



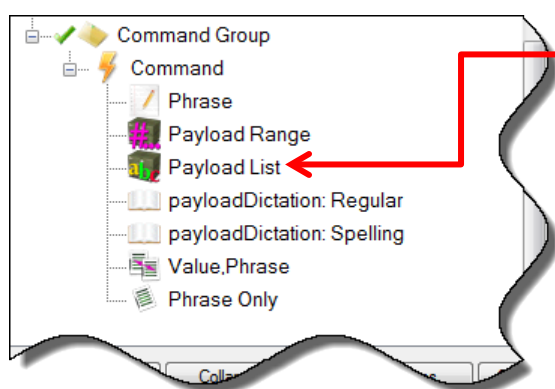
... mit Payloads in VoxCommando arbeitet - Teil 2

Payload List



Payloads sind wichtige Werkzeuge in Voxcommando. Wir können mit ihnen leicht komplexe Befehle erstellen. Ich möchte euch in diesem Tutorial mit Hilfe von Beispielen zeigen, wie Payloads in VoxCommando arbeiten und erstellt werden.

In diesem Tutorial lesen Sie oft das Wort "Phrase / Phrasen", Phrasen sind ein wichtiger Teil von VoxCommando, da diese bestimmen, welcher Befehl ausgeführt wird. Im Gegensatz zu einem Payload enthält die Phrase selbst keinen Wert (mit Ausnahme der Befehls id). Wenn Sie mehrere Phrasen durch Kommas getrennt (auch bekannt als Aliasnamen) einsetzen, ist es egal welche Sie sagen. Phrasen sind nur eine Zeichenfolge, die einen Befehl auslöst und enthalten keinen Wert.

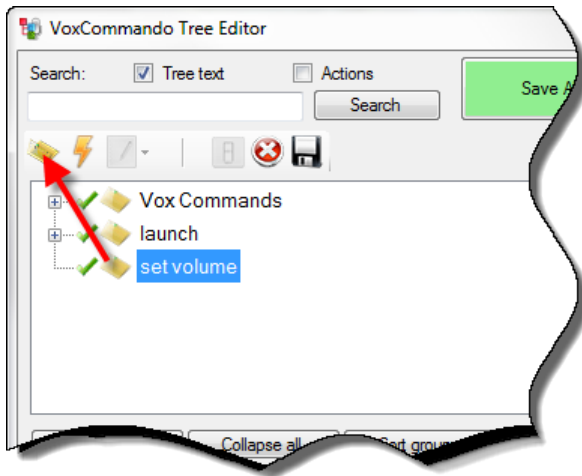


Im zweiten Teil geht es weiter mit der Payload List.

Im Bild links, sehen Sie die verschiedenen Icons für die Payloads

Payload List ist eine Liste von Zahlen, Wörter oder Buchstaben, die als Daten weiter gegeben werden. Die erkannte Zeichenfolge wird in den Platzhalter der Payload eingefügt.

Parameter: Eine Zeichenfolge die aus - Zahlen, Wörtern oder Buchstaben, getrennt durch ein Komma (hallo, 10, A, ...) besteht.

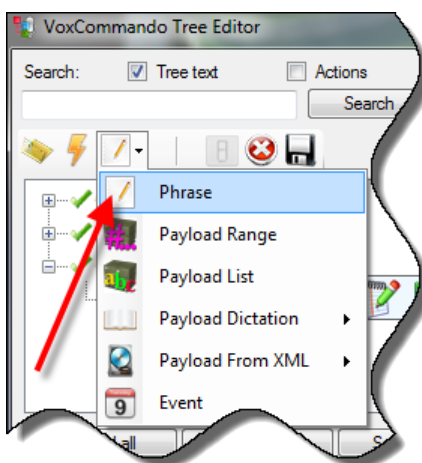
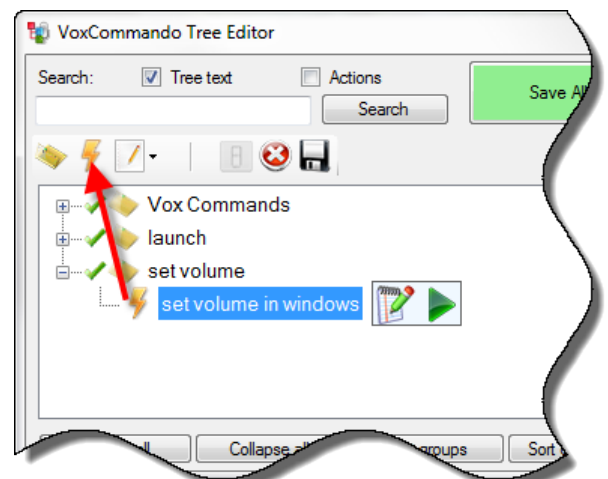


Wir beginnen hier mit einem einfachen Beispiel, wie Sie ihre Windows Systemlautstärke verändern, in dem Sie sagen “setze Lautstärke in Windows auf” gefolgt von “einem Wert zwischen 0 und 100 in 10er Schritten” (10, 20, 30, etc.).

Erstellen Sie im Editor eine neue Befehlsgruppe, in dem Sie auf das Ordnersymbol im Editor klicken und geben dieser einen Namen. (*set volume*).

In den folgenden Bildern sind die Einträge in Englisch zu sehen. Hier im Tutorial habe ich den einzugebenden Text in Deutsch geschrieben, welchen Sie auch nutzen sollten, es sei denn Sie bevorzugen die englische Spracheingabe.

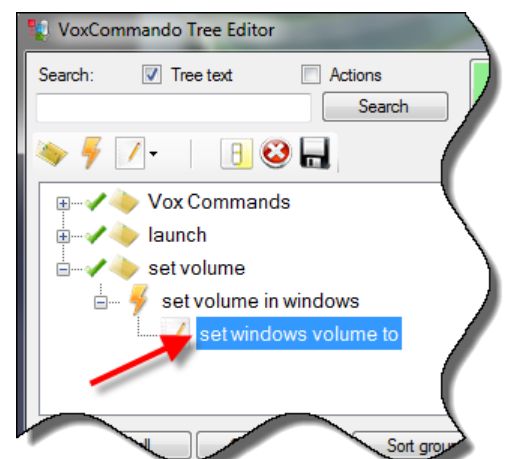
Im nächsten Schritt klicken Sie auf das Blitzsymbol um ein Befehl zu erzeugen und geben diesem einen Namen (*setze Lautstärke in Windows*).

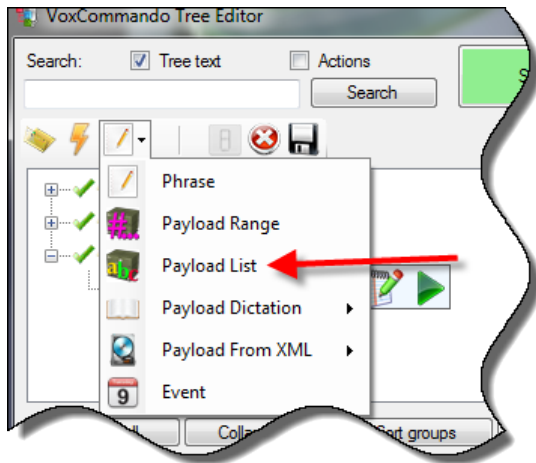


Nun erstellen wir eine Phrase die den Befehl ansteuert. Klicke auf das Bleistiftsymbol, wähle “Phrase” aus und schreibe die Wörter die den Befehl auslösen sollen “setze Lautstärke in Windows auf”.

Es ist wichtig, das Du nicht die gleichen Wörter verwendest, die schon in anderen aktiven Kommandogruppen existieren.

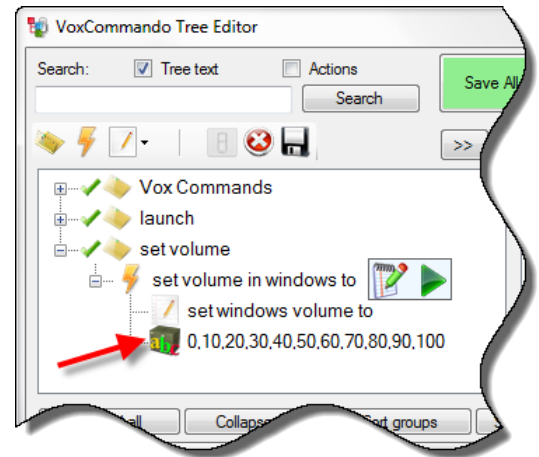
VoxCommando kann zwischen diesen **nicht** unterscheiden. Es ist also besser, wenn wir unser Kommando spezifizieren “setze Lautstärke in **Windows** auf”.



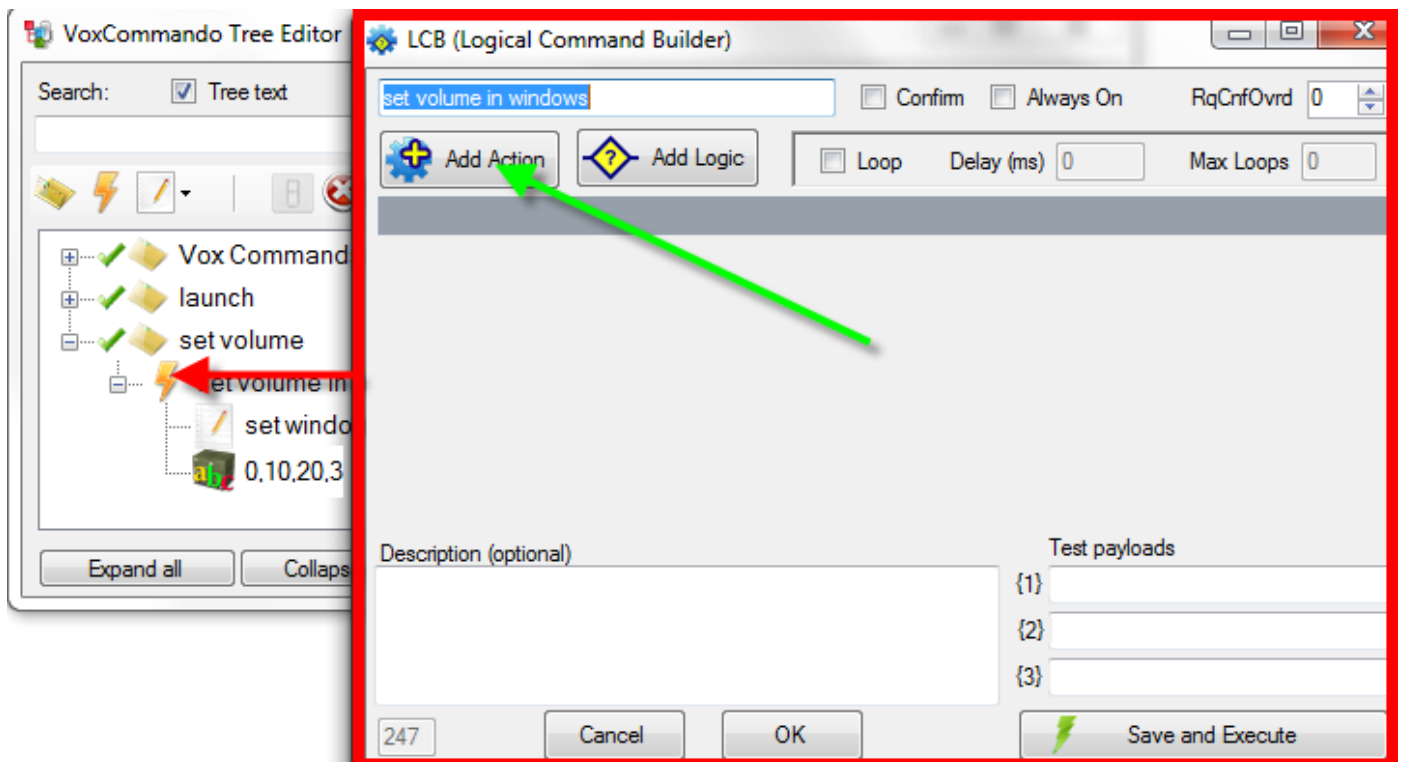


Als nächstes brauchen wir eine Payload List mit einer Zahlenfolge 0, 10, 20, 30, etc. bis 100. Klicke auf das Bleistiftsymbol und wähle "Payload List" aus.

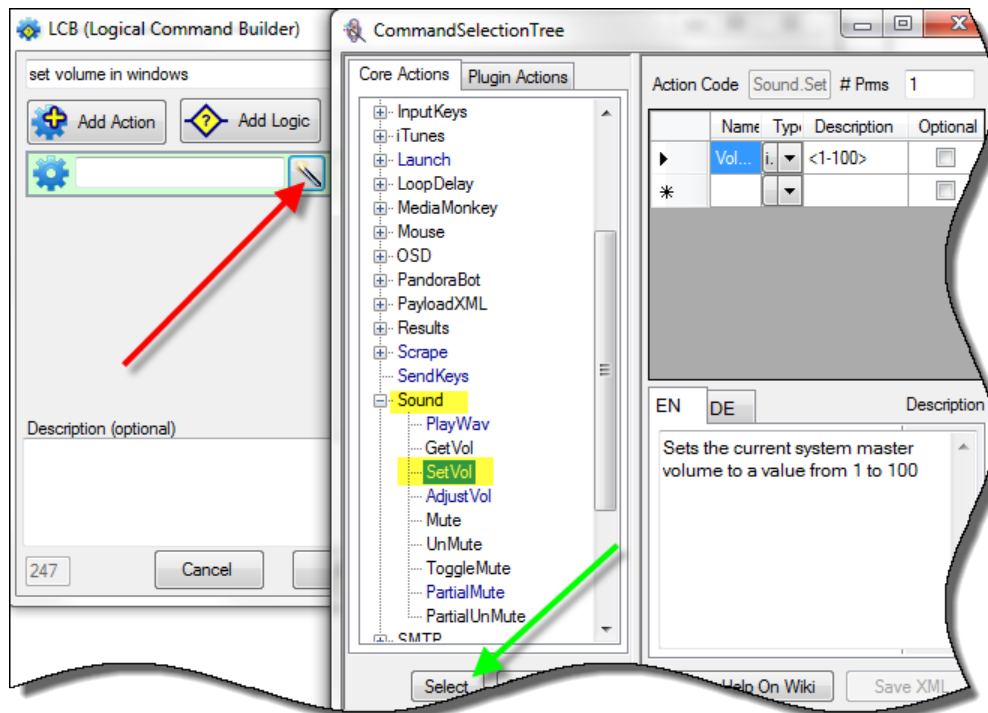
Tragen Sie hier Zahlen von 0 bis 100 in 10er Schritten ein. Sie könnten auch 0 bis 1000 eintragen, aber dies macht keinen Sinn, da hier Windows einen Wert von 0 für „Aus“ und 100 für maximale Laustärke vorgibt.



Jetzt sind die Vorarbeiten des Befehls abgeschlossen und wir können an die Programmierung des Befehls gehen. Hierzu klicken wir doppelt auf das „Blitzsymbol“ unserer erstellten Struktur, so das sich der „Logical Command Builder“ öffnet. Klicke auf "Add Action" um einen leeren Aktions-Befehl zu erstellen.

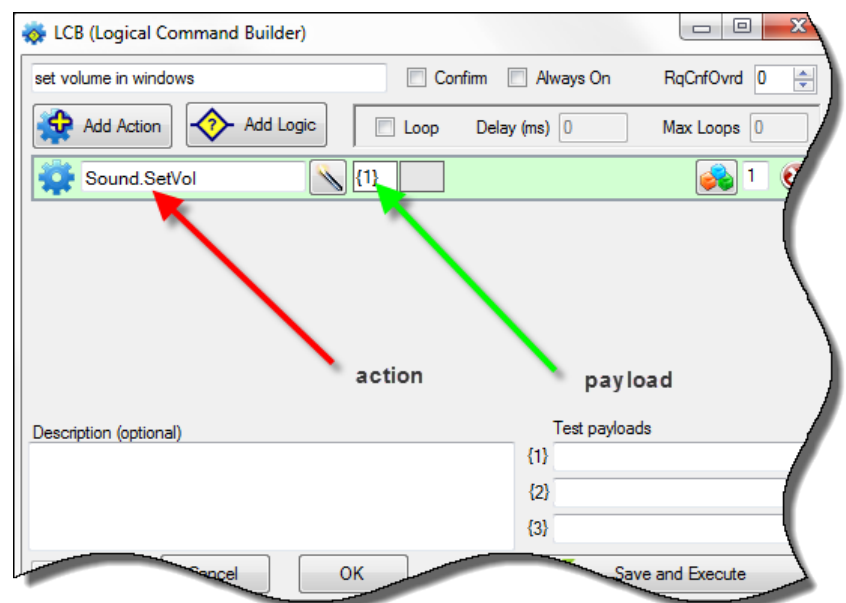


Klicke im Aktions-Befehl auf den „Zauberstab“, dieser öffnet den „Command Selection Tree“, klicke auf das Pluszeichen vor dem „Sound“ Eintrag und wähle die „SetVol“ Aktion. Klicke auf den „Select“ button um die Auswahl zu übernehmen.



Nun sollte es wie auf dem Bild rechts aussehen.

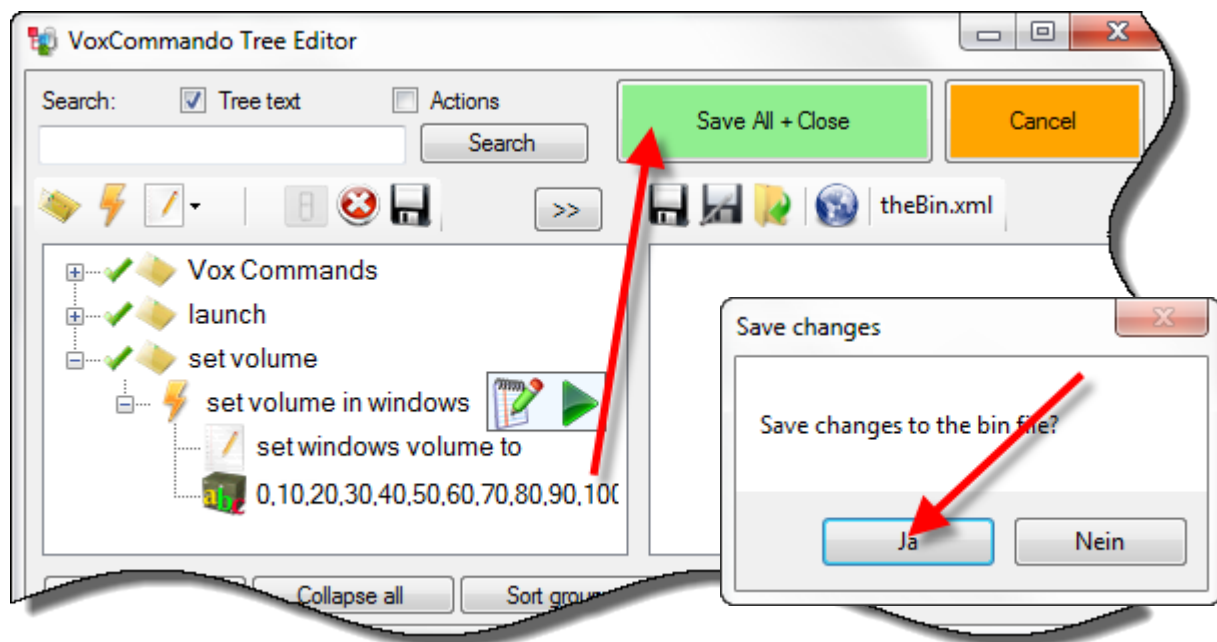
Was uns jetzt noch fehlt, ist die Einbindung der Payload Range, welche den Wert des Befehls festlegt. Auch dies ist sehr einfach, hierzu müssen wir nur wissen das jeder Payload in der Baumstruktur einer Befehlsfolge durchnummeriert ist, d.h. wenn mehrere Payloads in einem Befehl genutzt werden (was durchaus Sinn macht), hat der erste Payload die {1}, der zweite die {2} usw..



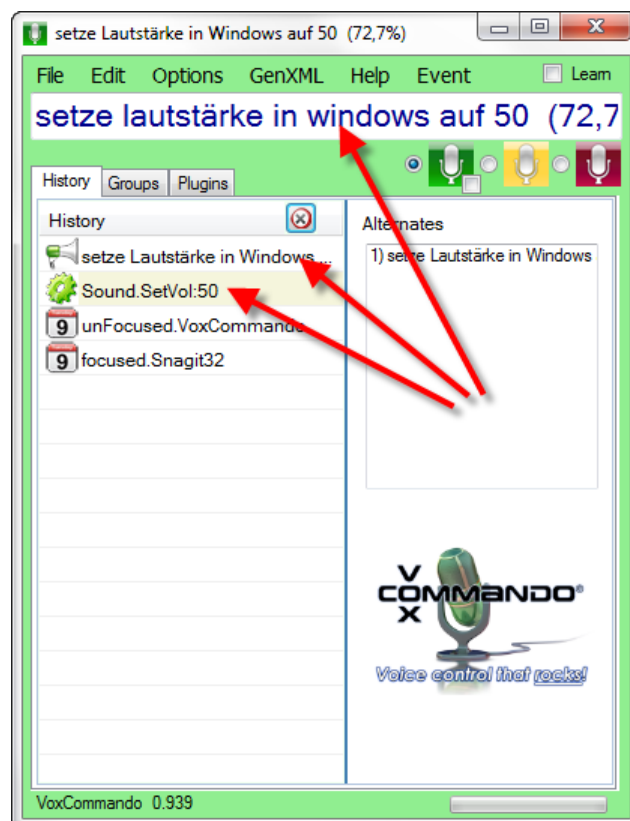
In unserem Fall gibt es nur einen Payload und dieser hat dem zufolge auch die {1}.

In Aktionen werden die Payloads immer in geschweiften Klammern geschrieben, also tragen wir im Parameterfeld {1} ein und klicken auf OK um den Wert zu speichern.

Nun schließen wir das Editorfenster mit „Save All“ und „Yes“. Danach macht VoxCommando ein „rebuild“ und dein Befehl ist fertig, um ihn zu nutzen.

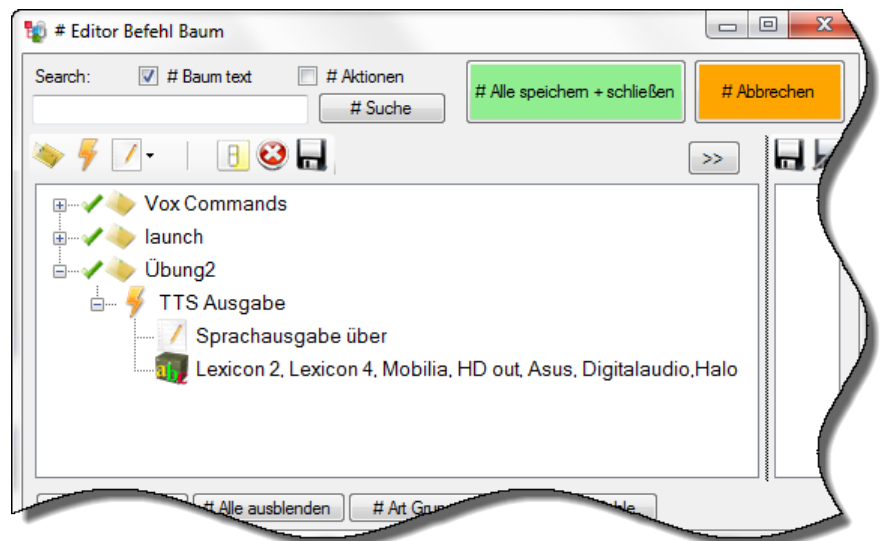


Spreche in dein Mikrofon „setze Lautstärke in Windows auf 50“ und VoxCommando sollte es so ausführen wie unten im Bild gezeigt.

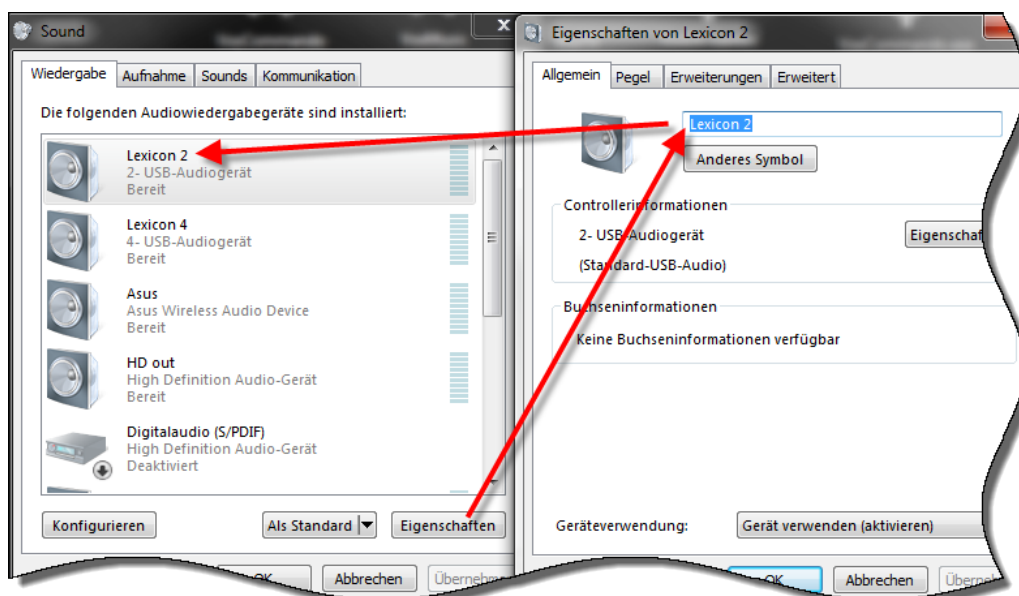
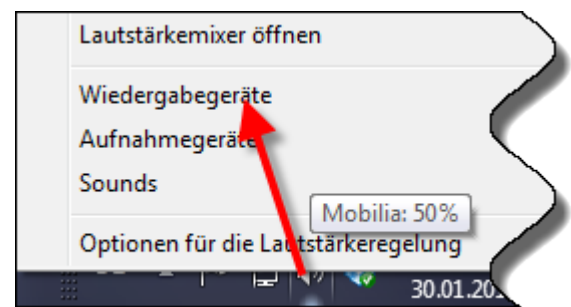


Als zweites Beispiel zur Payload List möchte ich Ihnen zeigen, wie Sie ihre Ausgänge der Soundkarte ansteuern können. Mit diesem Beispiel können Sie per Sprache einstellen, über welchen Ausgang die „Sprachausgabe“ von VoxCommando erfolgen soll.

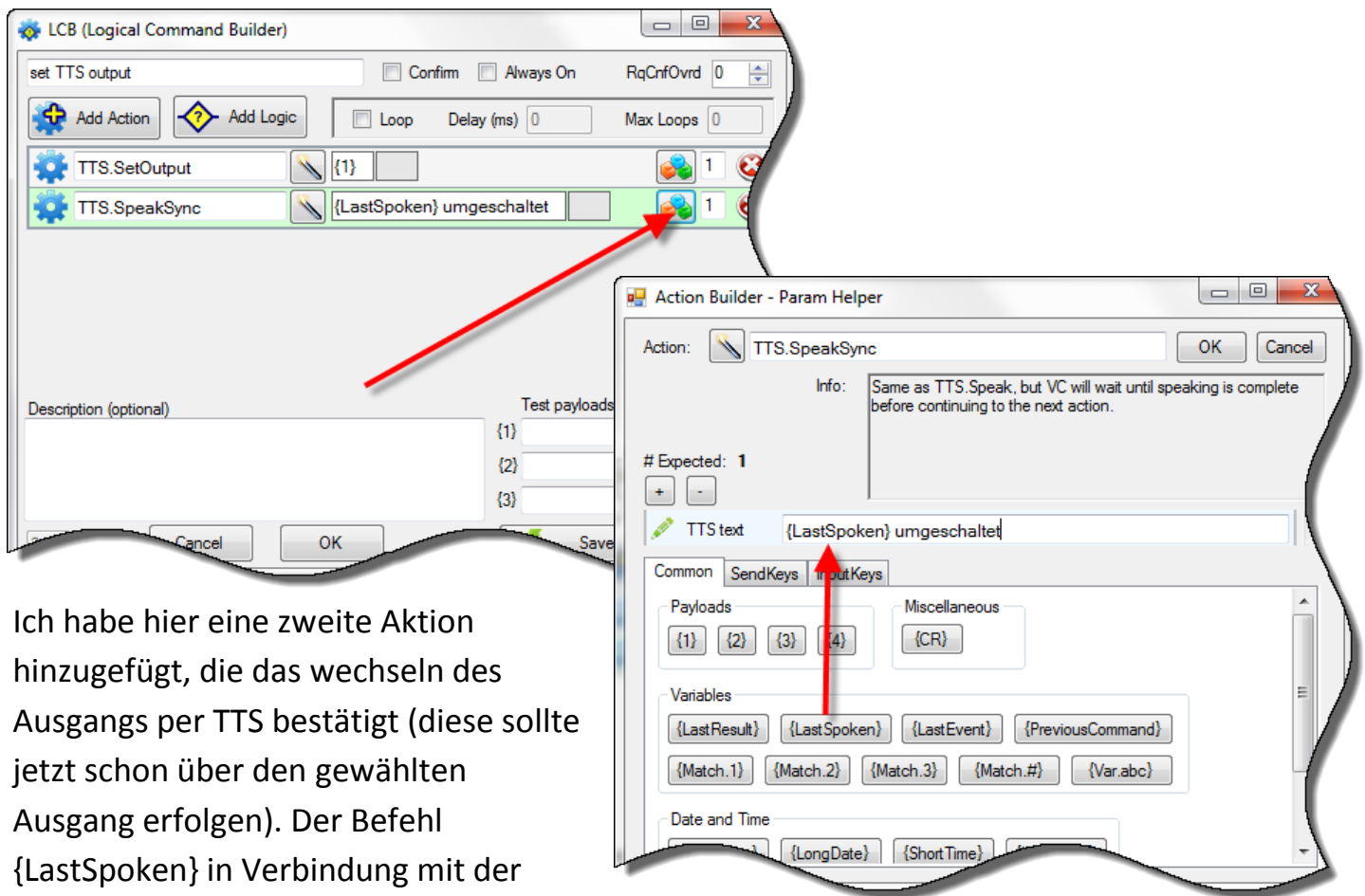
Die Struktur wird wie am Anfang des Tutorials beschrieben angelegt, nur nehmen wir als Phrase „Sprachausgabe über“ und in die Payload Liste tragen wir die Namen der Ausgänge von Ihrem Windows-System ein.



Diese können Sie aufrufen, in dem Sie mit der rechten Maustaste auf das Lautsprechersymbol klicken und „Wiedergabegeräte“ auswählen. Hier können die Ausgänge über Eigenschaften auch umbenannt werden.



Nun erstellen wir den Befehl mit einem Doppelklick auf das „Blitzsymbol“ unserer Baumstruktur und tragen folgende Werte, wie in den folgenden Bildern gezeigt, ein.



Ich habe hier eine zweite Aktion hinzugefügt, die das wechseln des Ausgangs per TTS bestätigt (diese sollte jetzt schon über den gewählten Ausgang erfolgen). Der Befehl {LastSpoken} in Verbindung mit der „TTS.SpeakSync“ Aktion, gibt den zuletzt gesprochenen Befehl an, in unserem Fall eine der Phrasen aus der „Payload List“. Ihr könnt das Ganze noch etwas abrunden, wenn ihr hier als Parameter:

„auf {LastSpoken} umgeschaltet“

einsetzt. Jetzt noch alles speichern und ausprobieren!

WEITER MIT PAYLOAD DICTATION REGULAR /SPELLING IN TEIL 3