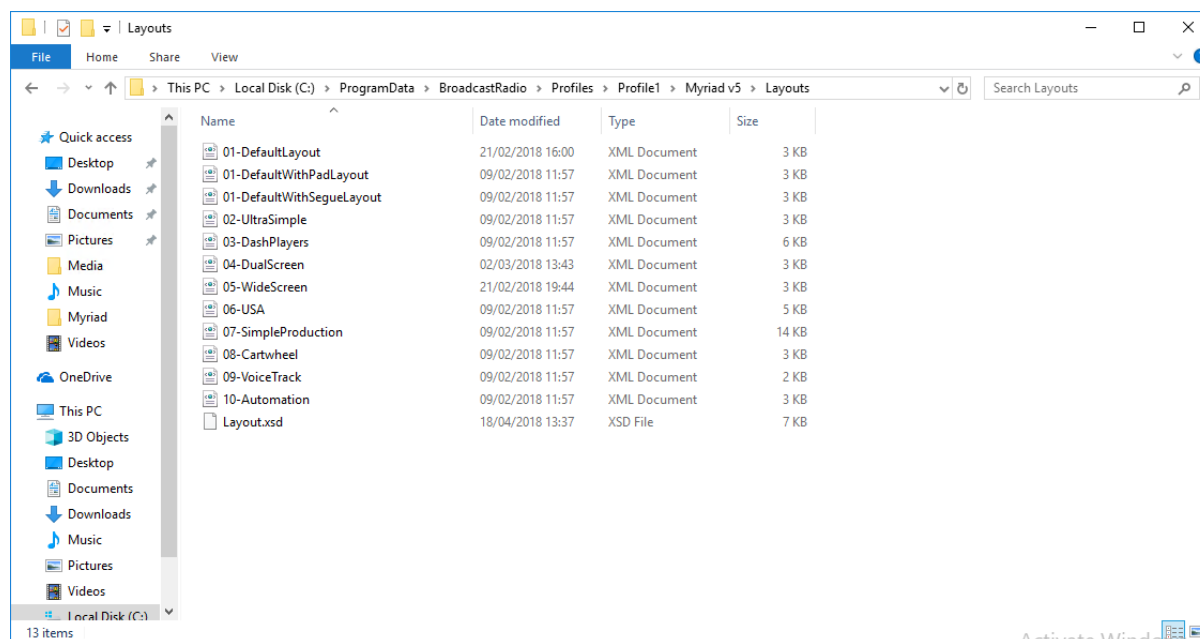
On your **C Drive** you should now see a folder called **Program Data** this is the folder we want to look inside.

	Program Files (x86)	07/06/2018 14:54	File folder
	ProgramData	25/05/2018 10:52	File folder
	PSQInstallers	07/07/2017 16:05	File folder
	Requard	21/04/2018 15:04	File folder

You now need to go into the **Program Data** folder and navigate down to the folder below.

C:\ProgramData\BroadcastRadio\Profiles\Profile1\Myriad v5\Layouts

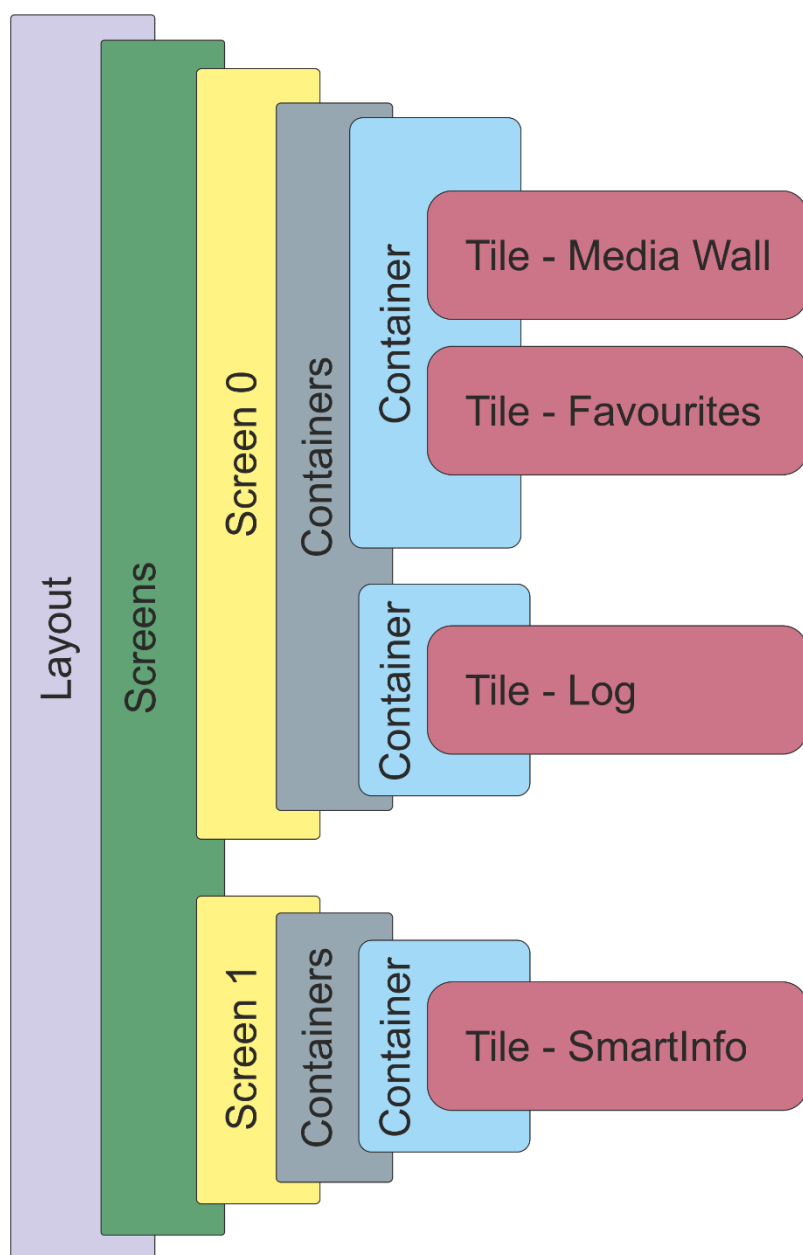
Please note that if you have multiple profiles setup then the 'Profile1' section will be replaced with the name of the profile you want to edit.



These are the Dynamic Layouts on your system. To add a new Dynamic Layout, you can copy or create a new XML file in this folder and Myriad 5 Playout will display it in the list of available Dynamic Layouts as soon as you have select the 'Refresh Layout List' option on the Layouts menu in Myriad 5 Playout.

The Basic Structure

Like all XML files, the Dynamic Layout files follow a simple but important structure. If you do not follow the structure then Myriad 5 Payout will not be able to use the Dynamic Layout file and it could cause problems with your Myriad system. The diagram below outlines the basic structure of the Dynamic Layout files.



Layout Node – This is the overall Dynamic Layout and should be the first and last tags in your Dynamic Layout File.

Screens Node – Like the Layout Node, the Screens node is really just a holder that contains all of the individual Screen node.

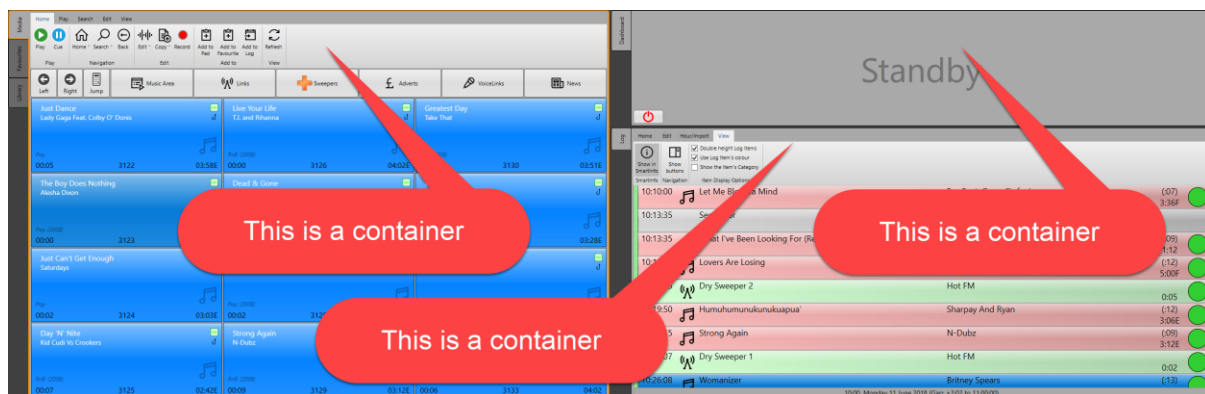
Screen Node – The Screen node contains everything that will be displayed on a single screen. So Screen Index '0' will be the first screen on your system, Screen Index '1' will be the second screen (ie dual screen) and so on.

So in the example structure above, we have defined two screens (0 & 1) and the first screen will display the Media Wall, Favoured and Log. The second screen (Screen 1) will only display the Smart Info tile.

With the Screen Node you also define the number of rows and columns that will be available.

Containers Nodes – This is a holder for the individual Container nodes within a Screen.

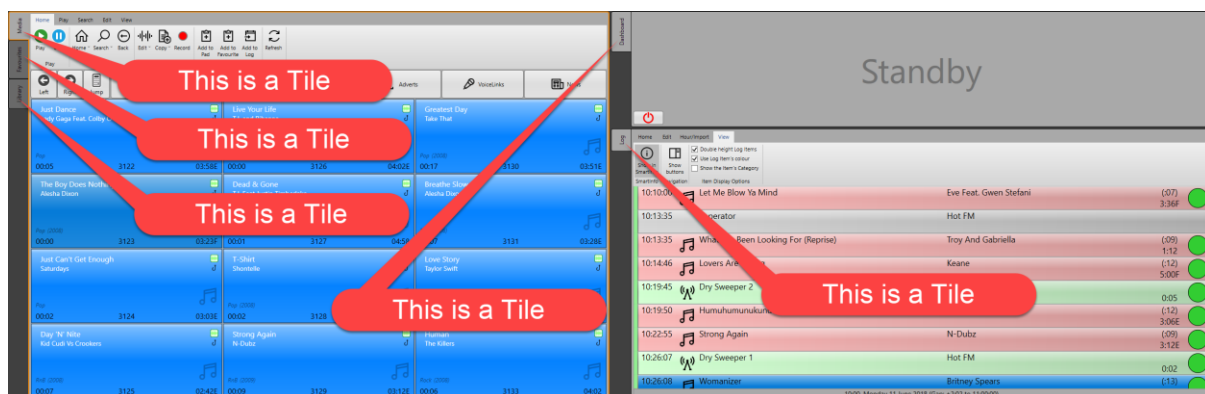
Container Nodes – Each container is given a position and a size (based on row and column). The container is effectively an empty window that will be displayed by Myriad 5 Playout. What goes in the Containers is defined by the tile nodes.



You can set the start position, width and height of Containers within the Node Header. If you put two containers on top of each other, the last one listed will be displayed on top.

Containers can also be set to **Tabs** or **Frames**. **Tabs** allow multiple Tiles to be within a single Container and the user will be able to switch between them using a tab strip on the left hand side. A **Frame** can only contain a single Tile which will be displayed at the full width of the Container.

Tile Nodes – The Tile Nodes live within the Container Nodes and are used to place Myriad 5 elements within the Containers.



You can place multiple Tiles within a Container (as long as it is defined as a 'Tab') and the user will be able to switch between them (see the image above on the left with the MediaWall, Favoured & Library Tiles all contained within a single Container).

You can also set a range of specific settings for each Tile with each Tile Node.

A Basic Dynamic Layout XML File

To start with, let's look at a simplified Dynamic Layout file that would achieve the layout outlined in the structure diagram in the previous section.

So in this case we want a very simple layout that consists of two screens with the following features:

Screen 0 (first screen)

- MediaWall & Favourites (in a single Container)
- Log (on it's own Container)

Screen 1 (first screen)

- SmartInfo (in it's own Container)

To achieve this we could use a very simple Dynamic Layout XML file like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<wl:layout header="Simple Layout" version="5.0"
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns:wl='http://schemas.broadcastradio.com/myriad/2016/windowLayout'
    xsi:schemaLocation="urn:Layout Layout.xsd">
    <wl:screens>
        <wl:screen index="0" totalRows="12" totalColumns="12">
            <wl:containers>
                <wl:container type="tab" >
                    <wl:position row="0" column="0" rowspan="12" columnSpan="6"/>
                    <wl:titles>
                        <wl:tile type="MediaGridView" header="Media" ribbonIsMinimised="false">
                            <wl:settings/>
                        </wl:tile>
                        <wl:tile type="FavouritesSetView" header="Favourites"
                            ribbonIsMinimised="false">
                            <wl:settings/>
                        </wl:tile>
                    </wl:titles>
                </wl:container>
                <wl:container type="tab" >
                    <wl:position row="0" column="6" rowspan="12" columnSpan="6"/>
                    <wl:titles>
                        <wl:tile type="LogView" header="Log" ribbonIsMinimised="false">
                            <wl:settings/>
                        </wl:tile>
                    </wl:titles>
                </wl:container>
            </wl:containers>
        </wl:screen>

        <wl:screen index="1" totalRows="12" totalColumns="12">
            <wl:containers>
                <wl:container type="tab" >
                    <wl:position row="0" column="0" rowspan="12" columnSpan="12"/>
                    <wl:titles>
                        <wl:tile type="SmartInfoView" header="SmartInfo">
                            <wl:settings/>
                        </wl:tile>
                    </wl:titles>
                </wl:container>
            </wl:containers>
        </wl:screen>
    </wl:screens>
</wl:layout>
```

This will result in a Myriad 5 Playout that looks a little like this:



Now let's take look at this in a little more detail to see what is going on.

```

<?xml version="1.0" encoding="utf-8" ?>
<wl:layout header="Simple Layout" version="5.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wl="http://schemas.broadcastradio.com/myriad/2016/windowLayout"
  xsi:schemaLocation="urn:Layout Layout.xsd">
  <wl:screens>
    <wl:screen index="0" totalRows="12" totalColumns="12">
      <wl:containers>
        <wl:container type="tab">
          <wl:position row="0" column="0" rowspan="12" colspan="6"/>
          <wl:titles>
            <wl:title type="MediaGridView" header="Media" ribbonIsMinimised="false">
              <wl:settings/>
            </wl:title>
            <wl:title type="FavouritesSetView" header="Favourites" ribbonIsMinimised="false">
              <wl:settings/>
            </wl:title>
          </wl:titles>
        </wl:container>
        <wl:container type="tab">
          <wl:position row="0" column="6" rowspan="12" colspan="6"/>
          <wl:titles>
            <wl:title type="LogView" header="Log" ribbonIsMinimised="false">
              <wl:settings/>
            </wl:title>
          </wl:titles>
        </wl:container>
      </wl:containers>
    </wl:screen>
    <wl:screen index="1" totalRows="12" totalColumns="12">
      <wl:containers>
        <wl:container type="tab">
          <wl:position row="0" column="0" rowspan="12" colspan="12"/>
          <wl:titles>
            <wl:title type="SmartInfoView" header="SmartInfo">
              <wl:settings/>
            </wl:title>
          </wl:titles>
        </wl:container>
      </wl:containers>
    </wl:screen>
  </wl:screens>
</wl:layout>

```

First screen is a grid of 12 x 12

First Container is setup as a Tab

This Container will start at row 0, column 0 (eg top left corner) and stretch 6 columns across and 12 rows down.

This is the Favorites Tile

The first Tile will be a MediaWall and the ribbon will not be minimised

The second Container will start at Row 0, Column 6 (eg half way across screen 1 but at top of screen). It has one Tile which is a

Screen 1 (second screen) only has one Container and inside that is a single SmartInfo Tile.

In practice, this Dynamic Layout is probably not that useful as it doesn't include any Media Players but it is a good illustration of a simple layout.

Please note that Containers can either be **Tabs** or **Frames**.

- **Tabs** can contain multiple Tiles and users switch between them using standard Tab strip.
- **Frames** can only contain one Tile and that Tile will fill the entire container (eg there will be no tab strip).

Empty Dynamic Layout Template Example

Whilst the above example shows a working Dynamic Layout, when you are creating your own Dynamic Layouts it is useful to start with an empty template that you can edit and fill as required. The example below includes all the basic structures you will need but doesn't include any actual **Tiles**.

```

<?xml version="1.0" encoding="utf-8" ?>
<wl:layout header="Dual Screen Example" version="5.0"
  suggestedAspectRatio="DualRegular"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wl="http://schemas.broadcastradio.com/myriad/2016/windowLayout"
  xsi:schemaLocation="urn:Layout Layout.xsd">
  <wl:screens>

```



```

<wl:screen index="0" totalRows="12" totalColumns="12">
  <wl:containers>
    <wl:container type="tab" >
      <wl:position row="0" column="0" rowSpan="10" columnSpan="6"/>
      <wl:tiles>

        </wl:tiles>
      </wl:container>

    <wl:container type="tab" >
      <wl:position row="0" column="6" rowSpan="10" columnSpan="6"/>
      <wl:tiles>

        </wl:tiles>
      </wl:container>

    <wl:container type="tab" >
      <wl:position row="10" column="0" rowSpan="3" columnSpan="12"/>
      <wl:tiles>

        </wl:tiles>
      </wl:container>
    </wl:containers>
  </wl:screen>

  <wl:screen index="1" totalRows="12" totalColumns="12">
    <wl:containers>
      <wl:container type="tab" >
        <wl:position row="0" column="0" rowSpan="12" columnSpan="12"/>
        <wl:tiles>

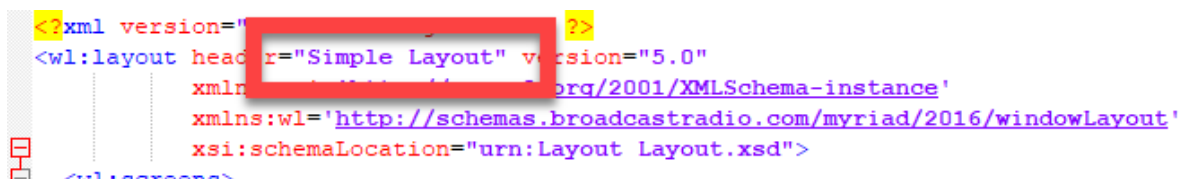
          </wl:tiles>
        </wl:container>
      </wl:containers>
    </wl:screen>
  </wl:screens>
</wl:layout>

```

You can copy the code above to be the basis for any Dynamic Layout. Without editing, the above example will result in a dual screen layout. On the first screen will be two **Containers** side by side with each occupying half of the screen and 5/6ths of the vertical screen. There is a third **Container** running along the bottom of the first screen. The Default Layout in Myriad 5 Payout uses a similar basic layout. The second screen only contains a single **Container** set to cover the full screen.

Explaining The Header Section

The only section of the header section that you should edit is the **Title** segment which is used to add a title to your Dynamic Layout which is the name that will appear in the menu in Myriad.



```

<?xml version="1.0" encoding="UTF-8" ?>
<wl:layout header="Simple Layout" version="5.0"
  xmlns="http://schemas.broadcastradio.com/myriad/2016/windowLayout"
  xsi:schemaLocation="urn:Layout Layout.xsd">
  <wl:screens>

```

Explaining The Screens Section

The Screens section dictates what screen the section should be displayed on and also sets up a grid on the screen which is used to position the Containers.

```
<wl:screen index="0" totalRows="12" totalColumns="12">
</wl:screen>
```

The screen that this section will be displayed on (0 is your first screen)

Creates a 12 x 12 grid on the screen for the Containers

We have not defined the size of the columns or rows, instead Myriad will take the screen resolution and divide it into the amount of rows and columns required. This means that the layout will work on standard and wide screen monitors and on monitors with different resolutions, however, you may wish to create different Dynamic Layouts for use on PC's with widely different screen types or configurations (for example screen in portrait mode).

Explaining The Containers Section

The **Container** is a section of the screen that can contain one or more **Tiles**. The Container node includes the type of Container (**Tab** or **Frame**) and the start position and size of the **Container**.

```
<wl:container type="tab" >
  <wl:position row="0" column="0" rowspan="10" colspan="6"/>
  <wl:tiles>

</wl:tiles>
</wl:container>
```

The Container will have Tabs on the left hand side

The Container will start at Row 0 and Column 0 (top left corner)

The Container will be 10 Rows high and 6 Columns wide

Here is an example of a Container set as a Frame. Frames can only contain a single **Tile** and the Tile will be displayed at the full size of the Container.

```
<wl:container type="Frame">
  <wl:position row="0" column="0" rowspan="4" colspan="16" />
  <wl:tiles>
    <wl:tile type="FavouritesSetView" header="Favourites">
      <wl:settings />
    </wl:tile>
  </wl:tiles>
```

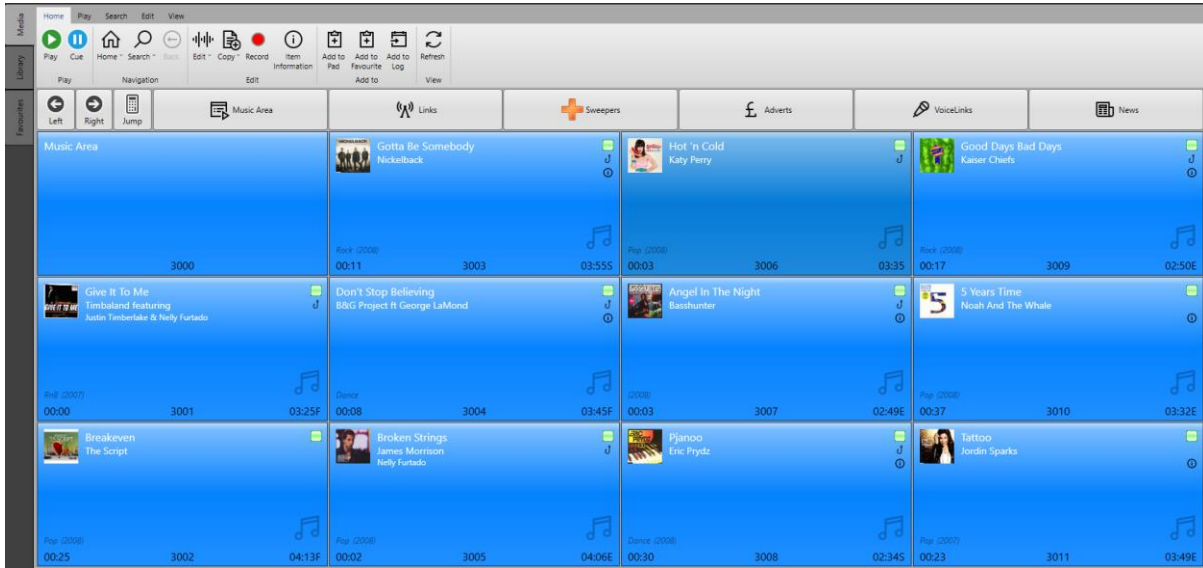
</wl:container>

Important Note: The XML node structure is very important as without correctly open and closed Nodes, Myriad 5 Playout will not be able to use your Dynamic Layout file. Tools like Notepad++ make this easier as they highlight nodes as you click on the starting block which makes spotting errors a lot easier.

Dynamic Layout Tile Types

Now that we have a basic idea about what a Dynamic Layout XML file should look like, it is time to look at the **Tile** types and the options available.

MediaWall Tile



Code Snippet

This Tile will display a MediaWall in a grid layout within the selected container.

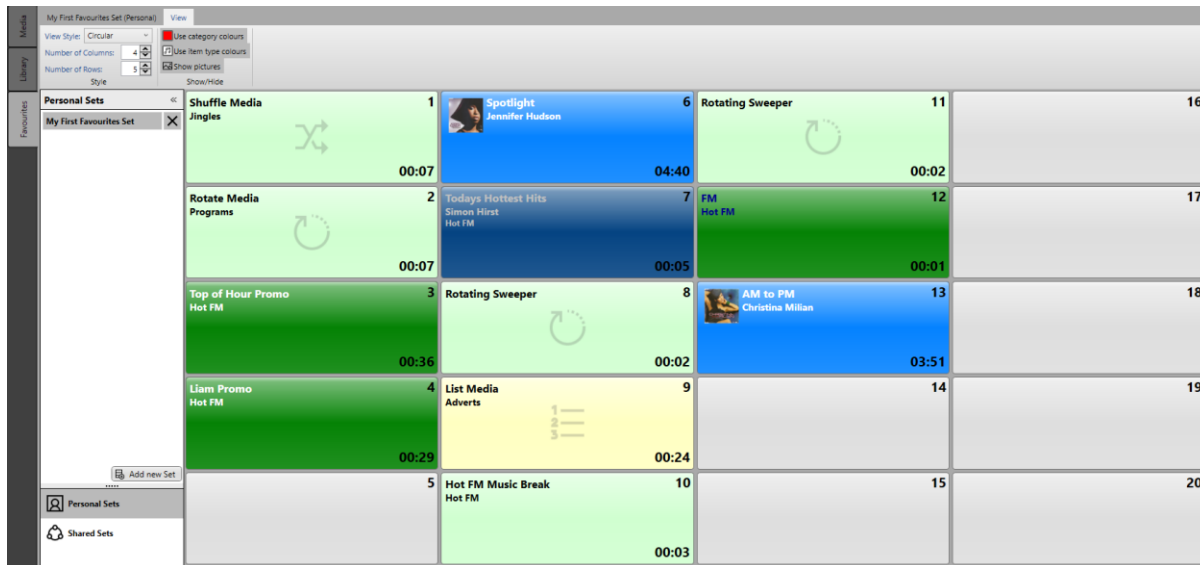
```
<wl:tile type="MediaGridView" header="MediaWall" ribbonIsMinimised="false">
  <wl:settings>
    <mediaGridViewSettings>
      <lockToStationId>-1</lockToStationId>
      <smartInfoEngineId>0</smartInfoEngineId>
    </mediaGridViewSettings>
  </wl:settings>
</wl:tile>
```

Options include:

- header – Label that will appear on the 'tab'.
- ribbonIsMinimised (True / False)– Whether ribbon is always visible (false) or only when clicked (true).
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).
- smartInfoEngineId – You can have multiple SmartInfo engines running and each SmartInfo Tile is set to display the results of the referenced engine. This field allows you to set which SmartInfo engine, this MediaWall should control. This is useful if you want to have two SmartInfo Tiles, one controlled from the MediaWall, the other from the Log (for example). In this example, the MediaWall Tile will

drive SmartInfo Engine ID '0' (which is the default). Note: Multiple Tiles can drive a single SmartInfo engine.

Favourites Tile



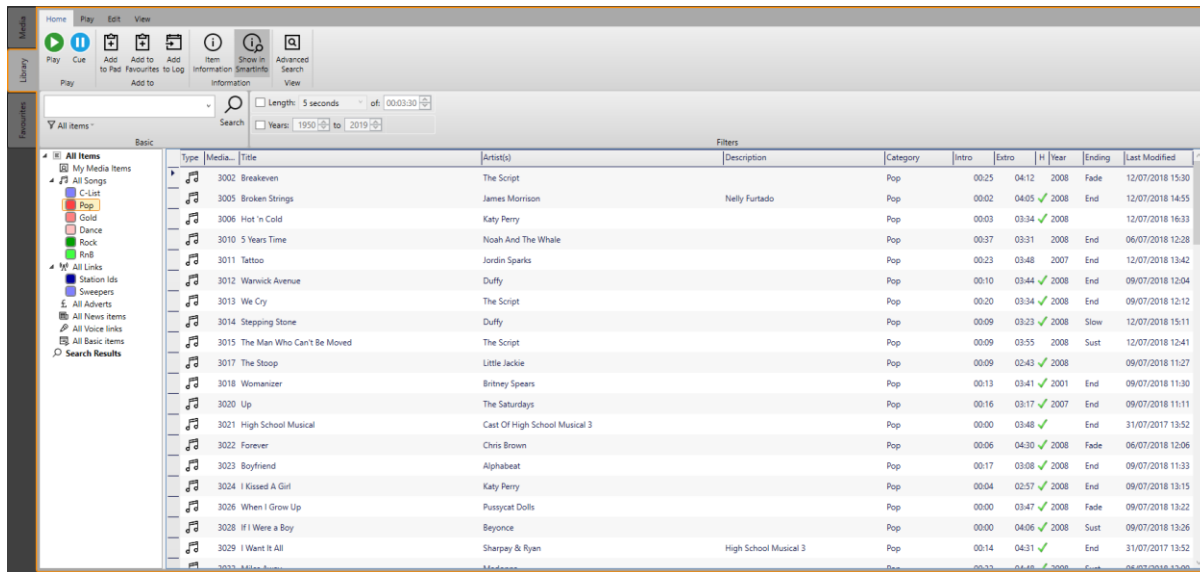
Code Snippet

```
<wl:tile type="FavouritesSetView" header="Favourites" ribbonIsMinimised="false">
  <wl:settings>
    <favouriteSetViewSettings>
      <lockToStationId>-1</lockToStationId>
    </favouriteSetViewSettings>
  </wl:settings>
```

This Tile will display a Favourite Set in the selected Container. The available options are:

- header – Label that will appear on the 'tab'.
- ribbonIsMinimised (True / False) – Whether ribbon is always visible (false) or only when clicked (true).
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).

Library Tile



Code Snippet

```
<wl:tile type="MediaLibraryView" header="Library" ribbonIsMinimised="false">
  <wl:settings>
    <mediaLibraryViewSettings>
      <lockToStationId>-1</lockToStationId>
      <smartInfoEngineId>0</smartInfoEngineId>
    </mediaLibraryViewSettings>
  </wl:settings>
</wl:tile>
```

This Tile will display a Library view in the selected Container. The available options are:

- header – Label that will appear on the ‘tab’.
- ribbonIsMinimised (True / False) – Whether ribbon is always visible (false) or only when clicked (true).
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).
- smartInfoEngineId – You can have multiple SmartInfo engines running and each SmartInfo Tile is set to display the results of the referenced engine. This field allows you to set which SmartInfo engine, this Library should control. This is useful if you want to have two SmartInfo Tiles, one controlled from the Library, the other from the Log (for example). In this example, the Library Tile will drive SmartInfo Engine ID '0' (which is the default). Note: Multiple Tiles can drive a single SmartInfo engine.

Log Tile

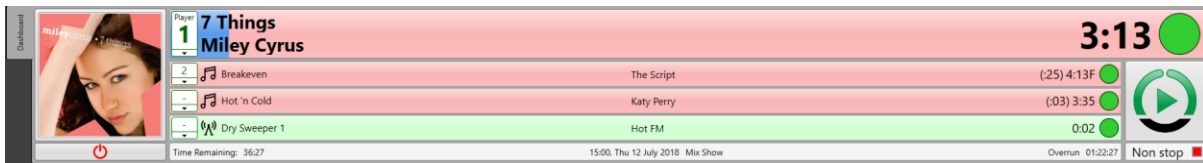


Code Snippet

```
<wl:tile type="LogView" header="Station Log" ribbonIsMinimised="false">
  <wl:settings>
    <logViewSettings>
      <lockToStationId>-1</lockToStationId>
      <smartInfoEngineId>0</smartInfoEngineId>
      <sendSelectedItemNotifications>true</sendSelectedItemNotifications>
    </logViewSettings>
  </wl:settings>
</wl:tile>
```

- header – Label that will appear on the ‘tab’.
- ribbonIsMinimised (True / False) – Whether ribbon is always visible (false) or only when clicked (true).
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).
- smartInfoEngineId – You can have multiple SmartInfo engines running and each SmartInfo Tile is set to display the results of the referenced engine. This field allows you to set which SmartInfo engine, this Log should control. This is useful if you want to have two SmartInfo Tiles, one controlled from the Log, the other from the MediaWall (for example). In this example, the Log Tile will drive SmartInfo Engine ID '0' (which is the default). Note: Multiple Tiles can drive a single SmartInfo engine.
- sendSelectedItemNotifications (True / False) – This controls whether the Log View send the User highlighted item to the **Segue Editor**, that way if you have multiple Log View Tile on screen, you can control which one will drive the Segue Editor. This is especially useful if you have a Dynamic Layout that includes a second Log View Tile that is ‘looking’ at the same Station as the primary Log View Tile (maybe for a guest screen) but you don’t want the Segue Editor on the main screen to be affected by scrolling around in the Log on the second screen.

Dashboard Tile



Code Snippet

```
<wl:tile type="DashboardView" header="Dashboard">
  <wl:settings>
    <dashboardViewSettings>
      <showPictures>true</showPictures>
      <stretchGoButtonToFill>true</stretchGoButtonToFill>
      <lockToStationId>-1</lockToStationId>
      <smartInfoEngineId>0</smartInfoEngineId>
    </dashboardViewSettings>
  </wl:settings>
</wl:tile>
```

The Dashboard Tile is used to display a Dashboard within the selected Container.

- header – Label that will appear on the 'tab'.
- showPictures (True / False) – Toggle whether album art is displayed (true) or not (false) on the Dashboard
- stretchGoButtonToFill (True / False) – Toggle whether the Go button will stretch vertically to fill all free space on the Dashboard. This is purely cosmetic.
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).
- smartInfoEngineId – You can have multiple SmartInfo engines running and each SmartInfo Tile is set to display the results of the referenced engine. This field allows you to set which SmartInfo engine, this Dashboard should control. This is useful if you want to have two SmartInfo Tiles, one controlled from the Dashboard, the other from the MediaWall (for example). In this example, the Dashboard Tile will drive SmartInfo Engine ID '0' (which is the default). Note: Multiple Tiles can drive a single SmartInfo engine.

Media Players



Code Snippet

```
<wl:tile type="MediaPlayersView" header="Players">
  <wl:settings>
    <mediaPlayersViewSettings>
      <lockToStationId>-1</lockToStationId>
      <firstPlayerIndex>1</firstPlayerIndex>
      <lastPlayerIndex>99</lastPlayerIndex>
      <orientation>horizontal</orientation>
      <showPlayers>true</showPlayers>
      <showRecorders>true</showRecorders>
    </mediaPlayersViewSettings>
  </wl:settings>
</wl:tile>
```

The Media Players Tile allows you to display one or more Media Players inside the selected Container. The available options are:

- header – Label that will appear on the ‘tab’.
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).
- firstPlayerIndex – Allows you to set the Media Player number to be the first to be displayed within the Tile. In the example above, the FirstPlayerIndex is set to 1 which means the first Media Player displayed will be player number 1.
- lastPlayerIndex – This allows you to set the Media Player number of the last Media Player to be displayed within the Tile. So if you wanted to only display Media Players 1 & 2 in this Tile you would set the FirstPlayerIndex to 1 and the LastPlayerIndex to 2. The system will only display as many Media Players as you have defined so even if this setting is set to 99, if you only have four Media Players setup then only four will be displayed.
- orientation (Vertical / Horizontal) – Set whether the Media Players should be stacked horizontally (side by side) or vertically (on top of each other).
- showPlayers (True / False) – Set whether Players should be visible.
- showRecorders (True / False) – This setting toggles whether the dedicated Recorder is displayed in the Container to the right of the Media Players. You would normally only have one Recorder on screen so if you have multiple Media Player Tiles you might turn the Recorder ‘off’ (false) on the additional Player Tiles.

Mini Media Players



Code Snippet

```
<wl:tile type="MiniMediaPlayersView" header="Players">
  <wl:settings>
    <miniMediaPlayersViewSettings>
      <lockToStationId>-1</lockToStationId>
      <firstPlayerIndex>1</firstPlayerIndex>
      <lastPlayerIndex>99</lastPlayerIndex>
      <orientation>horizontal</orientation>
      <showPlayers>true</showPlayers>
      <showRecorders>false</showRecorders>
    </miniMediaPlayersViewSettings>
  </wl:settings>
</wl:tile>
```

The Mini Media Players view are closely related to the main Media Players Tile but the Players are far more compact making them ideally suited to dense layouts or multi-station setups.

- header – Label that will appear on the ‘tab’.
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).
- firstPlayerIndex – Allows you to set the Media Player number to be the first to be displayed within the Tile. In the example above, the FirstPlayerIndex is set to 1 which means the first Media Player displayed will be player number 1.
- lastPlayerIndex – This allows you to set the Media Player number of the last Media Player to be displayed within the Tile. So if you wanted to only display Media Players 1 & 2 in this Tile you would set the FirstPlayerIndex to 1 and the LastPlayerIndex to 2. The system will only display as many Media Players as you have defined so even if this setting is set to 99, if you only have four Media Players setup then only four will be displayed.
- orientation (Vertical / Horizontal) – Set whether the Media Players should be stacked horizontally (side by side) or vertically (on top of each other).
- showPlayers (True / False) – Set whether Players should be visible.
- showRecorders (True / False) – This setting toggles whether the dedicated Recorder is displayed in the Container to the right of the Media Players. You would normally only have one Recorder on screen so if you have multiple Media Player Tiles you might turn the Recorder ‘off’ (false) on the additional Player Tiles.

SmartInfo Tile

The screenshot shows a SmartInfo Tile interface. At the top, there's a header with a small profile picture of Miley Cyrus and the title "7 Things (2008) Miley Cyrus". Below the header is a tabbed interface with tabs for "Summary", "Wikipedia", "Twitter", "Notes", and "Traffic". The "Wikipedia" tab is selected, showing a Wikipedia-style entry for Miley Ray Cyrus, including her birth date (November 23, 1992) and a brief biography. Below the Wikipedia text is a Twitter feed showing a tweet with a photo of Miley Cyrus blowing a kiss. At the bottom of the tile, there's a "Presenter Notes" section with the text "Taken from the album Breakout". The footer of the tile displays "Traffic and Travel - Highways England" with a status bar showing icons for traffic, travel, and highways, along with a "Last updated: 12/07/2018 16:43:03" timestamp.

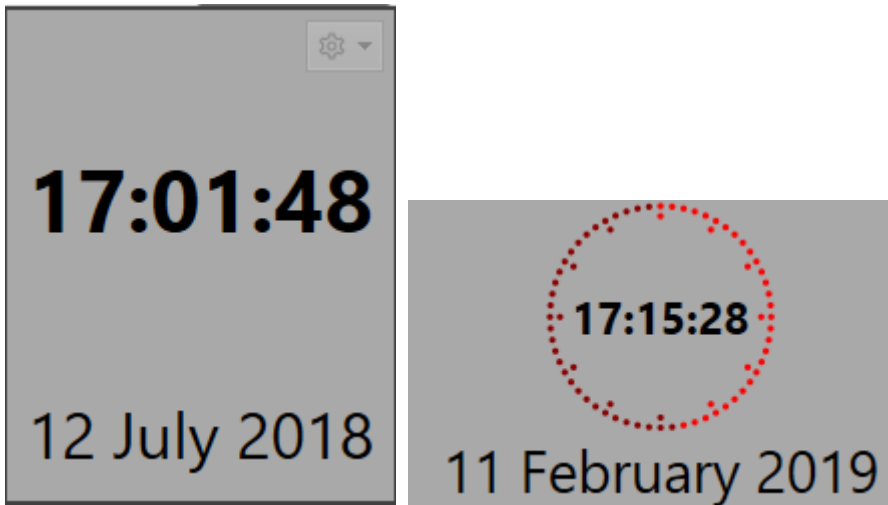
Code Snippet

```
<wl:tile type="SmartInfoView" header="SmartInfo">
  <wl:settings>
    <smartInfoViewSettings>
      <smartInfoEngineId>0</smartInfoEngineId>
    </smartInfoViewSettings>
  </wl:settings>
</wl:tile>
```

This will display a SmartInfo Tile in the selected Container. The options are:

- header – Label that will appear on the 'tab'.
- smartInfoEngineId – You can have multiple SmartInfo engines running and each SmartInfo Tile is set to display the results of the referenced engine. So in the above example, this Tile will show the results from SmartInfo engine 0 (the default). You can use this to have multiple SmartInfo Tiles on your Dynamic Layouts, that respond to input from different source Tiles so you could, for example' have one SmartInfo Tile that shows results for Media Items selected in the MediaWall and a totally separate SmartInfo Tile that always shows results for the selected item in the Log.

Clock Tile



Code Snippet

```
<wl:tile type="ClockView" header="Time">
  <wl:settings >
    <clockViewSettings>
      <backgroundColor>#000000</backgroundColor>
      <textColor>#FF0000</textColor>
      <secondTickOnColor>#FFF000</secondTickOnColor>
      <secondTickOffColor>#FF8B00</secondTickOffColor>
      <showTime>true</showTime>
      <showDate>true</showDate>
      <showSecondTicks>true</showSecondTicks>
      <timeFormat>T</timeFormat>
      <dateFormat>D</dateFormat>
      <showPreferences>true</showPreferences>
      <!--Valid Values for forceClockMode are HourCountDown and Clock-->
      <forceClockMode>HourCountDown</forceClockMode/>
    </clockViewSettings>
  </wl:settings>
</wl:tile>
```

The Clock Tile will display a simple clock / date in the selected Container. The options are:


- header – Label that will appear on the 'tab'.
- backgroundColor – Allows you to set a Hex colour for the background of the Title. You can also use WPF colour names (e.g. white etc).
- textColor – Allows you to set a Hex colour for the text of the Title. You can also use WPF colour names (e.g. white etc).
- secondTickOnColor – Allows you to set a Hex colour for the second 'ticks' that are 'on' (eg that seconds has passed in the minute). You can also use WPF colour names (e.g. white etc).
- secondTickOffColor – Allows you to set a Hex colour for the second 'ticks' that are 'off' (eg that seconds has not yet passed in the minute). You can also use WPF colour names (e.g. white etc).
- showTime (True / False) – Toggle whether the time should be displayed. Please note that this will be a simple digital clock.
- showDate (True / False) – Toggle whether the Date should be displayed below the time.
- showTicks (True / False) – Toggles whether clock should display circular second 'ticks' around the time (see second screen shot above).

- `timeFormat (T / t / HH:mm:ss.ff)` – Allows you to set the format of the time displayed. The options are:
 - `T` – Displays the default Time layout (set in Windows Control Panel) as the Long Time Format.
 - `t` – Displays the default Time layout (set in Windows Control Panel) as the Short Time Format
 - `HH:mm:ss.ff` - This uses the standard Microsoft Date / Time codes to allow you to build a time display format you choose. So for example `HH:mm:ss.ff` would display `15:35:22.88`. See table below for codes.

<code>d</code>	The day of the month, from 1 through 31.
<code>dd</code>	The day of the month, from 01 through 31.
<code>ddd</code>	The abbreviated name of the day of the week.
<code>dddd</code>	The full name of the day of the week.
<code>h</code>	The hour, using a 12-hour clock from 1 to 12.
<code>hh</code>	The hour, using a 12-hour clock from 01 to 12.
<code>H</code>	The hour, using a 24-hour clock from 0 to 23.
<code>HH</code>	The hour, using a 24-hour clock from 00 to 23.
<code>m</code>	The minute, from 0 through 59.
<code>mm</code>	The minute, from 00 through 59.
<code>M</code>	The month, from 1 through 12.
<code>MM</code>	The month, from 01 through 12.
<code>MMM</code>	The abbreviated name of the month.
<code>MMMM</code>	The full name of the month.
<code>s</code>	The second, from 0 through 59.
<code>ss</code>	The second, from 00 through 59.
<code>t</code>	The first character of the AM/PM designator.
<code>tt</code>	The AM/PM designator.
<code>y</code>	The year, from 0 to 99.
<code>yy</code>	The year, from 00 to 99.
<code>yyyy</code>	The year as a four-digit number.
<code>yyyyy</code>	The year as a five-digit number.

Please note that case is very important in these codes, for example 'm' means Minutes where as 'M' means Months! More information about building your own formats can be found on the Microsoft website at <https://docs.microsoft.com/en-us/dotnet/standard/base-types/standard-date-and-time-format-strings> and <https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings>

- `dateFormat (D / d / dddd, dd MMMM yyyy)`
 - `D`– Displays the default Date layout (set in Windows Control Panel) as the Long Date Format.
 - `d` – Displays the default Date layout (set in Windows Control Panel) as the Short Date Format
 - `dddd, dd MMMM yyyy` – This uses the standard Microsoft Date / Time codes to allow you to build a date display format you choose. So for example `dddd, dd MMMM yyyy` would display Thursday, 12 July 2018. See table above for codes.

- showPreferences (True / False) – This toggle options adds a small cog icon  to the top right of the Clock to allow the presenter to toggle between a standard clock and a count-down to the end of the current hour.
- forceClockMode (Clock / HourCountDown) – This option is disabled by default and looks like this in the example layout files.

```
<!--Valid Values for forceClockMode are HourCountDown and Clock-->
<forceClockMode/>
```

If you want to use this option you should change the above lines to look like this:

```
<forceClockMode>Clock</forceClockMode>
```

This will allow you to force the Clock Tile to either display a simple clock “Clock” or count down to the end of the current hour “HourCountDown”.

WARNING – If you use this then you should delete the showPreferences option as this will override the user options displayed when showPreferences is set to True which may confuse your users.

Leave this field blank if you would like the user to be able to switch between modes.

Pad Tile



```
<wl:tile type="PadView" header="Pad" ribbonIsMinimised="false">
  <wl:settings>
    <padViewSettings>
      <lockToStationId>-1</lockToStationId>
    </padViewSettings>
  </wl:settings>
</wl:tile>
```

This will display a Pad in the selected Container. The only options are:

- header – Label that will appear on the 'tab'.
- ribbonIsMinimised (True / False) – Whether ribbon is always visible (false) or only when clicked (true).
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).

Media Browser Tile



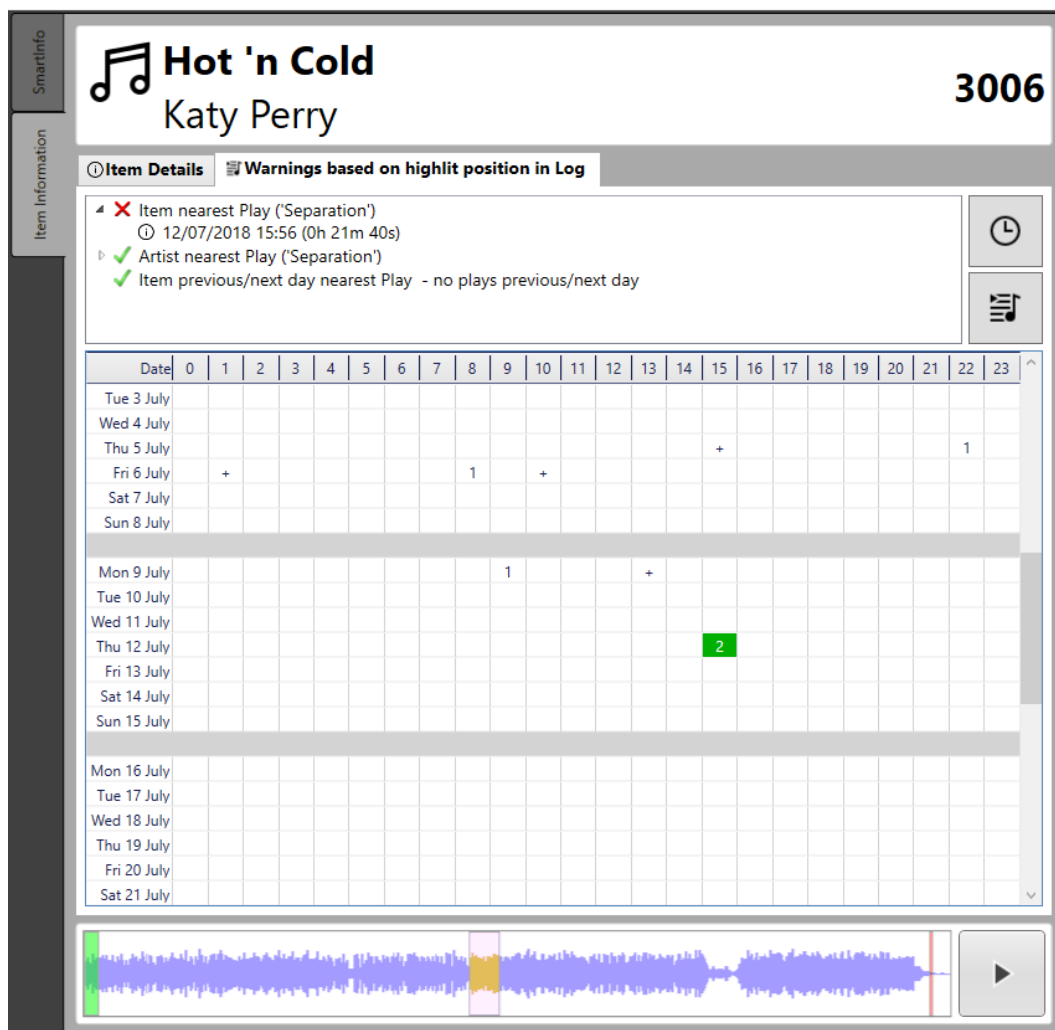
Code Snippet

```
<wl:tile type="MediaBrowserView" header="Media Browser" >
  <wl:settings>
    <mediaBrowserViewSettings>
      <lockToStationId>-1</lockToStationId>
    </mediaBrowserViewSettings>
  </wl:settings>
</wl:tile>
```

This will display a Media Browser tile in the selected Container. The Media Browser is a cut down version of the MediaWall tile and is useful in layouts where you want to be able to view and select Media Items without all the features of the full MediaWall. The options are:

- header – Label that will appear on the 'tab'.
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).

Media Item Information Tile

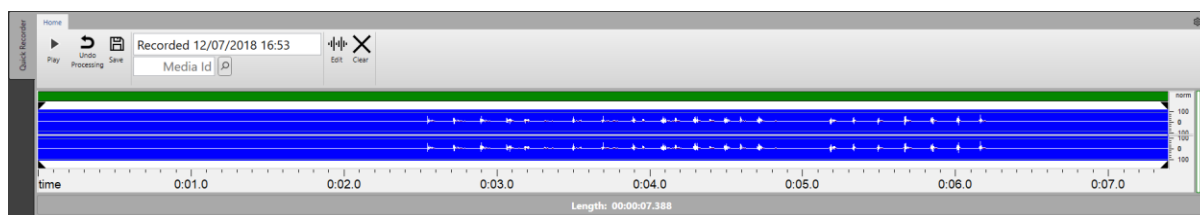


```
<wl:tile type="MediaItemInfoView" header="Item Information" ribbonIsMinimised="false">
  <wl:settings>
    <mediaItemInfoViewSettings>
      <lockToStationId>-1</lockToStationId>
    </mediaItemInfoViewSettings>
  </wl:settings>
</wl:tile>
```

The Media Item Information Tile is used to display general information about a selected Media Item as well as recent play information (based on the Log) and schedule warnings.

- header – Label that will appear on the 'tab'.
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).

QuickRecorder Tile



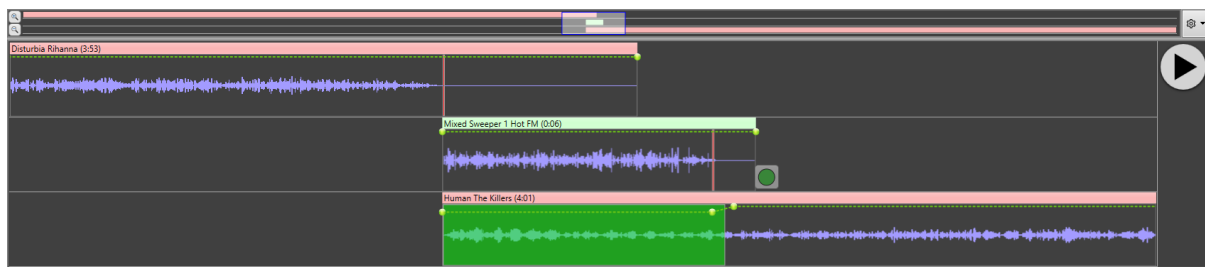
Code Snippet

```
<wl:tile type="QuickRecorderView" header="Quick Recorder" ribbonIsMinimised="false">
  <wl:settings>
    <quickRecorderSettings>
      <compactMode>False</compactMode>
      <lockToStationId>-1</lockToStationId>
    </quickRecorderSettings>
  </wl:settings>
</wl:tile>
```

The QuickRecorder is a simple recording tool designed to speed up the recording and editing process by automating as much as the workflow as possible. This tile allows you to place a QuickRecorder inside the selected Container. The options are:

- header – Label that will appear on the ‘tab’.
- ribbonIsMinimised (True / False) – Whether ribbon is always visible (false) or only when clicked (true).
- compactMode (True / False) – This allows you to toggle whether the QuickRecorder will display the waveform while recording and processing. With this option set to false, the QuickRecorder can be used in smaller Containers.
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).

Segue Editor Tile



Code Snippet

```
<wl:tile type="SegueEditorView" header="Segedit">
  <wl:settings>
    <segueEditorViewSettings>
      <lockToStationId>-1</lockToStationId>
    </segueEditorViewSettings>
  </wl:settings>
</wl:tile>
```

The SegueEditor Tile is used to place a SegueEditor inside the selected Container. The options are:

- header – Label that will appear on the 'tab'.
- lockToStationId – This parameter allows you to lock the tile to a specific Station ID which is useful in multiple Station environments. The default value is -1 which means the Tile is not locked and instead will always follow the currently open and active Station. If you enter a specific number (from 0 (zero) onwards) then this Tile will always be locked to that Station (whether it is open or not).

SmartDisplay Tile



Code Snippet

```
<wl:tile type="SmartDisplayView" header="SmartDisplay">
  <wl:settings>
    <smartDisplaySettings>
      <backgroundColor>Black</backgroundColor>
      <fillWith>Image</fillWith>
      <text>
        <position>
          <margin>0</margin>
          <edge>Bottom</edge>
          <verticalAlignment>Center</verticalAlignment>
          <horizontalAlignment>Center</horizontalAlignment>
        </position>
        <value></value>
        <fontSize>20</fontSize>
        <fontFamily>Arial</fontFamily>
        <fontWeight>bold</fontWeight>
        <backgroundColor>#660000</backgroundColor>
      </text>
      <image>
        <position>
          <margin>0</margin>
          <edge>Center</edge>
          <verticalAlignment>Center</verticalAlignment>
          <horizontalAlignment>Center</horizontalAlignment>
        </position>
        <location>c:\MyFiles\MyLogo.png</location>
      </image>
    </smartDisplaySettings>
  </wl:settings>
</wl:tile>
```

The SmartDisplay Tile allows you to add text and an image to a Container so that you can add messages or logos to your Dynamic Layouts. The options for SmartDisplay Tiles include:

- header – Label that will appear on the ‘tab’.
- backgroundColor – Sets the background colour for the panel. Choose a WPF colour name or hex value.

- fillWith (Text / Image) – This allows you to set whether the text block or the image block is the primary element within the tile. This affects the positioning of the opposite element. So if we set the 'FillWith' property to be 'Image' the then Image is the primary element and the text will be positioned relative to the Image using the Text Edge property (and the Image Edge property will be ignored).
- text – Allows you to add a block of text to the Container. There are a number of sub options you can use to adjust the look of the text you are adding.
 - margin – Set the margin (pixels) around the text.
 - edge (Bottom / Top / Left / Right) – Sets the position of the text relative to the image component. This property is only used if the FillWith property is set to 'Image'.
 - verticalAlignment – Set the vertical alignment of the text.
 - horizontalAlignment - Set the horizontal alignment of the text.
 - value – The actual text that you want to be displayed.
 - fontSize – Sets the font size.
 - fontFamily – Sets the font family.
 - fontWeight (Normal / Bold) – Sets the font weight.
- image – Allows you to add an image element to the Tile. There are a number of sub-options.
 - margin – Set the margin (pixels) around the image.
 - edge (Bottom / Top / Left / Right) – Sets the position of the image relative to the text component. This property is only used if the FillWith property is set to Text.
 - verticalAlignment – Set the vertical alignment of the Image.
 - horizontalAlignment - Set the horizontal alignment of the Image.
 - location – Set the location of the image file you want to use.

Simple Mixer View Tile (Mic Live for SRM / Webstation / Airlite / Capitol IP / Forum IP)



Code Snippet

```
<wl:tile type="SimpleMixerView" header="Mic Status">
  <wl:settings>
    <mixerViewSettings>
      <channel>
        <index>1</index>
      </channel>
      <textValueOff></textValueOff>
      <textValueOn></textValueOn>
      <backgroundColorOff>Black</backgroundColorOff>
      <backgroundColorOn>Black</backgroundColorOn>
      <imageLocationOff>c:\MyFiles\SR1Mic1Off.png</imageLocationOff>
      <imageLocationOn>c:\MyFiles\SR1Mic1On.png</imageLocationOn>
    </mixerViewSettings>
    <smartDisplaySettings>
      <fillWith>Image</fillWith>
      <text>
        <position>
          <margin>0</margin>
          <edge>right</edge>
          <verticalAlignment>Top</verticalAlignment>
          <horizontalAlignment>Right</horizontalAlignment>
        </position>
        <fontSize>20</fontSize>
        <fontFamily>Arial</fontFamily>
        <fontWeight>bold</fontWeight>
      </text>
      <image>
        <position>
          <margin>0</margin>
          <edge>right</edge>
          <verticalAlignment>Top</verticalAlignment>
          <horizontalAlignment>Right</horizontalAlignment>
        </position>
      </image>
    </smartDisplaySettings>
  </wl:settings>
```

</wl:tile>

The Simple Mixer View Tile is used to display realtime information from a range of compatible mixing consoles. The list of compatible consoles includes:

- Broadcast Radio SRM v1 & v2
- D&R Webstation
- AEQ Capitol IP
- AEQ Forum IP

This is primarily used to display 'mic live' information which gives a visual indication when one or microphone 'faders' are in the 'open' position.

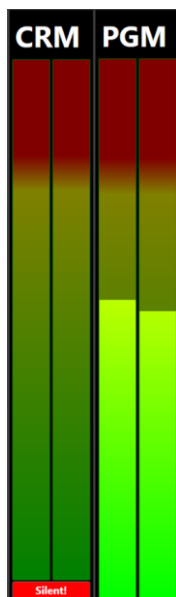
The Simple Mixer Tile is closely related to the Smart Display Tile and shares many of the same attributes and options.

These include:

- header – Label that will appear on the 'tab'.
- index (0 to 4) – This sets the hardware index channel that the Tile will respond to. If set to 0 then the Tile will activate when any microphone channel is in use. If you set the Index to a number (1-4) then the tile will only activate when the corresponding mic channel number is 'open'. So as an example, if you wanted this Tile to only activate when the third microphone is active, set the Index to 3.
- textValueOff – Set the text to be displayed when the Tile is not active.
- textValueOn - Set the text to be displayed when the Tile is active.
- backgroundColourOff – Sets the background colour of the tile when the Tile is not active. (Uses HTML colour names or WPF colour codes).
- backgroundColourOn – Sets the background colour of the tile when the Tile is active. (Uses HTML colour names).
- imageLocationOn – Allows you to set the location of an image to display when the SmartMixerView is active.
- imageLocationOff – Allows you to set the location of an image to display when the SmartMixerView is not active.
- fillWith (Text / Image) – This allows you to set whether the text block or the image block is the primary element within the tile. This affects the positioning of the opposite element. So if we set the FillWith property to be 'Image' the then image is the primary element and the text will be positioned relative to the image using the textEdge property (and the imageEdge property will be ignored).
- text – Allows you to add a block of text to the Container. There are a number of sub options you can use to adjust the look of the text you are adding.
 - margin – Set the margin (pixels) around the text.
 - edge (Bottom / Top / Left / Right) – Sets the position of the text relative to the image component. This property is only used if the FillWith property is set to Image.
 - verticalAlignment – Set the vertical alignment of the text.
 - horizontalAlignment - Set the horizontal alignment of the text.
 - value – The actual text that you want to be displayed.
 - fontSize – Sets the font size.
 - fontFamily – Sets the font family.
 - fontWeight (Normal / Bold) – Sets the font weight.

- image – Allows you to add an image element to the Tile. There are a number of sub-options.
 - margin – Set the margin (pixels) around the image.
 - edge (Bottom / Top / Left / Right) – Sets the position of the image relative to the text component. This property is only used if the FillWith property is set to Text.
 - verticalAlignment – Set the vertical alignment of the Image.
 - horizontalAlignment - Set the horizontal alignment of the Image.
 - location – Set the location of the image file you want to use.

Audio Monitor Tile (Line In / Broadcast Radio Audio Monitor / D&R Webstation / D&R Airlite)



Code Snippet

```
<wl:tile type="AudioMonitorView" header="Levels">
  <wl:settings>
    <audioMonitorViewSettings>
      <orientation>vertical</orientation>
      <ams:audioMonitorSettings
xmlns:ams="http://schemas.broadcastradio.com/audioMonitor/2016/audioMonitorSettings">
        <ams:display>
          <ams:defaultCaption>
            <![CDATA[My Station]]>
          </ams:defaultCaption>
          <ams:minimumDisplayLevel>10</ams:minimumDisplayLevel>
          <ams:maximumDisplayLevel>100</ams:maximumDisplayLevel>
        </ams:display>
        <ams:source type="LineIn">
          <ams:lineInDeviceId>-1</ams:lineInDeviceId>
          <ams:remoteTcpAddress>MyEncodingPC</ams:remoteTcpAddress>
          <ams:remoteTcpPort>6991</ams:remoteTcpPort>
          <ams:externalMixerSourceReference>controlroom</ams:externalMixerSourceReference>
        </ams:source>
        <ams:useShareTcpPort>false</ams:useShareTcpPort>
        <ams:shareTcpPort>6991</ams:shareTcpPort>
        <ams:useBlinkDevice>false</ams:useBlinkDevice>
        <ams:silenceThreshold>50</ams:silenceThreshold>
        <ams:silenceLengthStart>00:00:05</ams:silenceLengthStart>
        <ams:silenceLengthEnd>00:00:02</ams:silenceLengthEnd>

        <ams:notification>
          <ams:emailEnabled>true</ams:emailEnabled>
          <ams:emailServer>
            <serverDetails>
              <serverAddress>mymailserver.mystation.com</serverAddress>
              <serverPort>25</serverPort>
              <useSsl>false</useSsl>
              <requireLogin>false</requireLogin>
              <userName>silenceAlert@mystation.com</userName>
              <password>mypassword</password>
```

```

        </serverDetails>
    </ams:emailServer>
    <ams:emailSender>
        <emailAddress>
            <address>silenceAlert@mystation.com</address>
        </emailAddress>
    </ams:emailSender>
    <ams:emailRecipients>
        <emailAddresses>
            <emailAddress>
                <address>alerts@mystation.com</address>
            </emailAddress>
        </emailAddresses>
    </ams:emailRecipients>

</ams:notification>
<ams:events>
    <ams:onSilenceStart>
        <ams:launchApplication>
            <ams:enabled>false</ams:enabled>
            <ams:launchPath>notepad.exe</ams:launchPath>
            <ams:commandLine>silentStarted.txt</ams:commandLine>
        </ams:launchApplication>
        <ams:getUrl>http://myinternalserver/silentStarted</ams:getUrl>
    </ams:onSilenceStart>
    <ams:onSilenceEnd>
        <ams:launchApplication>
            <ams:enabled>false</ams:enabled>
            <ams:launchPath>notepad.exe</ams:launchPath>
            <ams:commandLine>silentEnded.txt</ams:commandLine>
        </ams:launchApplication>
        <ams:getUrl>http://myinternalserver/silentEnded</ams:getUrl>
    </ams:onSilenceEnd>
</ams:events>

    </ams:audioMonitorSettings>
</audioMonitorViewSettings>
</wl:settings>
</wl:tile>

```

The Audio Monitor Tile can be used to link with the Broadcast Radio Audio Monitor software (which is available for free) or from a local 'line in' audio source to provide some simple 'confidence' audio metering on your Dynamic Layout.

The Audio Monitor Tile is not intended as a replacement for your desk metering.

The options for the Audio Monitor Tile include:

- header – Label that will appear on the 'tab'.
- orientation (horizontal / vertical) – Set whether the VU will be displayed horizontally or vertically.
- audioMonitoringSettings – Leave this setting as the default.
- defaultCaption – You can set the name displayed on the VU. You must only change the section underlined below.

```
<![CDATA[My Station]]>
```

- minimumDisplayLevel – You can offset the minimum displayed level on the VU. It is recommend you set this to 40 as this will make the VU's show clearly when levels are too low.

- `maximumDisplayLevel` – This sets the maximum level displayed. Leave this at 100 in most cases.
- `source` (`LineIn` / `RemoteTcp` / `ExternalMixerSource`) – You can set whether the monitor should be monitoring audio from a remote Audio Monitor application, a compatible external mixer or from a local 'line in' audio source.
 - `lineInDeviceID` (-1 / 0-X) – This setting is only used if the previous setting is set to 'LineIn' or `externalMixerSource`. It is used to set the device ID of the audio input you want to use. If you set this to -1 then the default windows audio input will be used. If you specify another number then the corresponding device will be used. You may need to experiment with this.
 - `remoteTcpAddress` – This setting is only used if you are connecting to a remote Audio Monitor application and it is used to set the IP address (or computer name) of the PC running the Audio Monitor application.
 - `remoteTcpPort` – Only used when connecting to a remote Audio Monitor application and it sets the port that is used for communication. It should be left on the default 6991.
 - `externalMixerSourceReference` (`program` / `controlroom`) – Only used when using an `externalMixerSource` as the source (works with D&R Webstation & D&R Airlite). This can be set to 'program' or 'controlroom' depending on whether you want to VU's to display Program or Control Room outputs.
- `useShareTcpPort` (`true` / `false`) – Used to set whether to 'share' local audio monitor information (from `LineIn`) with remote connections (for example, a `SmartSign`).
- `shareTcpPort` (6991) – Used to specify the port that remote applications (such as `SmartSign`) will use to connect to this Audio Monitor. Default is 6991.
- `useBlinkDevice` (`true` / `false`) – This can be used to control a 'Blink USB' LED indicator. Contact sales for more information.
- `silenceThreshold` (0 – 100) - Set the level below which the system considers the input level to be 'silent'.
- `silenceLengthStart` – Set the time which the audio level needs to be considered silent before the display will switch to the 'silence' warning.
- `silenceLengthEnd` – Set the time duration that the audio level is above the Silence Threshold after which the 'silence' warning is cleared.

The Notification settings section some advanced options related to what to do when the Audio Monitor Tile detects silence. Please note there are many 'structural' node which can be ignored. The list below covers only the nodes that contain data that needs to be set.

- `emailEnabled` (`true` / `false`) – Enables sending an email when 'silence' is detected.
- `serverDetails` – Settings in this node refer to the email server to be used.
 - `serverAddress` – Add the address of your email server.
 - `serverPort` – Add in the port that should be used to communicate with the email server.
 - `useSsl` (`true` / `false`) – Set whether to use secure SSL protocol.
 - `requireLogin` (`true` / `false`) – Set whether you need to login to access email server.
 - `userName` – If login required then enter login name.
 - `password` – Set the password used to access email server.
- `emailSender` – Settings in this node refer to the sender of the alert email.
 - `address` – Type in the email address the alert email should be send to.
- `EmailRecipients` – Settings in this node refer to the recipient of the alert email.
 - `address` – The email address of the person you want to send the alert to.

The next section refers to actions that can occur when a silence notification starts (in the `onSilentStart` node)

- **launchApplication** – The settings in this node refer to the action of launching an application when a silence notification starts.
 - **enabled (true / false)** – Sets whether an applications should be launched when a silence alert starts.
 - **launchPath** – Sets the path for the application you want to launch (should be the full path to the .exe).
 - **commandLine** – Allows you to add a commandline 'switch' to the application you are launching.
- **getURL** – Allows you to specify an HTTP Get to be actioned when a silence starts.

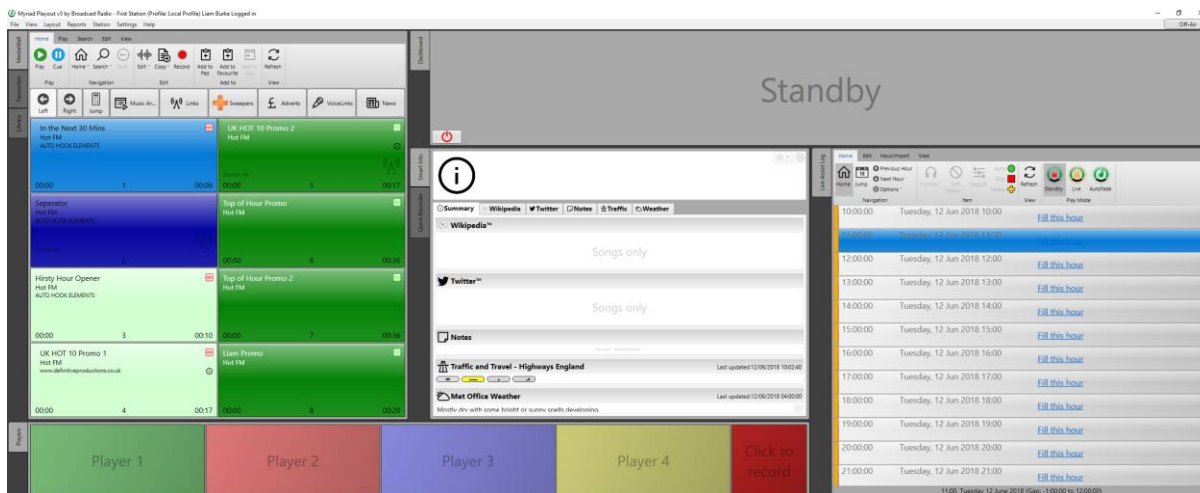
The next section refers to actions that can occur when a silence notification ends (in the onSilentEnd node)

- **launchApplication** – The settings in this node refer to the action of launching an application when a silence notification ends.
 - **enabled (true / false)** – Sets whether an applications should be launched when a silence alert ends.
 - **launchPath** – Sets the path for the application you want to launch (should be the full path to the .exe).
 - **commandLine** – Allows you to add a commandline 'switch' to the application you are launching.
- **getURL** – Allows you to specify an HTTP Get to be actioned when a silence ends.

This Tile is very flexible but because of the flexibility, it also has a large number of settings.

Real World Example

Now that we know what Tiles are available and how to create the Dynamic Layout XML file, let's take a look at building a real world example.



This is the CartWheel Dynamic Layout that is one of the 'pre-built' layouts included with Myriad 5 Playout. It is a single screen layout that combines many of the elements you would normally use in an 'on air' studio.

So let's see how it is built.

To start with we need the basic elements for our Dynamic Layout.

```
<?xml version="1.0" encoding="utf-8" ?>
<wl:layout header="Cartwheel" version="5.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wl="http://schemas.broadcastradio.com/myriad/2016/windowLayout"
  xsi:schemaLocation="urn:Layout Layout.xsd">
  <wl:screens>
    <wl:screen index="0" totalRows="12" totalColumns="12">
      <wl:containers>

        </wl:containers>
      </wl:screen>
    </wl:screens>
  </wl:layout>
```

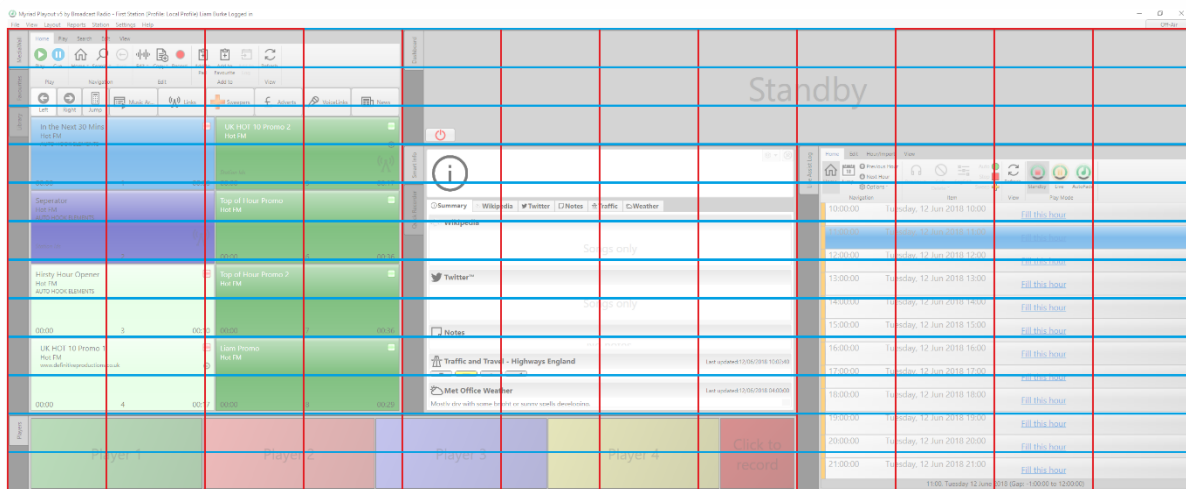
Grid View

The above code snippet show the basic starting point for all Dynamic Layouts. In this case we have a single **screen** with the **screen index = 0** (so everything within that node will be displayed on your first screen) and **Row = 12** and **Column = 12**. This means that Myriad 5 Playout will divide the screen into 12 rows and 12 columns when building the Dynamic Layout.

TIP: It is worth noting at this stage that we have not defined the size of the columns or rows, instead Myriad will take the screen resolution and divide it into the amount of rows and columns in the Dynamic Layout file which means that this layout will work on standard and wide screen monitors and on monitors with different

resolutions, however, you may wish to create different Dynamic Layouts for use on PC's with widely different screen types or configurations (for example screen in portrait mode).

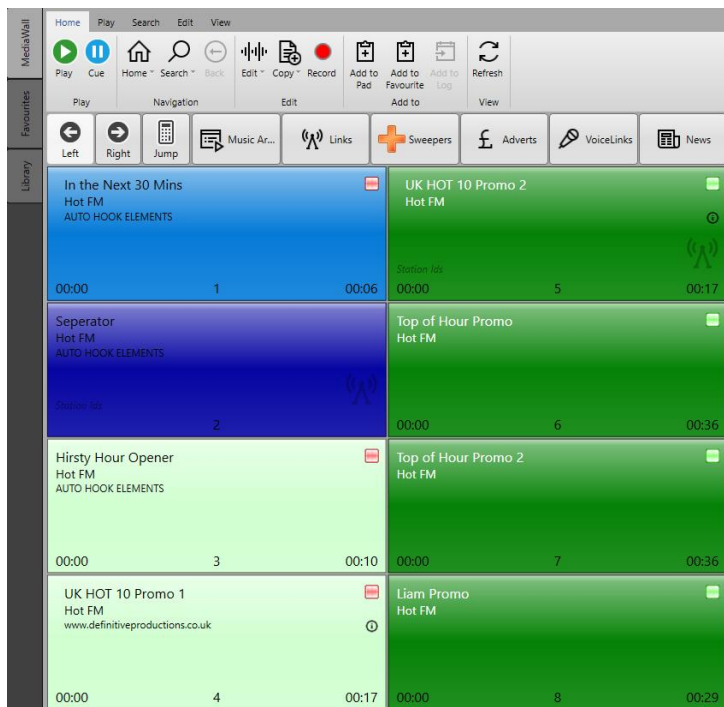
It is easier to visualise the next stage if we place a 12 x 12 grid onto of our design.



As you can see, each **Container** fits within the 12 x 12 grid. Let's add the first **Container** and the **Tiles** within.

Adding The First Container

The first **Container** we need to add contains three **Tiles**, the **Media Wall Tile**, **Favourites Tile** & **Library Tile**.



By referring to the grid view above, we can see that the top left corner of this **Container** needs to be on the **Top Row** and **Left Hand Column**. These will be **Row 0, Column 0**. We can also see from the grid view that that **Container** needs to **4 x Columns Wide** and **10 x Rows High**. So the XML we need to add to our Dynamic Layout File will look like this:

```
<wl:container type="tab" >
  <wl:position row="0" column="0" rowSpan="10" columnSpan="4"/>
  <wl:tiles>
```

```

<wl:tile type="MediaGridView" header="MediaWall">
  <wl:settings>
    <mediaGridViewSettings>
      <smartInfoEngineId>0</smartInfoEngineId>
    </mediaGridViewSettings>
  </wl:settings>
</wl:tile>
<wl:tile type="FavouritesSetView" header="Favourites">
  <wl:settings/>
</wl:tile>
<wl:tile type="MediaLibraryView" header="Library" ribbonIsMinimised="false">
  <wl:settings>
    <mediaLibraryViewSettings>
      <smartInfoEngineId>0</smartInfoEngineId>
    </mediaLibraryViewSettings>
  </wl:settings>
</wl:tile>

</wl:tiles>
</wl:container>

```

Container Position & Size

The position and size of the **Container** are defined in the <position> tag in the above example.

```
<wl:position row="0" column="0" rowSpan="10" columnSpan="4"/>
```

Here we can see that the **Container** will start at **row=0** and **column=0** which will be the top left of the grid. The **rowSpan=10** which means that it will be 10 Row high and the **columnSpan=4** means it will be 4 columns wide.

Adding In The Tiles

We know that this **Container** needs to host three different **Tiles** so it must be a **Tabbed** Container (which it is – see first line of code). Once you have defined the size and position of the Container, it is time to add the content as individual **Tile** nodes. The available **Tiles** were covered in the previous section and you can simply cut and paste each Tile into the

```

<wl:container type="tab" >
  <wl:position row="0" column="0" rowSpan="10" columnSpan="4"/>
  <wl:tiles>
    <wl:tile type="MediaGridView" header="MediaWall">
      <wl:settings>
        <mediaGridViewSettings>
          <smartInfoEngineId>0</smartInfoEngineId>
        </mediaGridViewSettings>
      </wl:settings>
    </wl:tile>
    <wl:tile type="FavouritesSetView" header="Favourites">
      <wl:settings/>
    </wl:tile>
    <wl:tile type="MediaLibraryView" header="Library" ribbonIsMinimised="false">
      <wl:settings>
        <mediaLibraryViewSettings>
          <smartInfoEngineId>0</smartInfoEngineId>
        </mediaLibraryViewSettings>
      </wl:settings>
    </wl:tile>
  </wl:tiles>
</wl:container>

```

This Tile node will add the MediaWall as the first Tab

Paste in the example **Tile** code for the **MediaWall**, **Favourites** & **Library**. Make sure all three sit within the **Tiles** node.

Next we want to add the **Dashboard** Container and Tile. The code below will create a new Container that is positioned to start at **Row 0** and **Column 5** and will be 8 columns wide and 3 rows high.

```
<wl:container type="tab" >
  <wl:position row="0" column="4" rowSpan="3" columnSpan="8"/>
  <wl:titles>
    <wl:tile type="DashboardView" header="Dashboard">
      <wl:settings>
        <dashboardViewSettings>
          <showPictures>true</showPictures>
          <stretchGoButtonToFill>true</stretchGoButtonToFill>
          <smartInfoEngineId>0</smartInfoEngineId>
        </dashboardViewSettings>
      </wl:settings>
    </wl:tile>
  </wl:titles>
</wl:container>
```

Now repeat the same process for each of the additional **Containers** used in the final layout. Take care to position and size each container to match the target layout. The final layout will should look something like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<wl:layout header="Cartwheel" version="5.0"
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:wl='http://schemas.broadcastradio.com/myriad/2016/windowLayout'
  xsi:schemaLocation="urn:Layout Layout.xsd">
  <wl:screens>
    <wl:screen index="0" totalRows="12" totalColumns="12">
      <wl:containers>

        <wl:container type="tab" >
          <wl:position row="0" column="0" rowSpan="10" columnSpan="4"/>
          <wl:titles>
            <wl:tile type="MediaGridView" header="MediaWall">
              <wl:settings>
                <mediaGridViewSettings>
                  <smartInfoEngineId>0</smartInfoEngineId>
                </mediaGridViewSettings>
              </wl:settings>
            </wl:tile>
            <wl:tile type="FavouritesSetView" header="Favourites">
              <wl:settings/>
            </wl:tile>
            <wl:tile type="MediaLibraryView" header="Library" ribbonIsMinimised="false">
              <wl:settings>
                <mediaLibraryViewSettings>
                  <smartInfoEngineId>0</smartInfoEngineId>
                </mediaLibraryViewSettings>
              </wl:settings>
            </wl:tile>
          </wl:titles>
        </wl:container>
        <wl:container type="tab" >
          <wl:position row="3" column="4" rowSpan="7" columnSpan="4"/>
          <wl:titles>
            <wl:tile type="SmartInfoView" header="Smart Info">
              <wl:settings>
```



```

        <smartInfoViewSettings>
            <smartInfoEngineId>0</smartInfoEngineId>
        </smartInfoViewSettings>
    </wl:settings>
</wl:tile>
<wl:tile type="QuickRecorderView" header="Quick Recorder">
    <wl:settings/>
</wl:tile>
</wl:tiles>
</wl:container>

```

```

<wl:container type="tab" >
    <wl:position row="0" column="4" rowSpan="3" columnSpan="8"/>
    <wl:tiles>
        <wl:tile type="DashboardView" header="Dashboard">
            <wl:settings>
                <dashboardViewSettings>
                    <showPictures>true</showPictures>
                    <stretchGoButtonToFill>true</stretchGoButtonToFill>
                    <smartInfoEngineId>0</smartInfoEngineId>
                </dashboardViewSettings>
            </wl:settings>
        </wl:tile>
    </wl:tiles>
</wl:container>

```

```

<wl:container type="tab" >
    <wl:position row="3" column="8" rowSpan="9" columnSpan="4"/>
    <wl:tiles>
        <wl:tile type="LogView" header="Live Assist Log">
            <wl:settings>
                <logViewSettings>
                    <lockToStationId>-1</lockToStationId>
                    <smartInfoEngineId>0</smartInfoEngineId>
                </logViewSettings>
            </wl:settings>
        </wl:tile>
    </wl:tiles>
</wl:container>

```

```

<wl:container type="tab" >
    <wl:position row="10" column="0" rowSpan="2" columnSpan="8"/>
    <wl:tiles>
        <wl:tile type="MediaPlayersView" header="Players">
            <wl:settings/>
        </wl:tile>
    </wl:tiles>
</wl:container>
</wl:containers>
</wl:screen>

```

```

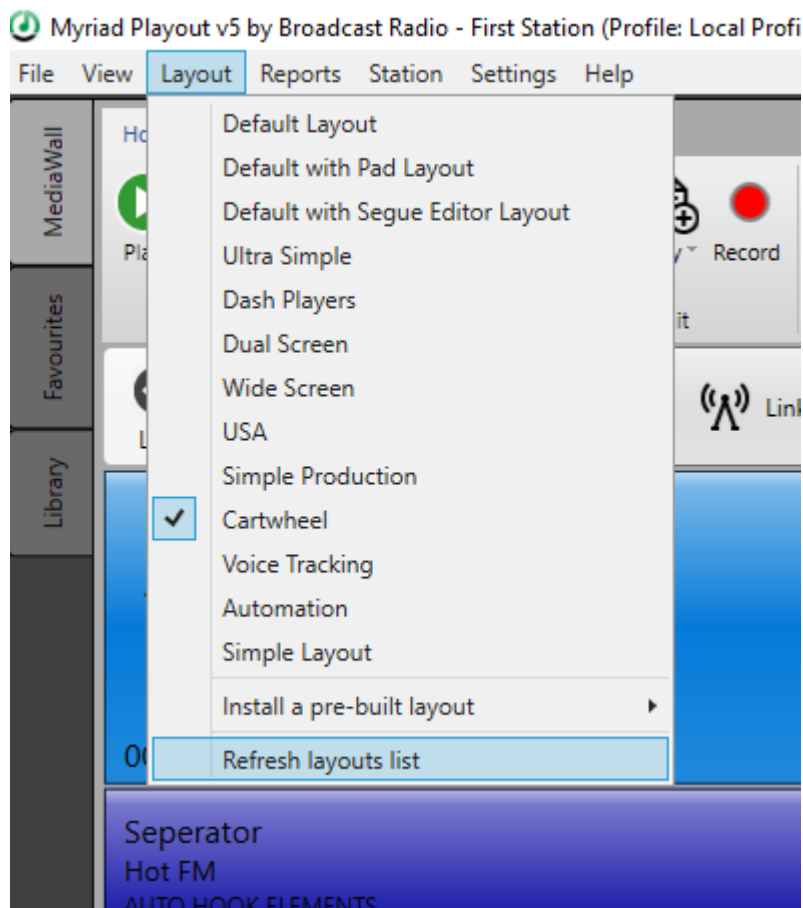
</wl:screens>
</wl:layout>

```

Reviewing Your Layouts In Myriad Playout

Hopefully, this document combined with the example layouts that are pre-installed with Myriad Playout will give you all the information you need to start modifying or creating your own Dynamic Layouts.

Once you have created your new Dynamic Layouts, just drop them in the Dynamic Layouts folder and click on the **Layouts Menu > Refresh Layouts List** and your new layout should appear in the menu. Load it up and enjoy your new Myriad 5 Dynamic Layout.



If you have modified one of the 'pre-built' layouts but need to restore to the originally shipped layout you can use the **Install A Pre-Build Layout** menu option to select and restore any of our originally shipped Dynamic layouts.



US Playout Style

Description: Designed to emulate US style playout systems with the focus on the next X items in the Dashboard.

Source: Broadcast Radio

Dual Screen: Yes

Notes: Works best when Dashboard set to display 6 items. VU set to display default sound input level.

Open link in new tab
Open link in new window
Open link in incognito window

Save link as...

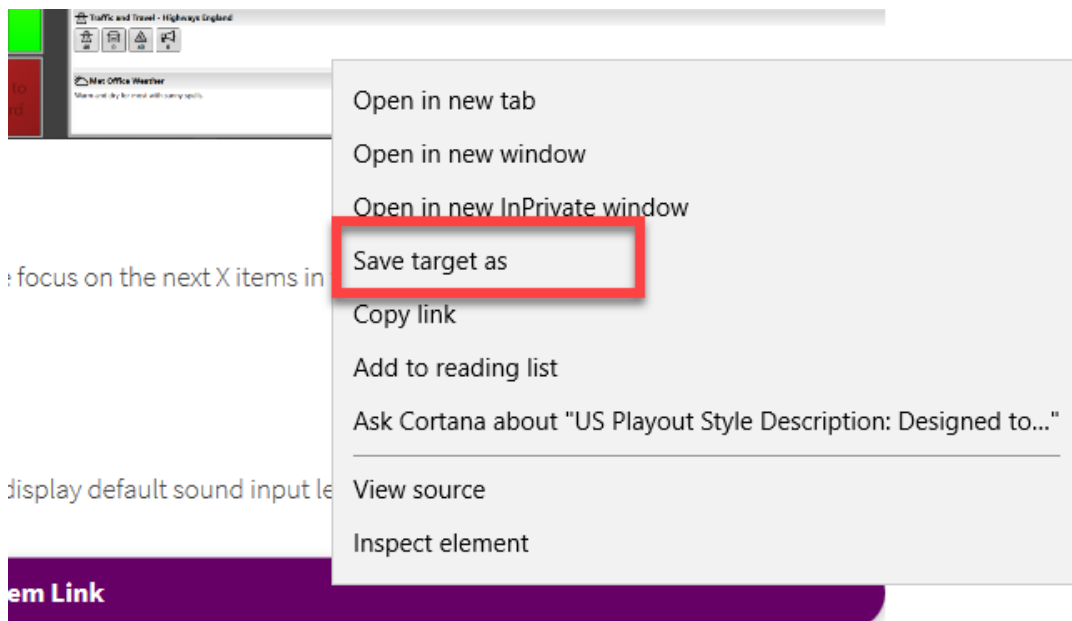
Copy link address

Inspect

Ctrl+Shift+I

Item Link

Please Note: The wording on the right click menu varies from browser to browser. On **Edge** or **Internet Explorer** the menu option you want is **Save Target As**.

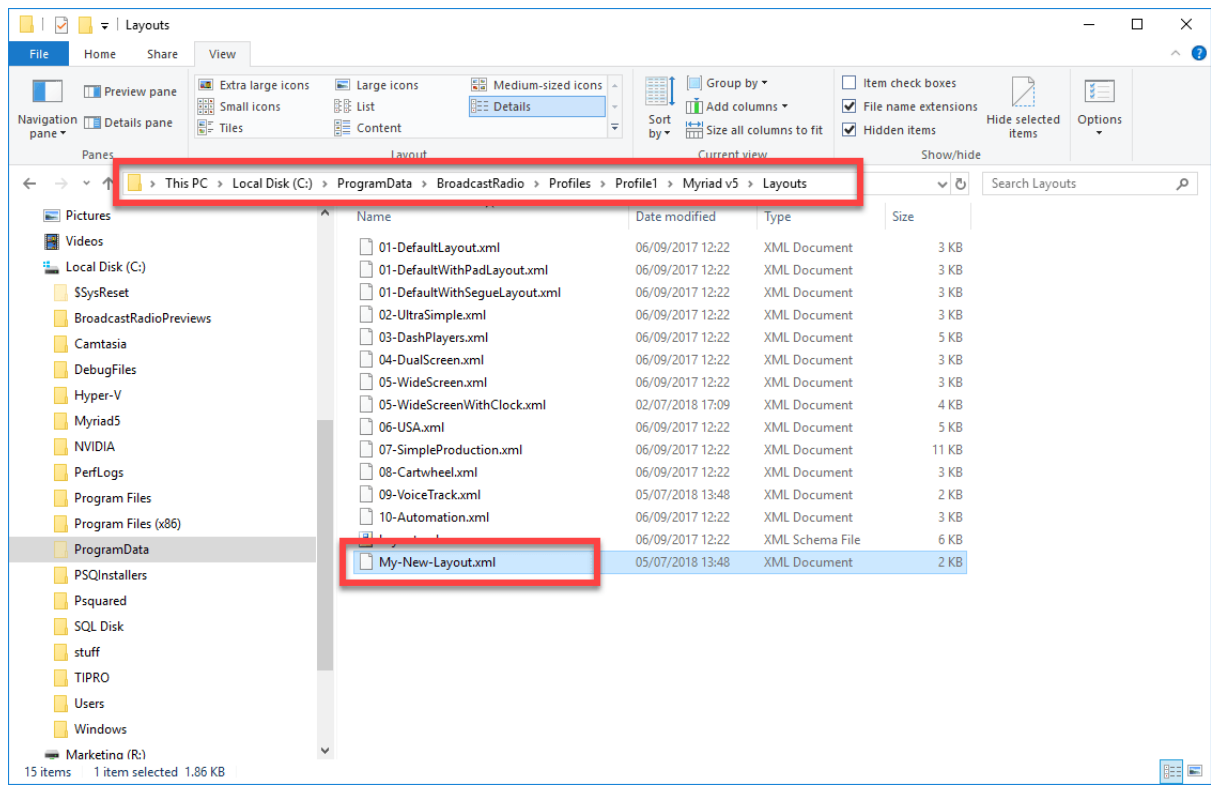


This step is needed as if you click on the button, most browsers will simply display the XML contained in the Dynamic Layout file.

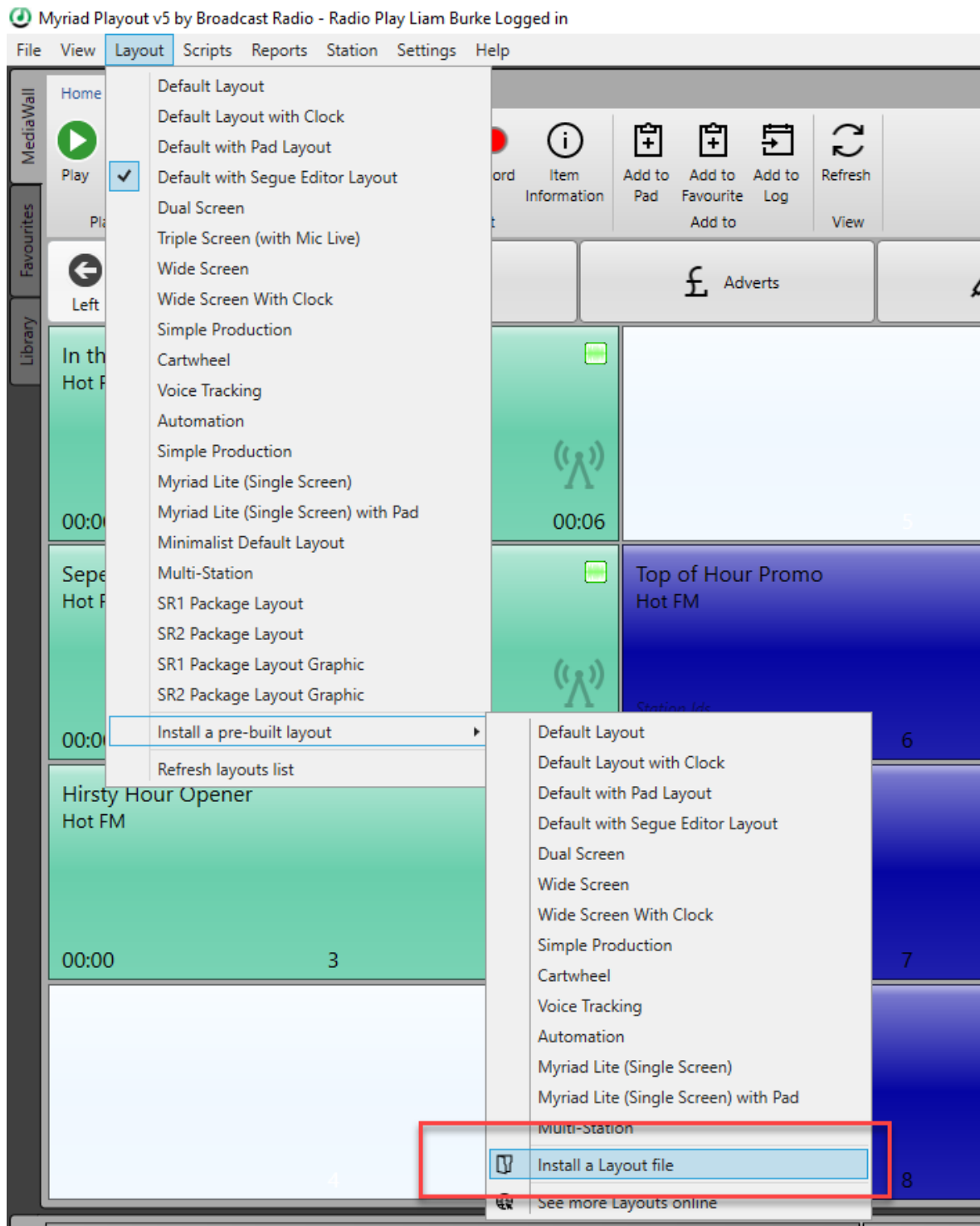
7. Save the file to the following location:

C:\ProgramData\BroadcastRadio\Profiles\Profile1\Myriad v5\Layouts

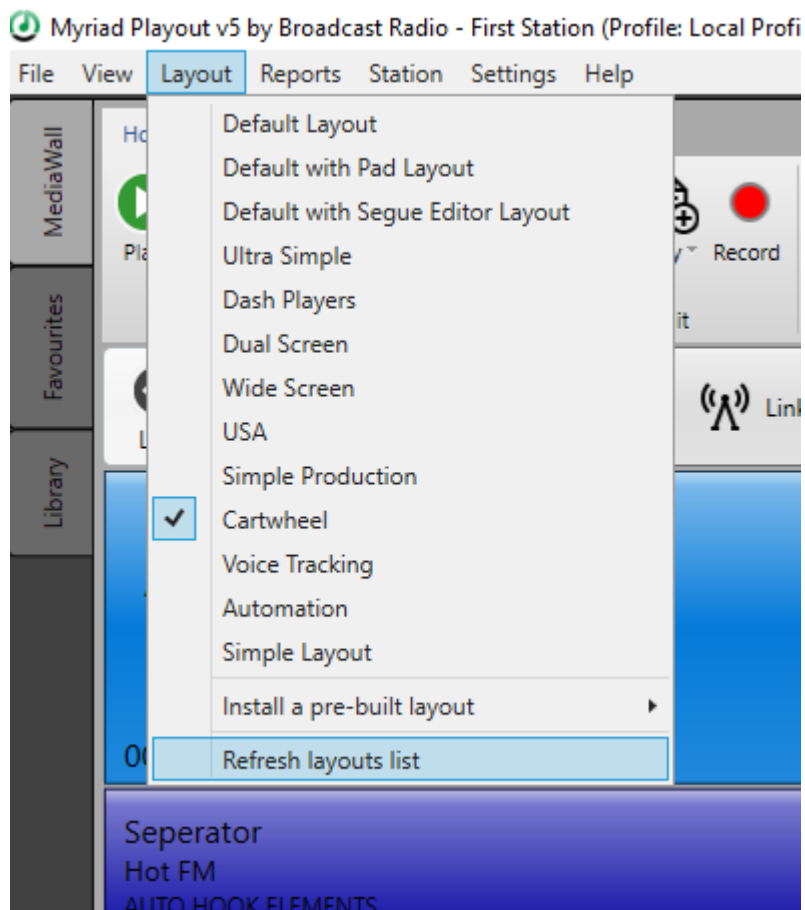
Where the **Profile1** should be replaced with an alternative Profile name if you are running multiple Profiles or have changed the default name for your Profile.



You can also save the file to your desktop and then import the Layout using the Install A Layout File option on in the Layout Menu > Install Pre-built Layouts menu option.



8. In **Myriad Playback** click on the **Layout** menu option and select **Refresh Layout List**. Your new Layout should now be available use.



If you have designed a Dynamic Layout that you think others would find useful and you would be willing to share then please send your Dynamic Layout XML file to support@trucastclient.com and we will be happy to add it to the Dynamic Layout Gallery.

More Information

For more information and the latest updates, please visit

<https://automation.trucastclient.com>

You can also watch a tutorial video on using Dynamic Layouts and other Myriad 5 Playout features by visiting.

<https://automation.trucastclient.com/videos>