

# MP4IPSymcon

Was man damit machen kann:

Ähnlich wie mit anderen Oberflächen, kann man über IP-Symcon steuern und visualisieren. Das ganze läuft im MediaPortal und läßt somit wie gewohnt bedienen.

Der Aufbau der Oberfläche ist etwas anders als bei WebFront oder auch anders als bei gewöhnlichen MP-Plugins. Die Oberfläche wird aktuell nicht generisch erzeugt, sondern man nutzt den MediaPortal SkinEditor (oder schreibt XML-Files).

Der Vorteil: man kann sich ganz individuelle Screens zusammen stellen und ist unabhängig von der konkreten Hierarchie in IPS. Nachteil: man muss etwas Handarbeit investieren.

Die IPS Objekte werden dann in die "Pages" von MediaPortal über ein *Binding* eingebunden. Hierzu wird das Feld Description der MP-Controls "missbraucht". Steht bei einem Button dort z.B. "RUN:4711" so wird das IPS Script 4711 gestartet.

Bindings gibt es aktuell für Button, ToggleButtons, Images, Labels.

Die aktuelle Version 0.9.1 unterstützt FS20-Aktoren, xComfort-Switch und Dimmer.

## Installation

Das Plugin-Verzeichnis ins MediaPortal-Stammverzeichnis kopieren (normalerweise C:\Programme\Team MediaPortal\MediaPortal). Den Skin-Folder nach C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Team MediaPortal\MediaPortal\ kopieren.

Das Beispiel enthält nur ein Skin-File für Blue3 und enthält Bindings, die so natürlich nur mit meiner IPS-Konfiguration funktionieren. Hier muss also jeder mit dem Editor ran.

Weiterhin fehlt aktuell ein Einstellungsdialog. Daher muss der IPS-Host „von Hand“ in die MediaPortal-Konfiguration eingetragen werden. Das File mediaportal.xml befindet auch im Ordner C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Team MediaPortal\MediaPortal\ ganz am Ende (vor </profiles>) fügt man eine neue Section ein:

```
<section name="MP4IPSymcon">
  <entry name="host">IPSHOSTNAME</entry>
</section>
```

## Bindings

### Label

```
<description>VAR:InstId:Format</description>
```

Das Binding ist durch drei Teile definiert:

1. VAR: konstant, definiert ein Variablen-Binding
2. InstId: Instanz-ID der Variable (ersetzen durch die konkrete IPS ID)
3. Format: Format-String (optional). Siehe C# Referenz `String.format()` z.B. hier <http://alexonasp.net/samples/stringformatting/> . Wenn man Format wegläßt, dann wird die Variable „roh“ geliefert.

Beispiel:

```

<control>
  <description>VAR:4711:Temperatur {0:F1} °C</description>
  <type>label</type>
  <posX>186</posX>
  <posY>405</posY>
  <label>Dummytext wird ersetzt</label>
</control>

```

## Image

```

<description>VAR:InstId:Filename-Format</description>

```

Das Binding ist durch drei Teile definiert:

1. VAR: konstant, definiert ein Variablen-Binding
2. InstId: Instanz-ID der Variable (ersetzen durch die konkrete IPS ID)
3. Filename-Format: Format-Ausdruck (siehe Label), welches den Filename bildet.

Beispiel:

```

<control>
  <id>14</id>
  <description>VAR:14980:leuchte_{0}.png</description>
  <type>image</type>
  <posX>406</posX>
  <posY>212</posY>
  <width>71</width>
  <height>68</height>
  <texture>ips\leuchte_True.png</texture>
</control>

```

Je nach Status der Boolean Variable wird nun leuchte\_True.png oder Leuchte\_False.png angezeigt. Über mehrere Werte lassen sich auch mehrere unterschiedliche Zustände anzeigen z.B. Temperatur oder Helligkeit des Dimmers.

## Button

Klick-Button, der beim Anklicken, die im Binding definierte Aktion ausführt.

Binding:

```

<description>Cmd:Id:Action</description>

```

Das Binding ist durch drei Teile definiert:

1. Cmd: ist entweder RUN, GO oder BUT.
2. Id: Instanz-, Window- oder Script-ID der Variable (ersetzen durch die konkrete IPS ID)
3. Action: 0 (default) toggle, 1 off, 2 on, 3 dim down, 4 dim up.

RUN startet ein Script, GO wechselt in ein neues MP-Fenster (-1 geht zurück), BUT definiert einen Klick-Button.

Beispiel:

```

<control>
  <id>6</id>
  <description>RUN:15096</description>
  <type>button</type>
  <label>Alles aus</label>
</control>

```

## ***Togglebutton***

Siehe Button allerdings ist nur die Action 0 = toggle sinnvoll. Der Status des Togglebuttons wird automatisch aktualisiert (mit der Status-Variable der Instanz).