

## Arbeiten mit „Events“ in VoxCommando

Events sind Systemereignisse die VoxCommando in der „History-Liste“ im Hauptfenster anzeigt (Abb.1)

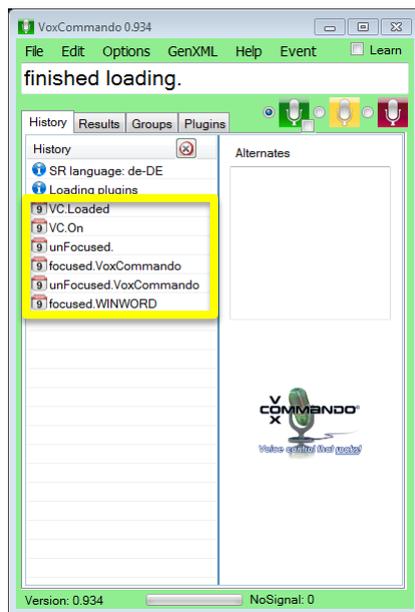


Abb.1

Diese Events können so genutzt werden, dass bei ihrem Erscheinen in VoxCommando eine bestimmte Aktion ausgeführt wird.

Als Beispiel wollen wir einen Befehl erzeugen, der nur die Gruppen aktiviert, die für das jeweils im Vordergrund sichtbare Programm gebraucht werden. So wird verhindert, dass mehrfach vergebene Befehle sich gegenseitig behindern (z.B. die Befehle: play, stop, pause usw. in XBMC, MediaMonkey, Mediaportal, iTunes und Windows Mediaplayer).

Als Beispiel nehmen wir XBMC:

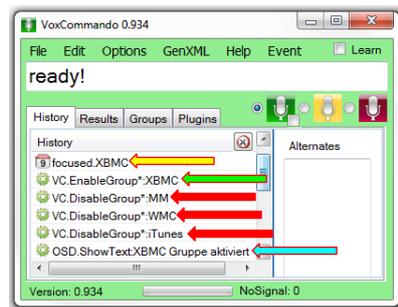


Abb.2

Wenn XBMC gestartet und im Vordergrund aktiv ist, wird dies im „History Fenster“ als „**focused.XBMC**“ Event angezeigt (Abb.2)

Unser Befehl **aktiviert** (falls nicht aktiv) die XBMC Gruppen und **deaktiviert** die Gruppen für MediaMonkey, Windows Mediaplayer sowie iTunes und zeigt den Text „**Gruppe aktiviert**“ in einem OSD-Fenster an.

**Schritt 1:** Wir rufen mit „Edit“ den Command Tree Editor auf und wählen das „Ordner-Icon“ aus um eine neue Befehlsgruppe zu erstellen.



Danach erscheint in der Baumstruktur eine neue Befehlsgruppe, die wir „Gruppe aktivieren“ nennen.



**Schritt 2:** Wir wählen jetzt das „Blitz-Icon“ aus, dieses erstellt einen neuen Befehl in der Gruppe und benennen diesen mit „Gruppen für XBMC aktivieren“ um.



**Schritt 3:** Nun verwandeln wir diesen Befehl in ein „Event-Befehl“, in dem wir auf das „Event-Icon“ klicken, „Event“ auswählen (Abb.3) und somit den Befehl für das Empfangen von „Events“ aktiviert haben (Abb.4).



Abb.4

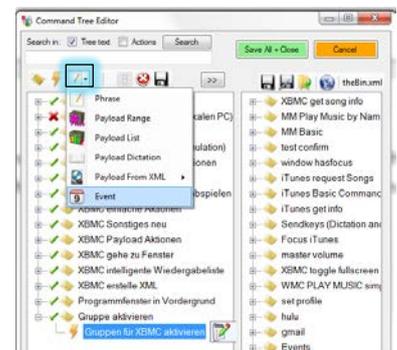


Abb.3

Der unter dem Befehl entstandene „Event“ (Abb.5) muss jetzt noch in den „Event“ umbenannt werden, den XBMC im „History-Fenster“ erzeugt, wenn das Programm im Vordergrund ist (Abb.6). In diesem Fall „focused.XBMC“



Abb.5



Abb.6

**Schritt 4:** Jetzt sind alle Vorarbeiten geleistet und wir können den Befehl editieren, in dem wir auf den „Event“ mit dem „Blitz-Symbol“ doppelt klicken und sich darauf der „Command-Builder“ öffnet (Abb.7).

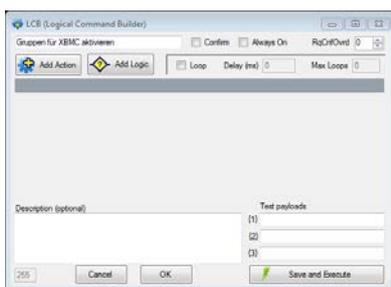


Abb.7

Im „Command-Builder“ klicken wir auf „Add Action“ und danach in der ersten Zeile auf den „Zauberstab“, so das sich der „Command Selection Tree“ öffnet (Abb.8) und wählen unter „VoxCommando“ -> „EnableGroup\*“ aus (der Stern steht für „Wildcard“), danach mit „Select“ bestätigen und als Parameter XBMC eintragen (XBMC ist dann die Wildcard, das heißt, alle Gruppen die XBMC enthalten werden aktiviert). Das Fenster sollte wie unten in Abb.9 aussehen.

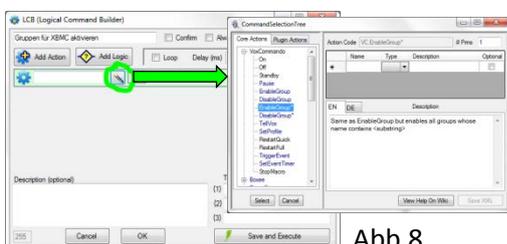


Abb.8



Abb.9

**Schritt 5:** Es wird der „Schritt 4“ mehrmals durchgeführt, nur das hier anstelle von „EnableGroup\*“ – „DisableGroup\*“ eingesetzt wird und als Wildcard-Parameter die Gruppen, die ihr deaktivieren möchtet wie in Abb.10 gezeigt, einträgt. Dies sind Beispiele für die VC-Standard-Gruppen, wenn ihr andere Gruppen-Namen verwendet „Mediaplayer“ usw. müsste ihr diese einsetzen. Der hier im Beispiel eingefügte OSD-Befehl schreibt in einem OSD-Fenster „XBMC Gruppen aktiviert“ welches in XBMC kurz sichtbar ist. Jetzt alles speichern und VC neu starten.

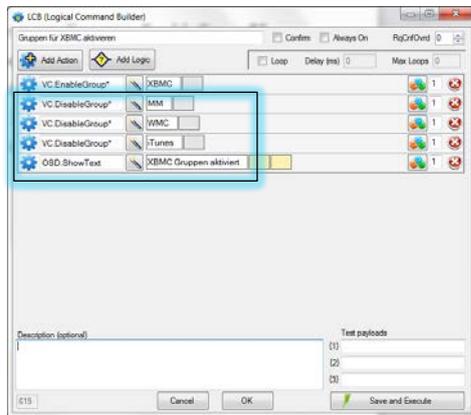


Abb.10

Dies zeigt euch einen von mehreren Wegen um Gruppen automatisch zu aktivieren bzw. deaktivieren, wenn Ihr nur eine Instanz von VoxCommando für mehrere Programme verwendet. Das ganze lässt sich auch mit einem Logik-Baustein (if-then-else) machen, probiert es einfach aus.

Kalle