Custom mixer layout configuration with a special markup language

Overview

Since build5.2.1009it is possible to to configure custom mixer layout with a special XML-based language. In this case, no need to develop custom Java class.

Custom mixer layout is a set of XML files with .mix extension placed in the same folder. Every file should describe layout for a certain number of participants: 1, 2, 3, 4 etc. File name should start from participants number followed by understrike character, the rest of file name should contain various alphanumeric characters, for example

```
1_test-mixer-layout-1-participant.mix
2_test-mixer-layout-2-participants.mix
3_test-mixer-layout-3-participants.mix
4_test-mixer-layout-4-participants.mix
...
```

One mixer layout description files should be placed in one folder. One participans count should be described by one, and only one, file. Mixing of two layouts in the same folder is not allowed.

A full path to mixer layout folder may be set with the following parameter inflashphoner.propertiesfile

```
mixer_layout_dir=/opt/test-mixer-layout
```

In this case, the layout will be applied by default to all the mixers on the server.

Also, the path to mixer layout folder can be passed in REST API query /mixer/startup:

```
POST /rest-api/mixer/startup HTTP/1.1
HOST: 192.168.1.101:8081
Content-type: application/json

{
    "uri": "mixer://mixer1",
    "localStreamName": "mixer1_stream",
    "hasVideo": true,
    "hasAudio": false,
    "mixerLayoutDir": "/opt/mixer1-layout"
}
```

Applying a custom layout to an active mixer on the fly

Since build 5.2.1480, a custom mixer layout may be applied to an active mixer on the fly

```
POST /rest-api/mixer/set_parameter HTTP/1.1
HOST: 192.168.1.101:8081
Content-type: application/json

{
    "uri": "mixer://mixer1",
    "mixerLayoutDir": "/opt/mixer1-layout"
}
```

To roll back to default mixer layout set the empty `mixerLayoutDir` parameter

```
POST /rest-api/mixer/set_parameter HTTP/1.1
HOST: 192.168.1.101:8081
Content-type: application/json

{
    "uri": "mixer://mixer1",
    "mixerLayoutDir": ""
}
```

In this case, a mixer layout class set in `mixerLayoutClass`parameter will be applied.

Layout description file format

Layout description XML file mustconform tho the following XSD scheme

Mixer layout XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="body" type="bodyType"/>
  <xs:complexType name="videoType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="value" type="xs:string"/>
        <xs:attribute type="xs:int" name="x" default="0"/>
        <xs:attribute type="xs:int" name="y" default="0"/>
        <xs:attribute type="xs:string" name="width" default="0"/>
        <xs:attribute type="xs:string" name="height" default="0"/>
        <xs:attribute type="xs:string" name="align" default="LEFT"/>
        <xs:attribute type="xs:string" name="watermark"/>
        <xs:attribute type="xs:int" name="padding-left" default="0"/>
        <xs:attribute type="xs:int" name="padding-right" default="0"/>
        <xs:attribute type="xs:int" name="padding-top" default="0"/>
        <xs:attribute type="xs:int" name="padding-bottom" default="0"/>
        <xs:attribute type="xs:boolean" name="crop" default="false"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="divType">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element name="video" type="videoType"/>
        <xs:element name="div" type="divType"/>
        <xs:element name="row" type="rowType"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute type="xs:int" name="x" default="0"/>
    <xs:attribute type="xs:int" name="y" default="0"/>
    <xs:attribute type="xs:string" name="width" default="0"/>
    <xs:attribute type="xs:string" name="height" default="0"/>
    <xs:attribute type="xs:string" name="align" default="LEFT"/>
    <xs:attribute type="xs:int" name="padding-left" default="0"/>
    <xs:attribute type="xs:int" name="padding-right" default="0"/>
    <xs:attribute type="xs:int" name="padding-top" default="0"/>
    <xs:attribute type="xs:int" name="padding-bottom" default="0"/>
  </xs:complexType>
  <xs:complexType name="rowType">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element name="video" type="videoType"/>
        <xs:element name="div" type="divType"/>
        <xs:element name="row" type="rowType"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute type="xs:int" name="x" default="0"/>
    <xs:attribute type="xs:int" name="y" default="0"/>
    <xs:attribute type="xs:string" name="width" default="0"/>
    <xs:attribute type="xs:string" name="height" default="0"/>
    <xs:attribute type="xs:string" name="align" default="LEFT"/>
    <xs:attribute type="xs:int" name="padding-left" default="0"/>
    <xs:attribute type="xs:int" name="padding-right" default="0"/>
    <xs:attribute type="xs:int" name="padding-top" default="0"/>
    <xs:attribute type="xs:int" name="padding-bottom" default="0"/>
  </xs:complexType>
  <xs:complexType name="bodyType">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element name="video" type="videoType"/>
        <xs:element name="div" type="divType"/>
        <xs:element name="row" type="rowType"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute type="xs:string" name="watermark"/>
  </xs:complexType>
</xs:schema>
```

All the layout descriptors are validated by this scheme, if validation fails for some descriptor, standard layout will be used for its participants count.

The following elements are supported:

- body - a root container to place all the pictures descriptions, repersenting mixer canvas. May be used once per file
- div - container to place one picture, a set of pictures or another container. May be used in any quantity per file, can also be nested
- video - a picture description, representing a stream in mixer. May be used in any quantity per file.

body element

The body element is a root container for all the children elements. The following attributes are supported:

- watermark - a picture file name to applt=y as watermark to whole mixer output stream (since build 5.2.1051)

div element

The div element is a container for other div or video elements. The following attributes are supported:

- x, y - left top corner coordinates on mixer canvas in pixels
- width - width on mixer canvas in pixels or parent element width percents
- height - height on mixer canvas in pixels or parent element height percents
- padding-left,padding-right,padding-top,padding-bottom - padding from corresponding side in pixels
- align - align on mixer canvas or in parent container

align attribute supports the following possible values

- LEFT - element is left aligned
- RIGHT -element is right aligned
- TOP -element is top aligned
- BOTTOM -element is bottom aligned
- CENTER - element is centered
- BOTTOM_CENTER - element is centered and bottom aligned
- TOP_CENTER -element is centered and top aligned
- INLINE_HORIZONTAL - element is adjusted to right top corner of closest left element
- INLINE_HORIZONTAL_CENTER - the same as above, with vertical centering
- INLINE_VERTICAL -element is adjusted to left bottom corner of closest element above this one
- INLINE_VERTICAL_CENTER -the same as above, with horizontal centering

If left top corner coordinates are set explicitly, align attribute is not applied

row element

Since build 5.2.1052 row element is supported to represent a row of one or more stream pictures. Supports the same attribuites as div

video element

Represents stream picture in parent container. The same attributes are supported as div element, with additional attribute

- crop - crop a picture around the center
- id - position identifier to place a stream (since build 5.2.1950)

cropattribute may be true или false (by default).

id attribute accepts any value which must be unique within the XML picture layout descriptor file

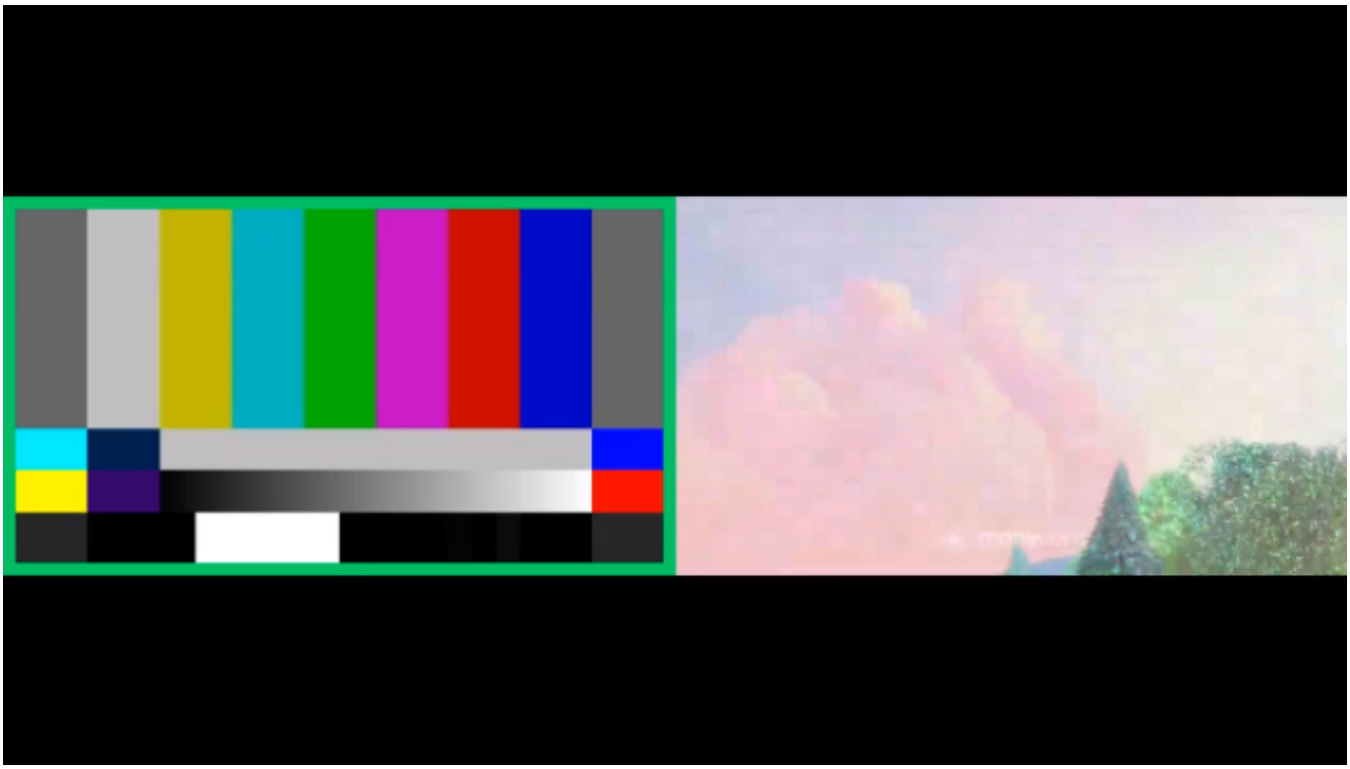
Stream picture width

Since build 5.2.1052, a stream picture width can be set in pixels, parent element width percents, or in columns. This is supported for row element children only, for example

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <row height="80%" align="CENTER">
    <video width="1col" align="INLINE_HORIZONTAL_CENTER"/>
    <video width="1col" align="INLINE_HORIZONTAL_CENTER"/>
  </row>
</body>
```

In this case, when parsing a custom layout descriptor, one column size is calculated as row width divided by cheild elements count of one level. Then all the child elements widths are set as columns count configured in width arrtibute.

The mixer output stream example for the descriptor above



It is not allowed to mix percents with columns in one level items!

Since build 5.2.1094, width may be set in columns for div child elements also.

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <div height="80%" align="CENTER">
    <video width="1col" align="INLINE_HORIZONTAL_CENTER"/>
    <video width="1col" align="INLINE_HORIZONTAL_CENTER"/>
  </row>
</body>
```

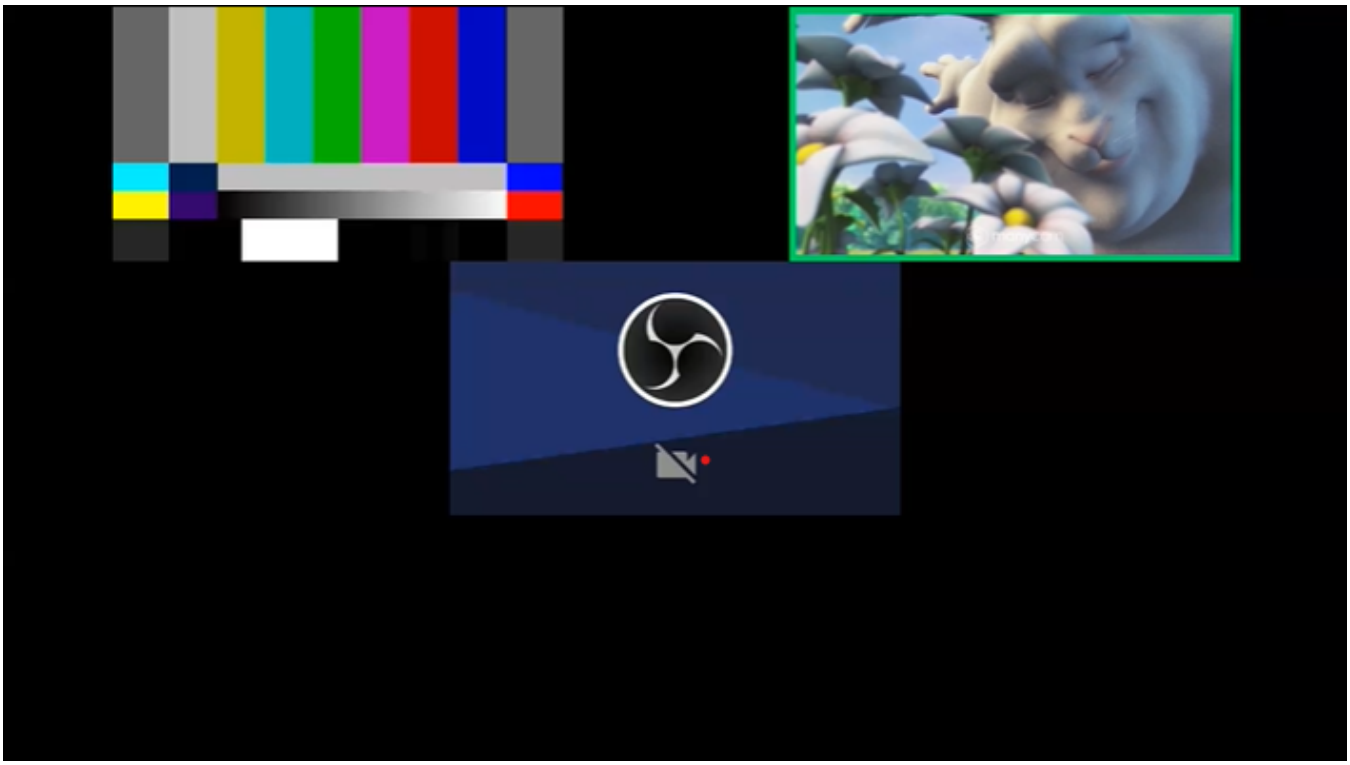
Stream picture height

Since build 5.2.1094, stream picture height can be set in pixels, parent element height percents, or in rows

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <row height="1row" align="INLINE_VERTICAL_CENTER">
    <div width="1col" height="100%" align="INLINE_HORIZONTAL_CENTER"><video width="100%" height="100%" align="
CENTER"/></div>
    <div width="1col" height="100%" align="INLINE_HORIZONTAL_CENTER"><video width="100%" height="100%" align="
CENTER"/></div>
  </row>
  <row height="1row" align="INLINE_VERTICAL_CENTER">
    <div width="1col" height="100%" align="INLINE_HORIZONTAL_CENTER"/>
    <div width="1col" height="100%" align="INLINE_HORIZONTAL_CENTER"><video width="100%" height="100%" align="
CENTER"/></div>
    <div width="1col" height="100%" align="INLINE_HORIZONTAL_CENTER"/>
  </row>
  <row height="1row" align="INLINE_VERTICAL_CENTER" />
</body>
```

In this case, when parsing a custom layout descriptor, one row size is calculated as element height divided by child elements count of one level. Then all the child elements heights are set as rows count configured in height arrtibute.

The mixer output stream example for the descriptor above



It is not allowed to mix percents with rows in one level items!

Stream name template definition

A stream name template can be set in video element. For example

```
<video>test</video>
```

will display only stream named 'test', and will not be applied to any stream with other name.

The template can be set as regular expression, for example

```
<video>test1.*</video>
```

In this case this element can display streams named test1, test1#room1, test11 etc

If the stream name does not meet any template in the file, and there is empty video element in the file, this element will be used for that stream. For example, here is a description for one participant

```
<?xml version="1.0" encoding="utf-8"?>
<body xsi:noNamespaceSchemaLocation="schema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="TOP_CENTER">
    <video>test1.*</video>
  </div>
  <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="BOTTOM_CENTER">
    <video>test2.*</video>
  </div>
  <div width="160" height="90" padding-left="0" padding-right="0" padding-bottom="0" align="RIGHT">
    <video>.*</video>
  </div>
</body>
```

In this case test3 will be displayed in last video element

Placing the stream picture to a certain position by identifier

Since build 5.2.1950 it is possible to place the stream picture to a certain position by identifier. This may be suitable if stream name template cannot be used. Foe instance there is the layout descriptor file for three participants

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <row height="100%" align="CENTER">
    <video width="100%" id="desktop" align="CENTER"></video>
    <row height="20%" align="BOTTOM">
      <div width="1col" height="100%" align="INLINE_HORIZONTAL"><video width="95%" height="95%" id="speaker"
align="RIGHT"/></div>
      <div width="1col" height="100%" align="INLINE_HORIZONTAL"><video width="95%" height="95%" id="
participant" align="LEFT"/></div>
    </row>
  </row>
</body>
```

In this case the stream can be placed to the position with `desktop` identifier using REST API query `/mixer/set_position` :

```
POST /rest-api/mixer/set_position HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "uri": "mixer://mixer1",
    "remoteStreamName": "stream3",
    "videoPositionId": "desktop"
}
```

The stream also can be added to the certain position:

```
POST /rest-api/mixer/add HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "uri": "mixer://mixer1",
    "remoteStreamName": "stream1",
    "videoPositionId": "speaker"
}
```

Whole mixer or participant stream watermarking

Since build 5.2.1051 it is possible to set PNG picture in custom layout descriptor, which will be applied as watermark to whole the mixer output stream or to a certain participant stream in mixer.

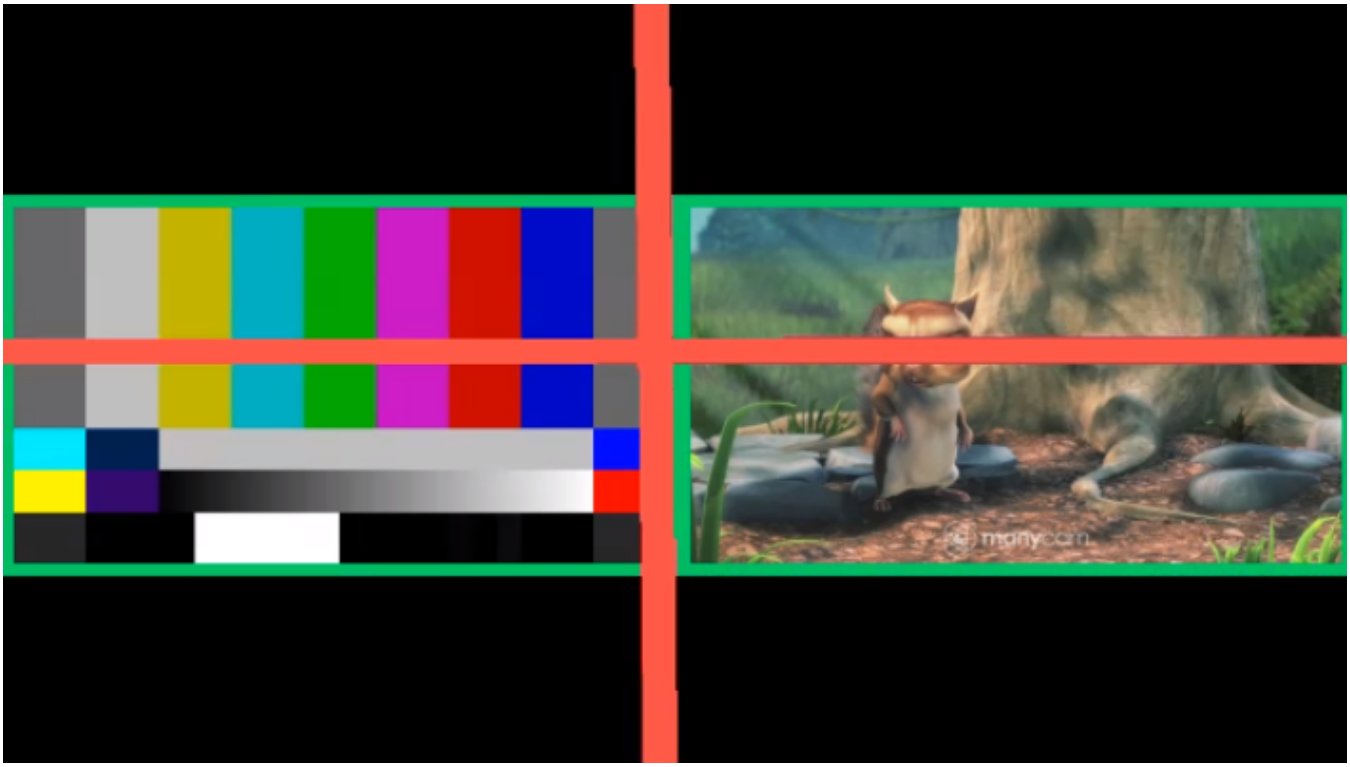
To apply watermark to the whole mixer output stream, add watermark attribute with relative PNG file name to body tag:

```
<body watermark="image.png">
  <row height="80%" align="CENTER">
    <video width="50%" align="INLINE_HORIZONTAL_CENTER"/>
    <video width="50%" align="INLINE_HORIZONTAL_CENTER"/>
  </row>
</body>
```

PNG file should be placed to the layout folder:

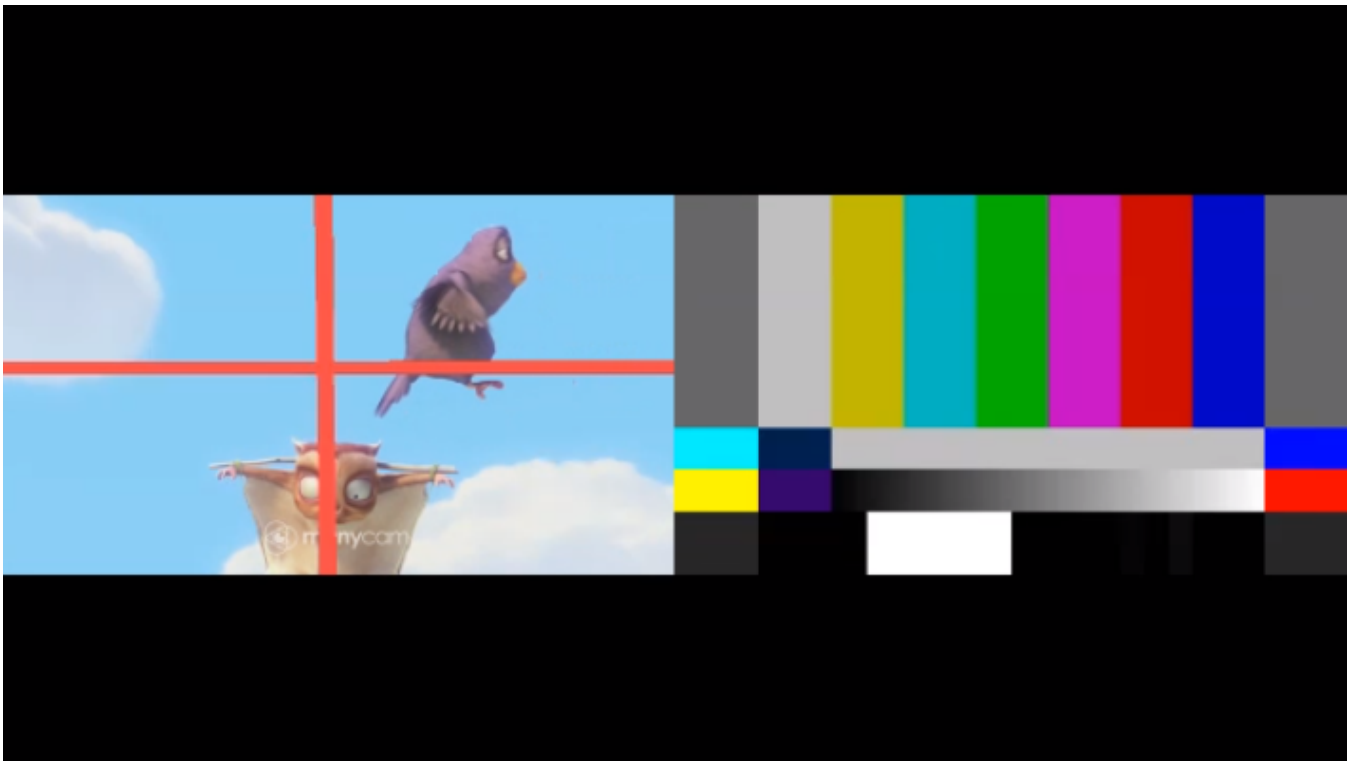
```
[root@demo ~]# ls -l /opt/GridLayout
total 20
-rw-r--r-- 1 root root  106 Sep 30 13:10 1_participants.mix
-rw-r--r-- 1 root root  204 Oct 11 12:11 2_participants.mix
-rw-r--r-- 1 root root  409 Oct  8 13:18 3_participants.mix
-rw-r--r-- 1 root root  591 Oct  8 13:00 4_participants.mix
-rw-r--r-- 1 root root 3434 Oct  7 08:57 image.png
```

The example of mixer output stream with watermark applied



Watermark can be applied to a certain participant stream in mixer. To do this, add watermark attribute withPNG file name to video tag:

```
<body>
  <row height="80%" align="CENTER">
    <video watermark="image.png" width="50%" align="INLINE_HORIZONTAL_CENTER"/>
    <video width="50%" align="INLINE_HORIZONTAL_CENTER"/>
  </row>
</body>
```



Screen sharing description for a certain number of participants

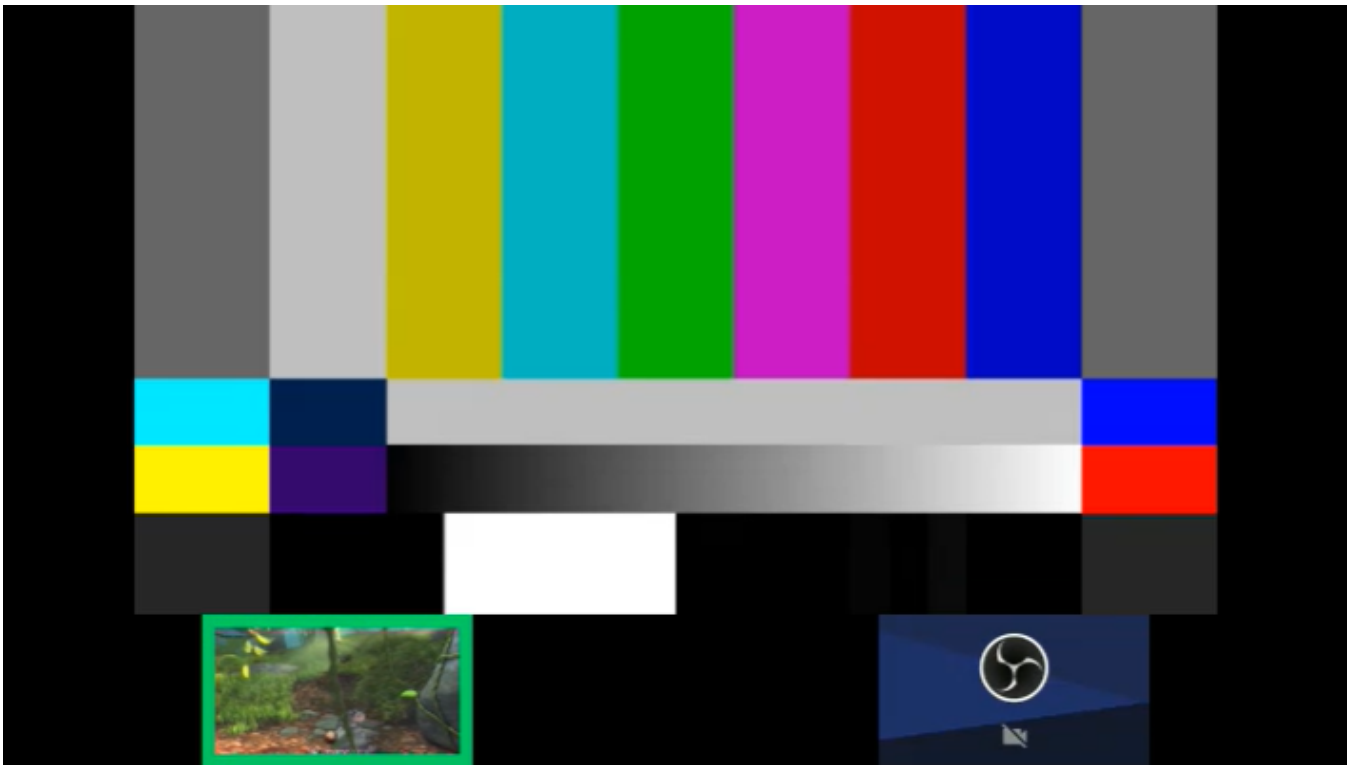
Since build 5.2.1091 it is possible to describe a custom desktop (screen sharing) layout for a certain number of participants. Desktop layout description files should have `.desktopmix` extension and should be placed in the same folder as usual descriptors:

```
1_participants.desktopmix
2_participants.desktopmix
3_participants.desktopmix
1_participants.mix
2_participants.mix
3_participants.mix
```

Desktop stream is detected by stream name template. This is the example descriptor for 2 participant streams and desktop stream

3_participants.desktopmix

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <row height="80%" align="TOP">
    <video width="100%" align="CENTER">.*_desktop.*</video>
  </row>
  <row height="20%" align="BOTTOM">
    <div width="1col" height="100%" align="INLINE_HORIZONTAL_CENTER"><video width="90%" align="CENTER"/></div>
    <div width="1col" height="100%" align="INLINE_HORIZONTAL_CENTER"><video width="90%" align="CENTER"/></div>
  </row>
</body>
```



Note that desktop stream is the same stream as participants one at mixer point, so 2 participants `video` elements and 1 desktop `video` element are defined in 3 participants descriptor file in this case.

If there is no `*.desktopmix` file foa a certain number of participants, and desktop stream is added to mixer, a standard desktop layout will be used.

Configuration errors handling

1. If the mixer layout contains no description file for a certain participants number, the layout class set in the following parameter will be applied

```
mixer_layout_class=com.flashphoner.media.mixer.video.presentation.GridLayout
```

By default, GridLayout will be used

2. If stream name does not meet any template in description file for a current number of participants, audio and video from this stream will not be added to mixer output stream

Custom layout examples

Placing stream pictures according to stream names

Lets' create a custom mixer layout for 640x360 mixer stream up to three participants. Note that all the picture sizes are set explicitly, and should not exceed mixer canvas size.

One participant description:

1_test.mix

```
<?xml version="1.0" encoding="utf-8"?>
<body xsi:noNamespaceSchemaLocation="schema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="TOP_CENTER">
    <video crop="false">test1.*</video>
  </div>
  <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="BOTTOM_CENTER">
    <video crop="false">test2.*</video>
  </div>
</body>
```

Two participans description

2_test.mix

```
<?xml version="1.0" encoding="utf-8"?>
<body xsi:noNamespaceSchemaLocation="schema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="TOP_CENTER">
    <video crop="false">test1.*</video>
  </div>
  <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="BOTTOM_CENTER">
    <video crop="false">test2.*</video>
  </div>
</body>
```

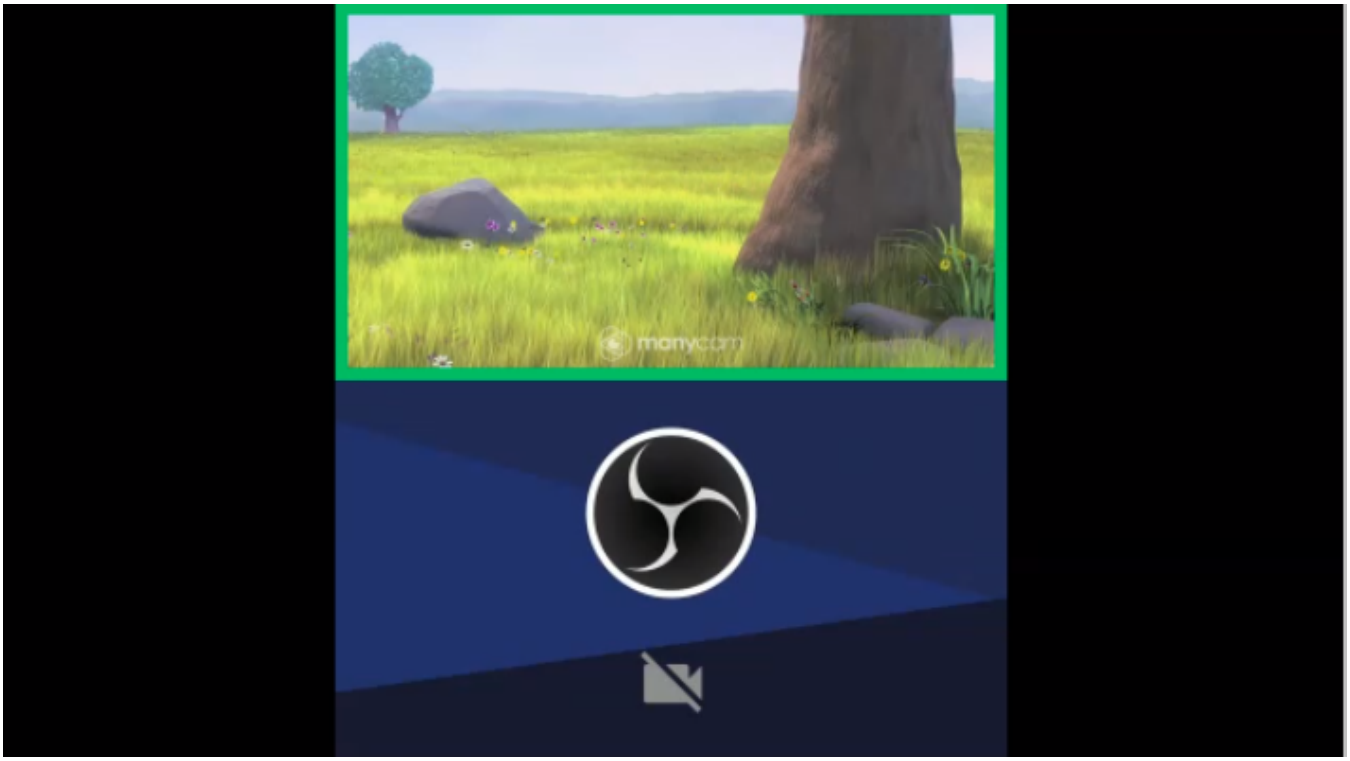
Three participants description

3_test.mix

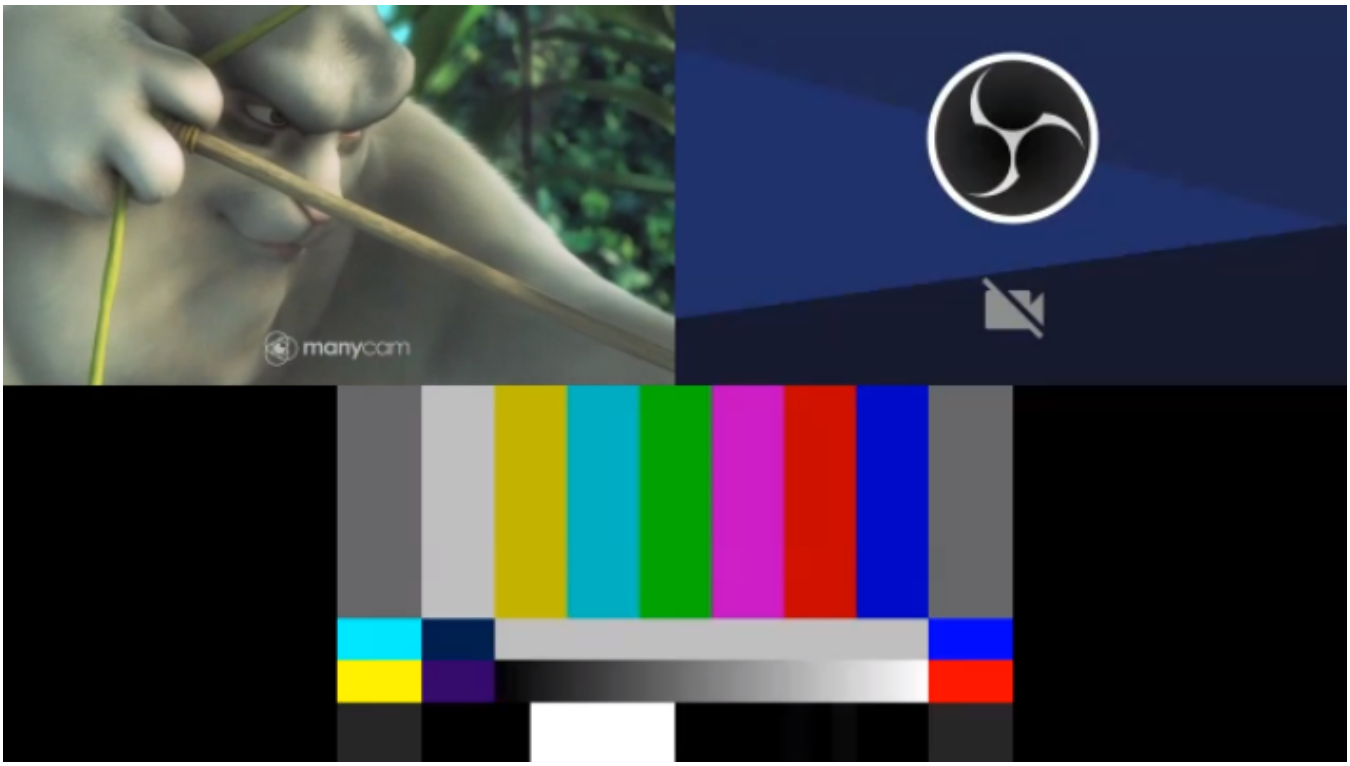
```
<?xml version="1.0" encoding="utf-8"?>
<body xsi:noNamespaceSchemaLocation="schema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="LEFT">
    <video crop="false">test1.*</video>
  </div>
  <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="RIGHT">
    <video crop="false">test2.*</video>
  </div>
  <div width="320" height="180" padding-left="0" padding-right="0" padding-bottom="0" align="BOTTOM_CENTER">
    <video crop="false">test3.*</video>
  </div>
</body>
```

The mixer with test1 stream output example

The mixer with test1 and test2 streams output example



The mixer with test1, test2, test3 streams output example



Placing stream pictures randomly without strict size limiting

One participant description:

1_participants.mix

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <row height="80%" align="CENTER">
    <video width="100%" align="CENTER"/>
  </row>
</body>
```

Two participants description:

2_participants.mix

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <row height="80%" align="CENTER">
    <video width="50%" align="INLINE_HORIZONTAL_CENTER"/>
    <video width="50%" align="INLINE_HORIZONTAL_CENTER"/>
  </row>
</body>
```

Three participants description:

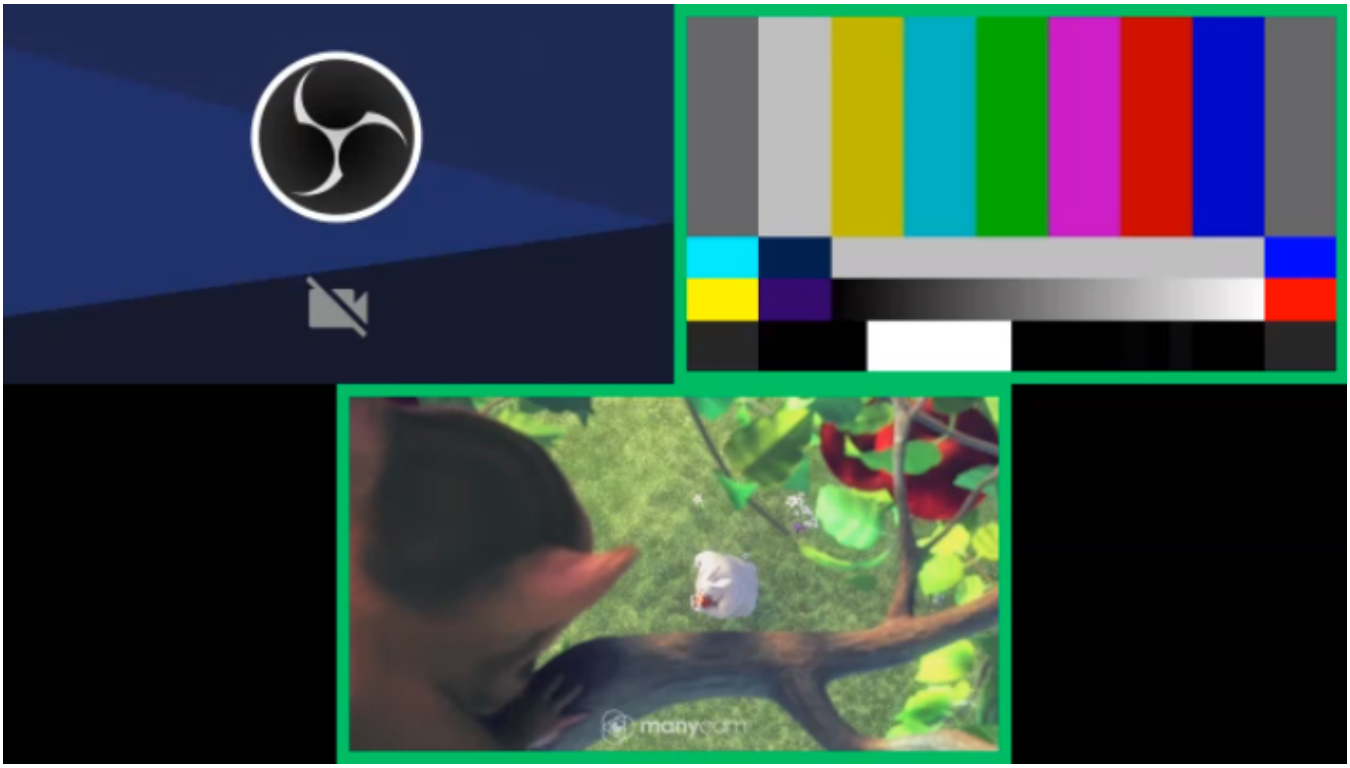
3_participants.mix

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <row height="50%" align="INLINE_VERTICAL_CENTER">
    <div width="50%" height="100%" align="INLINE_HORIZONTAL_CENTER"><video width="100%" align="CENTER"/></div>
    <div width="50%" height="100%" align="INLINE_HORIZONTAL_CENTER"><video width="100%" align="CENTER"/></div>
  </row>
  <row height="50%" align="INLINE_VERTICAL_CENTER">
    <video width="100%" align="CENTER"/>
  </row>
</body>
```

One participant displaying

Two participants displaying

Three participant displaying



Custom layout displaying tool

Since build 5.2.1035 custom mixer layout displaying tool is added. The tool should be launched from command line

```
cd /usr/local/FlashphonerWebCallServer/tools
bash ./mixer_layout_tool.sh /path/to/mixer_layout -o=/path/to/output
```

The tool writes mixer layout pictures to the folder defined, one PNG picture per one layout descriptor file

The following parameters are supported:

- /path/to/mixer_layout - path to layout folder, mandatory
- -o=/path/to/output - path to pictures output folder
- -n=1 - participants count to display a picture; if this is not set, pictures for all participants count will be written
- -N=test1,test2,test3 - stream names list to use for pictures rendering; if streams count in list is lower than participants count, automatically generated names will be used: stream0, stream1 etc
- -p=test - perfix to generate stream names; in this case, stream names wuill be generated subsequently, starting from 0, for example test0, test1, test2 etc
- -a - draw speech indicator frame around all the pictures
- -s=640:360 - mock participant picture size, equal to mixer picture size by default
- -h - shown the tool command line parameters list

The tool uses current mixer settings from flashphoner.properties file

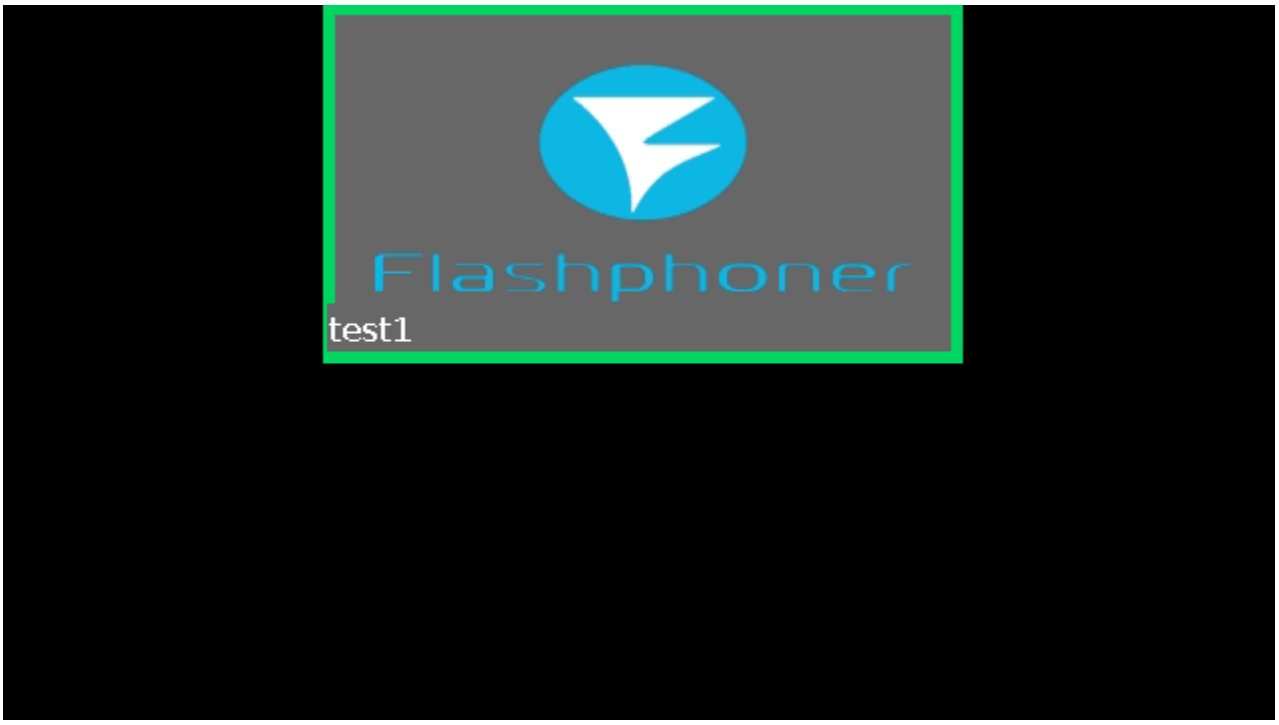
Displaying examples

Look at pictures displaying examples for the custom layout described above

1. The picture for one participant with speech indicator displaying, stream names are set explicitly

```
bash ./mixer_layout_tool.sh /opt/mixer_layout -o=/tmp -N=test1,test2,test3 -n=1 -a
```

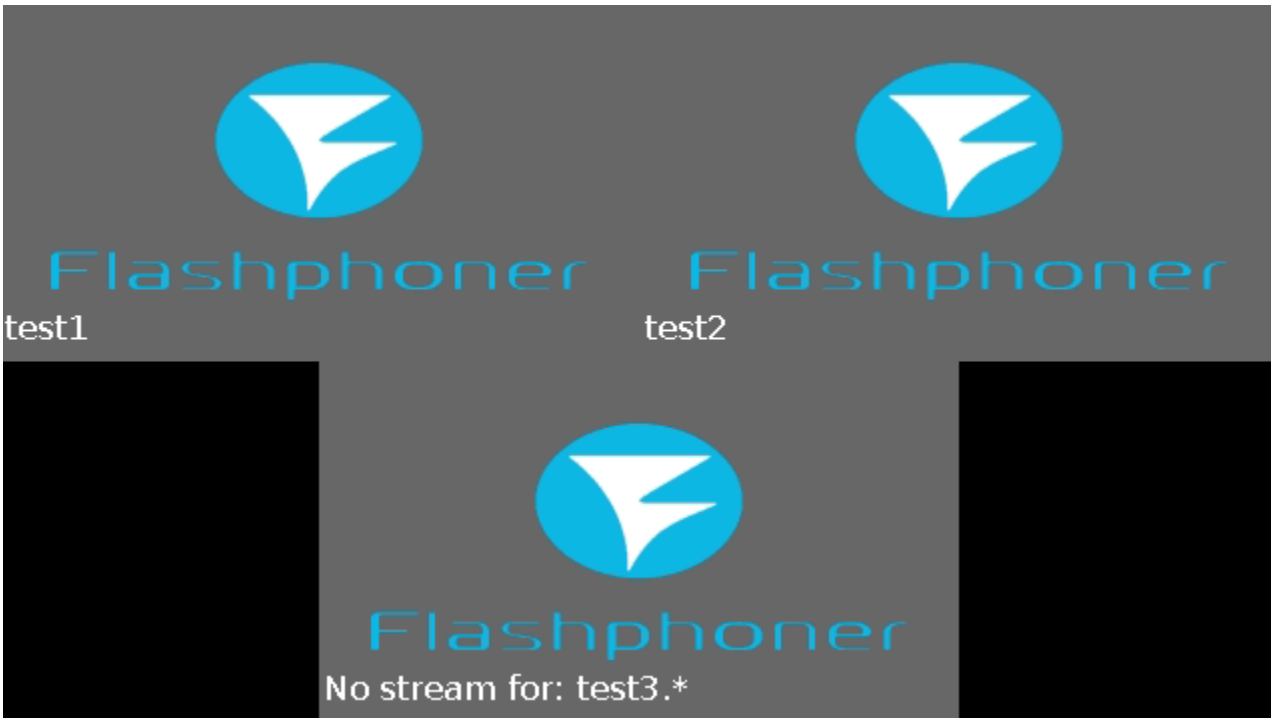
Output file /tmp/1_test.png



2. The picture for three participants, stream names are formed automatically with the prefix set

```
bash ./mixer_layout_tool.sh /opt/mixer_layouts -o=/tmp -p=test -n=3
```

Output file /tmp/3_test.png



Stream names templates in this layout descriptors are set explicitly as `test1.*`, `test2.*`, `test3.*`. When stream names are generated automatically, streams count starts from 0, therefore the following names were generated for this participants count: `test0`, `test1`, `test2`. In this case, for missed stream `test3` name is displayed as `No stream for: test3.*`

Error handling

1.If one of the pictures size in custom layout descriptor exceeds mixer canvas size, the tool will display the following error message, and PNG file will not be generated:

```
13:54:49,232 INFO  toryLayoutController - Mixer got 2 frames. Using 2_test.mix descriptor
Computed layout would produce exception: java.lang.RuntimeException: Computed layout element: Layout{point=java.
awt.Point[x=-106,y=0], dimension=java.awt.Dimension[width=852,height=478], frame:true} out of bounds
Please check configuration for this set of participants: [test1, test2]
```

If a mixer is created with such custom layout, the mixer will be closed with the same error message.

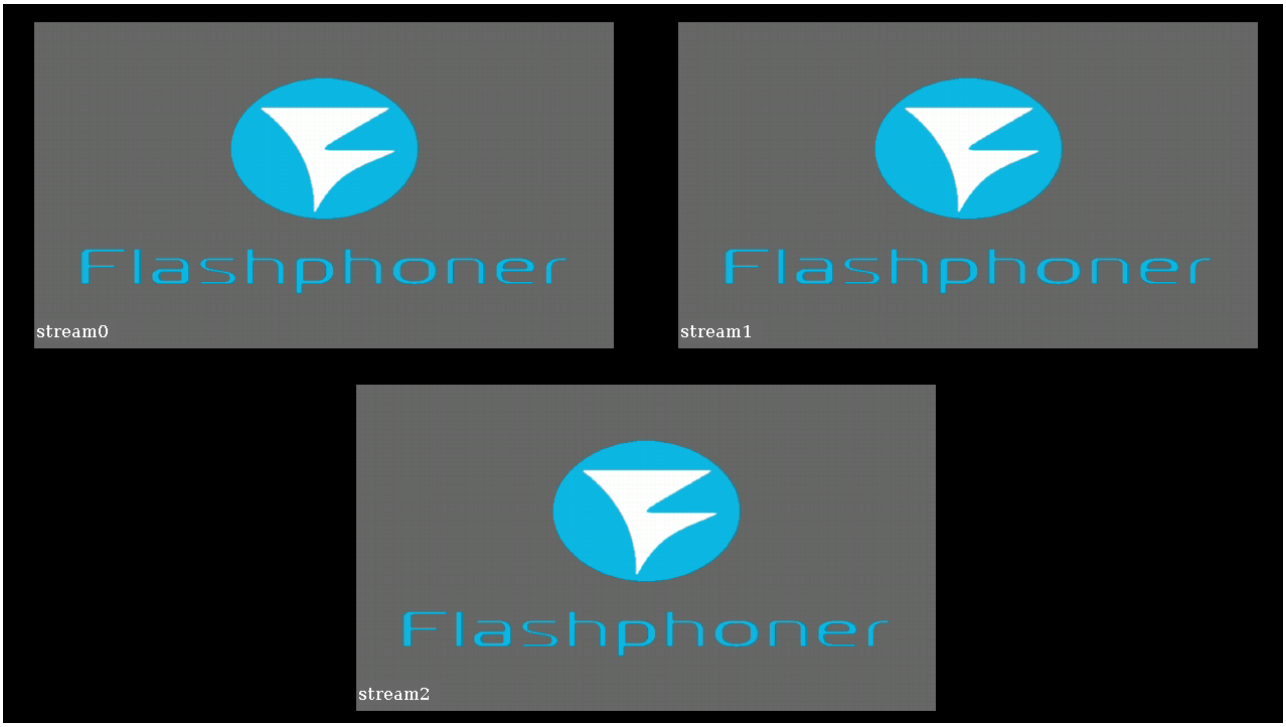
2. If there are no `*.mix` and `*.desktopmix` descriptors for a certain number of participants, the tool will generate a picture file with `_fallback` suffix using a standard mixer layout from WCS settings.

Standard layouts implementation in XML markup language

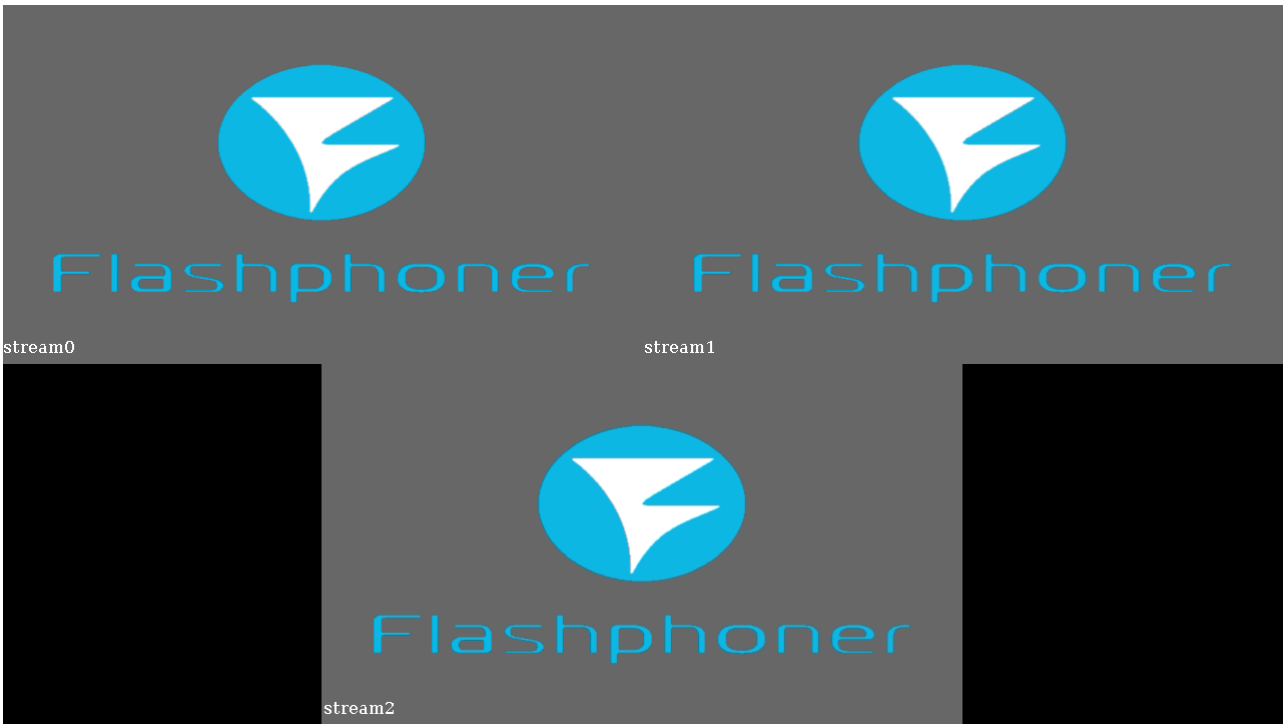
The following archive file `Layouts.tar.gz` contains examples of standard mixer layouts implementation compatible with WCS builds since 5.2.1094 and newer

The archive contains the following folders:

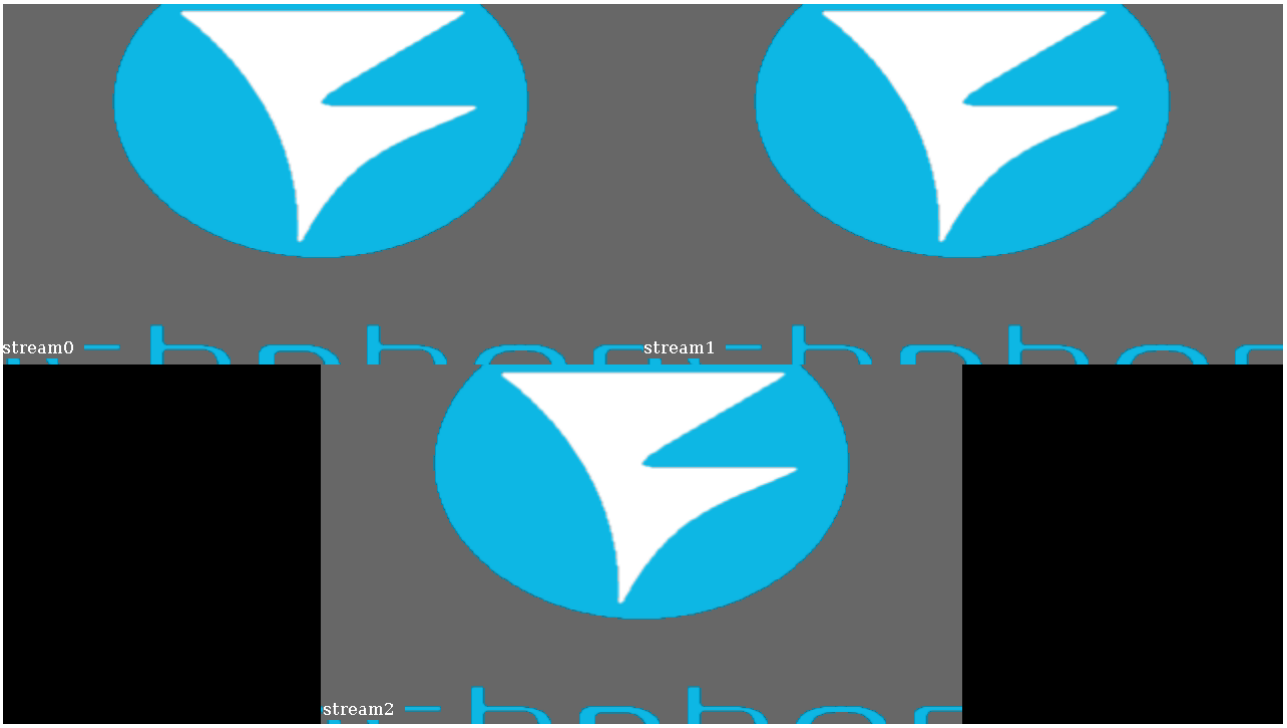
- GridLayout - grid layout implementation with paddings



- CenterNoPaddingGridLayout - grid layout implementation with no paddings between picture frames



- CropNoPaddingGridLayout - grid layout implementation with no paddings between picture frames and images crop



- samples - mixer output picture samples folder

Usage

1. Unpack the archive to /opt folder

```
cd /opt
tar -xzf ~/Layouts.tar.gz
```

2. Set default layout in flashphoner.properties file as needed

```
mixer_layout_dir=/opt/GridLayout
```

or set the layout while creating a mixer by REST API

```
POST /rest-api/mixer/startup HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "uri": "mixer://mixer1",
    "localStreamName": "mixer1_stream",
    "hasVideo": true,
    "hasAudio": true,
    "mixerLayoutDir": "/opt/CenterNoPaddingGridLayout"
}
```

"Picture-in-picture" custom layout implementation

To place one stream picture, for example, desktop translation, as a background for participants web cameras streams pictures, the `row` tag containing desktop stream should be parent for `row` tags with participants pictures, for example:

```
<?xml version="1.0" encoding="utf-8"?>
<body>
  <row height="100%" align="CENTER">
    <video width="100%" align="CENTER">.*_desktop.*</video>
    <row height="20%" align="BOTTOM">
      <div width="1col" height="100%" align="INLINE_HORIZONTAL"/>
      <div width="1col" height="100%" align="INLINE_HORIZONTAL"><video width="95%" height="95%" align="CENTER"
/></div>
      <div width="1col" height="100%" align="INLINE_HORIZONTAL"><video width="95%" height="95%" align="CENTER"
/></div>
      <div width="1col" height="100%" align="INLINE_HORIZONTAL"><video width="95%" height="95%" align="CENTER"
/></div>
      <div width="1col" height="100%" align="INLINE_HORIZONTAL"/>
    </row>
  </row>
</body>
```

The sample custom picture in picture layout up to 10 participants + 1 desktop is in the archive file `FullscreenDesktopLayout.tar.gz`

A desktop stream should have a name with `_desktop` suffix, for example `user1_desktop-room123456`