

Conditional Visibility

Table of Contents

- [1 Overview](#)
- [2 How They Work](#)
- [3 List of Boolean Conditions](#)
 - [3.1 Some pointers on boolean logic](#)
- [4 Changelog](#)
- [5 Conditions](#)
 - [5.1 Player conditions](#)
 - [5.2 Weather conditions](#)
 - [5.3 System conditions](#)
 - [5.4 Musicplayer conditions](#)
 - [5.5 Video OSD conditions](#)
 - [5.6 Playlist conditions](#)
 - [5.7 Musicpartymode conditions](#)
 - [5.8 Listitem conditions](#)
 - [5.9 Visualisation conditions](#)
 - [5.10 Skin conditions](#)
 - [5.11 Plugin conditions](#)
 - [5.12 Window conditions](#)
 - [5.13 Facadeview conditions](#)
 - [5.14 Control conditions](#)
 - [5.15 String conditions](#)
 - [5.16 Topbar conditions](#)
- [6 Examples](#)

Overview

All controls can benefit from “conditional visibility” support. This means that, instead of specifying just “yes” or “no” for the `<visible>` tag, you can now provide one of the many preset boolean conditions. Not only that, you can also specify how MediaPortal should transition between a visible state and a hidden state.

For example, if you add a `<visible>Player.HasMedia</visible>` tag to a control then this control will only be visible if MediaPortal is playing a media file. When its not playing anything, the control is hidden.

How They Work

The condition given in the `<visible>` tag is evaluated during the control's `Render()` function. MP decides whether or not the condition is true, and updates the control's visibility accordingly. Thus, it all happens without MP having to do the extra chores of remembering which controls need to be shown at which points in time. The controls automatically update themselves.

List of Boolean Conditions

You can combine two (or more) of the above settings by using '+' as an AND operator, '|' as an OR operator, '!' as a NOT operator, and '[' and ']' to bracket expressions. For example,

```
<visible>Player.HasVideo + Player.Rewinding8x</visible>
```

will only show the control when the player is rewinding a video at 8x, whereas

```
<visible>Player.HasVideo | Player.IsRecording</visible>
```

will show the control if a video is playing, or if we are recording something.

The AND operator takes precedence over the OR operator when evaluating the logic, and operators are read from left to right. So if you want to show something when

```
condition1 OR condition2 is true AND condition3 is true
```

you can do:

```
<visible>[condition1 | condition2] + condition3</visible>
```

Note that if you missed the brackets, then it would actually be reading

```
if condition1 or (condition2 and condition3)
```

due to AND taking precedence over OR.

Some pointers on boolean logic

The following holds true:

```
A + (B | C) = (A + B) | (A + C)
A | (B + C) = (A | B) + (A | C)

!(A + B) = (!A) | (!B)
!(A | B) = (!A) + (!B)
```

A common mistake is to do something like this:

```
(!A) | (!B) | (!C)
```

This is false only if A, B and C are all simultaneously true (as it's the same as $!(A + B + C)$), and you may have possibly been after

```
(!A) + (!B) + (!C)
```

which is false if A is true, or B is true, or C is true.

One thing you will notice, is that when a control is hidden, it can't be focused. This means you can't move to a control and have it automatically become visible (eg even if it had `Control.HasFocus(myID)`, it wouldn't come back on as MP wouldn't allow navigation to the hidden control). To solve this issue, you can use:

```
<visible allowhiddenfocus="true">Control.HasFocus(21)</visible>
```

on a control of `<id> 21`. This will allow the control to be focusable even when it's hidden. When the user moves to the hidden control, it will automatically unhide itself, due to the `Control.HasFocus(21)` visibility condition.

Changelog

Change	Date	Version
Auto Hide Pause OSD	2010/11/19	1.1.0 to 1.2.0
Visibility Condition System Time	2010/11/29	1.1.0 to 1.2.0
Visibility Condition System Date	2011/01/08	1.1.0 to 1.2.0
AudioVideoDelayPossible	2011/08/06	1.1.0 to 1.2.0
Topbar Visibility Conditions	2011/08/23	1.1.0 to 1.2.0

Conditions

The conditions listed in the following sections can be used in the `<visible>` tag. Note that the names of the conditions are not case sensitive, and you can code them in the mixture of upper and lower case that you prefer. All of the following styles are valid:

```
player.hasmedia
player.hasMedia
player.HasMedia
Player.HasMedia
```

Player conditions

```
player.hasmedia
player.hasaudio
player.hasvideo
player.playing
player.paused
player.rewinding
player.forwarding
player.rewinding2x
player.rewinding4x
player.rewinding8x
player.rewinding16x
player.rewinding32x
player.forwarding2x
player.forwarding4x
player.forwarding8x
player.forwarding16x
player.forwarding32x
player.canrecord           (not implemented)
player.recording           (not implemented)
player.displayafterseek   (not implemented)
player.seekbar            (not implemented)
player.seektime           (not implemented)
player.progress           (not implemented)
player.seeking            (not implemented)
player.showtime           (not implemented)
player.showcodec          (not implemented)
player.showinfo           (not implemented)
player.time               (not implemented)
player.timerremaining     (not implemented)
player.duration           (not implemented)
player.volume             (not implemented)
player.muted
player.hasduration
```

For details of the player conditions implemented, refer to the file "GUIInfoManager.cs" in the MediaPortal source.

Weather conditions

```
weather.conditions
weather.temperature
weather.location
weather.isfetched
```

System conditions

```
system.date
system.date.isbetween(mm.dd,mm.dd)
system.time
system.time.isbetween(hh:mm,hh:mm)
system.cputemperature
system.gputemperature
system.fanspeed
system.freespace(c) // d,e,f,g are also avail
system.buildversion
system.builddate
system.hasnetwork
system.fps
system.kalconnected
system.kaienabled
system.hasmediadvd
system.dvdready
system.trayopen
system.autodetection
system.freememory
system.screenmode
system.screenwidth
system.screenheight
system.currentwindow
system.currentcontrol
system.xboxnickname
system.dvdlabel
system.haslocks
```

Musicplayer conditions

None implemented

Video OSD conditions

```
#VideoOSD.AudioVideoDelayPossible
```

true when ffdshow is in graph otherwise to false e.g. <visible>string.equals(#VideoOSD.AudioVideoDelayPossible, false)</visible>

Playlist conditions

None implemented.

Musicpartymode conditions

```
musicpartymode.enabled
musicpartymode.songsplayed
musicpartymode.matchingsongs
musicpartymode.matchingsongspicked
musicpartymode.matchingsongsleft
musicpartymode.relaxedsongspicked
musicpartymode.randomsongspicked
```

Listitem conditions

None implemented.

Visualisation conditions

```
visualisation.locked
visualisation.preset
visualisation.name
visualisation.enabled
```

Skin conditions

```
skin.hassetting(setting)
skin.hastheme(theme)
```

Plugin conditions

```
plugin.isenabled(Plugin Name)
```

Window conditions

```
window.isosdvisible
window.ispauseosdvisible
window.isactive(window id)
window.istopmost(window id)
window.isvisible(window id)
window.previous(window id)
window.next(window id)
```

Facadeview conditions

```
facadeview.album
facadeview.filmstrip
facadeview.list
facadeview.largeicons
facadeview.playlist
facadeview.smallicons
```

Control conditions

```
control.hasthumb(controlId)
control.hasfocus(controlId)
control.hastext(controlId)
control.isvisible(controlId)
```

String conditions

```
string.equals(val1,val2)    will return true if val1 equals val2
string.equals(val1)         will return true if val1 has a length > 0
string.contains(val1,val2)  will return true if val1 contains val2
string.starts(val1,val2)    will return true if val1 starts with val2
```

Topbar conditions

```
topbar.focused    will return true if any topbar control (i.e. button) is focused
topbar.visible    will return true if topbar is visible on screen
```

Be sure to check the source code [GUIInfoManager](#) for updates.

Examples

1. Fade in an image when a player starts playing, and fade out the image when the player stops playing:

```
<control>
  <description>your image</description>
  <type>image</type>
  <id>1</id>
  <posX>0</posX>
  <posY>130</posY>
  <width>338</width>
  <height>375</height>
  <texture>image.png</texture>
  <visible>Player.HasMedia</visible>
  <animation effect="fade" time="250">visiblechange</animation>
</control>
```

For more information on the `<animation>` tag, see [Animations](#).

2. Fade in an image when the button with id 13 is focused, and fade out the image when the button is unfocused:

```
<control>
  <description>your image</description>
  <type>image</type>
  <id>1</id>
  <posX>0</posX>
  <posY>130</posY>
  <width>338</width>
  <height>375</height>
  <texture>image.png</texture>
  <visible>Control.HasFocus(13)</visible>
  <animation effect="fade" time="250">visiblechange</animation>
</control>
```

3. Fade in an image when a control with id 13 becomes visible, and fade out the image when the control becomes hidden:

```
<control>
  <description>your image</description>
  <type>image</type>
  <id>1</id>
  <posX>0</posX>
  <posY>130</posY>
  <width>338</width>
  <height>375</height>
  <texture>image.png</texture>
  <visible>Control.IsVisible(13)</visible>
  <animation effect="fade" time="250">visiblechange</animation>
</control>
```

4. Fade in an image when facadeview listview is selected, and fade out the image when another facadeview is chosen:

```
<control>
  <type>image</type>
  <id>1</id>
  <posX>1045</posX>
  <posY>150</posY>
  <width>280</width>
  <height>280</height>
  <texture>thumb.png</texture>
  <visible>facadeview.list</visible>
  <animation effect="fade" time="250">visiblechange</animation>
</control>
```