

1 und Tastatur Recorder – Anleitung

1.1 Maus und Tastatur Recorder

Anleitung für 'Maus und Tastatur Recorder' Version ~~6.7.0~~

Weitere Informationen zum Maus und Tastatur Recorder sind erhältlich im Internet:

<http://www.kratronic.com/tools-de/recorder/>

(Diesen Link in den Internet Explorer kopieren um die Anleitung dort zu öffnen)

Diese Anleitung herunterladen (MS Word 97/2000/XP/2003 Format)

Anleitung abspeichern: Mit der rechten Maustaste auf den Link klicken und 'Ziel speichern unter..' wählen.

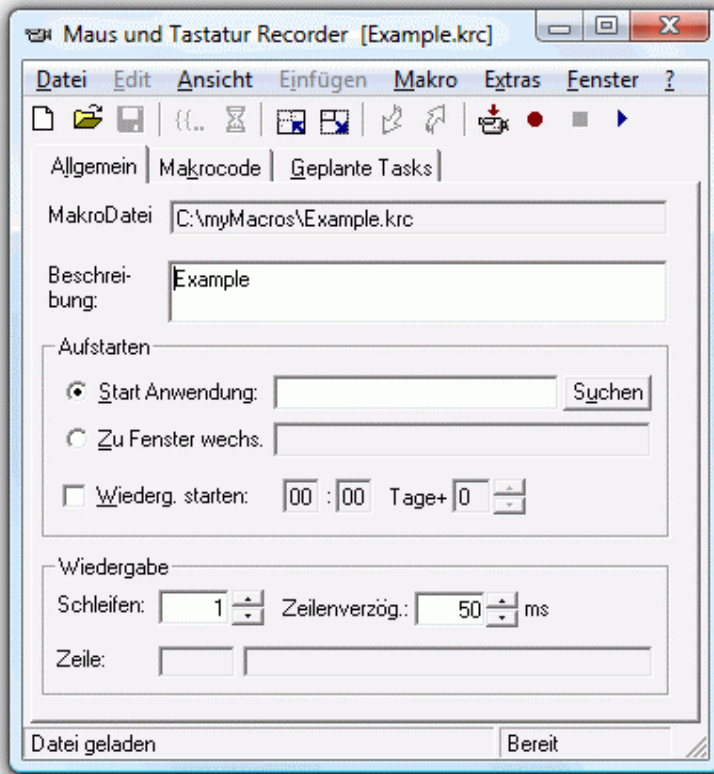
Falls das nicht funktioniert, auf den obigen Link klicken. Nach einiger Zeit erscheint das Manual im Help-Fenster. Nun mit 'CTRL a' den Text markieren, mit CTRL c den Text kopieren, ein neues Word-Dokument öffnen und mit 'CTRL v' den Text in das Word Dokument einfügen. Nun kann man die Anleitung speichern oder drucken. (CTRL wird auch als STRG bezeichnet)

1.2 Einleitung

Vor langer, langer Zeit, als wir noch mit Windows 3.1 arbeiteten, hatten wir Zugriff auf ein praktisches Programm, den Microsoft Recorder. Mit diesem Recorder konnte man Tastenklicks und Maus-Bewegungen aufzeichnen. Die daraus entstandenen einfachen Makros konnte man mittels Doppelklick auf eine Verknüpfung starten.

Mit dem Erscheinen von Windows 95 und Windows NT 4.0 verschwand der Microsoft Recorder ohne jeglichen Ersatz. Die einzige Alternative waren nun Batch Dateien im alten DOS. Später erschien dann VBS (Visual Basic Scripting), aber VBS ist viel zu kompliziert für schnelle und einfache Makros.

Der 'Maus und Tastatur Recorder' für Windows ~~98/ME und Windows NT 4.0/2000~~/XP/Vista/Win7, geschrieben von John Krauer, bietet nun einen Ersatz für den verlorenen Microsoft Recorder. Das nächste Bild zeigt dieses Werkzeug:



1.3 'Maus und Tastatur Recorder' Anforderungen

1.3.1 Betriebssystem:

Der 'Maus und Tastatur Recorder' kann auf folgenden Betriebssystemen eingesetzt werden;

~~Windows 98 / ME~~

~~Windows 2000~~

- Windows XP
- Windows Vista
- Windows 7 (32 Bit und 64 Bit)

~~Nicht getestet wurden Windows Server Versionen, aber es gibt keinen Grund warum der Recorder dort nicht laufen sollte.~~

~~Für Windows 95 und Windows NT4.0 muss die alte 'Maus und Tastatur Recorder' Version V4.5 verwendet werden.~~

~~Internet Explorer 5.0 oder neuer muss auf dem PC vorhanden sein.~~

1.3.2 Bildschirmauflösung:

Der 'Maus und Tastatur Recorder' benötigt eine Bildschirmauflösung von mindestens: ~~600 * 800.~~

~~Soll die grosse Schriftart verwendet werden (Bildschirmeinstellungen), dann ist eine Auflösung von 768 * 1024 nötig~~

~~Für die Arbeit auf kleineren Bildschirmen können 'Virtual-Screen'-Programme verwendet werden. Solche Programme stellen bis zu 2'500 Bildschirme zur Verfügung. Fenster, welche zu gross sind für einen Bildschirm, werden dann auf zwei Bildschirmen dargestellt. Hier die Adresse für ein solches Programm:~~

~~<http://enablevirtualdesktop.com/virtualdesktop/index.htm> (Virtual Desktop)~~

Festplatte und Direktzugriffsspeicher (RAM)

Der 'Maus und Tastatur Recorder' benötigt 6 MB Festplatte und im Minimum 64 MB RAM.

1.4 Lizenzen

Der 'Maus und Tastatur Recorder' ist ein 'Shareware' Programm. Das bedeutet, dass der Recorder während 3 Monaten getestet werden kann. Erst dann muss eine lizenzierte Version gekauft werden. Mit der Test-Version können nach 3 Monaten neue Makros nicht mehr abgespeichert werden. (Abspielen funktioniert immer noch). Die lizenzierte Version kann über das Internet bezogen werden. Dazu klickt man im geöffneten 'Maus und Tastatur Recorder' auf den Knopf '?' und dann auf 'Lizenz bestellen'. Dabei wird folgende Adresse angewählt: <http://www.kratronic.com/tools-de/recorder/>

Die lizenzierte Version kann mit Kreditkarte, Schecks und auch Banküberweisung ~~Barzahlung~~ bezahlt werden.

Eine Einzel-Lizenz (single license) für den 'Maus und Tastatur Recorder' kostet 23 US\$. (Eine Firmenlizenz (site-license) kostet 100.- US\$ und eine weltweite (world-wide) Firmen Lizenz kostet 300 US\$.)

Lizenz Bedingungen

- Eine Einzel-Lizenz darf nur für einen PC verwendet werden.
- Eine Firmen-Lizenz gilt für eine ganze Firma in einer Stadt.
- Eine weltweite Lizenz ist nötig, wenn der 'Maus und Tastatur Recorder' in verschiedenen Zweigfirmen in verschiedenen Städten verwendet werden soll.

Eine Firmen-Lizenz und eine weltweite Lizenz erlaubt die Verwendung überall in dieser Firma. (Aber nicht ausserhalb der Firma)

Der Status des 'Maus und Tastatur Recorders' (Test-Version oder Lizenzierte Version) wird mit dem Menu '?' und dann 'Information' angezeigt.

1.5 Funktionen

Der 'Maus und Tastatur Recorder' kann Tastaturklicks, Mausklicks und Maus-Bewegungen aufnehmen und wieder abspielen. Aber das ist nur die Grundfunktion. Der Recorder kann noch viel mehr:

- Abspeichern (und laden) der Makros in normale Dateien und einen Zwischenspeicher
- Programm starten oder zu laufenden Programmen umschalten
- DOS (Eingabeaufforderung) Befehle ausführen
- Dateien löschen, kopieren, vergleichen und finden
- Verzeichnisse erstellen, kopieren und löschen
- Inhalt von Laufwerk-Verzeichnissen lesen und in Datei speichern
- Verschiedene Arten von Verzögerungen verwenden
- Befehle kommentieren
- Den Start von Makros um Minuten, Stunden oder gar Tage verzögern
- Mittels Zeitschaltprogramm Makros einmalig oder regelmässig ausführen
- Makros automatisch wiederholen (beliebig oft)
- Konditionen verwenden: (If then else / Wenn.. dann)
- Texte vergleichen (Direkt oder mit Zwischenablage)
- Erzeugen von Zufallszahlen und Zufalls-Text
- Laden und ausführen von mehreren Makros, gesteuert durch ein Meister Makro
- Verwenden von Zahlen- und Textspeichern. (Variable)

- Befehle nur während ersten, zweiten... letzten Wiederholung ausführen
- Verwendung von Import-Dateien
- Benutzer-Abfragen
- Prüfen von aktiven Fenstern (Warten auf Fenster, warten auf das Verschwinden)
- Prüfen von Bildschirmbereichen auf Veränderungen (Warten auf Veränderung)
- Suchen von Mustern auf dem Bildschirm
- Warten auf vollständigen Seitenaufbau im Internet Explorer
- Diverse mathematische und Text Befehle
- Bearbeiten der Makros mit integriertem Editor
- Makros mittels 'Befehlszeile' starten (incl. Variablen-Parameter)
- Makros (und Aufnahmemodus) mit Schnellzugriffs-Tasten (Hotkeys) starten

~~Verwendung des Recorders im Hintergrund als Log-Buch~~

1.6 Arbeiten mit dem 'Maus und Tastatur Recorder'

Anders als der alte Microsoft Recorder erlaubt es der 'Maus und Tastatur Recorder' die aufgezeichneten Makros zu bearbeiten. Der Recorder zeichnet alle Aktivitäten der Tastatur und der Maus in allen Programmen auf. Weit über 140 zusätzliche Befehle erlauben es fast jede Aufgabe im Bereich der Automation zu meistern.

1.6.1 Wichtige Hinweise

Der 'Maus und Tastatur Recorder' ist ein mächtiges Werkzeug. Wie bei jedem starken Werkzeug, muss man bei dessen Verwendung vorsichtig sein. Der 'Maus und Tastatur Recorder' spielt alle Befehle, welche vorher aufgezeichnet wurden, wieder ab. Dabei kontrolliert der 'Maus und Tastatur Recorder' **nicht** automatisch, ob die Situation auf dem Bildschirm immer noch die gleiche ist, wie damals bei der Aufnahme. Das bedeutet, dass der Recorder bei z.B. zu schneller Wiedergabe die Synchronisation verlieren kann. In so einer Situation können Schäden angerichtet werden. (Falsche Knöpfe können gedrückt werden, falsche Befehle ausgeführt, usw...)

Wie können solche ungemütliche oder gar gefährliche Situationen vermieden werden? Es folgen ein paar Hinweise, auf was geachtet werden sollte:

- Lese die Anleitung genau, damit alle Befehle und Möglichkeiten des 'Maus und Tastatur Recorders' klar sind.
- Teste Makros sorgfältig, bevor sie schnell oder unbeaufsichtigt ausgeführt werden.
- Platziere den 'Maus und Tastatur Recorder' sichtbar in eine Ecke des Bildschirms. (Dazu die Option 'Recorder Fenster immer sichtbar' einschalten.) Auf diese Weise kann auch die Anzeige auf dem Recorder überwacht werden. Zudem ist der Recorder dann schnell erreichbar, wenn das Makro gestoppt werden muss.
- In einer "Notsituation" kann ALT-Pause/Break verwendet werden um den Recorder zu stoppen. (Bei Notebooks: Fn Alt Pause)
- Benutze die 'Zeilen Verzögerung' von z.B. 2000 (2 sec) um den Aktivitäten des Recorders Zeile um Zeile folgen zu können. Wenn das Makro nicht die erwarteten Schritte ausführt, kann mit Stop-Knopf der Makroablauf gestoppt werden.
- Überlege beim Erstellen eines Makros gut, wo und wie grosse Verzögerungen eingebaut werden müssen. Beispiel: Wenn der Recorder auf eine langatmige Grafik-Konvertierung warten muss, sollte die Verzögerungszeit so eingestellt sein, dass auch für das grösste Bild genügend Zeit bleibt. (Reservezeit einplanen) Eine andere Möglichkeit auf das Ende einer Operation zu warten ist die Benutzung der Befehle: `{{WaitForFocus or {{WaitForEndFocus {{WaitForScreenChange`
- Schalte alle Geräte aus, welche den Makro-Ablauf stören können. Bildschirmschoner verdecken die zu bedienenden Fenster. Hintergrundprogramme benötigen oft viel Rechenleistung. Dann

sind oft die Verzögerungszeiten plötzlich zu kurz. Heraufspringende Fenster können zu massiven Störungen führen. (Ein Enter wird plötzlich, statt für den normalen Makroablauf, für das neue Fenster gebraucht). Wechselpatten gehen oft in einen Schlafmodus. Wenn sie wieder aufstarten, blockieren sie den PC für einige Sekunden.

- Während des Makroablaufes kann der PC nicht anderweitig benutzt werden. Wenn Mausclicks benutzt werden, muss auch sichergestellt sein, dass die echte Maus nicht bewegt wird. Alternative: `LockUI True` verwenden.

Der 'Maus und Tastatur Recorder' wurde von uns eingehend auf Fehler getestet. Der Autor des 'Maus und Tastatur Recorder' kann nicht verantwortlich gemacht werden für Schäden, welche durch die Benutzung des 'Maus und Tastatur Recorders' auf dem PC des Benutzers auftreten könnten.

1.7 Aufzeichnen von Makros

Nach dem Starten des 'Maus und Tastatur Recorders' erscheint das am Anfang der Anleitung abgebildete Fenster. Normalerweise will man mit Recorder Aktivitäten in anderen Programmen wie z.B. Microsoft Word oder Excel ausführen. Das bedeutet, dass zuerst ein Befehl zum Starten oder zum Umschalten auf das gewünschte Programm notwendig ist. Dieser Start- oder Umschaltbefehl kann folgendermassen eingefügt werden: Register 'Makrocode' wählen, Knopf 'Befehl einfügen' (`{...}`) klicken und start oder switch ¹⁾ eintippen.

Der Recorder zeigt nun eine Information über den Befehl an. Dazu gehört eine Erklärung und Beispiele. Mit Hilfe dieser Info kann nun der Befehl vervollständigt werden. Durch den Knopf OK wird der Befehl dem Makro hinzugefügt. Als nächstes wird der Knopf 'Aufzeichnen' (Roter Punkt) angeklickt. Nun ist der Recorder 'scharf'. Bevor wir die Aktivitäten im Word oder Excel aufzeichnen können, müssen wir auf das Programm umschalten. Dies muss mit der Tastenkombination ALT TAB (Tabulator) erfolgen. Grund: Ein Mausclick auf das Programm-Fenster würde ja auch aufgezeichnet. (ALT TAB wird nicht aufgezeichnet.) Nun können im Programm diejenigen Tastenklicks / Mausclicks ausgeführt werden, welche aufgezeichnet werden sollen. Aufgezeichnet werden alle Tasten, Mausclicks mit allen 3 Tasten, Maus Doppelclicks und mit der Maus ziehen. Der Recorder zeichnet die Tasten und Maus Aktivitäten völlig ohne Zeitinformationen auf. Das bedeutet, dass das Makro beim Abspielen viel schneller abläuft, als es geschrieben wurde. Daher müssen bei Bedarf Verzögerungen eingefügt werden. Das kann auch wieder mit dem Knopf 'Befehl einfügen' erreicht werden. (Befehl: `Delay`) Zusätzlich stehen Schnellzugriffs-Tasten (Hotkeys) zur Verfügung. Dazu wird CTRL SHIFT ALT F12 (STRG UMSCHALT ALT F12) gedrückt und dann einer der folgenden Tasten: **1:** 300 ms, **2:** 500 ms, **3:** 750ms, **4:** 1min, **5:** 2min, **6:** 5min, **7:** 10min, **8:** 15min, **9:** 50min, **0:** 1std.; **q:** 1sec, **w:** 2sec.....**o:** 9sec; **p:** 10sec; **a:** 11sec, **s:** 12sec....**l:** 19sec; **y/z:** 21sec, **x:** 22sec....**m:** 27sec.

Nachdem alle nötigen Tasten (und Mausclicks) im Programm ausgeführt sind, mit ALT TAB wieder auf den Recorder zurückschalten. Mit einem erneuten Klick auf den Knopf 'Aufzeichnen' wird der Aufzeichnungs-Modus wieder verlassen.

Wenn das Makro bearbeitet werden soll, kann dies durch einen Klick auf die Register (Tab): 'Makrocode' erfolgen. Eine umfassende Beschreibung der einzelnen Befehle kann im Kapitel: 'Funktions-Beschreibung' gefunden werden.

Hinweis: Der 'Maus und Tastatur Recorder' behandelt Tastendrücke und Maus Klicks unterschiedlich. Im Normal-Text-Modus (`{SendTextNormal}`) warten Tastendrücke auf die Anwendung. Das heisst, ein zweiter Tastendruck wird erst gesendet, wenn der erste Tastendruck von der Anwendung abgearbeitet ist. ~~(Das ist aber nicht mehr so bei Windows Vista/Win7).~~ Maus Klicks werden teilweise genau dann abgearbeitet, wenn sie vom Recorder gesendet werden. Das bedeutet, dass man bei Maus Klicks die Wartezeiten viel sorgfältiger einstellen muss. (Genug Wartezeit zwischen den einzelnen Maus Klicks). Schlussfolgerung: Wenn immer möglich Normal-Text-Modus Tastendrücke verwenden und nicht Mausclicks ~~bei Windows XP und früheren Versionen.~~

¹⁾ Ein `Switch` - Befehl kann auch durch die Tastenkombination; CTRL-ALT-SHIFT-F12 F12 (STRG-ALT-UMSCHALT-F12 F12) generiert werden. Siehe auch `Switch` Befehlsbeschreibung.

1.8 Das erste 'Maus und Tastatur Recorder' - Makro

Dieses Kapitel gibt ein erster Eindruck wie einfach es ist mit dem 'Maus und Tastatur Recorder' ein Makro zu erstellen.

Versuche die folgenden Schritte:

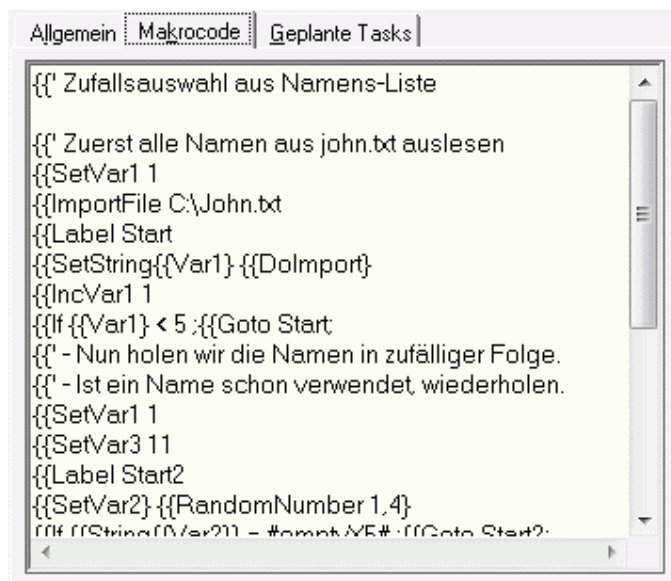
- Textverarbeitung starten (z.B. Word)
- Maus und Tastatur Recorder starten
- Den Knopf: 'Aufzeichnen' (Roter Punkt) drücken
- Mit ALT TAB (Tabulator) auf Word wechseln
- Die Tasten STRG ALT UMSCHALT gedrückt halten, dann F12 drücken. Dann die Tasten loslassen und nochmals F12 drücken. (Dadurch wird ein Umschaltbefehl zum Textverarbeitungs-Fenster eingefügt)
- Beliebigen Text eingeben
- Mit der Maus Text markieren und umformatieren
- Mit 'STRG a' (CTRL a) alles markieren und löschen
- Mit ALT TAB zum Recorder zurückwechseln
- Den 'Aufzeichnen' Knopf drücken. (Aufzeichnen wird beendet)

Jetzt den Wert in 'Zeilenverzögerung' (im Register 'Allgemein') auf 1000 stellen und den Knopf: 'Abspielen' (Blaues Dreieck) drücken und zusehen was der Recorder tut...

Nun kann die 'Zeilenverzögerung' kürzer eingestellt werden. Wenn man die Zeit aber zu kurz wählt, dann kann es zu Fehlern kommen (Mausklicks kommen schneller als die Textverarbeitung diese verarbeiten kann)

1.9 Funktions-Beschreibung

1.9.1 Makro-Fenster und Tasten Eingabe



```
Algemein | Makrocode | Geplante Tasks |
{{ Zufallsauswahl aus Namens-Liste
{{ Zuerst alle Namen aus john.txt auslesen
{{SetVar1 1
{{ImportFile C:\John.txt
{{Label Start
{{SetString{{Var1}} {{DolImport}}
{{IncVar1 1
{{If {{Var1}} < 5 :{{Goto Start
{{ - Nun holen wir die Namen in zufälliger Folge.
{{ - Ist ein Name schon verwendet, wiederholen.
{{SetVar1 1
{{SetVar3 11
{{Label Start2
{{SetVar2}} {{RandomNumber 1,4}}
{{If (String{{Var2}} = #empty#) :{{Goto Start2}}
```

Das Makrofenster zeigt das aufgenommene (oder geladene) Makro. Um ein Makro aufzunehmen, muss der Knopf 'Aufzeichnen' (Roter Knopf) angeklickt werden. Tasten und Maus-Klicks welche im Aufzeichnungs-Modus gedrückt werden, erscheinen automatisch im Hauptfenster. Die

aufgenommenen Klicks (und Befehle) werden automatisch nach oben gerollt, sodass man immer die letzten aufgezeichneten Zeilen sieht.

Beim Verwenden von Sondertasten werden folgende Befehle in das Makro geschrieben:

Taste	Code
Buchstaben wie Abc	Abc
BACKSPACE <=	{BACKSPACE}
BREAK (Unterbrechen)	{BREAK}
CAPS LOCK (Grossb.) *)	{CAPSLOCK}
NUM LOCK (Zahlen) *)	{NUMLOCK}
SCROLL LOCK *)	{SCROLLLOCK}
PRINT SCREEN (Druck) *)	{PRTSC}
SHIFT (Umschalt) *)	{SHIFT}
CTRL (STRG) *)	{CTRL}
ALT *)	{ALT}
LOESCHEN	{DELETE} oder {DEL}
ENDE	{END}
ENTER (RETURN)	{ENTER}
ESC	{ESC}
HELP (Hilfe)	{HELP}
ANFANG	{HOME}
EINFUEGEN	{INSERT}
NACH UNTEN	{DOWN}
NACH LINKS	{LEFT}
SEITE UNTEN	{PGDN}
SEITE OBEN	{PGUP}
NACH OBEN	{UP}
NACH RECHTS	{RIGHT}
TAB (Tabulator)	{TAB}
F1	{F1}
F2	{F2}
F3	{F3}
.....	
F11	{F11}
F12	{F12}
ENTER auf NumPad	{{NumPadEnter}}
/ auf NumPad	{{NumPadSlash}}
* auf NumPad	{{NumPadAsterisk}}
- auf NumPad	{{NumPadMinus}}
+ auf NumPad	{{NumPadPlus}}
. auf NumPad	{{NumPadDot}}
0 auf NumPad	{{NumPad0}}
1 auf NumPad	{{NumPad1}}
2 auf NumPad	{{NumPad2}}
3 auf NumPad	{{NumPad3}}
4 auf NumPad	{{NumPad4}}
5 auf NumPad	{{NumPad5}}
6 auf NumPad	{{NumPad6}}
7 auf NumPad	{{NumPad7}}
8 auf NumPad	{{NumPad8}}
9 auf NumPad	{{NumPad9}}

Windows Taste:	{{WindowsKey}}
ContextMenü Taste	{{WindowsDropDown}}

Befehle welche kursiv geschrieben sind werden vom Recorder nicht aufgezeichnet. Befehle welche kursiv geschrieben sind und ein *) enthalten funktionieren nicht so wie erwartet. Als Ersatz gibt es Recorderbefehle wie {{ScollLockOn, {{Shift.. etc. resp. ^,%,+ können nur im Text-Modus: {{SendTextSpecial} verwendet werden.

Die Sondertasten CTRL (STRG), ALT und SHIFT (UMSCHALT) können beim Aufzeichne wie-normal benutzt werden

Der 'Maus und Tastatur Recorder' fügt dafür folgende Zeichen ein:

Taste	Code
SHIFT (UMSCHALT)	+
CTRL (STRG)	^
ALT	%

Beispiele: +^h = Shift Ctrl h, +^%{F1} = Shift Ctrl Alt F1, ^%e = € (^% entspricht Alt Gr)
Für weitere CAPS LOCK, NUM LOCK, CTRL, Shift, Alt Möglichkeiten siehe die Befehle {{CapsLock, {{NumLock, {{CTRL, {{Alt und {{Shift in Kapitel: 'Befehle hinzufügen'

Einige Buchstaben werden im 'Maus und Tastatur Recorder' speziell behandelt. Es sind dies folgende:

Das Plus-Zeichen (+), Dächlein (^), Prozent-Zeichen (%), Tilde (~), Klammern (), Eckige Klammern ([]) und geschwungene Klammern ({}). Alle dieses Buchstaben müssen mit geschwungenen Klammern eingefasst werden. {+},{^},{%},{~},{(},{)},{{},{}},{{{},{}}}. Hinweis: Beim deutschen Windows wird das Zeichen {^} als & ausgegeben... Dies ist ein bekannter Fehler im deutschen Windows :- (Workaround: Zwischenablage und CTRL v verwenden.

Das Anführungszeichen (") wird in Makro-Dateien als Kontroll-Buchstabe verwendet. Daher muss es mit einem anderen Buchstaben ersetzt werden. Der 'Maus und Tastatur Recorder' benutzt ein (â) anstelle des Anführungszeichens.

Alle obigen Umwandlungen werden vom 'Maus und Tastatur Recorder' automatisch gemacht. Nur wenn man ein Makro manuell nachbearbeitet, muss man diese Regeln beachten.

Hinweise:

- Der 'Maus und Tastatur Recorder' kann im Normal-Text-Modus keine (oder nur unvollständige) Tastendrücke an Programme senden, welche nicht vollfür Microsoft Windows-konform sind. Es muss dann der erstellt wurden. Eines dieser Nicht-Windows-Programme ist DOS. Das heisst, der 'Maus und Tastatur Recorder' kann normalerweise nicht direkt mit DOS-Boxen arbeiten. Für DOS-Befehle enthält der Recorder den Befehl {{DOS. Beispiel {{DOS rename c:\temp\snap.jpg-snap2.jpg. Eine andere Möglichkeit Texte in DOS-Boxen hineinzubringen ist die Verwendung des Befehles: {{SendTextSpecial} verwendet werden. Beispiele: Alte SAP Programme, Spezielle Web-Formulare.
- Die oben aufgeführten Befehle welche *kursiv* geschrieben sind, können nicht über die Tastatur eingegeben werden. Für diese Befehle muss der Knopf 'Befehl einfügen' oder das Register (Tab) 'Makrocode' verwendet werden. Befehle welche kursiv geschrieben sind und ein *) enthalten können nur im Text-Modus: {{SendTextSpecial} verwendet werden.
Die Taste 'Print Screen' (Bildschirm in Zwischenspeicher ablegen) kann im Normal-Text-Modus nicht verwendet werden. Im {{SendTextSpecial-Modus aber funktioniert dieser Befehl. Für Bildschirm-Photos kann aber auch ein externes Programm angewendet werden. Ein sehr gutes Programm für diesen Zweck ist HyperSnap. <http://www.hyperionics.com>
Achtung: Die Schnellzugriffstasten des HyperSnap können nicht verwendet werden. Daher HyperSnap Aufzeichnung mit der Maus auslösen
- Die folgenden Tasten-Kombinationen können vom Recorder nicht erfasst und wiedergegeben werden: Ctrl-Alt-Del (Strg-Alt-Löschen), Ctrl-Esc (Strg-Esc), Alt-Tab, Alt-Esc. Für die Funktion

- Alt-Esc muss der Befehl `{{Switch` verwendet werden. Ctrl-Alt-Del kann vom Maus und Tastatur Recorder nicht ausgegeben werden. Grund: Microsoft Sicherheits-Massnahme.
- Tasten-Wiederholungen können folgendermassen definiert werden `{DOWN 10}` = 10 mal Mauszeiger nach unten. **Einschränkung: Max. 8000 Wiederholungen sind möglich in einem Befehl.**
 - Das Senden von Texten welche `+^%~()[]{}` enthalten, kann zu Problemen führen. Beispiel: `{{SetVarPower {{Power 5 | 300}}`. Das Senden von `{{VarPower` würde folgendes ausgeben: 4.90909346529773E+209. Das + würde als Umschalt (SHIFT) angesehen und die Applikation würde 4.90909346529773E"09 erhalten (auf einer Schweizer Tastatur). Dies kann vermieden werden bei der Verwendung der Zwischenablage (Clipboard) und von Strg/Ctrl v (^v). Siehe auch Kapitel: Hinweise / Einschränkungen: ([BracesReplace](#)). [Siehe auch {{SendTextSpecial}.ReplaceBraces](#).
 - Bei Spielen ist es hin und wieder nötig, eine Taste für längere Zeit zu drücken. Der Befehl `{{KeyDownTime` erlaubt es diese Zeit zu setzen. Beispiel: `{{KeyDownTime 3000}`. Jeder Tastendruck hält nun die Taste für 3 Sekunden unten. `{{KeyDownTime 0}` stellt die Taste-unten Zeit wieder auf den Normalwert zurück. WICHTIG: Dieser Befehl funktioniert nur zusammen mit `{{SendTextSpecial}!`"
 - Siehe 'Makros Bearbeiten' für weitere Hinweise.
Einschränkung: Windows erlaubt "bloss" 8100 {RIGHT}, {LEFT}... Befehle auf einmal. Das heisst dass z.B. {RIGHT 9000} einen Fehler erzeugt.

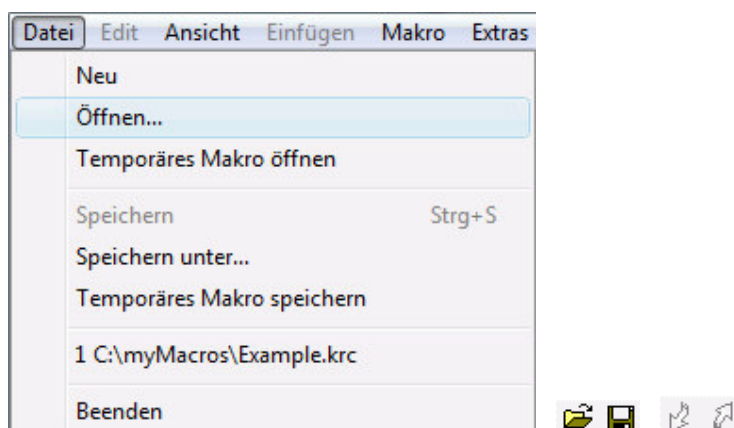
In der Grundeinstellung zeichnet der 'Maus und Tastatur Recorder' Tastatur-Klicks und Maus-Klicks auf. Sollen nur Tastatur-Klicks aufgezeichnet werden, kann die Maus-Aufzeichnung ausgeschaltet werden. Vorgehen: 'Extras' / 'Optionen' / 'Aufzeichnen' anklicken und dann 'Mausbewegungen aufzeichnen' ausschalten.

Während dem Aufnehmen von Makros können mittels Schnellzugriffs-Tasten (Hotkeys) Verzögerungs-Befehle in das Makro eingefügt werden. Schnellzugriffs-Tasten: CTRL SHIFT ALT F12 (STRG UMSCHALT ALT F12) und dann eine Taste zwischen 0 und 'm'. Wenn die voreingestellte Schnellzugriffs-Taste bereits durch ein anderes Programm benutzt wird, kann diese Schnellzugriffs-Taste umdefiniert werden. (Menü: 'Extras' / 'Optionen' / 'Aufzeichnen', 'Schnellzugriffs-Tasten für....'). Wichtig: Damit die Schnellzugriffs-Tasten funktionieren müssen die Makro-Schnellzugriffs-Tasten in den Optionen eingeschaltet sein. (Menu: Extras / Optionen / Allgemein)

Verzögerungs-Tasten Zuordnungen: **1**: 300 ms, **2**: 500 ms, **3**: 750ms, **4**: 1min, **5**: 2min, **6**: 5min, **7**: 10min, **8**: 15min, **9**: 50min, **0**: 1std., **q**: 1sec, **w**: 2sec.....**o**: 9sec; **p**: 10sec; **a**: 11sec, **s**: 12sec.....**l**: 19sec; **y/z**: 21sec, **x**: 22sec....**m**: 27sec.

Beispiel für 16 sec: CTRL SHIFT ALT F12 h

1.9.2 Laden und Speichern von Makros



Existierende Makros können mit dem Menu Datei und Öffnen geladen werden. Bereits früher geladene oder gespeicherte Makros findet man über dem Untermenü 'Beenden'. In der Symbolleiste kann mit dem grau-gelben Ordner ebenfalls ein Makro geladen werden. Für das schnelle Öffnen und Speichern eines Versuchs-Makros stehen die beiden Untermenüs: 'Temporäres Makro öffnen' und 'Temporäres Makro speichern' zur Verfügung. In der Werkzeugliste sind diese beiden Befehle auch nochmals als graue Pfeile vorhanden. Hinweis: Beim Speichern und Laden von temporären Makros werden keine Warnungen beim Überschreiben von Änderungen ausgegeben.

Zum Speichern stehen die Befehle 'Speichern' und 'Speichern unter...' zur Verfügung. Mit 'Speichern unter...' kann auch ein bestehendes Makro unter einem neuen Namen abgespeichert werden. In der Symbolleiste übernimmt die kleine Diskette das Speichern.

1.9.3 Starten oder Umschalten zu Programmen



Der 'Maus und Tastatur Recorder' arbeitet immer mit anderen Programmen. Deshalb muss zu Beginn eines Makros ein Programm gestartet werden, oder es muss zu einem bereits laufenden Programm gewechselt werden. Für das Wechseln auf ein Programm muss derjenige Name im Feld 'Zu Fenster umsch.' eingetragen werden, welcher beim Umschalten mit ALT TAB erscheint.

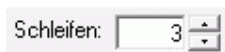
Beispiel: *Dokument2 - Microsoft Word*. Wenn nur ein Microsoft Word offen ist reicht auch *Microsoft Word* um umzuschalten.

Soll eine Anwendung gestartet werden, muss der ganze Pfad zu dieser Anwendung in das Feld 'Start Applikation' eingegeben werden. Beispiel: C:\PROGRAMS\MSOFFICE\WINWORD\WINWORD.EXE. Sind die beiden Felder leer, wird kein Fenster-Wechsel und auch kein Programmstart ausgelöst.

Befehle: {{Switch und {{Start

Das im Feld 'Start Anwendung:' eingetragene Programm wird nur einmal am Anfang des Makros gestartet (resp. auf das Programm in 'Zu Fenster wechs.' umgeschaltet). Soll ein Programm während des Programmablaufes gestartet (oder darauf umgeschaltet) werden, muss man die Befehle: {{Start oder {{Switch verwenden.

1.9.4 Wiederholungen



Oft muss eine aufgezeichnete Aktivität mehrmals wiederholt werden. Beispiel: Kopieren von 100 E-Mail Adressen in ein Word Dokument. Das Feld 'Schleifen' erlaubt es die gewünschten Makro-Wiederholungen einzugeben. Für die Anzahl Loops können max. 9 Stellen verwendet werden. Im obigen Beispiel wird das gesamte Makro 3 mal wiederholt. Manchmal sollten bestimmte Befehle nur während der ersten oder der letzten Wiederholung ausgeführt werden. Dazu können die folgenden Befehle verwendet werden:

{{1}Start C:\MSO\Word.exe. Hier wird Word nur während der ersten Wiederholung gestartet.
{{n}%db. Während dem letzten Durchlauf wird Word geschlossen.
{{2}Ich bin Linie 2.. Nur während der 2. Wiederholung wird der Text ausgegeben
{{<10}Dateiname00 Bei allen Wiederholungen kleiner als 10...

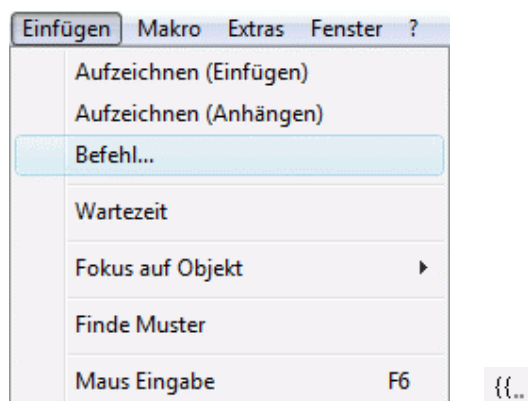
Manchmal sollte die Wiederholungs-Nummer verwendet werden können. Beispiel: Die Dateien File1.txt, dann File2.txt... sollen geöffnet werden. Dafür ist der Befehl {{Loop geeignet. Dieser Befehl gibt die Wiederholungszahl aus. Beispiel: {{Start Notepad.exe File{{Loop}.txt.

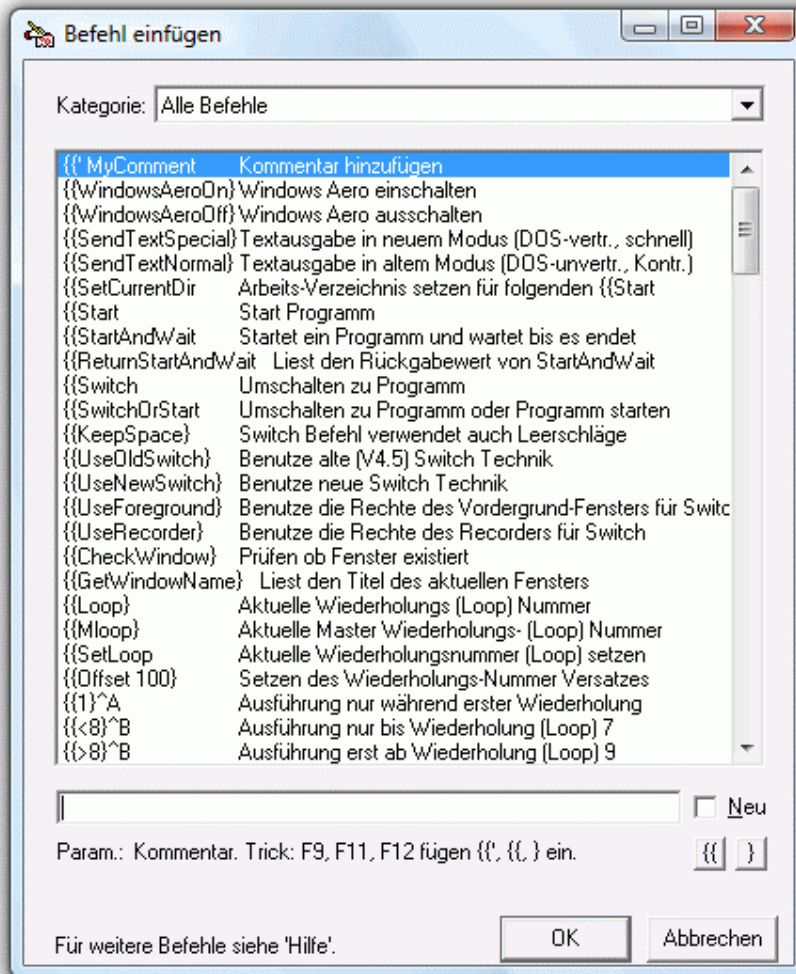
Sollte mit der Datei-Nummer 101 begonnen werden, kommt der Befehl Offset zum Einsatz. Beispiel: `{{Offset 100}}`. Während der ersten Wiederholung wird die Datei: File101.txt geöffnet.
Nachteil: Wenn die total Anzahl Loops z.B. 150 ist, läuft das Makro trotz `{{Offset 100}}` 150 mal durch. Der Befehl `{{SetLoop 101}}` vermeidet diesen Nachteil.

Eine andere oft benötigte Funktion ist das Bewegen des Positionszeigers abhängig von der Wiederholung. Beispiel: Es sind alle Dateien in einem Verzeichnis zu bearbeiten. Das heisst bei jeder Wiederholung soll die Markierung beim Dateiöffnen um einen Platz mehr verschoben werden. Dazu wird folgender Befehl verwendet: `{{DownLoop}}`. Bei Wiederholung 1 wird der Positionszeiger um einen Platz verschoben, bei Wiederholung 2 um zwei Plätze.... Der Befehl `{{RightLoop}}` macht das selbe, aber verschiebt den Positionszeiger nach rechts.

Wiederholungen können auch mittels dem Befehl `{{Goto}}` gebildet werden.

1.9.5 Befehle hinzufügen





Der Menü Befehl: 'Einfügen' / 'Befehl' (resp. der Knopf 'Befehl einfügen' {...}) öffnet ein Befehls-Eingabe-Fenster. Hier können spezielle Befehle ausgewählt werden. Wenn einer der Befehle angeklickt wird, erscheint dieser Befehl im untersten Feld. Ein weiterer Text zeigt an, auf welche Art der Befehl ergänzt werden muss. Eine ev. erscheinende 3. Textzeile gibt weitere Hinweise. Gleichzeitig erscheint ein Informations-Fenster mit einer Befehlsbeschreibung und Beispielen. Nach der Eingabe des ersten Befehls kann der Schalter 'Neu' betätigt werden. Dadurch wird es möglich einen weiteren Befehl auszuwählen, welcher dann im untersten Feld angefügt wird. Beispiel : `{{SetVar1 {{RandomNumber 1,10}}`

Das Befehls-Eingabe-Fenster hat noch eine weitere eingebaute Automatik. Es ist möglich, ein Befehl direkt in das unterste Feld einzutippen. Während des Tippens wird laufend der geeignetste Befehl markiert. Wenn der **richtigeRichtige** Befehl markiert ist (Beispiel: setg => SetGlobalVar) wird Ctrl m (Strg m) gedrückt. Dadurch wird der Befehl vervollständigt. Auch hier kann der Schalter 'Neu' angewendet werden. Die Eingabe-Automatik hat einen Seiteneffekt. Die 'nach links' Taste arbeitet nicht mehr ganz richtig. Um wieder Buchstaben in bereits eingegebenen Text einfügen zu können muss Ctrl n (Strg n) gedrückt werden.

Falls der gewählte Befehl einen Pfad und Dateinamen benötigt, wird der Knopf 'Suchen' eingeblendet. Damit kann ein Pfad und Dateiname gewählt werden, welcher dann in das unterste Feld eingetragen wird.

Hinweise:

Wenn ein Befehl ohne {{ am Anfang eingegeben wird, ersetzt der Recorder die Spezialzeichen wie z.B. [,], +, %, ^ ... mit folgenden Zeichen: {{}}, {{}}, {{+}}, {{%}}, {{^}}....

Das Fenster 'Befehle Einfügen' enthält nicht alle existierenden Befehle (nur diejenigen die man am häufigsten braucht.) Weitere Befehle sind im Kapitel: 'Spezielle Befehle' beschrieben.

Die Leerschläge in den Befehlen (siehe Beispiele) sind wichtig. Wenn man sie weglässt kriegt man Fehlermeldungen.

Die Befehl-Reihenfolge im 'Befehle einfügen' - Fenster kann in den Optionen auf alphabetisch geändert werden.

1.10 Alle Befehle

Verwendung von Parametern in Befehlen

Der 'Maus und Tastatur Recorder' erlaubt es die Ausgabe von bestimmten Befehle als Parameter für andere Befehle zu verwenden. Beispiel: `{{Start {{StringDateiname}}`. Der Befehl `{{Start` startet ein Programm. Der Text-Speicher `{{StringDateiname}}` enthält den Programm-Namen. Weitere Beispiele:

- `{{RenameFile {{String1}}{{Var2}}.{{WriteAsk1}} | {{String2}}{{Loop}}.TXT`
- `{{Switch {{String1}} - {{Date}}`
- `{{SetVar1}} {{Loop}}`

Regeln:

- Alle Befehle, welche einen Eingabewert benötigen, können den Ausgabewert eines anderen Befehles als Parameter benutzen. Hier eine Liste der Befehle, welche als Parameter benutzt werden können: `{{AppDataPath}, {{AppPath}, {{Asc}, {{Calc}, {{CheckFocus}, {{CheckWindow}, {{Chr}, {{Clipboard}, {{CompareFiles}, {{CR}, {{Date}, {{DateTime}, {{Div}, {{DoImport}, {{False}, {{FindFile}, {{FindPattern}, {{FileSize}, {{GetAeroState}, {{GetPpixel}, {{GetWindowName}, {{GlobString}, {{GlobVar}, {{ImportNextLine}, {{InStr}, {{IsNumeric}, {{Lcase}, {{Left}, {{Len}, {{LF}, {{Loop}, {{MacroPath}, {{Mid}, {{MkrFileName}, {{MLoop}, {{Mod}, {{Power}, {{RandomNumber}, {{RandomString}, {{Replace}, {{ReturnStartAndWait}, {{RGB}, {{Right}, {{SerialTime}, {{SetIEFocus}, {{String}, {{Tab}, {{Time}, {{Trim}, {{True}, {{Ucase}, {{Var}, {{WaitForEndFocus}, {{WaitForFocus}, {{WaitForIReady}, {{WaitForScreenChange}, {{WriteAsk}, {{WriteGlobAsk}}`
- Befehle welche als Parameter dienen, können selber auch wieder Parameter-Befehle enthalten. Beispiele:
 - `{{If {{FindFile {{String1}} = {{True}} ;;{{ShowError}}`
 - `{{SetGlobVarRandom}} {{RandomNumber {{VarUnten}, {{VarOben}}}`.
 - `{{CR}, {{LF}}` und `{{Tab}}` können beliebig oft in einer Zeile verwendet werden.
 - Parameter-Befehle müssen immer mit einer geschweiften Klammer geschlossen werden. (Siehe obige Beispiele)
 - Es ist auch möglich die obigen Parameter-Befehle in einem Ausgabe-Text zu verwenden. Beispiel:
`Benzinverbrauch in Monat: {{StringMonat}: {{VarBenzin}}`

Wichtig: Es ist nicht möglich weiteren Text oder Befehle nach einem Befehl, welcher am Zeilenanfang steht, zu verwenden. Beispiele (FALSCH):

`{{String1}} ist ein Mitglied der Partei: {{String2}}.`

Diese Zeile muss folgendermassen geändert werden:

`{{String1}}`
ist ein Mitglied der Partei: `{{String2}}`

Wichtig: Befehle welche nicht als Parameter verwendet werden und einen Text enthalten, dürfen nicht mit geschwungener Klammer abgeschlossen werden. Beispiele:

`{{SetString1}} MyText`
`{{Switch Document2 Microsoft Word}}`

1.10.1 Befehls-Beschreibungen

1.10.1.1 `{' Text, {' Text`

Dieser Befehl erlaubt es Kommentare einzufügen. Kommentare können an beliebiger Stelle in der Zeile platziert werden. Alle Texte nach einem `{'` oder `{{'` werden ignoriert. Alle Zeichen vor einem `{{'` werden verwendet. Das heisst, Leerzeichen vor einem `{{'` werden auch verwendet !! Dies ist normalerweise nicht erwünscht und so wird allermeistens der Befehl `{'` verwendet.

Hinweis: Für den Kommentar-Befehl `{{'` darf nicht das Anführungszeichen verwendet werden. Statt dessen müssen zwei Apostrophe zur Anwendung kommen.

1.10.1.2 `SendTextSpecial`, `SendTextNormal`

Diese beiden Befehle erlauben das Wählen des Text-Ausgabe Modus. Standardmässig (Normal) werden alle Texte mit dem internen VB-Befehl 'SendKeys' ausgegeben. Mit dem Befehl `SendTextSpecial` wechselt die Ausgabe zum API Befehl 'keybd_event'. Beide Modus haben Vor- und Nachteile:

`SendTextNormal`:

Vorteil: Nach jedem Buchstaben/Zeichen wird gewartet bis das Zeichen verarbeitet ist, erst dann wird das nächste Zeichen ausgegeben. Dadurch muss man sich nicht um die Wartezeiten bei Text-Ausgaben kümmern. (~~Hinweis: Das ist nicht mehr so bei Windows Vista/\$win7. Microsoft hat dieses Warten teilweise einfach entfernt :-(-)~~)

Nachteil: Die Normal-Ausgabe funktioniert zum Teil nicht für die DOS-Box und auch zunehmend nicht mehr in exotischenspeziellen Internet Web-Formularen, in modernen Windows oder alten Programmen wie z.B. in alten SAP Fenstern. Spezielle Zeichen wie +, %, ^, etc müssen eingeklammert werden.

Beispiel {+}Formularen.

`SendTextSpecial`:

Vorteil: Die Textausgabe funktioniert auch in alten Programmender-DOS-Box und in allen Internet Formularen.

Nachteil: Die Zeichen werden sofort ausgegeben. (Es wird nicht gewartet bis die Applikation das Zeichen abgeholt hat). Daher istmuss man bei diesem Ausgabe-Modus ein sorgfältigeressorgfältiges Delay-Management nötiganwenden.

Hinweise:

Der `SendTextSpecial` Modus ist wirksam ab der Stelle wo der Befehl im Makro steht, und wirkt bis zum Befehl `SendTextNormal` oder bis zum Ende des Makros (wenn `SendTextNormal` fehlt). Mittels `LoadMacro` aufgerufene Makros arbeiten wieder im Normal-Modus.

Sinnvollerweise verwendet man `SendTextSpecial` nur für die kritischen Teile wie z.Bdie-DOS-Box, widerspenstige Internet Formulare.

`SendTextSpecial+`

Mit `SendTextSpecial+` kann man Texte ganz normal ohne Spezialbehandlung der Sonderzeichen ausgeben. Die Sonderzeichen: +, ^, %, ~, (,), [,], {, } werden ganz normal ausgegeben. Mit

`SendTextSpecial+` haben die obigen Zeichen dann auch nicht die sonstigen Funktionen wie ALT, Shift, CTRL, Enter.. etc. Ausnahmen: Doppelte geschwungene Klammern werden nach wie vor als Befehl interpretiert. Beispiel: Zeile 1`{CR}`Zeile 2. Das `{CR}` erzeugt einen Zeilenumbruch.

Die Sonderzeichen ^, ~, ` werden nur richtig ausgegeben wenn der Schalter Schnellzugriffstasten (in Optionen) nicht aktiviert ist.

Mit `SendTextNormal` oder `SendTextSpecial` kann man diesen Spezial+ Modus wieder ausschalten; Office 2007, manchmal Internet Explorer 7 (IE7) und Windows Vista/Win 7.

1.10.1.3 `WindowsAeroOff`, `WindowsAeroOn`,

Diese Befehle erlauben das ein- und ausschalten von Windows Aero (Aero-Glass-Modus). ~~Das Ausschalten von Windows Aero ist nötig für die alte Version von {FindPattern. Hintergrund: Microsoft~~

~~hat den Befehl 'GetPixel', welcher für den obigen Befehl nötig ist, schrecklich langsam gemacht im Aero-Modus.~~

1.10.1.4 `GetAeroState`

Dieser Befehl liest den Zustand des Aero-Glass-Modus. Der Befehl kann benutzt werden um zu entscheiden ob `WindowsAeroOff` und `WindowsAeroOn` nötig sind. Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehle darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `if GetAeroState = True ;.....`

1.10.1.5 `Start`, `StartAndWait`

Diese Befehle erlauben das Starten eines Programms. Dazu muss der volle Pfadname und der Dateiname angegeben werden.

Beispiel: `Start "C:\Program Files\MeinPfad\MeinProgramm.exe"`

Ausnahme: Wenn der Pfad zum gewünschten Programm in der Windows Umgebung eingetragen ist, kann der Pfad weggelassen werden. Beispiele für solche Programme: Notepad, Calc, Write.

Hinweise: `Start` kann nur Programme mit folgenden Endungen starten: .exe, .bat, .com.

Will man eine Daten-Datei z.B. MeinText.DOC starten, kann man dies als Befehlszeile hinzufügen.

Beispiel: `Start C:\MSOFFICE\WORD.EXE "D:\Eigene Dateien\MeinText.DOC"`

Es ist auch möglich die Applikation erst zu starten (z.B. Winword) und dann mit z.B. ALT d f... die Dokument-Datei zu laden.

Pfade und Dateinamen welche einen Leerschlag enthalten, müssen mit Anführungs- und Schlusszeichen eingefasst werden. Beispiel: `"C:\Program Files\MSOFFICE\WORD.EXE" "D:\Eigene Dateien\MeinText.DOC"`

Verlangt das zu startende Programm ein Arbeits-Verzeichnis kann dieses vor dem Befehl `Start` mit `SetWorkingDir` definiert werden.

`StartAndWait`: Wenn auf das Ende eines gestarteten Programms gewartet werden soll, kann der Befehl `StartAndWait` verwendet werden. Beispiel: `StartAndWait Notepad`. Resultat: Der Notepad Editor wird gestartet und das Makro wartet, bis der Notepad geschlossen wird.

Hinweis: Mit `StartAndWait` kann man nur das Programm starten, aber keine Befehlszeile (CommandLine) mitgeben. Allerdings kann man einfach z.B eine TXT Datei angeben, dann wird diese mit dem zugehörigen Programm gestartet (hier Notepad). `StartAndWait C:\MeineDateien\John.TXT`

Einige Programme liefern einen "Return-Code" (Rückgabe-Wert). Auf diesen Code kann mit dem Befehl `ReturnStartAndWait` zugegriffen werden. Beispiel: `MessageBox ReturnStartAndWait`. Wichtig: Der Befehl: `ReturnStartAndWait` kann nur als Parameter verwendet werden. (Das bedeutet er darf nicht am Anfang einer Makro-Zeile stehen.)

1.10.1.6 `SetCurrentDir`

Einige Programme benötigen ein Arbeits-Verzeichnis, damit sie richtig arbeiten. Oft wird dieses Verzeichnis für das Vorgabe-Verzeichnis beim Öffnen und Speichern von Dateien benutzt. Beispiel: `SetWorkingDir C:\Programs\Office`. Beim der nächsten Verwendung von `Start` oder `SwitchOrStart` wird C:\Programs\Office als Arbeits-Verzeichnis verwendet.

1.10.1.7 **`DOS` (Eingabeaufforderung)**

Dieser Befehl erlaubt das Ausführen von DOS (Eingabeaufforderung) Befehlen. Beispiel: `Dos DEL C:\DATEI.TXT`. **HinweisWichtig:** `Dos` unterstützt nur **8.3** Pfade und Dateinamen. (Max. 8 Zeichen für den Dateinamen und 3 Zeichen für die Erweiterung). Diese Einschränkung kann umgangen werden mit `Start CMD`. Dies öffnet eine modernere DOS-Box die auch längere Pfade und Dateinamen unterstützt.

Hinweise: Für normale Datei-Befehle wie delete / rename / copy / create dir / delete dir sollten die eingebauten 'Maus und Tastatur Recorder' Befehle: `{{CopyFile}`, `{{DeleteFile}`, `{{RenameFile}`, `{{CreateDir}`, `{{CopyDir}`, `{{DeleteDir}` verwendet werden.

~~Siehe auch Befehl: `{{SendTextSpecial}`. Dieser Befehl erlaubt es direkt in DOS-Boxen hineinzuschreiben.~~

~~Der DOS-Befehl der für `{{Dos}` verwendet wird, enthält nur eine eingeschränkte Menge von DOS-Befehlen. Das folgende Beispiel zeigt wie man die volle Funktionalität der DOS-Box nutzen kann:~~

```
{{SendTextSpecial}
{{WindowsKey+r
cmd
{ENTER}
{{Delay 1000}
Echo {%}COMPUTERNAME{%} >> C:\Recorder.log
{ENTER}
{{exit
{ENTER}
```

~~Dieses Beispiel speichert den Computer Namen in die Datei: C:\Recorder.log. Das wäre nicht möglich mit dem Befehl `{{Dos}`.~~

1.10.1.8 `{{Switch}`

Dieser Befehl erlaubt das Umschalten zu einem anderen Programm. Für `{{Switch}` muss derjenige Name verwendet werden, welcher beim Drücken von ALT Tabulator erscheint. Es ist auch möglich nur einen Teil des Namens zu verwenden, solange er eindeutig ist. Beispiel: `{{Switch Microsoft}`. Manchmal ist auch noch ein Gedankenstrich notwendig, damit Windows das Fenster findet. (- Microsoft Word) Mit Probieren kann man herausfinden, was Windows akzeptiert.

Hinweise: Der `{{Switch}` Befehl entfernt Leerschläge vor und nach dem Programmnamen. Wenn diese Leerschläge benötigt werden, ist der Befehl `{{KeepSpace}` vor dem `{{Switch}` Befehl zu verwenden.

Schnellzugriffs-Taste:

`{{Switch}` ist ein Befehl, der recht oft benötigt wird. Daher bietet der 'Maus und Tastatur Recorder' eine Schnellzugriffs-Taste an, um solche `{{Switch}` Befehle rascher hinzuzufügen zu können. Vorgehen: Aufzeichnen aktivieren, zum gewünschten Fenster wechseln (ALT TAB) und dann die folgenden Tasten miteinander drücken: CTRL-ALT-SHIFT-F12 (Deutsche Tastatur: STRG-ALT-UMSCHALT-F12) und dann F12. Es ist möglich die Schnellzugriffs-Taste umzudefinieren und einen Fenster-Filter zu verwenden. Siehe Kapitel 'Optionen' für mehr Informationen.

Neue Umschalt-Technik:

Seit Version V5.0 wird eine neue Technik für das Umschalten verwendet. Diese sollte auch bei schwierigen Programmen besser funktionieren. Falls ein alter `{{Switch}` Befehl mit der neuen Version nicht mehr funktionieren sollte, kann mit dem Befehl: `{{UseOldSwitch}` für das aktuelle Makro wieder auf die alte Technik umgeschaltet werden. `{{UseNewSwitch}` erlaubt es die Umschaltung auf die alte Umschaltung aufzuheben.

Zweites Fenster:

Wenn der 'Maus und Tastatur Recorder' eine Applikation startet und diese Applikation öffnet dann ein weiteres Programmfenster, kann hin und wieder mit `{{Switch}` nicht mehr auf dieses Fenster umgeschaltet werden. In dieser Situation kann der Befehl `{{UseForeground}` helfen. (Für die Berechtigung, das Fenster hochzubringen wird dann die Berechtigung des Vordergrund-Fensters verwendet) Mit `{{UseRecorder}` kann man wieder den Originalzustand herstellen.

Falsches Fenster gefunden:

Wenn ein Fenster einen kurzen Namen hat, (z.B. Editor) kann es passieren, dass ein falsches Fenster welches auch diesen Namen enthält, aktiviert wird. (z.B. Microsoft Visual Basic Editor - MKR). Dies kann mit folgenden zwei zusätzlichen Befehlen verhindert werden: `<=` und `=>`. `<=Editor` bedeutet dass nur ein Fenster dessen Name mit Editor beginnt, aktiviert wird. `Editor=>` bedeutet, dass nur ein Fenster dessen Name mit Editor endet, aktiviert wird.

Namenloses Fenster:

Mittels `{{Switch #emptyX5#` kann auch zu einem namenlosen Fenster umgeschaltet werden.
(Bei `#emptyX5#` ist Gross-Kleinschreibung relevant)

Maus und Tastatur Recorder:

Ab V5.4 kann mit `{{Switch` nicht mehr direkt zum Recorder umgeschaltet werden. Wenn man den Recorder aktivieren will, muss man den symbolischen Namen: `#MKRX5#` verwenden. Beispiel:
`{{Switch #MKRX5#`

1.10.1.9 `{{SwitchOrStart`

Dieser Befehl versucht erst auf das Programm mit dem angegebenen Fenstername umzuschalten. Wenn das Fenster nicht existiert, wird das Programm gestartet. Beispiel: `{{SwitchOrStart Microsoft Word | C:\Programs\Office\Word.exe`. Wenn Word schon gestartet ist, wird zum Word Fenster umgeschaltet, wenn nicht, wird Word.exe gestartet.

Hinweise: `{{SwitchOrStart` kann nur Programme mit folgenden Endungen starten: .exe, .bat und .com.

Der `{{SwitchOrStart` Befehl entfernt Leerschläge vor und nach dem Programmnamen. Wenn diese Leerschläge benötigt werden, ist der Befehl `{{KeepSpace` vor dem `{{Switch` Befehl zu verwenden.

Siehe auch die Befehle `{{Switch` und `{{Start` für weitere Optionen.

1.10.1.10 `{{Loop}`

Dieser Befehl gibt die Schleifen-Wiederholungs-Nummer aus. Hinweis: Ein `{{Loop` ist immer ein völlig wiederholtes Makro. Die Wiederholungs-Nummer (wie oft das Makro wiederholt werden soll) kann im 'Maus und Tastatur Recorder' - Fenster im Feld 'Schleifen' eingestellt werden. Beispiel:

```
{{Switch Microsoft Word
This is loop number {{Loop}
```

Hinweise: Der Befehl `{{Loop` hat nicht die gleiche Funktion wie in VisualBasic. Die Anzahl Wiederholungen muss im Recorder-Feld 'Schleifen' eingestellt werden. Der Befehl `{{Loop` gibt nur die Wiederholungs-Zahl aus. (Bei der dritten Wiederholung eine 3)._

Statt Schleifen kann auch der Befehl `{{Goto` für die Wiederholung von ganzen Makros oder Teile davon verwendet werden.

1.10.1.11 `{{LockUI`

Der Befehl `{{LockUI` (Lock User Interface) erlaubt es Maus und Tastatur vorübergehend zu blockieren. Bei Makros, welche Mausbefehle verwenden, kann eine unbeabsichtigte Mausbewegung durch den Benutzer das Makro so stören, dass an einem falschen Ort geklickt wird. Anwendungs-Beispiel: `{{LockUI {{True}`: Tastatur und Maus werden blockiert. `{{LockUI {{False}`: Tastatur und Maus werden wieder freigegeben. Beim Beenden des Makros werden Tastatur und Maus ebenfalls freigegeben. Zusätzlich kann durch CTRL-Alt-Delete (STRG-Alt-Entfernen) die Tastatur und Maus Blockade beendet werden.

Wichtig: `{{LockUI` funktioniert bei Windows Vista und neuer nur wenn der Recorder als Administrator gestartet wird. Dies wird erreicht mittels Rechtsklick auf das Icon und dann "Als Administrator ausführen". In den Eigenschaften des Icons (Erweitert) kann auch eingestellt werden dass beim Doppelklicken auf das Icon automatisch als Administrator gestartet wird.

Hinweise: Um den Recorder per Windows Aufgabenplanung (Task Scheduler) als Administrator zu starten muss beim Reiter: Allgemein (General) die Option "Mit höchsten Privilegien ausführen" ("Run with highest privileges") aktiviert sein.

Wenn man den Recorder mittels Icon als Administrator (ohne Admin-Abfrage) starten will, kann man folgenden Trick anwenden: Einen Eintrag in die Aufgabenplanung eintragen mit Option "Mit höchsten Privilegien ausführen". Beispiel: Ordnername\Aufgabenname: MeineTasks\Recorder. Dann per Icon diesen Eintrag folgendermassen aufrufen: C:\Windows\System32\schtasks.exe /RUN /TN "MeineTasks\Recorder"

1.10.1.12 `SetLoop`

Dieser Befehl erlaubt es den aktuellen Schleifen-Zähler zu verändern. Beispiel `SetLoop 100`. Dadurch wird beim nächsten `Loop` Befehl (in der gleichen Schleife) 100 ausgegeben. Bei einer Schleifen-Endzahl von 150 stoppt das Makro nach 50 Durchläufen.

1.10.1.13 `Offset-100`

Dieser Befehl zählt den Versatz (Offset) Wert zur Wiederholungsnummer (Loop) hinzu. Beispiel: `Offset 100`: Während der ersten Wiederholung ergibt der Befehl `Loop` das Resultat 101. Bei einer Schleifenendzahl von 150 stoppt das Makro erst nach 150 Durchläufen. (`Loop` zeigt dann 250)

1.10.1.14 `1`

Befehle nach `1` werden nur während der ersten Wiederholung (Schleife) des Makros ausgeführt. Beispiel: `1SetVarCounter 1`. Der Zahlenspeicher 'Counter' wird nur beim ersten Durchlauf (Schleife) auf 1 gesetzt.

Hinweis: Der Befehl `Offset` beeinflusst die Grösse des Wertes 'Wiederholung'. Der Wert von `Offset` wird zum Wert Wiederholung hinzugezählt. Das bedeutet, dass ein Befehl nach `1` bei der Verwendung von z.B. `Offset 5` nie ausgeführt wird !

1.10.1.15 `8`

Befehle nach `8` werden nur während der achten Wiederholung (Schleife) des Makros ausgeführt. Beispiel: `8SetStringHinweis 8. Schleife erreicht`. Der Textspeicher 'Hinweis' wird nur beim 8. Durchlauf (Schleife) mit dem Text gefüllt.

Hinweis: Der Befehl `Offset` beeinflusst die Grösse des Wertes 'Wiederholung'. Der Wert von `Offset` wird zum Wert Wiederholung hinzugezählt. Wenn der `Offset` den Wert 5 hat, dann werden Befehle nach dem `8` während der 3. Wiederholung ausgeführt.

1.10.1.16 `<8`

Befehle nach `<8` werden nur in Wiederholungen (Schleifen) kleiner als 8 ausgeführt (1, 2, 3.. 7). Beispiel: `<8IncVarCounter 1`. Der Zahlenspeicher 'Counter' wird nur bei den ersten 7 Durchläufen (Schleifen) um je 1 erhöht.

Hinweis: Der Befehl `Offset` beeinflusst die Grösse des Wertes 'Wiederholung'. Beispiel: `Offset 5`: Befehle nach `<8` werden nur während den Wiederholungen 1 und 2 ausgeführt.

1.10.1.17 `>8`

Befehle nach `>8` werden nur in Wiederholungen (Schleife) grösser als 8 ausgeführt. (9, 10, 11, etc.). Beispiel: `>8IncVarCounter 1`. Der Zahlenspeicher 'Counter' wird erst nach den ersten 8. Durchläufen (Schleifen) um je 1 erhöht.

Hinweis: Der Befehl `Offset` beeinflusst die Grösse des Wertes 'Wiederholung'. Beispiel: `Offset 5`: Befehle nach `>8` werden nur während den Wiederholungen 4, 5, 6 etc. ausgeführt.

1.10.1.18 `n`

Befehle nach `n` werden nur in der letzten Wiederholung ausgeführt.).

Beispiel: `nMessageBox Letzter Durchlauf`. Der Text: 'Letzter Durchlauf' wird während der letzten Schleife angezeigt.

Hinweis: Der Befehl `Offset` beeinflusst die Grösse des Wertes 'Wiederholung' NICHT. Befehle nach `n` werden immer mit der letzten Wiederholung ausgeführt.

1.10.1.19 {{Beep}}

Dieser Befehl erzeugt einen Ton. `{{Beep}}` kann z.B. dazu benutzt werden das Ende eines Makros anzuzeigen.

1.10.1.20 {{MessageBox}}

Dieser Befehl zeigt dem Benutzer einen Text. Beispiel: `{{MessageBox BITTE CD EINLEGEN. {{CR}}CD DARF NICHT KOPIERGESCHUETZT SEIN.` Der Befehl `{{CR}}` erlaubt mehrere Zeilen. Wichtig: Nach dem Befehl `{{MessageBox}` muss ein `{{Switch}` programmiert werden, damit der Fokus wieder auf das gewünschte Programm umgeschaltet wird.

1.10.1.21 {{FlashBox}}

`{{FlashBox}` zeigt wie `MessageBox` dem Benutzer einen Text. Der Benutzer muss aber nicht OK klicken. Der Befehl erlaubt es einen Text (und optional Knöpfe) zu zeigen, den Text zu ändern und die Textbox mit dem Befehl wieder zu entfernen. Beispiele: `{{FlashBox Das ist mein erster Text | 1.` Die Textbox erscheint und zeigt den Text. `{{FlashBox Nun zeige ich einen zweiten Text | 1.` Die Textbox bleibt, aber der neue Text wird angezeigt. `{{FlashBox Dummy | 0.` Die Textbox verschwindet. (Für das Ausschalten der Box kann irgend ein Text verwendet werden. Im Beispiel verwenden wir 'Dummy')

Optional können weitere Parameter verwendet werden. Es sind dies: Box-Titel, einen OK-Knopf und einen Abbrechen-Knopf. Beispiel: `{{FlashBox Bitte Resultat prüfen. Ist es gut so ? | 1 | Abfrage | JA | NEIN.` Es erscheint nun eine Box mit dem Titel 'Abfrage', dem Text: 'Bitte Resultat prüfen. Ist es gut so?', einem Knopf JA und einem Knopf NEIN. Drückt der Benutzer den Knopf JA (den 1. Knopf) dann wird der Speicher `VarFlashBoxAnswerX55` auf `{{True}}` gesetzt. Bei NEIN (zweiter Knopf) auf `{{False}}`. Zudem wird beim drücken von JA oder NEIN der Speicher `VarFlashBoxEndedX55` auf `{{True}}` gesetzt. Wird weder JA noch NEIN gedrückt und die Flashbox beendet, dann enthält `VarFlashBoxAnswerX55` den Wert `{{True}}`.

Hier nun ein Beispiel wie dies verwendet werden könnte:

```

{{SetVarZaehler 0
{{FlashBox Bitte Resultat prüfen. Ist es gut so ? | 1 | Abfrage | JA | NEIN
{{Label Nochmals
{{Delay 1000
{{IncVarZaehler 1
{{If {{VarZaehler} >= 10 ;{{Goto Weiter;
{{If {{VarFlashBoxEndedX55} = {{True}} ;{{Goto Nochmals
{{Label Weiter
{{FlashBox Dummy | 0
{{If {{VarFlashBoxAnswerX55} = {{True}} ;{{MessageBox JA gedrückt oder 10 sec sind abgelaufen;
{{If {{VarFlashBoxAnswerX55} = {{False}} ;{{MessageBox NEIN gedrückt;

```

Der Erscheinungsort der `FlashBox` wird abgespeichert. Dadurch erscheint die Box immer am letzten Ort. Wenn die `FlashBox` ausserhalb des **primären** Bildschirms platziert wurde, wird sie automatisch in den Bildschirm geschoben. Ist diese Funktion nicht gewünscht (z.B. bei mehreren Bildschirmen) kann sie ausgeschaltet werden. Einfach folgende Datei starten:

www.kratronic.com/tools/recorder/sw/FlashBoxBorderOff.reg. (Wieder einschalten:

`FlashBoxBorderOn.reg`)

Hinweis: Wenn der Recorder ein anderes Programm gestartet hat, kann die Flashbox unter dessen Fenster erscheinen. In diesem Falle die Box ausserhalb des Fensters platzieren.

1.10.1.22 {{ShowInfo}}

`{{ShowInfo}` erlaubt es einen Text im Makro Beschreibungsfeld anzuzeigen (Reiter 'Allgemein'). Befehlsstruktur: `{{ShowInfo MeinText.` Der Original-Beschreibungsfeld Text wird nach dem Ende des Makros wieder hergestellt. Dieser Befehl kann benutzt werden um zusätzliche Status Informationen anzuzeigen wie z.B. Zähler Werte, Schleifenzähler bei Benutzung von `{{Goto}`, etc.

1.10.1.23 {{WaitForKey}}

Dieser Befehl erlaubt es auf irgend einen Tastendruck zu warten. Das Makro wartet, bis irgend eine Taste (auch z.B. Umschalttaste) gedrückt wird. Dann fährt das Makro weiter. Der Inhalt der gedrückten Taste wird im Speicher: `{{StringWaitForKeyX55}}` abgelegt.

Hinweis: Nur ein Tastendruck wird gespeichert z.B. a, ü, RIGHT, F3, ALT. Kombinationen wie ALT-a sind nicht möglich.

1.10.1.24 `{{Stop}}`

Dieser Befehl erlaubt es ein Makro zu stoppen. Der Befehl `{{Stop}}` zeigt ein Fenster mit folgenden Möglichkeiten: 'Stoppen', 'Weiterfahren', 'Weiterfahren mit nächstem Befehl', 'Weiterfahren mit nächster Schleifen-Wiederholung (Loop)' oder 'Weiterfahren mit der nächsten `{{Master}}` Befehl'.

Hinweise: Wenn man nach dem `{{Stop}}` weiterfahren möchte, muss man als nächsten Befehl ein `{{Switch}}` einfügen. Grund: `{{Stop}}` setzt den Fokus auf den Recorder.

Siehe auch Kapitel: Entwanzen (Debugging): (Notstopp mittels ALT Break, Pause: Shift Break)

Wichtig: Der `{{Stop}}` Befehl darf nicht verwendet werden, wenn zuvor der Befehl `{{Minimize}}` ausgeführt wurde. Sonst kann das Stop-Fenster auf bestimmten Betriebssystemen nicht mehr geschlossen werden.

1.10.1.25 `{{Exit}}`

Dieser Befehl beendet den 'Maus und Tastatur Recorder'. Wenn am Makro eine Änderung gemacht wurde, wird der Benutzer erst gefragt, ob das Makro gespeichert werden soll. Dieser Befehl ist besonders praktisch, wenn ein Makro über Befehlszeile mit einem Icon gestartet wird. `{{Exit}}` ermöglicht es dann, dass der Recorder sich nach getaner Arbeit wieder entfernt.

1.10.1.26 `{{Var}}`, `{{GlobVar}}`, `{{GlobVar1}}`

Diese Befehle erlauben es Zahlenspeicher zu verwenden. Zahlenspeicher müssen mit `{{SetVar}}` gefüllt werden, bevor man sie verwenden kann. Beispiel: `{{VarMeineVariableVar5}}`: Der Inhalt des Zahlenspeichers `"MeineVariable"5` wird ausgegeben. Die Anzahl Zahlenspeicher ist nicht begrenzt. Beispiel: `{{Var56345}}`. ~~Es ist auch möglich Zahlenspeicher mit Namen zu verwenden. Beispiel: `{{VarAnzahlMitglieder}}`.~~ (Gross-Kleinschreibung ist nicht relevant für Zahlenspeicher-Namen)

Ein nicht gefüllter Speicher enthält eine 0. Der Befehl `{{Var}}` / `{{GlobVar}}` (ohne Nummer oder Name) darf nicht mehr verwendet werden. (Inkompatibilität zu Versionen kleiner V5.0)

Hinweis: `{{Var}}`-Speicher werden zusammen mit dem Makro gespeichert. `{{GlobVar}}`-Speicher werden für alle Makros nur einmal gespeichert. Das bedeutet, dass man einen `{{GlobVar}}`-Speicher in allen Makros verwenden kann. Alle Speicher überleben einen PC-Neustart.

1.10.1.27 `{{SetVar}}`, `{{SetGlobVar}}`, `{{SetGlobVar1}}`

Diese Befehle füllen Zahlen in Zahlenspeicher. Beispiel: `{{SetVarMeineZahlSetVar15}}` 25: Zahlenspeicher `"MeineZahl"15` enthält nun die Zahl 25. Für Dezimalwerte muss ein Punkt verwendet werden (kein Komma). Mit diesem Befehl können nur Zahlen, keine Texte gespeichert werden. Die Anzahl Zahlenspeicher ist nicht begrenzt. Beispiel: `{{SetVar56345}}`. ~~Es ist auch möglich Zahlenspeicher mit Namen zu verwenden. Beispiel: `{{SetVarAnzahlMitglieder}}`.~~ (Gross-Kleinschreibung ist nicht relevant für Zahlenspeicher-Namen)

Der Befehl `{{SetVar}}` / `{{SetGlobVar}}` (ohne Nummer oder Name) darf nicht mehr verwendet werden. (Inkompatibilität zu Versionen kleiner V5.0)

Hinweise: `{{SetVar}}`-Speicher werden zusammen mit dem Makro gespeichert. `{{SetGlobVar}}`-Speicher werden für alle Makros nur einmal gespeichert. Das bedeutet, dass man einen `{{SetGlobVar}}`-Speicher in allen Makros verwenden kann. Alle Speicher überleben einen PC-Neustart.

Bestimmte Speichernamen sind reserviert für Befehle welche mehr als einen Wert zurückgeben wie z.B. `{{GetWindowPos}}`. Folgende Namen sollten nicht für eigene Speicher verwendet werden:
`{{VarGetWinCoord_X}`, `{{VarGetWinCoord_Y}`, `{{VarGetWinCoord_X_End}`, `{{VarGetWinCoord_Y_End}`,
`{{VarGetCursorCoord_X}`, `{{VarGetCursorCoord_Y}`, `{{VarFindPatternCoord_X_...`
`{{VarFindPatternCoord_Y_...`, `{{VarFindPattern_Numbers}`, `{{VarFlashBoxEndedX55}`,
`{{VarFlashBoxAnswerX55}`, ~~`{{VarFlashBoxAnswerX55}`~~, `{{StringWaitForKeyX55}`,
`{{VarScanDirectoriesX55}`.

1.10.1.28 *Inc/Dec/Mul/Div - Befehle*

~~`{{IncVar}`, `{{IncGlobVar}`~~
~~`{{DecVar}`, `{{DecGlobVar}`~~
~~`{{MulVar}`, `{{MulGlobVar}`~~
~~`{{DivVar}`, `{{DivGlobVar}`~~
~~`{{IncVar1}`, `{{IncGlobVar1}`~~
~~`{{DecVar1}`, `{{DecGlobVar1}`~~
~~`{{MulVar1}`, `{{MulGlobVar1}`~~
~~`{{DivVar1}`, `{{DivGlobVar1}`~~

Diese Befehle erlauben es, Speicher zu verändern. Beispiele:

`{{IncVarZaehlerIncVar1} 5`: Zählt zum Speicher "Zaehler"⁴ eine fünf hinzu
`{{DecGlobVarZaehlerDecGlobVar1} 5`: Zählt vom globalen Speicher "Zaehler"⁴ eine fünf ab
`{{MulVar1} 10`: Multipliziert Speicher 1 mit 10
`{{DivVarAnzahlSchüler} 10`: Dividiert Speicher 'AnzahlSchüler' mit 10. (Real-Division)
 Siehe auch Befehl: `{{Calc}}`.

Hinweis: Die obigen Befehle arbeiten mit einem Punkt (.) als Dezimalpunkt, auch wenn die Regional-Einstellungen ein anderes Zeichen definieren. Wenn nötig muss mittels `{{Replace der Dezimalpunkt angepasst werden}`.

`{{String}`, `{{GlobString}`

1.10.1.29 *~~{{String1}~~, ~~{{GlobString1}~~*

Dieser Befehl erlaubt es Text-Speicher zu verwenden. Text-Speicher müssen mit `{{SetString}` gefüllt werden, bevor man sie verwenden kann. Beispiel: `{{StringMeinTextString5}`: Der Inhalt des Text-Speichers "MeinText"⁵ wird ausgegeben. Die Anzahl Text-Speicher ist nicht begrenzt. Beispiel: `{{String56345}`. ~~Es ist auch möglich Text Speicher mit Namen zu verwenden. Beispiel: `{{StringMitgliederName}`.~~ (Gross-Kleinschreibung ist nicht relevant für Textspeicher-Namen). Ein nicht gefüllter Speicher enthält eine "0"⁰.

Der Befehl `{{String}` / `{{GlobString}` (ohne Nummer oder Name) darf nicht mehr verwendet werden. (Inkompatibilität zu Versionen kleiner V5.0)

Hinweise: `{{String}`-Speicher werden zusammen mit dem Makro gespeichert. `{{GlobString}`-Speicher werden für alle Makros nur einmal gespeichert. Das bedeutet, dass man einen `{{GlobString}`-Speicher in allen Makros verwenden kann. Alle Speicher überleben einen PC-Neustart.

Wenn man grössere Mengen von Text speichert/abrufen kann es bei der Ausgabe (`{{String..}`) zu Fehlern kommen. Lösung: `{{SendTextSpecial}` verwenden oder ~~noch besser~~ den Text in den Zwischenspeicher (Clipboard) kopieren und mittels CTRL/STRG v ausgeben.

1.10.1.30 *~~{{SetString}~~, ~~{{SetGlobString}~~* ~~*`{{SetString1}`, `{{SetGlobString1}`*~~

Dieser Befehl speichert Texte in Text-Speicher. Beispiel: `{{SetStringMeinGlobalerTextSetString15}` MEIN TEXT. Der Text-Speicher "MeinGlobalerText"¹⁵ enthält nun 'MEIN TEXT'. ~~Es ist auch möglich Text Speicher mit Namen zu verwenden. Beispiel: `{{SetStringMitgliederName}`.~~ (Gross-Kleinschreibung ist nicht relevant für Textspeicher-Namen). Zeilenumbrüche können eingefügt werden mit den Befehlen: `{{CR}` und `{{LF}`. Beispiel: `{{SetString1} Zeile eins{{CR}}{{LF}}Zeile zwei`. Hinweise: Je nach Ausgabeart ist ev. nur ein `{{CR}` notwendig. Ein Leertext ("") kann mit der folgenden Konstante erzeugt werden: `#emptyX5#`. Beispiel: `{{SetString1} #emptyX5#`. Ein einzelner Leerschlag (" ") kann mit der

folgenden Konstante erzeugt werden: #spaceX5#. Beispiel: `{{SetString1} #spaceX5#`.
(Bei #emptyX5# und #spaceX5# ist Gross-Kleinschreibung relevant)

Hinweise: Der Befehl `{{SetString}` / `{{SetGlobString}` (ohne Nummer oder Name) darf nicht mehr verwendet werden. (Inkompatibilität zu Versionen kleiner V5.0)

`{{String-Speicher}` werden zusammen mit dem Makro gespeichert. `{{GlobString-Speicher}` werden für alle Makros nur einmal gespeichert. Das bedeutet, dass man einen `{{GlobString-Speicher}` in allen Makros verwenden kann. Alle Speicher überleben einen PC-Neustart.

Bestimmte Speichernamen sind reserviert für Befehle welche einen Wert im Text-Speicher zurückgeben wie z.B. `{{WaitForKey}`. . Folgender Name sollte nicht für eigene Speicher verwendet werden: `{{StringWaitForKeyX55}`. Im Weiteren darf der Ask-Speichernamen kein = (Gleichheitszeichen) enthalten.

1.10.1.31 ~~`{{Ask}`, `{{GlobAsk}`, `{{WriteAsk}`, `{{WriteGlobAsk}}`~~`{{Ask1}`, `{{GlobAsk1}`, `{{WriteAsk1}`, `{{WriteGlobAsk1}`~~~~

`{{Ask1}` fragt nach einer Eingabe. Das heisst `{{Ask}` fragt den Benutzer während des Makroablaufes nach einem Text. Befehlsaufbau: `{{AskName} Angezeigter_Text |Vorgabewert`. Beispiel: `{{AskMeineFrageAsk3} NAME {{CR}} VORNAME NICHT VERGESSEN |Peter Müller`. Der Befehl `{{CR}}` erlaubt es mehrzeilige Fragen zu stellen. Der Text nach dem | wird als Vorgabe in das "`{{Ask-Fenster}`" eingetragen, und kann auch weggelassen werden. Der mit `{{Ask}` erhaltene Text kann später mit `{{WriteAsk}` wieder verwendet werden. Beispiel: `{{If {{WriteAskMeineFrageWriteAsk1} = JA;{{Goto Ende`

Die Anzahl `{{Ask-Speicher}` ist nicht begrenzt. Beispiel: `{{Ask56345}`. ~~Es ist auch möglich Ask-Speicher mit Namen zu verwenden. Beispiel: `{{AskMitgliederName}`.~~ (Gross-Kleinschreibung ist nicht relevant für Ask-Speicher-Namen).

Ein nicht gefüllter `{{Ask-Speicher}` enthält eine `"0"`. `'0'`.

Hinweis: `{{Ask - Fragespeicher}` werden zusammen mit dem Makro abgespeichert. . `{{GlobAsk-Speicher}` werden für alle Makros nur einmal gespeichert. Das bedeutet, dass man einen `{{GlobAsk-Speicher}` in allen Makros verwenden kann.

NumLock wird automatisch eingeschaltet während der Abfrage. (Wenn es vor dem Start des Macros schon eingeschaltet war.

Wichtig: Nach einer Abfrage muss ein `{{Switch}` eingefügt werden, damit der Fokus wieder zum Programm zurückgeschaltet wird.

Der Ask-Speichernamen darf kein = (Gleichheitszeichen) enthalten.

1.10.1.32 `{{SetClipboard}`, `{{Clipboard}`

`{{SetClipboard}` schreibt Text vom Makro in die Zwischenablage. Beispiel: `{{SetClipboard MEINTEXT{{CR}}{{LF}}MEINE ZWEITE ZEILE`. `{{SetClipboard}` wird benötigt um Sonderzeichen, erzeugt mit `{{RandomString}`, auszugeben. ~~Im weiteren kann `{{SetClipboard}` auch für das Schreiben von Befehlen in die DOS-Box verwendet werden. (ALT Leerzeichen | Bearbeiten | Einfügen).~~

`{{Clipboard}` hat fast die gleiche Wirkung wie Ctrl v (Strg v). Ausnahme: Wenn die Zwischenablage die Zeichen: +, ^, %, ~, (,), [,] enthält, dann sollte Ctrl v benutzt werden.

`{{Clipboard}` wandelt die Zeichen {, } um in {{} und }}. Wenn das nicht gewünscht ist kann der Befehl `{{Clipboard0}` (ClipboardNull) verwendet werden. `{{Clipboard}` kann auch verwendet werden um Text aus der Zwischenablage in einen Text-Speicher zu kopieren. Beispiel: `{{SetString1} {{Clipboard}`.

Hinweis: Das mehrfache Lesen der Zwischenablage führt auf einzelnen Rechnern zu Problemen z.B. Runtime Error 524 (internes Windwos Problem). Dies kann umgangen werden, indem der Inhalt der Zwischenablage erst in einen Textspeicher gespeichert wird und dann dieser Textspeicher für weitere Zugriffe verwendet wird. Manchmal verhindert auch ein kurzes Delay zwischen den Befehlen dieses Problem.

1.10.1.33 *DownLoop, UpLoop, RightLoop, LeftLoop*

Diese Befehle bewegen den Positionszeiger abhängig von der Anzahl Makro-Wiederholungen. Wenn die 5. Makrowiederholung ausgeführt wird, bewegt sich der Positionszeiger 5 Zeilen nach unten. (oder oben, links, rechts) Eine Zahl nach *DownLoop* wird mit der Anzahl Makro-Wiederholungen multipliziert. Beispiel: *UpLoop 5*. Wenn die 7. Makrowiederholung läuft, bewegt sich der Positionszeiger 35 Zeilen nach oben. **Einschränkung: Max. 8000 Bewegungen sind möglich in einem Befehl.**

Down = nach unten
Up = nach oben
Right = nach rechts
Left = nach links

1.10.1.34 *DownVar, UpVar, RightVar, LeftVar*

Diese Befehle bewegen den Positionszeiger nach unten (oder oben, links, rechts) abhängig vom Inhalt des Zahlenspeichers oder einer einzelnen Zahl. Beispiele: *DownVar {Var1}* oder *DownVar {RandomNumber 1,10}* oder *DownVar 15*. Beim letzten Beispiel wird der Positionszeiger 15 Zeilen nach unten bewegt. **Einschränkung: Max. 8000 Bewegungen sind möglich in einem Befehl.**

Down = nach unten
Up = nach oben
Right = nach rechts
Left = nach links

1.10.1.35 *DateTime*

Dieser Befehl gibt einen Text aus, welcher das Datum und die Zeit enthält. Beispiel: ~~2010~~2002-12-13-23-51-12. Dieser Befehl kann verwendet werden wenn während eines Makros viele Dateien abgespeichert werden müssen. Jede Sekunde erhält man so einen anderen Dateinamen.

1.10.1.36 *Date*

Dieser Befehl gibt einen Text aus, welcher das Datum enthält. Beispiel: ~~2010~~2002-12-13

1.10.1.37 *Time*

Dieser Befehl gibt einen Text aus, welcher die Zeit enthält. Beispiel: 23-45-59

1.10.1.38 *SerialTime*

Der Befehl *SerialTime* gibt eine Nummer zurück, welche der Summe aller Sekunden seit dem 1. Jan. 1970 entspricht. Beispiel für 20. Aug. 2011, 17:40: 1313862025~~12. Feb. 2005, 22:10: 1108246222~~.

Dieser Befehl kann benutzt werden um die Zeit zwischen Befehlen zu messen. Ein Beispiel befindet sich im Kapitel: Beispiel-Makros.

Wichtig: Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehl darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: *SetVarStartTime {SerialTime}*

1.10.1.39 *CreateSerialTime*

Der Befehl *CreateSerialTime* gibt eine Nummer zurück, welche der Summe aller Sekunden seit dem 1. Jan. 1970 und einem gegebenen Datum/Zeit entspricht. Beispiel *CreateSerialTime 2011,12,08,18,00,00*. Resultat: 1323367200. Dieser Befehl kann benutzt werden um zu einer gewissen Zeit eine Aktion anzustossen.

Wichtig: Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehl darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{SetVarStartTime {{CreateSerialTime ...}}`

1.10.1.40 `{{AddSerialTime`

Dieser Befehl erlaubt es Jahre, Monate, Tage, Stunden, Minuten und Sekunden zu der aktuellen SerialTime hinzuzuzählen. Beispiel: `{{AddSerialTime 0001,03,04,05,15,45}`. Dieser Befehl zählt alle Sekunden für 1 Jahr, 3 Monate, 4 Tage, 5 Stunden, 15 Minuten und 45 Sekunden zur aktuellen SerialTime hinzu. Resultat: Wenn der aktuelle DateTime Wert 1323550505 beträgt, dann ergibt dies: 1363313450.

Wichtig: Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehl darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{SetVarNewTime {{AddSerialTime ...}}`

1.10.1.41 `{{PCIdle}`

Der Befehl `{{PCIdle}` misst die PC-Leerlauf-Zeit in Sekunden. Die PC-Leerlauf-Zeit ist die Zeit seit dem letzten Maus Klick oder der letzten Tastatur-Betätigung. Wichtig: Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehl darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{SetVarIdleTime {{PCIdle}}`

1.10.1.42 `{{Delay}`, `{{ExtDelay}`

Der Befehl `{{Delay}` kann mittels dem Icon Sanduhr oder mit dem Menu: 'Einfügen' Wartezeit' eingefügt werden. Der Vorgabewert für `{{Delay}` kann in Menu 'Extras', 'Optionen', 'Bearbeiten', 'Einfüge Wartezeit' geändert werden. Die Einheit für `{{Delay}` ist Millisekunden. Beispiel: `{{Delay 2000}`: Das Makro wartet 2 Sekunden und läuft dann weiter.

Der Befehl `{{ExtDelay}` erlaubt es grosse Verzögerungen einfach anzugeben. Befehlsaufbau: `{{ExtDelay Tage:Stunden:Minuten:Sekunden}`. Beispiel: `{{ExtDelay 10:23:59:40}`. Dies ergibt folgende Verzögerung: 10 Tage, 23 Std., 59 min. und 40 sec. Anderes Beispiel: `{{ExtDelay 10:15}` = 10 min., 15 sec. `{{ExtDelay 15}` = 15 sec. Maximale Verzögerung: 3650 Tage.

Hinweis: In Windows 98 und ME hat die Verzögerung eine Auflösung von 55 ms. Das bedeutet, dass eine Einstellung von 10 ms eine Verzögerung von 55 ms auslöst. (60 ms ergeben 110 ms). Die moderneren Betriebssysteme haben eine Auflösung von 1 ms.

1.10.1.43 `{{WaitForTime`

Wenn man wie mit einem Zeitschaltprogramm zu genau einer gegebenen Zeit ein Makro-Teil weiterlaufen lassen will, kann man den Befehl `{{WaitForTime}` verwenden. Befehlsstruktur: `{{WaitForTime Tage:Stunde:Minute: Sekunde}`. Beispiel: `{{WaitForTime 2:17:15:30}`. Das heisst die Befehle nach `{{WaitForTime}` werden in zwei Tagen um 17:15 und 30 Sekunden ausgeführt. Ein anderes Beispiel: `{{WaitForTime 20:30:10}` Heute um 20:30 und 10 Sekunden. (Teile die man weglässt (hier Tage) werden als 0 angenommen). Hinweis: Tage wechseln um Mitternacht.

Beim den obigen Beispielen kann die Zeit nicht in der Vergangenheit liegen. z.B heute um 14:00 Uhr erzeugt ein Befehl `{{WaitForTime 07:30:10}` einen Fehler (Zielzeit liegt in der Vergangenheit). Mit einem X bei Tage: kann man das umgehen. Beispiel: `{{WaitForTime X:00:15:30}`. Nun wartet der Recorder bis zur nächsten (oder gleichen) Stunde um 15 Min. 30 Sec. Andere Beispiele: `{{WaitForTime X:00:00:10}`. Der Recorder wartet bis zur nächsten (oder gleichen) Minute wenn die Uhr den Sekundenwert von 10 hat. `{{WaitForTime X:14:15:00}`: Der Recorder wartet auf 14 Uhr 15. (Heute oder morgen je nachdem was eher kommt)

1.10.1.44 `{{RandomNumber`

Dieser Befehl erzeugt Zufallsnummern. `{{RandomNumber}}` benötigt zwei Parameter: Untere Grenze und Obere Grenze. Die Zufallsnummer liegt zwischen den beiden Grenzen (inklusive die Grenznummern). Beispiel: `{{RandomNumber 100, 200}}`. Mögliches Resultat: 126.

1.10.1.45 `{{RandomString}}`

Dieser Befehl erzeugt einen Zufallstext. Befehlsaufbau: `{{RandomString Anzahl_Zeichen, Start_Zeichen, End_Zeichen}}`. Der Text wird 'Anzahl Zeichen' lang und enthält die Zeichen zwischen Start- und Ende-Zeichen. (Incl. Grenzzeichen). Beispiel: `{{RandomString 12, a, z}}`. Mögliches Resultat: rhnqlkddcyhu. Um grosse und kleine Buchstaben zu erhalten können die Parameter az,a verwendet werden. Beispiel: `{{RandomString 12, az, a}}` erzeugt 12 Zeichen von A..z. (Nur Gross- und Kleinbuchstaben). Für Vokal-dominante Zufallstexte kann man den Parameter a2 verwenden. Beispiel: `{{RandomString 12, a2, a}}` erzeugt 12 Zeichen von A..z. (Nur Gross- und Kleinbuchstaben), wobei jeder zweite Bustabe ein Vokal (Grossbuchstabe) ist: kUdAhOrUkUdU.

Wichtig: Enthält der Zufallstext Sonderzeichen, muss der Text mittels Zwischenablage `{{SetClipboard}}` `{{RandomString 20, !, ~}}` und Ctrl v (Strg v) ausgegeben werden. Siehe auch Kapitel: Hinweise / Einschränkungen: ([BracesReplace](#)). Es ist auch möglich `{{SendTextSpecial+}}` zu verwenden[ReplaceBraces](#).

Hinweise: Der Befehl `{{RandomString}}` arbeitet mit der ASCII table. Das bedeutet, dass ausserhalb 0..9, A..Z and a..z folgende Zeichen generiert werden können: `!..0 = !"#%&'()*+,-./0 9..A = 9; <=> ?@A Z..a = Z[\]^_`az..~ = z{ } ~`. Leerschläge (und die Konstante `#spaceX5#`) können nicht verwendet werden für den `{{RandomString}}` Befehl.

1.10.1.46 `{{ImportFile, {{ImportSeparator, {{DoImport, {{ImportNextLine}}`

Diese Befehle erlauben die Verwendung von Listen in einer Datei. Siehe Kapitel 'Datei Import' für mehr Informationen.

1.10.1.47 `{{If}}`

Dieser Befehl erlaubt es Zahlen und Texte zu vergleichen. Befehlsformat: `{{If Ausdruck Kondition Ausdruck; Wahr-Aktion; Falsch-Aktion}}`. Beispiel: `{{If {{VarAuswahl} = 5; {{Switch Microsoft Word; {{Switch Microsoft Excel}}`. Resultat: Wenn der Zahlenspeicher 'Auswahl' eine 5 enthält wird zu Word umgeschaltet, wenn der Zahlenspeicher 'Auswahl' eine andere Zahl enthält wird zu Excel umgeschaltet.

Der Befehl `{{If}}` erlaubt die Verwendung der folgenden Konditionen: =, <, >, <=, >=, <>, ||, &&. =: gleich, <: kleiner als, >: grösser als, <=: kleiner oder gleich >=: grösser oder gleich, <>: ungleich, ||: oder, &&: und.

Wichtig: Die Kondition muss immer von Leerschlägen umgeben sein. `{{If 1 = 2 ;... Richtig. Falsch wäre: {{If 1=2 ;...}} und && können nur für logische Vergleiche benutzt werden: {{True, {{False}}. (Siehe auch- Beispiel: {{If {{True} && {{False} ; {{MessageBox Beide Ausdrücke sind wahr;}}`

Seit Version 7 sind mehrere Befehle in den Aktionen möglich. Das Trennzeichen dafür ist: \$. Beispiel: `{{If {{Var1 = {{True} ; {{SetVar10 15$ {{SetVarEntscheid {{True}$ {{Beep}; SetVar10 0$ {{SetVarEntscheid {{False}$ {{Goto Ende}}`

Wichtig:

- Die Ausdrücke dürfen keine Leerschläge enthalten: Beispiel (FALSCH): `{{If {{String1} = Mein Text ;...}` Ausdrücke mit Leerschlägen müssen in Speicher gelegt werden und so verwendet werden: (RICHTIG): `{{If {{String1} = {{String2} ;...}` Die gleiche Lösung (`{{String2}`) muss für Texte, welche folgende Zeichen enthalten, angewendet werden: =, <, >, <=, >=, <>, ||, &&.
- Der Text zwischen den ';' wird genau so verwendet wie er dort geschrieben steht. Beispiel ...; JA; NEIN. Resultat: Die Leerschläge vor JA und NEIN werden auch verwendet ! Bei z.B. ; `{{Beep}`; führt dieser Leerschlag zu einem Fehler im Makroablauf.

~~In den Wahr- und Falsch-Aktionen darf nur jeweils ein Befehl verwendet werden. Das folgende Beispiel ist FALSCH: {{If {{String1} = Test ;{{SetString2 Test2}}{ENTER}};. Sind mehr als ein Befehl nötig, muss {{Goto verwendet werden.~~

- Die folgenden Befehle dürfen nicht verwendet werden in den Wahr- und Falsch-Aktionen: {{If, {{',', {{'1', {{>1}, {{<1}, {{n}, {{Master, {{Label. Bei {{GoSub muss berücksichtigt werden, dass mit dem {{Return} der Subroutine das Makro mit der Zeile nach dem {{If weiterfährt (und nicht mit dem nächsten Befehl in der Wahr- oder Falsch-Aktion.)

Hinweise:

- Wenn das Zeichen ';' in den Wahr oder Falsch-Aktionen verwendet werden muss, kann dieses Zeichen umdefiniert werden. Beispiel:

```
{{IfSep, {{GlobString1} <> {{String1},Text: Name; Adresse; Stadt = falsch,OK.
```

 In diesem Beispiel wird nun ein Komma als Separator verwendet und somit kann der Strichpunkt in der Wahr-Aktion verwendet werden. Wichtig: Zwischen IfSep und dem Separator kein Leerzeichen einfügen.
- Ist eine Wahr-Aktion oder Falsch-Aktion nicht nötig, kann sie einfach leer gelassen werden. Beispiele: Keine Falsch-Aktion:

```
{{If {{WriteAsk1} = x ;{{Ask1 Gültiger Text eingeben! Ein x ist nicht erlaubt.;
```

 Keine Wahr-Aktion:

```
{{If {{VarAntwort} = JA ;{{Goto NoPart
```
- Der Befehl {{If kann Zahlen und Texte vergleichen. Sind beide Ausdrücke Zahlen, wird ein Zahlenvergleich gemacht. Beispiel: 071 > 17. Resultat: Wahr. Sind einer oder beide Ausdrücke Texte, dann wird ein Textvergleich gemacht. Beispiel: 071 > 17Euro. Resultat: Falsch.
- Für einige Befehle kann das Schlüsselwort **{{ShowError}** benutzt werden. Beispiel:

```
{{If {{FindFile C:\MyFile.txt} = {{True}};{{ShowError}}.
```

 Wenn der Recorder die Datei: C:\MyFile.txt findet, wird die nächste Makrozeile ausgeführt. Wird die Datei nicht gefunden, gibt der Recorder einen Fehlertext aus und das Makro wird abgebrochen. **{{ShowError}** kann für folgende Befehle verwendet werden:

```
{{FindFile, {{CompareFiles, {{CheckFocus, {{WaitForFocus, {{WaitForEndFocus, {{SetIEFocus, {{WaitForIEReady, {{WaitForScreenChange, {{ImportNextLine, {{FindPattern. Hinweis: Wenn {{ShowError} verwendet werden soll, muss {{ShowError} alleine stehen (nicht mehrere Befehle im Aktionsteil).

```
- Es ist möglich die beiden Ausdrücke **{{True}** und **{{False}** in einen Zahlenspeicher zu speichern. Beispiel:

```
{{SetVar1 {{True} ...und später im Makro... {{if {{Var1} = {{True} ;{{Beep;
```
- Ein leerer Text ("") kann mit folgender Konstante geprüft werden: #emptyX5#. (Bei #emptyX5# ist Gross-Kleinschreibung relevant)
 Beispiel:

```
{{If {{Var1} = #emptyX5# ;{{Beep;
```
- Ein einzelner Leerschlag (" ") kann mit folgender Konstante geprüft werden: #spaceX5#. (Bei #spaceX5# ist Gross-Kleinschreibung relevant)
 Beispiel:

```
{{If {{Var1} = #spaceX5# ;{{Beep;
```

1.10.1.48 **{{True}, {{False}**

Diese beiden Befehle werden bei Vergleichen verwendet z.B.

```
{{If, {{CompareFiles, {{CheckWindow,
```

 etc.). **{{True}** (bedeutet Wahr) wird repräsentiert durch eine eins (1), **{{False}** (bedeutet: Nicht Wahr) wird repräsentiert durch eine Null (0).

1.10.1.49 **{{FindFile**

Dieser Befehl sucht nach einer Datei. Beispiel:

```
{{FindFile C:\MEINTEXT.TXT. {{FindFile}
```

 gibt eine 1 zurück, wenn die Datei gefunden wurde, sonst eine 0. Normalerweise wird **{{FindFile}** zusammen mit **{{If}** verwendet. Beispiel:

```
{{if {{FindFile C:\Abc.exe} = {{True}};{{Start C:\Abc.exe}};{{goto Error.
```

 Als Aktion kann auch der Befehl **{{ShowError}** benutzt werden. Dieser zeigt im Fehlerfall einen Fehlertext an und das Makro endet. **{{FindFile}** erlaubt die Verwendung von Jokern (z.B. *.txt). Es wird dann die erste passendeTXT- Datei gefunden (Nicht unbedingt nach abc*)

1.10.1.50 **{{CompareFiles**

Dieser Befehl vergleicht Dateien. Wenn beide Dateien gleich sind wird eine '1' zurückgegeben, sonst eine 0. Normalerweise wird **{{CompareFiles}** zusammen mit **{{If}** verwendet. Beispiel:

```
{{If {{CompareFiles
```

`C:\DATEI1.TXT | C:\DATEI2.Txt | 10000} = {{False}} ;{{DeleteFileDeleteFile C:\DATEI2.TXT}; Resultat: Wenn die Dateien unterschiedlich sind, wird Datei2 gelöscht. Im Beispiel werden die ersten 10'000 Byte verglichen. Sind die Dateien kleiner als 10'000 Byte werden die ganzen Dateien verglichen. Als Aktion kann auch der Befehl {{ShowError}} benutzt werden. Dieser zeigt im Fehlerfall einen Fehlertext an und das Makro endet.`

1.10.1.51 `{{ScanDirectories}}`

Dieser Befehl durchsucht ein Verzeichnis (und Unterverzeichnisse). Das Resultat dieser Suche (gefundene Dateien) wird in einer Text-Datei abgespeichert. Befehlsaufbau: `{{ScanDirectories Durchsuchter_Pfad | Suchfilter | Unterverzeichnisse_einbeziehen | Resultat_Datei_Name | Mehrere_Teile_in_der_Resultat_Datei}}`. Der letzte Teil des Befehles (Mehrere Teile in der Resultat Datei) ist optional, wenn er weggelassen wird, wird `{{False}}` verwendet.

Beispiel: `{{ScanDirectories C:\MeineDaten | *.Doc | {{True}} | C:\Temp\DirList.txt | {{True}}`. In diesem Beispiel wird im Verzeichnis: C:\MeineDaten nach allen Doc-Dateien gesucht. Der Befehl sucht auch nach Doc-Dateien in den Unterverzeichnissen (`{{True}}`). Die gefundenen Dateien (Pfad und Dateiname) werden in der Datei: C:\Temp\DirList.txt abgespeichert. Aufgrund des letzten `{{True}}` im Befehl wird nicht nur der Dateiname (mit Pfad) abgespeichert, sondern die folgenden Teile: Pfad und Datei mit Erweiterung, Datei mit Erweiterung, Datei ohne Erweiterung, Erweiterung (z.B. .Doc), Pfad ohne Datei. Die Teile in der Resultat Datei sind mit dem Separator > getrennt. Das bedeutet, dass für das Einlesen der Teile aus der Resultat Datei mittels `{{DoImport}}`, der Befehl `{{ImportSeparator >}}` verwendet werden muss. Dieser Separator ist auch nötig, wenn die Resultat Datei nur Pfad und Datei enthält. Grund: Ein Dateiname kann ein Komma enthalten und das wäre dann der Standard Separator für `{{DoImport}}`. Die Anzahl durch `{{ScanDirectories}}` gefundener Dateien wird im Zahlenspeicher `{{VarScanDirectoriesX55}}` abgelegt.

Der Suchfilter kann folgende Zeichen für das Filtern der Dateien enthalten: *, ?, #, !, [und].

Art des Vergleichs	Muster	Passend	Nicht passend
Mehrere Zeichen	a*a	aa, aBa, aBBBa	aBC
	ab	abc, AABB, Xab	aZb, bac
Spezial Charakter	a[*]a	a*a	aaa
Mehrere Zeichen	ab*	abcdefg, abc	cab, aab
Einzelne Zeichen	a?a	aaa, a3a, aBa	aBBBa
Einzelne Zahlen	a#a	a0a, a1a, a2a	aaa, a10a
Zeichen-Bereich	[a-z]	f, p, j	2, &
Ausserhalb Bereich	[!a-z]	9, &, %	b, a
Keine Zahl	[!0-9]	A, a, &, ~	0, 1, 9
Kombiniert	a[!b-m]#	An9, az0, a99	abc, aj

1.10.1.52 `{{CreateDir}}`

Dieser Befehl erzeugt ein Verzeichnis. Existiert das Verzeichnis schon, oder ist der Pfad ungültig, wird nichts gemacht. Es kann ein einzelnes Verzeichnis, oder auch ein ganzer Baum erzeugt werden. Beispiel: `{{CreateDir C:\MeinVerz\MeinTieferesVerz\MeinTiefstesVerz}}`

1.10.1.53 `{{CopyDir}}`

Dieser Befehl kopiert ein Verzeichnis. Beispiel: `{{CopyDir C:\MeinVerz1 | C:\MeinVerz2}}`. Existiert das Zielverzeichnis noch nicht, wird es erstellt. Bereits im Zielverzeichnis existierende Dateien werden überschrieben. Möchte man nur Dateien kopiert haben, welche im Zielverzeichnis noch nicht existieren, sieht der Befehl so aus: `{{CopyDir C:\MeinVerz1 | C:\MeinVerz2 | {{False}}`. Hinweis: Das ganze Verzeichnis mit Unterverzeichnissen wird kopiert.

`{{CcopyDir}}` copies a directory. Example: `{{CopyDir C:\MyDir1 | C:\MyDir2}}`. If the target directory not yet is existing, it will be created. Files already existing in the target directory are overwritten. If just files shall be copied which are not existing in the target directory then the following syntax can be used: `{{CopyDir`

C:\MyDir1 | C:\MyDir2 | {{False}}. Hint: The whole directory with subdirectories will be copied.

1.10.1.54 **{}DeleteDir**

Dieser Befehl löscht ein Verzeichnis. Beispiel: {}DeleteDir C:\MeinVerz. Existiert das Verzeichnis nicht, oder ist der Pfad ungültig wird nichts gemacht. Normalerweise wird das

Wichtig: Das Verzeichnis wird nur gelöscht, wenn es leer ist. Will man ein nicht-leeres Verzeichnis löschen, kann folgender Befehl verwendet werden: {}DeleteDir C:\MeinVerz | {{True}}. Wenn das Verzeichnis schreibgeschützte oder blockierte Dateien enthält, wird das Verzeichnis und diese Dateien nicht gelöscht. Mit einem Trick kann man den Grund des Nichtlöschens anzeigen lassen: {}DeleteDir C:\MeinVerz | 2 entspricht {}DeleteDir C:\MeinVerz mit Fehleranzeige, {}DeleteDir C:\MeinVerz | 3 entspricht {}DeleteDir C:\MeinVerz | {{True}} mit Fehleranzeige!

1.10.1.55 **{}CopyFile**

Dieser Befehl kopiert eine Datei und/oder benennt sie um. Befehlsbeschreibung: {}CopyFile QuellenDatei | ZielDatei | Joker: Ja/Nein. Beispiel: {}CopyFile C:\Eigene Dateien\Info.txt | C:\Privat\Hinweis.txt | {{False}}. Bei Joker: {{False}} ist immer ein Pfad und ein ganzer Dateiname nötig. (Keine Abkürzungen *). Bei Joker: {{True}} können Joker (Wildcards) verwendet werden. Wichtig: Wenn Joker verwendet werden, muss auch in der Zieldatei ein *.* verwendet werden (Ansonsten werden alle gefundenen Quelldateien auf die eine ZielDatei kopiert.) Vereinfachung: Bei * und ? in der Quelldatei wird automatisch ein *.* in der Zieldatei verwendet (Nicht aber bei z.B. #). Der Parameter Joker ist optional. Wird er weggelassen wird {}False verwendet.

Die Quelldatei kann folgende Zeichen für das Filtern der Dateien enthalten: *, ?, #, !, [und].

Art des Vergleichs	Muster	Passend	Nicht passend
Mehrere Zeichen	a*a	aa, aBa, aBBa	aBC
	ab	abc, AABb, Xab	aZb, bac
Spezial Charakter	a[*]a	a*a	aaa
Mehrere Zeichen	ab*	abcdefg, abc	cab, aab
Einzelne Zeichen	a?a	aaa, a3a, aBa	aBBa
Einzelne Zahlen	a#a	a0a, a1a, a2a	aaa, a10a
Zeichen-Bereich	[a-z]	f, p, j	2, &
Ausserhalb Bereich	[!a-z]	9, &, %	b, a
Keine Zahl	[!0-9]	A, a, &, ~	0, 1, 9
Kombiniert	a[!b-m]#	An9, az0, a99	abc, aj

1.10.1.56 **{}RenameFile**

Dieser Befehl benennt eine Datei um oder verschiebt sie. Beispiel: {}RenameFile C:\Data\Info.txt | C:\Data\Hinweis.txt. Wenn der Pfad unterschiedlich ist, wird die Datei auch noch verschoben. Es ist immer ein Pfad und ein ganzer Dateiname nötig. (Keine Abkürzungen *). Hinweis. Ev. Ist es nicht möglich Dateien auf andere Laufwerke zu verschieben. (Betriebs-System abhängig)

1.10.1.57 **{}DeleteFile**

Dieser Befehl löscht Dateien. Beispiel: {}DeleteFile C:\Data\Info.txt. Wenn die Datei nicht existiert wird kein Fehler angezeigt. Es ist immer ein Pfad und ein Dateiname (oder Abkürzung) nötig. Abkürzungen sind möglich. Beispiel: {}DeleteFile C:\Data*.txt. Löscht alle Dateien in Verzeichnis Data welche die Erweiterung TXT haben.

1.10.1.58 **{}FileAttrib**

Dieser Befehl erlaubt es Datei-Eigenschaften zu ändern. Befehlsaufbau: {}FileAttrib Pfad | Filter | Unterverzeichnisse | Schreibgeschützt | Versteckt | Archiv. Beispiel: {}FileAttrib C:\Temp | *.TXT |

{{True} | +R | -H | +A. Nun werden im Verzeichnis und Unterverzeichnissen von C:\Temp alle Dateien mit der Endung *.TXT auf ReadOnly (Schreibgeschützt), auf nicht Versteckt (not Hidden) und auf Archivierbar (Archive) gestellt. Weiteres Beispiel: {{FileAttrib C:\Temp | John.txt | {{False} | -R. Nur im Verzeichnis C:\Temp (ohne Unterverzeichnisse) wird bei der Datei John.txt der Status ReadOnly entfernt.

Hinweise: Die 3 Status: | +R | -H | +A können vertauscht werden. + bedeutet den Status setzen, - bedeutet den Status entfernen. Die letzten beiden Parameter können weggelassen werden (sind optional)

Warnung: Der Status +R und +H muss mit Vorsicht angewendet werden. Wird z.B. das ganze Laufwerk C:\ auf ReadOnly gestellt, funktionieren die allermeisten Programme auf dem PC nicht mehr richtig.

1.10.1.59 **}}**Goto

Mit diesem Befehl springt der Maus und Tastatur Recorder zu einem Bezeichner (**}}**Label). Die **}}**GoTo Funktion kann dafür verwendet werden auf verschiedene Situationen während eines Makroablaufes zu reagieren. Beispiel: **}}**If **}}**Var1 = 15 ;**}}**GoTo WEITER; => Wenn der Speicher 1 den Wert 15 enthält springt der Recorder zur Marke **}}**Label WEITER. **}}**Goto erlaubt es in einem Makro vorwärts und rückwärts zu springen.

1.10.1.60 **}}**Gosub, **}}**Return}

Gosub erlaubt die Verwendung von Unterprogrammen. Unterprogramme sind Makro-Teile, welche öfters verwendet werden sollen. Beispiel: Die Zahlen einer Tabelle sollen mit immer der gleichen Formel zusammengerechnet werden. Der Makro-Teil welcher die Berechnung vornimmt wird in ein Unterprogramm verlegt. Befehlsaufbau: **}}**Gosub Label. Wenn der Recorder auf den Befehl **}}**Gosub trifft, springt er zur Zeile mit dem angegebenen Label. Dort bearbeitet der Recorder die folgenden Zeilen. Trifft er auf den Befehl **}}**Return springt der Recorder zurück auf die Zeile nach **}}**Gosub.

Beispiel:

```
}}Gosub MeinSubProgr
```

```
.....
```

```
}}Goto Ende
```

```
}}Label MeinSubProgr
```

```
.....
```

```
}}Return}
```

```
.....
```

```
}}Label Ende
```

Hinweise:

- Vor dem Unterprogramm muss ein **}}**Goto ... eingefügt werden. Der Recorder darf nie "normal" in ein Unterprogramm 'hineinlaufen'. (Nur per **}}**Gosub Befehl).
- Die Anzahl **}}**Gosub und **}}**Return müssen übereinstimmen.
- Unterprogramme im Recorder enthalten keine lokalen Speicher (wie in anderen Programmiersprachen). Die Var und String Speicher sind die gleichen wie im übrigen Makro.
- Eine Gosub Unterprogramm kann ein weiteres Gosub Unterprogramm aufrufen oder kann auch "sich selbst" aufrufen (Rekursion). Es sind max 500 Verschachtelungen möglich.

1.10.1.61 **}}**Label

}}Label bestimmt das Ziel für die Befehle **}}**Goto und **}}**Gosub. Beispiel: **}}**Label WEITER. Wichtig: Marken unterscheiden Gross- und Kleinschreibung. Das heisst: **MeineMarke** ist nicht das Gleiche wie **meineMarke**. Hinweis: Hinter dem Label ist kein Doppelpunkt (:) nötig.

1.10.1.62 **}}**LoadMacro

Dieser Befehl lädt ein anderes Makro und startet es sofort. Beispiel: **}}**LoadMacro C:\DATA\RECORDER\MACRO.KRC. Befehle die hinter dem Befehl **}}**LoadMacro stehen, werden nicht

ausgeführt ! Dieser Befehl erlaubt es, ein grosses Makro in kleinere Teile aufzubrechen. Kleinere Makros sind einfacher zu unterhalten.

Hinweis: Der Befehl `{{Master}}` erlaubt es mehrere Makros nacheinander in einem `{{Master}}`-Makro aufzurufen.

1.10.1.63 `{{Master}}`

Dieser Befehl erlaubt es mehrere Makros nacheinander zu laden und laufen zu lassen. Nach dem Befehl `{{Master}}` können z.B. mehrere `{{LoadMacro}}` Befehle folgen. Dadurch werden die Makros geladen und gleich ausgeführt. Beispiel:

```
{{Master}}
{{LoadMacro C:\MyMacros\Macro1.krc}}
{{LoadMacro C:\MyMacros\Macro2.krc}}
```

Nach dem Befehl `{{Master}}` sind nicht nur `{{LoadMacro}}` sondern fast alle Befehle möglich. Ausnahmen: `{{Goto}}`, `{{GoSub}}`, `{{ImportFile}}`, `{{ImportNextLine}}` und `{{DoImport}}`.

Ein `{{Master}}` Makro darf nicht mittels `{{LoadMacro}}` ein zweites Makro laden, welches selber einen `{{Master}}`-Befehl enthält. ~~Ersatz-Möglichkeit: `{{StartAndWait}}` einen zweiten Recorder mit dem gewünschten Makro.~~

Makro-Wiederholungen (Schleifen) sind erlaubt in Makros welche den Master-Befehl enthalten. Es existieren aber einige Einschränkungen:

- Ein Makro welches den Master-Befehl enthält wird erst ab dem Befehl `{{Master}}` wiederholt. (Alle Befehle vor `{{Master}}` werden nur einmal ausgeführt.)
- Wenn ein Makro den Master-Befehl enthält und die Schleifenzahl grösser 1 ist, können folgende Befehle nicht im Master-Makro benutzt werden: `{{Offset}}`, `{{1}}`, `{{2}}` .. `{{n}}`, `{{DownLoop}}` .. `{{LeftLoop}}`. (Diese Befehle können aber in Makros welche per `{{LoadMacro}}` geladen wurden, verwendet werden.)
- Soll in einem Master-Makro der Schleifenzähler verwendet werden, ist der Befehl `{{MLoop}}` anzuwenden. (`{{Loop}}` zeigt den Loop des per `{{LoadMacro}}` geladenen Makros)
- Wenn ein Makro den Master-Befehl enthält zeigt der Schleifenzähler im Allgemein Register die Schleifen des Master-Makros, auch wenn per `{{LoadMacro}}` weitere Makros geladen wurden und diese eigene Schlaufen fahren.

Wichtig: In einem `{{Master}}`-Makro dürfen keine normalen Speicher verwendet werden. Nur die globalen Speicher funktionieren in Master-Makros. (`{{GlobVar}}`, `{{GlobString}}`, `{{GlobAsk}}`)

1.10.1.64 `{{WriteToFile}}`, `{{WriteLineToFile}}`

Dieser Befehl schreibt Texte in eine Datei. Beispiel: `{{WriteToFile C:\Logdatei.txt |Test bestanden LOGDATEI.TXT |TEST BESTANDEN}}`. `{{WriteToFile}}` schreibt eine Zeile ohne Zeilenumbruch, `{{WriteLineToFile}}` schreibt die Zeile mit Zeilenumbruch.

1.10.1.65 `{{Minimize}}`, `{{Normalize}}`

`{{Minimize}}` minimiert den 'Maus und Tastatur Recorder' zu einem Balken in der Aufgabenleiste (TaskBar)

`{{Normalize}}` bringt den 'Maus und Tastatur Recorder' aus der Aufgabenleiste (TaskBar) zurück zur normalen Grösse.

Siehe auch Kapitel: 'Befehlszeilen Funktion' um den Recorder minimiert zu starten. (MKRMinimize)

Achtung Im minimierten Zustand keine `{{Stop}}` und `{{MessageBox2}}` Fenster anzeigen !! Sonst können diese Fenster auf bestimmten Betriebssystemen nicht mehr geschlossen werden.

1.10.1.66 *{{SetIndex1}}, {{Index1}}*

Indexe können für das indirekte Adressieren verwendet werden. Beispiel: `{{SetIndex1} 55` und dann `{{Var{{Index1}}}`. Der zweite Befehl (Var) gibt nun den Inhalt des Zahlenspeichers 55 aus. Es können 10 Indexe verwendet werden: 0..9.

Hinweis: Der Befehl Index ist ein "Überbleibsel" aus den frühen Tagen des Recorders. Seit langem ist es möglich normale Speicher für den Indirekten Zugriff auf Speicher zu verwenden. Beispiel: `{{Var{{Var15}}}`. Die Zahl in Var15 bestimmt, welcher Zahlenspeicher angesprochen wird. Die indirekte Adressierung kann für alle Speicher verwendet werden. (Var, String, Ask)

1.10.1.67 *{{NumLockOn}}, {{NumLockOff}}*

~~Schalter Der 'Maus und Tastatur Recorder' schaltet NumLock ein oder automatisch aus während des Makro Ablaufes. Am Ende des Makros wird dann wieder der Originalzustand hergestellt. Diese Massnahmen macht den Ablauf von Makros auf Windows 98 und ME bedeutend schneller. Wenn NumLock während dem Makroablauf benötigt wird, dann kann dies mit {{NumLockOn}}/-{{NumLockOff}} erreicht werden.~~

Hinweis: Das ein und Ausschalten von NumLock benötigt etwas Zeit. ~~(Vor allem auf Windows NT 4.0 Windows 2000, XP und neuer).~~ Wenn NumLock kurz nacheinander umgeschaltet werden soll, muss ein `{{Delay}}` dazwischen eingefügt werden.

1.10.1.68 *{{SetGenDelay}}*

Dieser Befehl überschreibt die Generelle Zeilen Verzögerung welche im 'Maus und Tastatur Recorder' Fenster eingestellt ist. `{{SetGenDelay}}` ist besonders praktisch für die Fehlersuche. Er ermöglicht die 'Verlangsamung' eines Teiles eines Makros. Hinweis: Ein `{{SetGenDelay}}` in einem `{{Master Makro}}` welches `{{LoadMacro}}` enthält bleibt gültig bis das geladene Makro endet.

1.10.1.69 *{{WaitForIEReady}}*

WaitForIEReady erlaubt es auf das vollständige Herunterladen einer Internetseite zu warten. Beispiel: `{{If {{WaitForIEReady 10} = {{True}} ;{{Goto OK}};{{Goto error}}`. Wenn der Seitenaufbau mehr als 10 Sekunden benötigt, springt das Makro zum Label error, sonst zum Label OK. Wenn kein Internet Explorer offen ist, wird False zurückgegeben. Als Aktion kann auch der Befehl `{{ShowError}}` benutzt werden. Dieser zeigt einen Fehlertext an und das Makro endet.

Zusätzliche Information:

Dieser Befehl kann nur für den Microsoft Internet Explorer V4.0 oder höher verwendet werden. Der Befehl funktioniert nicht für Netscape oder andere Internet-Browser.

Das Erkennen des Endes eines Internet-Seiten Herunterladens ist schwierig. Wenn auf der Internet Seite mehrere Frames vorhanden sind, erzeugt das WebBrowser Object mehrere Ready (Fertig) Signale. Manchmal erscheint das Ready (Fertig) Signal für eine kurze Zeit zwischen den einzelnen Frames. Um diese Probleme zu überwinden wiederholt der Befehl WaitForIEReady die Abfrage 8 mal. (Mit einer Wartezeit von 0.15 sec. zwischen jeder Abfrage) Wenn diese 8 Abfragen nicht reichen (oder zu hoch sind) kann die Anzahl Wiederholungen eingestellt werden.

Beispiel: `{{If {{WaitForIEReady 10,30} = {{True}} ;{{Goto OK}}; {{ShowError}}`. In diesem Beispiel wird 30 mal abgefragt. Dies benötigt eine Zeit von 4.5 sec.

Hinweise: Wenn mehrere Internet Explorer aktiv sind wartet WaitForIEReady bis alle Ready (Fertig) sind. Wenn ein Firefox Browser verwendet wird, dann kann der Stop Knopf mittels `{{GetPixel}}` oder `{{Find Pattern}}` geprüft werden (Roter Knopf mit weissem Kreuz). Der Stop Knopf ändert die Farbe erst, wenn alle Teile der Internetseite geladen sind.

1.10.1.70 *{{SetIEFocus}}*

`{{SetIEFocus}}` setzt den Fokus im Internet Explorer (IE) auf das erste Element. Dies wird erreicht indem ein ALT n (%n) ausgegeben wird. Das ist ein legacy (Altlast) Befehl. Einfacher ist es direkt %n zu verwenden.

Hinweis: Unglücklicherweise funktioniert dieser Befehl nicht für alle IE Versionen und Internet Seiten. In solchen Fällen ist ein Maus Klick in eine "leere" Fläche des Browser Fensters nötig. (Ein- und Ausschalten der Favoriten in IE8: CTRL Shift i (STRG Umschalt i)). Für weitere Tastenkombinationen suche bei Google mit "~~IEIE8~~ Tastenkombinationen"

1.10.1.71 `{{WaitForScreenChange}}`

`WaitForScreenChange` erlaubt es auf eine Änderung auf dem Bildschirm zu warten. Befehlsformat: `{{If {{WaitForScreenChange Maximale_Wartezeit, Pixel-Rechteck, Wiederholungen}}`

Beispiel:

`{{RelativeMouse Corel PHOTO-PAINT, 1, 1, 688, 434, 1.}}` Vor Verwendung von `WaitForScreenChange` muss der Cursor an die Stelle platziert werden, wo die Bildschirm-Änderung passieren wird. (1 bedeutet Maus nur platzieren, nicht klicken)

`{{If {{WaitForScreenChange 30,15,1}} = {{True}} ;{{Goto OK}};{{Goto error.}}` Wenn das Rechteck hinter den Cursor (15x15 Pixel) nicht innert 30 sec 1 mal ändert, springt das Makro zum Label error, sonst zum Label OK. Als Aktion kann auch der Befehl `{{ShowError}}` benutzt werden. Dieser zeigt einen Fehlertext an und das Makro endet.

Bei einer Wiederholung von z.B. 5 `{{WaitForScreenChange 30,15,5}}` muss eine Änderung für 5 x 150 ms konstant bleiben, bis sie akzeptiert wird.

Das Rechteck hinter dem Cursor wird von der Spitze aus berechnet. Beispiel für ein 15x15 Pixel Rechteck: 

Wichtig: Der Cursor selbst wird nicht auf eine Änderung geprüft, nur die Pixel unter dem Cursor. Das bedeutet, dass eine Änderung des Cursors von Pfeil auf Sanduhr von dieser Funktion nicht registriert wird!

`{{WaitForScreenChange}}` ist nur für kleine Rechtecke gedacht (2..100 Pixel). Werte grösser 100 benötigen eine lange Zeit für die Verarbeitung. (Minuten oder bei grösseren Werten gar Stunden). Daher wird bei Werten >100 eine Warnung angezeigt.

Hinweise: `{{WaitForScreenChange}}` funktioniert nicht mit allen Grafik Karten.

`{{PrepareScreenChange}}`

Manchmal ändert sich der Bildschirm schneller als `{{WaitForScreenChange}}` dies erfassen kann. Dann wartet `{{WaitForScreenChange}}` bis zur Max. Wartezeit. Um das zu verhindern kann

`{{PrepareScreenChange}}` verwendet werden. Beispiel:

`{{Mouse 1, 1, 871, 854, 1}}` ' Cursor-Zeiger platzieren

`{{PrepareScreenChange}}` ' Das Rechteck (15x15 Pixel) hinter den Cursor wird gespeichert

{ENTER} ' In diesem Beispiel verursacht das Enter meistens sofort eine Änderung hinter dem Cursor

`{{If {{WaitForScreenChange 30,15,1}} = {{True}};{{ShowError}}`

Das folgende `{{WaitForScreenChange}}` verwendet nun das vor dem {ENTER} abgespeicherte Rechteck für den Vergleich. Das bedeutet, dass er selbst wenn der Bildschirm augenblicklich nach dem Enter ändert, dies von `{{WaitForScreenChange}}` trotzdem bemerkt wird. Verwendet man

`{{PrepareScreenChange}}` wird grundsätzlich immer ein Rechteck von 15x15 Pixel verwendet, auch

wenn man im Befehl `{{WaitForScreenChange 30,99,1}}` etwas anderes definiert. Das Rechteck das

mit `{{PrepareScreenChange}}` gespeichert wird, wird nur vom nächsten `{{WaitForScreenChange}}`

benutzt. Nachher wird es gelöscht und ein weiteres `{{WaitForScreenChange}}` arbeitet wieder normal.

1.10.1.72 `{{GetPixel}}`

`{{GetPixel}}` liest die Farbe eines Pixel-Punktes auf dem Bildschirm. Befehlsaufbau: `{{GetPixel X-Koodinate, Y-Koodinate.}}` Um die Farbe des Pixel-Punktes hinter dem Cursor-Zeiger zu lesen kann der

folgende Befehl verwendet werden: `{{GetPixel -1, -1`. Mit `GetPixel` kann man auch auf eine Aenderung auf dem Bildschirm warten. Beispiel: `{{If {{GetPixel 100, 200} = 13160660;{{Goto MachsNochmal;` Um auf die Aenderung der Farbe zu warten muss die Original-Farbe bekannt sein. Diese kann man mit folgendem Befehl in Erfahrung bringen: `{{SetClipboard {{GetPixel 150, 288}`. Diesen Befehl laufen lassen und dann die Zahl in das Makro einfügen. Falls die Farbe schon im RGB Format bekannt ist, kann der Befehl `{{RGB` verwendet werden. Beispiel: `{{If {{GetPixel 100, 200} = {{RGB 100,200,300} ;{{Goto MachsNochmal;`

Technischer Hinweis: `{{GetPixel` gibt folgende Werte zurück: `0x00bbggrr` (b = blau, g = grün, r = rot, Hexadezimal). Für die Umwandlung kann der Befehl `{{RGB` verwendet werden. Im Internet gibt es viele RGB Tools zum hin und her rechnen. (Vorsicht, die Farbreihenfolge ist nicht immer bei allen Autoren gleich..)

Hinweis: `{{Mouse 1, 1, 430, 283, 1` und `{{Mouse 1, 1, 430, 283, 0` platzieren den Mauszeiger nicht immer an den gleichen Ort. Unterschied: Manchmal 1 Pixel. Grund: Die beiden Befehle bauen auf zwei unterschiedlichen Koordinaten Systemen auf. Das führt zu Rundungsfehlern.

1.10.1.73 {{RGB

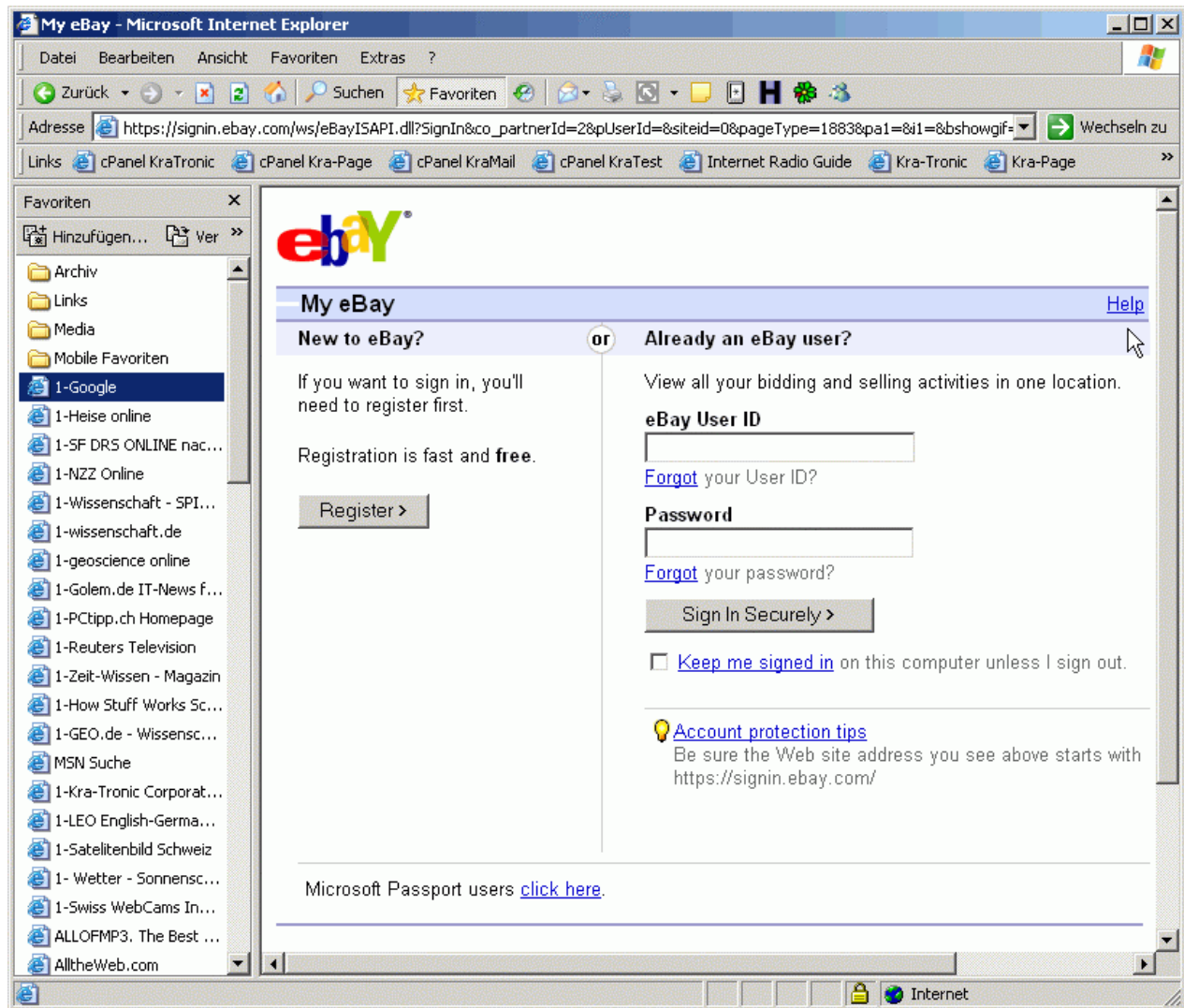
`{{RGB` wandelt Farbinformationen vom RGB-Format ins allgemeine Format um, welches von `{{GetPixel` geliefert wird. Beispiel: `{{RGB 128,100,255}`. RGB entspricht Rot, Gelb,Grün. Die einzelnen Farben können die Werte 0-255 haben. Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehl darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern:

`{{MessageBox {{RGB 128,100,255}`

Hinweis: Einige Konverter-Programme benutzen BRG statt RGB. Bei Problemen einfach die Farben austauschen.

1.10.1.74 {{FindPattern

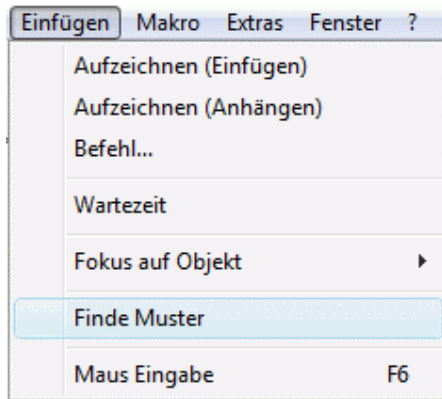
`{{FindPattern` erlaubt es, ganz bestimmte Muster auf dem Bildschirm zu finden, auch wenn sie nicht immer am gleichen Ort erscheinen. Hier ein Beispiel:



Die ID und das Passwort sind einzugeben. Die Felder für die Eingaben können aber aus verschiedenen Gründen "wandern".

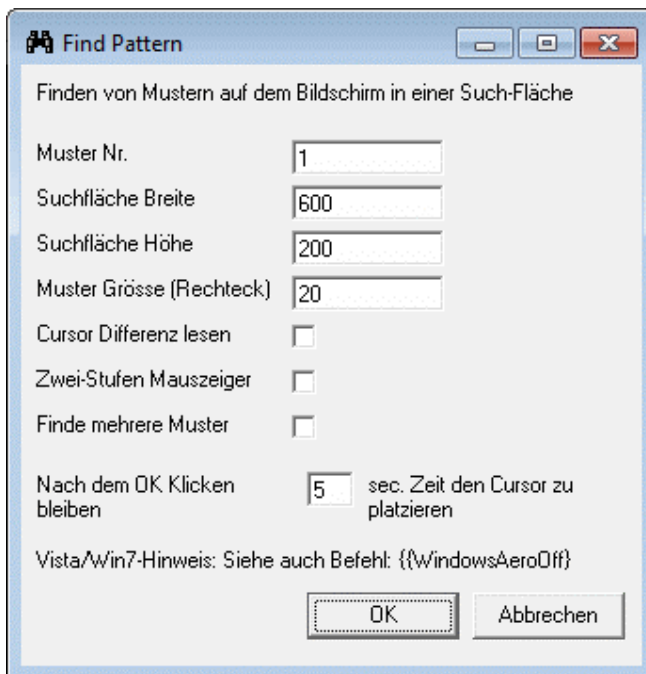
- Favoriten sind ein- oder ausgeschaltet.
- Die Links sind ein- oder ausgeschaltet.
- ebay vergrößert seinen farbigen Titel-Schriftzug.
- Verschieden grosse Werbefenster werden eingefügt.

Mit `FindPattern` haben wir nun ein Auge, das in einem bestimmten Suchbereich nach einem bestimmten Muster sucht. Bei diesem Beispiel suchen wir z.B nach dem Text 'eBay User ID' und platzieren dann den Zeiger in das Eingabefenster. Vorgehen:



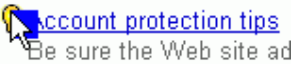
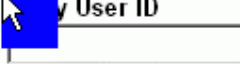
Menu Einfügen und dann auf 'Finde Muster' klicken

Folgendes Fenster erscheint:



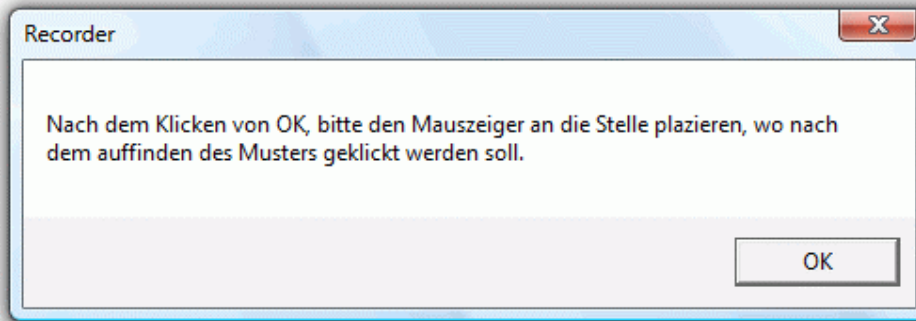
Es können mehrere Muster in einem Makro verwendet werden (max. 200). Auf der ersten Zeile wird die Muster Nummer definiert. Die zweite und dritte Zeile definieren die Suchfläche. {{FindPattern sucht nicht auf dem ganzen Bildschirm (das dauert recht lange) sondern nur in einer definierten Suchfläche. Je nachdem, wie gross die erwarteten Verschiebungen des Musters auf dem Bildschirm sind, kann die Suchfläche grösser oder kleiner gewählt werden. Das obige 'Find Pattern' Fenster z.B. ist 600 Pixel breit und 200 Pixel hoch.

Das gesuchte Muster ist immer ein Quadrat. Bei kleineren Muster-Quadraten geht die Suche rascher, aber die Gefahr, dass man was Falsches findet, ist grösser. Ein idealer Grössenbereich ist 15 - 30

Beispiel 15 Pixel:  [account protection tips](#) Be sure the Web site ad  [User ID](#)

Befindet sich das Muster nicht genau dort, wo man nachher hinklicken will, kann man eine Cursor-Differenz verwenden.

Nach dem Klicken von OK hat man 5 Sekunden Zeit um den Cursor zu platzieren. (Nur Maus platzieren, es ist kein Klick nötig). Nach 5 Sekunden erscheint für kurze Zeit das Muster-Quadrat und dann wird nach dem Differenz-Ziel gefragt:



Bei unserem 30 Pixel Beispiel würden wir den Cursor in das Eingabefeld platzieren.

Windows 7 verwendet vermehrt ändernde Hintergrundfarben wenn die Maus über einem Icon schwebt. Beispiel:



Mit der Option "Zwei-Stufen Mauszeiger" kann man Pattern aufzeichnen welche nicht eingefärbt sind. Dabei wird eine weitere Wartezeit verwendet, in der man den Cursor wegbewegen kann.

Nach dem OK wird folgender Code in das Recorder Fenster eingetragen:

```

{{If {{FindPattern 1, 626, 489, 600, 200, 0} = {{True}};{{ShowError}}
{{IncVarFindPatternCoord_X_1} 60
{{IncVarFindPatternCoord_Y_1} 25
{{Mouse 1, 1, {{VarFindPatternCoord_X_1}, {{VarFindPatternCoord_Y_1}, 0

```

Beim Ausführen des Makros passiert folgendes:

- Das Makro sucht innerhalb der Suchfläche von 600x200 Pixel nach dem Muster.
- Wird das Muster nicht gefunden zeigt der Rekorder einen Fehler.
- Ansonsten wird zu den gefundenen Muster-Koordinaten die Cursor-Differenz hinzugezählt und dann wird auf diese Koordinate geklickt.

Manchmal müssen identische Muster gefunden werden. Beispiel: Liste mit Aufzählungszeichen. Um mehrere solcher identischer Muster finden zu können, kann die Option: 'Finde mehrere Muster' (Nach: Einfügen/Finde Muster) eingeschaltet werden. Dann findet {{FindPattern alle identischen Muster innerhalb der Suchfläche. Die gefundenen Koordinaten werden nummeriert:

```

{{VarFindPatternCoord1_X_1}, {{VarFindPatternCoord1_Y_1}}
{{VarFindPatternCoord2_X_1}, {{VarFindPatternCoord2_Y_1}}

```

Die Anzahl der gefundenen Muster wird im Zahlenspeicher: {{VarFindPattern_Numbers} abgelegt.

{{FindPattern kann im Maximum 200 identische Muster finden. Ein Befehl zum Finden von identischen Mustern sieht folgendermassen aus: {{FindPattern 1, 626, 489, 200, 200, 2}

Wichtig: Nur Muster welche nicht genau horizontal ausgerichtet sind, werden gefunden. (Vertikal oder diagonal ist ok.)

Hinweise:

~~Mit Version V6.7 wurde eine schnellere Muster-Suche eingebaut. Die Suchgeschwindigkeit ist ca. 7-mal grösser als mit der alten Suche. Falls dies zu Problemen führt (zu kurze Wartezeit für Hintergrundfunktionen, oder ein Muster wird aufgrund spezieller Videokarte nicht gefunden) kann auf die alte Pattern-Suche umgeschaltet werden. Befehl: {{OldPattern {{True}}~~

Die Geschwindigkeit der Suche hängt von 4 Dingen ab: Leistung des PCs, Grösse der Suchfläche, Grösse des Musters und Aussehen des Musters. Das Aussehen des Musters hat deshalb einen

Einfluss, da bei jeder Koordinate zuerst nur die 4 Ecken-Pixel und der Pixel in der Mitte geprüft werden. Ist einer dieser Pixel ungleich zum Hintergrund wird der Mustervergleich abgebrochen und die nächste Koordinate geprüft. Ansonsten wird begonnen alle Pixel des Musters zu prüfen. Das obige Beispiel (30 Pixel) ist z.B. nicht optimal. Alle 4 Ecken sind weiss wie die sonstige Fläche. Vielleicht rettet uns hier der Mitte-Pixel...

Mit einem Trick kann man die Suchfläche sichtbar machen. Dazu ändert man die letzte Zahl zu einem 1: `{{If {{FindPattern 1, 626, 489, 200, 200, 1} = {{True}};{{ShowError}}`. Der Befehl führt dann zwar zu einem Fehler aber die Suchfläche wird mit blauen Streifen angezeigt.

Ist die Suchfläche klein, dann ist die Anzeige der Streifen nur sehr kurz sichtbar. Mit einer **3** als letzte Zahl wird der Suchbereich blau ausgefüllt. Diese blaue Fläche bleibt bestehen bis sie vom PC überschrieben wird. Beispiel: `{{FindPattern 1, 626, 489, 20, 20, 3}` Auch mit Parameter 3 zeigt {{FindPattern immer einen Fehler, daher nachher wieder auf 0 umstellen.

Wichtig:

Ragt die Suchfläche zum Bildschirm heraus, wird die Suchfläche in den Bildschirm hinein verschobener überragende Teil weggeschritten und die Cursor-Position in die Mitte der neu positionierten resultierenden Suchfläche verschoben. Dadurch kann die Suchfläche verschoben kleiner werden als beim Aufzeichnen definiert. Wird das Muster nicht gefunden, dann mit dem den obigen Trick `...200, 200, 1}` die Suchfläche überprüfen und ev. anpassen. Hinweis: Der Befehl `{{ScreenSize}` zeigt die Bildschirmgröße und den Bildschirm-Mittelpunkt des primären Bildschirms an.

Mit der Option 10 kann unabhängig von der Suchflächenangabe im Befehl der ganze Bildschirm abgesucht werden. Beispiel: `{{If {{FindPattern 1, 0, 0, 0, 0, 10}}`. Die Suche beginnt dabei im Bildschirm-Mittelpunkt. Die Option 10 ist kombinierbar mit den Optionen 1, 2 und 4. Dazu verwendet man die Optionen: 11, 12 und 14. Beispiel: 11 entspricht Optionen 1 und 10.

Die Muster werden als Dateien abgelegt. Sie erhalten den gleichen Namen wie das Makro mit einer Ergänzung. Beispiel für Muster 1: MyMacro.krc ergibt: MyMacro_FP_1.fpd. Das bedeutet, falls Makros welche `{{FindPattern}` enthalten, umbenannt werden, müssen auch diese Muster-Dateien umbenannt werden.

Bei Situationen auf dem Bildschirm (hin und wieder bei Internet Seiten) wo sich Knöpfe oder Eingabefelder bei jedem Aufruf ein wenig verändern, kann `{{FindPattern}` diese dann nicht finden. In diesem Fall macht es Sinn erst einen konstanten Text oder Bild zu suchen und dann mit Cursor Differenz den Knopf oder das Eingabefeld anzuklicken.

Eine andere Möglichkeit ist die Verwendung der Option 4: `{{FindPattern 1, 626, 489, 200, 200, 4}` Wenn diese Option gewählt ist, werden immer nur 4 Pixel geprüft (die 4 Pixel in den Ecken). Wenn man Glück hat, befindet sich die Veränderung an anderen Stellen..

Wichtig: Das gesuchte Muster muss auf dem Bildschirm vorhanden sein, wenn die Suche beginnt. Die Suche beginnt bei der Maus-Position und bewegt sich nach aussen. Wenn das Muster ein wenig zu spät kommt, kann `{{FindPattern}` dies nicht mehr erkennen.

`{{FindPattern}` arbeitet nur mit dem primären Bildschirmpositiven Koordinaten. Das bedeutet, wenn mehr als ein Bildschirm benutzt wird, und der zweite Bildschirm negative Koordinaten verwendet, dann kann {{FindPattern auf dem zweiten Bildschirm dort keine Muster finden.

Der Speicher-Name: FindPatternCoord_X/Y_1, 2.. ist reserviert für `{{FindPattern}` und sollte daher nicht für eigene Zwecke gebraucht werden.

`{{Mouse 1, 1, 430, 283, 1}` und `{{Mouse 1, 1, 430, 283, 0}` platzieren den Mauszeiger nicht immer an den gleichen Ort. Unterschied: Manchmal 1 Pixel. Grund: Die beiden Befehle bauen auf zwei unterschiedlichen Koordinaten Systemen auf. Das führt zu Rundungsfehlern.

Mit Windows Aero (Vista/Win7) funktioniert die alte Version von {{FindPattern nicht richtig (dauert ewig lange). In diesem Falle kann der Befehl {{WindowsAeroOff verwendet werden um Windows Aero auszuschalten.

{{ShowPattern: Wenn man viele Muster verwendet, wird es oft schwierig die Übersicht zu behalten. (Welche Nummer enthält welches Muster) Dafür gibt es den Befehl `{{ShowPattern 5}`. Nach dem Auftreten dieses Befehls im Makro werden bei jedem Befehl `{{FindPattern}` alle Muster links oder rechts

vom Suchbereich angezeigt. Das aktuell gesuchte Muster ist mit zwei blauen Balken markiert. Die Zahl nach `[[ShowPattern 5` definiert wie lange die Muster-Übersicht angezeigt werden soll (hier 5 Sekunden). Es werden alle Muster in Reihe angezeigt. Beispiel `..FP_1.fpd, ..FP_3.fpd, ..FP_4.fpd`. (Keine Lücke für das fehlende `..FP_2.fpd`).

1.10.1.75 `[[CheckWindow`

Dieser Befehl prüft ob ein Programm Fenster existiert. Beispiel: `[[CheckWindow Microsoft Word`. Wenn das Fenster existiert schaltet der Recorder zu diesem Fenster um und gibt eine '1' aus, sonst eine '0'. Für `[[CheckWindow` muss derjenige Name verwendet werden, welcher beim Drücken von ALT Tabulator erscheint. Es ist auch möglich nur einen Teil des Namens zu verwenden, solange er eindeutig ist. Beispiel: `[[CheckWindow Microsoft`. Manchmal ist auch noch ein Gedankenstrich notwendig, damit Windows das Fenster findet. (- Microsoft Word) Mit Probieren kann man herausfinden, was Windows akzeptiert.

Normalerweise wird `[[CheckWindow` zusammen mit einem `[[If` verwendet. Beispiel: `[[If [[CheckWindow Microsoft Word] = [[True] ;[[Goto Existiert;[[Goto ExistiertNicht`.

Siehe auch Befehl `[[Switch` für weitere Optionen. (Neue Umschalt-Technik und `=>`, `<=>`)

Wichtig: Wenn `[[CheckWindow` in einem `[[If` Befehl verwendet wird und die Zeichen `<=>` oder `=>` enthält, muss ein Text-Speicher verwendet werden. Beispiel:

```
[[SetStringTitle Oeffne Report=>
[[If [[CheckWindow [[StringTitle]] = [[True] ;[[Goto ReportOk;[[Goto ReportNotOk
```

1.10.1.76 `[[GetWindowPos`

Dieser Befehl liest die Position eines Fensters. Befehlsaufbau: `[[GetWindowPos FensterName`. Beispiel: `[[GetWindowPos Microsoft Word`. Die gefundenen Koordinaten werden als Zahlenspeicher abgelegt. Folgende Speichernamen werden verwendet:

```
[[VarGetWinCoord_X, [[VarGetWinCoord_Y (Koordinaten links oben)
[[VarGetWinCoord_X_End, [[VarGetWinCoord_Y_End (Koordinaten rechts unten)
```

Hinweis: Der Befehl `[[GetWindowPos` schaltet automatisch zum FensterName um.

Wichtig: Der Befehl `[[GetWindowPos` kann nicht als Parameter verwendet werden. (Muss zuvorderst in der Zeile stehen)

1.10.1.77 `[[GetCursorPos`

Dieser Befehl liest die Position des Mauszeigers. Befehlsaufbau: `[[GetCursorPos`. Die gefundenen Koordinaten werden als Zahlenspeicher abgelegt. Folgende Speichernamen werden verwendet:

```
[[VarGetCursorCoord_X, [[VarGetCursorCoord_Y
```

Wichtig: Der Befehl `[[GetCursorPos` kann nicht als Parameter verwendet werden. (Muss zuvorderst in der Zeile stehen)

1.10.1.78 `[[Decode`

Der Recorder ist sehr praktisch um z.B. Name und Passworte einzugeben. Bestimmte Programme in der Büroumgebung verlangen jedes Mal wieder ein Login, trotzdem man sich ja in den PC eingelogged hat. Wenn man allerdings die Passworte in Klartext in die Makros reinschreibt, hat oft das IT (Information Technology Abteilung) keine Freude. Der Befehl `[[Decode` kann für diese Passworte verwendet werden. Die Kodierung erfolgt mit einem kleinen Tool, das bei Kra-Tronic Corp. angefordert werden muss (Es wird gratis an registrierte Benutzer abgegeben).

Beispiel: `[[Decode 212232404250164402210372375201334516206455376504`. Der kodierte Text kann alle Zahlen, Zeichen und Sonderzeichen enthalten.

Wichtig: Der Befehl `[[Decode` kann nicht als Parameter verwendet werden. (Muss zuvorderst in der Zeile stehen)

1.10.1.79 *Sort*

Der Befehl `Sort` kann benutzt werden um nummerierte Speicher zu sortieren. Befehlsaufbau: `Sort SpeicherTyp, SpeicherName, StartNummer, EndNummer, EntferneDoppelteEinträge`. Beispiel: `Sort VAR, John, 1, 20, 1`. Dieser Befehl sortiert die Speicher `VarJohn1 ... VarJohn20` und entfernt doppelte Einträge. SpeicherTyp: VAR und STRING sind möglich. EntferneDoppelteEinträge: 0 = Doppelte Einträge NICHT entfernen, 1 = Doppelte Einträge entfernen. Wenn doppelte Einträge entfernt wurden, dann enthalten die überzähligen VAR Speicher eine 0 und die überzähligen STRING Speicher sind leer (`#emptyX5#`).

1.10.1.80 *Weitere Befehle*

Weitere Befehle, welche besonders praktisch für Software Tests sind, können im Kapitel: 'Spezielle Funktionen: Automatisierter Software Test' gefunden werden. Es sind dies: `CheckFocus`, `WaitForFocus`, `WaitForEndFocus`, `MessageBox2`

1.10.2 *Spezielle Befehle*

Der 'Maus und Tastatur Recorder' enthält weitere spezielle Befehle, welche bei der Befehlsauswahl mittels Menü zum Teil nicht erscheinen. Es sind dies Befehle die man selten benutzt und deshalb auch von Hand eingeben muss.

Die folgenden Befehle sind in der Beschreibung des Befehles: `Switch` zu finden:
`KeepSpace`, `UseOldSwitch`, `UseNewSwitch`, `UseForeground`, `UseRecorder`

Die folgenden Befehle sind in der Beschreibung des Befehles: `Start` zu finden:
`StartAndWait`, ~~`ReturnStartAndWait`~~

Der folgende Befehl ist in der Beschreibung des Befehles: `WaitForScreenChange` zu finden:
`PrepareScreenChange`

1.10.2.1 *CTRL, Shift, Alt*

Diese 3 Befehle erlauben es die Control-Tasten CTRL (STRG), Alt und Umschalt alleine zu drücken (Nicht in Kombination mit anderen Tasten). Die 3 Befehle können auf einer Zeile beliebig kombiniert werden. z.B. nur ein Befehl, oder alle 3.

Beispiele: `CTRL` Die Taste CTRL (STRG) wird während 50 ms (Millisekunden) gedrückt. (50 ms ist der Standardwert, wenn keine Dauer angegeben wird)

`Alt 700` Die Taste Alt wird während 700 ms (0,7s) gedrückt.

`Shift 300 CTRL Alt` Die drei Tasten CTRL (STRG), Alt und Umschalt werden für 300 ms gedrückt.

Hinweis: Es wird nur die Zeit nach dem ersten Befehl ausgewertet. `Shift 300 CTRL 600` bedeutet dass Umschalt und CTRL (STRG) für 300 ms gedrückt werden.

1.10.2.2 *CTRL+/-, Shift+/-, Alt+/-*

Die Befehle `CTRL+/-`, `Shift+/-`, `Alt+/-` erlauben es diese 3 Tasten gezielt ein und auszuschalten.

Beispiele:

`CTRL+` schaltet die CTRL Taste ein. Die CTRL bleibt eingeschaltet bis ein `CTRL` oder `CTRL-` Befehl benutzt wird.

`CTRL+ ALT+` schaltet die CTRL und ALT Taste ein. Die Tasten bleibt eingeschaltet bis ein `CTRL` `ALT` oder `CTRL ALT` - Befehl benutzt wird.

Wichtig: Wenn `CTRL+`, `Shift+`, `Alt+` im Makro benutzt wurden, MUESSEN diese vor dem Ende des Makros ausgeschaltet werden! Sonst verhält sich die Tastatur sehr ungewöhnlich.

1.10.2.3 *Space*

Dieser Befehl erlaubt es SPACE (Leerschlag) eine definierte Zeit lang zu drücken. Beispiele: `SPACE`. Die Taste SPACE wird während 50 ms gedrückt. `Space 700`. Die Taste SPACE wird während 700

ms (0,7s) gedrückt. Hinweise: Dieser Befehl kann nur für den SPACE Tastendruck verwendet werden, nicht aber für Befehle wie `{{If` oder `{{SetString`. Dort muss `#SpaceX5#` verwendet werden. Der Befehl `{{Space` löst keine Tastatur-Wiederholung aus wie wenn man manuell die Space Taste drückt. `{{Space` ist eine Funktion die vor allem bei Spielen (Games) benötigt wird.

1.10.2.4 `{{CR}`, `{{LF}`, `{{Tab}`

Diese 3 Befehle erlauben es Steuerzeichen in den Text einzubauen: CR = Wagenrücklauf, LF = Zeilenvorschub, Tab = Tabulator. Diese Befehle können nur als Parameter verwendet werden. Das heisst diese Befehle dürfen nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{MessageBox {{Erste Zeile{{CR}Zweite Zeile}` oder `Erster Teil{{Tab}Zweiter Teil`. Anstelle von `{{Tab}` kann auch `{TAB}` verwendet werden. `{TAB}` darf auch am Anfang der Zeile stehen.

1.10.2.5 `{{MouseDownTime`

Bei Spielen ist es hin und wieder nötig, die Maustaste für längere Zeit zu drücken. Der Befehl `{{MouseDownTime` erlaubt es diese Zeit zu setzen. Beispiel: `{{MouseDownTime 3000`. Jeder Mausklick hält nun die Maustaste für 3 Sekunden unten. `{{MouseDownTime 0` stellt die Maus-unten Zeit wieder auf den Normalwert zurück.

1.10.2.6 `{{KeyDownTime`

Bei Spielen ist es hin und wieder nötig, eine Taste für längere Zeit zu drücken. Der Befehl `{{KeyDownTime` erlaubt es diese Zeit zu setzen. Beispiel: `{{KeyDownTime 3000`. Jeder Tastendruck hält nun die Taste für 3 Sekunden unten. `{{KeyDownTime 0` stellt die Taste-unten Zeit wieder auf den Normalwert zurück. **WICHTIG:** Dieser Befehl funktioniert nur zusammen mit `{{SendTextSpecial}!`

1.10.2.7 `{{MacroPath}`

Dieser Befehl liefert den Pfad des aktuell geladenen Makros. Beispiel: Makro: C:\Eigene Dateien\Recorder\MeinMakro.krc. `{{MacroPath` liefert: C:\Eigene Dateien\Recorder\

1.10.2.8 `{{AppDataPath}`

Diese Konstante liefert den Pfad zum Verzeichnis wo folgende Dateien abgelegt werden: Fehler-Such-Log-Buch, Globale Variablen, HotKey-Datei, Temporäre Makros. Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehle darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{MessageBox {{AppDataPath}`

1.10.2.9 `{{AppPath}`

Dieses Konstante liefert den Pfad zum Verzeichnis wo das Recorder Programm liegt (Recorder.exe). Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehle darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{MessageBox {{AppPath}`

1.10.2.10 `{{FileSize}`

Dieser Befehl liefert die Grösse einer Datei in Byte. Kann eine Datei nicht gefunden werden, liefert dieser Befehl die Dateigrösse -1. Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehle darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{MessageBox {{FileSize C:\MyFile.txt}`

Hinweis: Dieser Befehl ist im Fenster "Befehl einfügen" nicht vorhanden (Muss manuell eingegeben werden).

1.10.2.11 `{{WindowsKey}`

Dieser Befehl wird eingefügt, wenn die Windows-Taste gedrückt wird (~~nur auf neueren Tastaturen vorhanden~~). Der Befehl `{{WindowsKey}` erlaubt auch Parameter wie z.B. R oder E. Beispiel: `{{WindowsKey R`. Dieser Befehl Oeffnet das 'Ausführen' Fenster. `{{WindowsKey E` öffnet den Windows Explorer, `{{WindowsKey M` minimiert alle Fenster, `+{{WindowsKey M` bringt die Fenster wieder zurück, `{{WindowsKey L` logged den aktuellen Benutzer aus. Hinweis: Der Parameter wird nicht automatisch aufgezeichnet, sondern muss nachträglich im Editor hinzugefügt werden.

1.10.2.12 `{{NoFocus}}`

Wenn ein Makro endet geht der Fokus zurück an den Maus und Tastatur Recorder. (Der Recorder ist im Vordergrund). Normalerweise ist diese Reaktion erwünscht. In bestimmten Situationen möchte man aber gerne, dass die Anwendung, auf der vom Recorder gearbeitet wurde, im Vordergrund bleibt. Dieses Verhalten kann durch den Befehl `{{NoFocus}}` erreicht werden. Befindet sich irgendwo im Makro der Befehl: `{{NoFocus}}`, dann kommt der Recorder nach Beenden am Ende des Makros NICHT in den Vordergrund. (Ausnahme: Wenn der Recorder während des Makro-Ablaufes bereits im Vordergrund war, bleibt er am Ende des Makros im Vordergrund.)

Hinweis: `{{NoFocus}` sollte am Ende des Makros platziert werden. Grund: Ansonsten kann, wenn der Recorder minimiert ist und ein Fehler auftritt, der Recorder nicht mehr normalisiert werden.

1.10.2.13 `{{GetWindowName}}`

Dieser Befehl gibt den Namen des aktiven Fensters zurück. (Das aktive Fenster ist das oberste Fenster, das mit dem farbigen Balken.) Gleichzeitig wird zu diesem Fenster umgeschaltet. Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehle darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{MessageBox {{GetWindowName}}`

1.10.2.14 `{{CapsLockOn}}`, `{{CapsLockOff}}`

Dieser Befehl erlaubt es die Festhalte-Taste (CapsLock) ein- oder auszuschalten. (CapsLock verursacht, dass alle Buchstaben gross geschrieben werden). Wenn man `{{CapsLockOff}}` an den Anfang eines Makros setzt, dann läuft das Makro auch dann richtig, wenn der Benutzer vergisst CapsLock auszuschalten. Hinweis: Der Zustand von CapsLock vor dem Beginn des Makros wird nach dem Ende des Makros nicht automatisch wiederhergestellt.

1.10.2.15 `{{DebugOn}}`, `{{DebugOff}}`

Der Befehl `{{DebugOn}}` zeigt ein Debug-Fenster an. Bei jedem Lesen und Schreiben eines Speicherwertes wird der Name des Speichers, der Wert und der dazugehörige Befehl angezeigt. Die neuesten Einträge erscheinen immer zuoberst, die restlichen Einträge werden nach unten weggeschoben. `{{DebugOff}}` beendet das Melden der Speicherwerte, lässt aber das Debug Fenster offen.

Einschränkung. Das Debug Fenster wird bei einem Aufruf des Recorder-Makros aus einer anderen Anwendung ev. nicht angezeigt. Beispiel: Aufruf aus Excel Makro. Das Debug Fenster wird auch nicht angezeigt wenn der Recorder Full Screen (bildschirmfüllend) verwendet wird.

1.10.2.16 `{{ResetLogFile}}`

Während grossen Makros (oder Makros mit vielen Schleifen) kann das die LogBuch-Datei sehr gross werden und den Makro-Ablauf verlangsamen (oder gar zu Störungen im Makro Ablauf führen). Mit dem Befehl `{{ResetLogFile}}` kann die LogBuch-Datei umkopiert und gelöscht werden. Maximum 5 LogBuch-Dateien werden gespeichert. `{{ResetLogFile}}` macht das folgende: Die LogBuch-Datei 4 wird in LogBuch-Datei 5 kopiert, Datei 3 in 4, Datei 2 in 3 und die original LogBuch-Datei wird in LogBuch-Datei 2 kopiert. Die original LogBuch-Datei wird dann gelöscht und neu gestartet. LogBuch-Datei und LogBuch-Datei 2 können über das Menu: Extras / Optionen / Abspielen / 'Logbuch öffnen' geöffnet werden. Zum öffnen der restlichen LogBuch-Dateien kann folgender Befehl verwendet werden: `{{Start notepad {{AppDataPath}}\Logfile3.txt}`

1.10.2.17 `{{BracesReplace}}`

"Geschwungene Klammern" ({ }) in Textspeichern: Weil der Recorder geschwungene Klammern als Start- und End-Marken bei Befehlen verwendet, können geschwungene Klammern in Textspeichern Probleme verursachen. Solche Klammern können Teil eines Textes sein, wenn man z.B. Seiten aus dem Internet in Textspeicher kopiert.

Lösung: Mit der Verwendung des folgenden Makro-Codes können geschwungene Klammern in andere Zeichen (und zurück) umgewandelt werden.

^c

`{{BracesReplace § | ° | {{True}}`
`^v`

Der erste Parameter definiert den Ersatz für {, Der zweite Parameter definiert den Ersatz für }, der dritte Parameter definiert ob Klammern ersetzt oder zurück gewandelt werden ({{True} => {} werden ersetzt, {{False} => {} werden wieder hergestellt).

Zusätzliche Funktion von BracesReplace: Die Buchstaben: +, ^, %, ~, (,), [,], {, } werden vom Recorder nicht ausgegeben (es sind Steuer-Zeichen). Der folgende Befehl erlaubt es auch solche Zeichen direkt vom Recorder ausgeben zu lassen (ohne CTRL v) `{{BracesReplace x | x | 2}`. Der Befehl ersetzt die Steuerzeichen in der Zwischenablage durch: {+, {^}, {%}, {~}, {(,)}, {)}, {{}}, {{}}, {{}}. Dadurch können sie vom Recorder ausgegeben werden.

`{{BracesReplace}` arbeitet immer mit der Zwischenablage. Hinweise: Die geschwungenen Klammern können auch durch Texte wie z.B. #MeinStartZeichen# ersetzt werden.

Frühere Versionen des Recorder enthielten den Befehl: ReplaceBraces.exe. Dieser Befehl ist nicht mehr unterstützt. Falls dieser Befehl verwendet wurde, muss er entsprechend abgeändert werden.

1.10.2.18 `{{HDFreeSpace}`

Der Befehl `{{HDFreeSpace}` zeigt den freien Platz auf einem Laufwerk (in MegaByte). Beispiel `{{MessageBox {{HDFreeSpace C:}}` Wichtig: Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehl darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{SetVarFreeSpace {{HDFreeSpace C:}}`

1.10.2.19 `{{HDUsedSpace}`

Der Befehl `{{HDFreeSpace}` zeigt den benutzten Platz auf einem Laufwerk (in MegaByte). Beispiel `{{MessageBox {{HDUsedSpace C:}}` Wichtig: Dieser Befehl kann nur als Parameter verwendet werden. Das heisst dieser Befehl darf nie am Anfang einer Zeile stehen. Beispiel für die Verwendung von Parametern: `{{SetVarUsedSpace {{HDUsedSpace C:}}`

1.10.2.20 `Mathematik und Text Befehle`

Die folgenden Befehle sind spezial Befehle für Zahlen und Texte. Diese Befehle können nur als Parameter verwendet werden. Das heisst diese Befehle dürfen nie am Anfang einer Zeile stehen.

Beispiel für die Verwendung von Parametern: `{{MessageBox {{String1}}`

Die Mathematik und Text Befehle verwenden einen | als Spearator. Dieser Separator kann geändert werden mit dem Befehl: `{{MathStrSeparator}`. Beispiel: `{{MathStrSeparator *`

Nicht alle Fehler verursachen eine Fehlermeldung. Bei mathematischen Fehlern wie z.B. bei ~~einem Ueberlauf~~ (~~{{Power 5|500}}~~) oder bei einer Division durch 0 wird eine Null zurückgegeben.

Die Mathematik und Text Befehle können keine Berechnungen innerhalb des Befehls durchführen. Beispiel (FALSCH) `{{MessageBox {{Left Recorder | {{VarLimite} + 5}}`. Berechnungen müssen vorher gemacht werden. `{{IncVarLimite 5} {{MessageBox {{Left Recorder | {{VarLimite}}}}`.

Ausnahme: Die vorherige Limitation gilt nicht für den Befehl `{{Calc}`.

Hinweis: Die Verwendung von geschwungenen Klammern in Textspeichern kann Probleme verursachen. Ursache: {} werden als Start- und End-Zeichen bei Befehlen verwendet. Lösung: Mit der Verwendung von "BracesReplaceReplaceBraces" kann dieses Problem umgangen werden. Siehe Kapitel: Hinweise / Einschränkungen.

1.10.2.21 `{{Power}`

Dieser Befehl gibt ErsteZahl hoch ZweiteZahl zurück. Beispiel: 5^2 : `{{MessageBox {{Power 5 | 2}}`.

Resultat 25. Wichtig: Die erste Zahl darf nur dann negativ sein, wenn die zweite Zahl (Exponent) eine ganze Zahl ist. (~~Bei Windows 98 und ME darf nur eine der beiden Zahlen negativ sein~~)

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.22 `{{Mod}`

Dieser Befehl führt eine Modulodivision durch. Der Rest, der bei der Division bleibt, wird zurückgegeben. Beispiel: `{{MessageBox {{Mod 14 | 5}}`. Resultat 4. Hinweis: Der Mod Befehl rundet zuerst Real-Zahlen in Integer und führt dann die Mod-Funktion aus. Beispiel: `{{MessageBox {{Mod 19 | 6.7}}` Resultat 5. Maximal-Zahl für `{{Mod: $2^{64}/22'147'483'647$` .

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.23 **Div**

Führt eine Integerdivision durch, d.h. gibt nur ganze Zahlen zurück und ignoriert den Rest, der bei der Division eventuell auftritt. Beispiel: `{{MessageBox {{Div 14 | 5}}`. Result 2. Hinweis: Der Div Befehl rundet zuerst Real-Zahlen in Integer und führt dann die Div-Funktion aus. Maximal-Zahl für `Div`: [2⁶⁴/22'147'483'647](#).

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.24 **IsNumeric**

Die IsNumeric-Funktion überprüft, ob eine gesamte Zeichenfolge als Zahl interpretiert werden kann.

Beispiel: `{{If {{IsNumeric 23f3} = {{True} ;{{Beep}}`;

Resultat: Es ertönt kein Beep (23e3 würde ein Beep erzeugen).

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.25 **Len**

Ermittelt die Anzahl der Zeichen einer Zeichenfolge und gibt sie als Zahl zurück. Beispiel:

`{{MessageBox {{Len {{String1}}}`. Führende und nachfolgende Leerzeichen werden nicht berücksichtigt.

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.26 **InStr**

Die InStr-Funktion gibt die Position des ersten Zeichens einer Zeichenfolge innerhalb einer anderen Zeichenfolge zurück. Befehlsaufbau: `{{InStr StartNummer | Text der durchsucht wird | Text welcher gesucht wird`. Beispiel: `{{MessageBox {{InStr 1 | Recorder | der}}` Resultat: 6. Wenn nichts gefunden wird erhält man eine Null zurück.

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.27 **Mid**

Die Mid-Funktion gibt eine bestimmte Anzahl zusammenhängender Zeichen aus einem vorgegebenen String zurück.. Befehlsaufbau: `{{Mid Text | StartNumber | Länge`. Beispiel: `{{MessageBox {{Mid Recorder| 3 | 5}}` Resultat: corde. Für die Länge kann auch der Wert -1 eingegeben werden. Dadurch wird dann ein Text beginnend von der StartNumber bis zum Textende zurückgegeben. Beispiel:

`{{MessageBox {{Mid Recorder| 3 | -1}}` Resultat: corder.

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.28 **Left**

Dieser Befehl gibt eine vorgegebene Anzahl von Zeichen, ausgehend von der linken Seite einer Zeichenfolge zurück. Befehlsaufbau: `{{Left Text | AnzahlCharakter`. Beispiel: `{{MessageBox {{Left Recorder | 3}}` Resultat: Rec

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.29 **Right**

Dieser Befehl gibt eine vorgegebene Anzahl von Zeichen, ausgehend von der rechten Seite einer Zeichenfolge zurück. Befehlsaufbau: `{{Right Text | AnzahlCharakter`. Beispiel: `{{MessageBox {{Right Recorder | 6}}` Resultat: corder

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.30 **Trim**

Die Trim-Funktion löscht Leerzeichen links und rechts von einer Zeichenfolge.. Beispiel: `{{MessageBox {{Trim Text mit Leerschlägen }}`. Resultat: Text mit Leerschlägen. Dieser Befehl wird selten benötigt, da die meisten Recorder Befehle automatisch ein Trim machen.

Siehe also Hinweise im Kapitel: Mathematik und Text Befehle.

1.10.2.31 **Ucase**

Die UCase-Funktion wandelt die Kleinbuchstaben einer Zeichenfolge in Grossbuchstaben um. Beispiel: `{{MessageBox {{UCase Recorder}}`. Resultat: RECORDER.

1.10.2.32 `{{LCase`

Die LCase-Funktion wandelt die Grossbuchstaben einer Zeichenfolge in Kleinbuchstaben um. Beispiel: `{{MessageBox {{LCase Recorder}}`. Resultat: recorder.

1.10.2.33 `{{Chr`

Die Chr-Funktion gibt das Zeichen für einen bestimmten ASCII Zeichencode zurück.. Beispiel: `{{MessageBox {{Chr 33}}`. Resultat: !

1.10.2.34 `{{Asc`

Die Asc-Funktion gibt den Zeichencode, der dem ersten Buchstaben der mitgegebenen Zeichenfolge entspricht zurück. Beispiel: `{{MessageBox {{Asc Hallo}}`. Resultat: 72

1.10.2.35 `{{Replace`

Die Replace-Funktion ersetzt Zeichenketten in einem Text. Befehlsaufbau: `{{Replace OriginalText | Zu ersetzender Text | Neuer Text | Startpunkt im OriginalText | AnzahlErsetzungen}`. Beispiel: `{{MessageBox {{Replace Recorder | c | k | 1 | -1}}` Resultat: Rekorder. -1 bedeutet, dass alle vorkommenden c ersetzt werden.

Hinweis: Es ist nicht möglich mit dem obigen Befehl einen Buchstaben durch } zu ersetzen. Grund: } wird als Endmarke im Befehl verwendet. Lösung: Mit der Verwendung von "`BracesReplaceReplaceBraces`" kann dieses Problem umgangen werden. Siehe Kapitel: Hinweise / Einschränkungen.

1.10.2.36 `{{Calc`

Die vorherigen mathematischen Befehle haben einen grossen Nachteil. Diese Befehle können nicht kombiniert werden wie z.B. $(5+2)*\sin(1.54)/3^2$. Der Befehl `{{Calc` erlaubt es nun ~~die bis zu 200~~ Operationen in einem Befehl zu kombinieren. Zusätzlich unterstützt er mehr als ~~70+30~~ mathematische Funktionen. Befehlsaufbau: `{{Calc ListeDerOperationen}`. Beispiele: `{{Calc (5+2)/Power(3,^2)}`. Resultat: 0,777. `{{Calc` verwendet die normalen mathematischen Regeln (* vor -). Recorder Variablen können mit diesem Befehl ebenfalls verwendet werden: `{{MessageBox {{Calc 5*{{VarMyValue}}+(25+{{VarMyValue2}})*Power({{VarMyValue3},^2)}}`

Hier eine Liste aller möglichen Operationen:

Function	Description	Note
+	addition	
-	subtraction	
*	multiplication	
/	division	35/4 = 8.75
<code>GT (x, y) %</code>	x greater than y (>) Percentage (Prozent)	return 1 (true) 0 (false) 35% = 0.35
<code>GE (x, y) \</code>	x equal or greater than y (>=) integer division	return 1 (true) 0 (false) 35\4 = 8
<code>LT (x, y) ^</code>	x less than y (<) raise-to-power (hoch x)	return 1 (true) 0 (false) 3^1.8 = 7.22467405584208
<code>LE (x, y) + +</code>	x equal or less than y (<=) absolute-value	return 1 (true) 0 (false) -5 =5 (the same as abs)
<code>EQ (x, y) !</code>	x equal y (=) factorial	return 1 (true) 0 (false) 5!=120 (the same as fact)
<code>NE abs (x, y)</code>	x not equal y (<>) absolute-value	return 1 (true) 0 (false) abs(-5) = 5
<code>Percent (x, y) atan(x) , -atan(x)</code>	Percentage (Prozent) inverse tangent	Percent(50,10) = 5 (5% from 50) atan(pi/4) = +
<code>Diveos (x, y)</code>	integer division cosine	Div(35,4) = 8 argument in radiant
<code>mod (a, b) sin(x)</code>	Division remainder (Divisions-Rest) sin	mod(29,6) = 5 mod(-29,6) = -5 argument in radiant
<code>Power (n, exp(x))</code>	raise-to-power (hoch x) exponential	Power(3,1.8) = 7.22467405584208 exp(+)

<u>Absfix(x)</u>	absolute value integer part (ganzzahliger Teil)	= 2.71828182845905 Abs(-5) = 5 fix(-3.8) = 3
<u>Factint(x)</u>	factorial integer part (ganzzahliger Teil)	Fact(5) = 120 int(-3.8) = -4
<u>expdec(x)</u>	exponential decimal part	exp(1) = 2.71828182845905 dec(-3.8) = -0.8
<u>Int(x)</u>	integer part (ganzzahliger Teil)	int(-3.8) = -4
<u>Ceil(x)</u>	integer part (ganzzahliger Teil)	Ceil(-3.8) = 3
<u>Floor(x)</u>	integer part (ganzzahliger Teil)	Floor(-3.8) = -4
<u>decim(x)</u>	decimal part (Dezimalteil)	decim(-3.8) = -0.8
<u>Trunc(x)</u>	Part before decimal point	Trunc(-3.8) = -3
<u>round(x,d)</u>	round a number with d decimal places	round(1.35712, 2) = 1.36
<u>ln(x), log(x)</u>	logarithm natural	argument x > 0
<u>logN(x,n)</u>	N-base logarithm	logN(16,2) = 4
<u>Log10rnd(x)</u>	10-base logarithm Random (Zufallszahl)	Log10(1.2) = 0.699 returns a random number between x and 0
<u>rnd</u>	Random (Zufallszahl)	returns a random number between x and 0
<u>sgn(x)</u>	Sign (Vorzeichen)	returns 1 if x > 0, 0 if x = 0, -1 if x < 0
<u>sqr(x) / sqrt(x)</u>	square root (Quadratwurzel)	sqr(2) = 1.4142135623731, also 2^(1/2)
<u>cbr(x)</u>	cube root (Kubik Wurzel)	$\sqrt[3]{x}$, example cbr(2) = 1.2599, cbr(-2) = -1.2599
<u>root(x,n)</u>	n-th root (the same as $x^{1/n}$)	argument n \neq 0, x \geq 0 if n even, $\forall x$ if n odd
<u>min(a,b,)</u>	Minimum (5 parameters necessary)	min(13,24) = 13
<u>max(a,b,)</u>	Maximum (5 parameters necessary)	max(13,24) = 24
<u>mcd(a,b)</u>	maximum common divisor	mcd(4346,174) = 2
<u>mcm(a,b)</u>	minimum common multiple	mcm(1440,378) = 30240
<u>cos(x)</u>	cosine	argument in radian
<u>sin(x)</u>	sin	argument in radian
<u>tan(x)</u>	tangent	argument (in radian) $x \neq k\pi/2$ with $k = \pm 1, \pm 2, \dots$
<u>atn(x), atan(x)</u>	inverse tangent	atn(pi/4) = 1
<u>acos(x)</u>	inverse cosine	argument $-1 \leq x \leq 1$
<u>asin(x)</u>	Inverse sine	argument $-1 \leq x \leq 1$
<u>cosh(x)</u>	hyperbolic cosine	$\forall x$
<u>sinh(x)</u>	hyperbolic sine	$\forall x$
<u>tanh(x)</u>	hyperbolic tangent	$\forall x$
<u>acosh(x)</u>	Inverse hyperbolic cosine	argument $x \geq 1$
<u>asinh(x)</u>	Inverse hyperbolic sine	$\forall x$
<u>atanh(x)</u>	Inverse hyperbolic tangent	$-1 < x < 1$
<u>root(x,n)</u>	n-th root (the same as $x^{1/n}$)	argument n \neq 0, x \geq 0 if n even, $\forall x$ if n odd
<u>mod(a,b)</u>	Division remainder (Divisions-Rest)	mod(29,6) = 5 mod(-29,6) = -5
<u>fact(x)</u>	factorial	argument $0 \leq x \leq 170$
<u>comb(n,k)</u>	combinations	comb(6,3) = 20, comb(6,6) = 1
<u>perm(n,k)</u>	permutations	perm(8,4) = 1680, -
<u>min(a,b,...)</u>	minimum	min(13,24) = 13
<u>max(a,b,...)</u>	maximum	max(13,24) = 24
<u>mcd(a,b,...)</u>	maximum common divisor	mcd(4346,174) = 2
<u>mcm(a,b,...)</u>	minimum common multiple	mcm(1440,378,1560,72,1650) = 21621600
<u>gcd(a,b,...)</u>	greatest common divisor	The same as mcd
<u>lcm(a,b,...)</u>	lowest common multiple	The same as mcm
<u>csc(x)</u>	cosecant	argument (in radian) $x \neq k\pi$ with $k = 0, \pm 1, \pm 2, \dots$
<u>sec(x)</u>	secant	argument (in radian) $x \neq k\pi/2$ with $k = \pm 1, \pm 2, \dots$
<u>cot(x)</u>	cotangent	argument (in radian) $x \neq k\pi$ with $k = 0, \pm 1, \pm 2, \dots$
<u>acsc(x)</u>	inverse cosecant	-
<u>asec(x)</u>	inverse secant	-
<u>acot(x)</u>	inverse cotangent	-
<u>esch(x)</u>	hyperbolic cosecant	argument $x > 0$
<u>sech(x)</u>	hyperbolic secant	argument $x > 1$
<u>coth(x)</u>	hyperbolic cotangent	argument $x > 2$
<u>aesch(x)</u>	inverse hyperbolic cosecant	-

<u>asech(x)</u>	inverse hyperbolic secant	argument $-0 \leq x \leq 1$
<u>asech(x)</u>	inverse hyperbolic cotangent	argument $-x < 1$ or $x > 1$
<u>rad(x)</u>	radiant conversion	converts radiant into current unit of angle
<u>deg(x)</u>	degree sess. conversion	convert sess. degree into current unit of angle
<u>grad(x)</u>	degree cent. conversion	converts cent. degree into current unit of angle
<u>Pi</u> <u>round(x,d)</u>	Constant Pi round a number with d-decimal	<u>3.1415</u> <u>round(1.35712, 2) = 1.36</u>
<u>Land</u> >	logic and greater than	<u>Land(a,b) = return 1 (true) 0 (false) if a=0 or b=0</u>
<u>Lor</u> >=	logic equal or greater than	<u>Lor(a,b) = return 1 (true) 0 (false) only if a=0 and b=0</u>
<u>Lnot</u> <	logic not less than	<u>Lnot(a) = return 1 (true) 0 (false) if a ≠ 0, else 1</u>
<u>Lxor</u> <=	logic exclusive or equal or less than	<u>Lxor(a,b) = return 1 (true) only if a ≠ b 0 (false)</u>
<u>Lnand</u> =	logic nand equal	<u>Lnand(a,b) = return 1 (true) if a=1 or b=1, 0 (false)</u>
<u>Lnor</u> <>	logic nor not equal	<u>Lnor(a,b) = return 1 (true) only if a= 0 and b=0 (false)</u>
<u>Lnxor</u> and	logic exclusive nor and	<u>Lnxorand(a,b) = return 1 (true) only 0 (false) if a=0 or b=0</u>
<u>Year(n)</u> or	<u>Year (Jahr)</u> logic or	<u>Year(now) or(a,b) = return 0 (false) only if a=0 and b=0</u>
<u>Month(n)</u> not	<u>Month (Monat)</u> logic not	<u>Month(now) not(a) = return 0 (false) if a ≠ 0, else 1</u>
<u>Day(n)</u> xor	<u>Day (Tag)</u> logic exclusive or	<u>Day(now) xor(a,b) = return 1 (true) only if a ≠ b</u>
<u>Hour(n)</u> nand	<u>Hour (Stunde)</u> logic nand	<u>Hour(now) nand(a,b) = return 1 (true) if a=1 or b=1</u>
<u>Minute(n)</u> nor	<u>minute</u> logic nor	<u>Minute(now) nor(a,b) = return 1 (true) only if a=0 and b=0</u>
<u>Second(n)</u> nxor	<u>second</u> logic exclusive nor	<u>Second(now) nxor(a,b) = return 1 (true) only if a=b</u>
<u>Weekday(n)</u> Psi(x)	<u>Weekday (Wochentag)</u> Function psi-system timestamp (Systemzeit)	<u>Weekday(now)</u>
<u>NowDNorm(x,μ,σ)</u>	<u>Normal density function</u>	$\forall x, \mu > 0, \sigma > 0$
<u>TodayCNorm(x,m,d)</u>	<u>system datestamp (Systemdatum)</u> Normal cumulative function	$\forall x, \mu > 0, \sigma > 1$
<u>PadLeft(s,l,c)</u> DPoisson(x,k)	Padding left side of a text Poisson density function	<u>PadLeft("John", 7, "x") = xxxJohnx</u> $> 0, k = 1, 2, 3 \dots$
<u>PadRight(s,l,c)</u> CPoisson(x,k)	Padding left side of a text Poisson cumulative function $k = 1, 2, 3 \dots$	<u>PadRight("John", 7, "x") = Johnxxx</u> $> 0, k = 1, 2, 3 \dots$
<u>Wcase(s)</u> DBinom(k,n,x)	First letter uppercase Binomial density for k successes for n trials	<u>Wcase("john") = John</u> $k, n = 1, 2, 3 \dots, k < n, x \leq 1$
<u>Money(x)</u> CBinom(k,n,x)	Formats number as currency Binomial cumulative for k successes for n trials	<u>Money(12.346) = Fr. 12.35k</u> $n = 1, 2, 3 \dots, k < n, x \leq 1$
<u>FormatSi(x)</u>	<u>Format</u> Sine integral	<u>format(1234.5, "#.###.00") = 1'234.50</u> $\forall x$
<u>Ci(x)</u>	Cosine integral	<u>format(today, "dd. MMM. yyyy") = 15. Aug. 2011</u> $x > 0$
<u>FresnelS(x)</u>	Fresnel's sine integral	$\forall x$
<u>FresnelC(x)</u>	Fresnel's cosine integral	$\forall x$
<u>J0(x)</u>	Bessel's function of 1st kind	$x \geq 0$
<u>Y0(x)</u>	Bessel's function of 2st kind	$x \geq 0$
<u>I0(x)</u>	Bessel's function of 1st kind, modified	$x > 0$
<u>K0(x)</u>	Bessel's function of 2st kind, modified	$x > 0$
<u>BesselJ(x,n)</u>	Bessel's function of 1st kind, nth order	$x \geq 0, n = 0, 1, 2, 3 \dots$
<u>BesselY(x,n)</u>	Bessel's function of 2st kind, nth order	$x \geq 0, n = 0, 1, 2, 3 \dots$
<u>BesselI(x,n)</u>	Bessel's function of 1st kind, nth order, mod.	$x > 0, n = 0, 1, 2, 3 \dots$
<u>BesselK(x,n)</u>	Bessel's function of 2st kind, nth order, mod.	$x > 0, n = 0, 1, 2, 3 \dots$
<u>HypGeom(x,a,b,c)</u>	Hypergeometric function	$-1 < x < 1 - a, b > 0 - c \neq 0, -1, -2 \dots$

PolyCh(x,n)	Chebyshev's polynomials	$\forall x$, orthog. for $-1 \leq x \leq 1$
PolyLe(x,n)	Legendre's polynomials	$\forall x$, orthog. for $-1 \leq x \leq 1$
PolyLa(x,n)	Laguerre's polynomials	$\forall x$, orthog. for $0 \leq x \leq 1$
PolyHe(x,n)	Hermite's polynomials	$\forall x$, orthog. for $-\infty \leq x \leq +\infty$
AiryA(x)	Airy function Ai(x)	$\forall x$
AiryB(x)	Airy function Bi(x)	$\forall x$
Ell1(x)	Elliptic integral of 1st-kind	$\forall \phi$, $0 < k < 1$
Ell12(x)	Elliptic integral of 2st-kind	$\forall \phi$, $0 < k < 1$
Erf(x)	Error-Gauss's function	$x > 0$
gamma(x)	Gamma-function	$\forall x, x \neq 0, -1, -2, -3, \dots$ ($x > 172$ overflow-err)
gammaLn(x)	Logarithm Gamma function	$x > 0$
gammaI(a,x)	Gamma-Incomplete function	$\forall x, a > 0$
digamma(x) psi(x)	Digamma-function	$x \neq 0, -1, -2, -3, \dots$
beta(a,b)	Beta-function	$a > 0, b > 0$
betaI(x,a,b)	Beta-Incomplete function	$x > 0, a > 0, b > 0$
Ei(x)	Exponential integral	$x \neq 0$
Ein(x,n)	Exponential integral of n-order	$x > 0, n = 1, 2, 3, \dots$
zeta(x)	zeta Riemman's function	$x < 1$ or $x > 1$
Clip(x,a,b)	Clipping function	return a if $x < a$, return b if $x > b$, otherwise ret x.
WTRI(t,p)	Triangular-wave (Dreieck-Kurve)	t = time, p = period
WSQR(t,p)	Square wave (Rechteck-Kurve)	t = time, p = period
WRECT(t,p,d)	Rectangular wave	t = time, p = period, d = duty-cycle
WTRAPEZ(t,p,d)	Trapez. wave	t = time, p = period, d = duty-cycle
WSAW(t,p)	Saw wave (Sägezahn)	t = time, p = period
WRAISE(t,p)	Rampa wave	t = time, p = period
WLIN(t,p,d)	Linear wave	t = time, p = period, d = duty-cycle
WPULSE(t,p,d)	Rectangular pulse wave	t = time, p = period, d = duty-cycle
WSTEPS(t,p,n)	Steps wave	t = time, p = period, n = steps-number
WEXP(t,p,a)	Exponential pulse wave	t = time, p = period, a = dumping factor
WEXPB(t,p,a)	Exponential bipolar pulse wave	t = time, p = period, a = dumping factor
WPULSEF(t,p,a)	Filtered pulse wave	t = time, p = period, a = dumping factor
WRING(t,p,a,fm)	Ringing wave	t = time, p = period, a = dumping factor, fm = frequency
WPARAB(t,p)	Parabolic pulse wave	t = time, p = period
WRIPPLE(t,p,a)	Ripple wave	t = time, p = period, a = dumping factor
WAM(t,fo,fm,m)	Amplitude modulation	t = time, p = period, fo = carrier freq., fm = modulation freq., m = modulation-factor
WFM(t,fo,fm,m)	Frequency modulation	t = time, p = period, fo = carrier freq., fm = modulation freq., m = modulation factor
Year(d)	year	d = dateserial
Month(d)	month	d = dateserial
Day(d)	day	d = dateserial
Hour(d)	hour	d = dateserial
Minute(d)	minute	d = dateserial
Second(d)	second	d = dateserial
DateSerial(a,m,d)	Dateserial from date	a = year, m = month, d = day
TimeSerial(h,m,s)	Timeserial from time	h = hour, m = minute, s = second
time#	system time	
date#	system date	
now#	system timestamp	
Sum(a,b,...)	Sum	sum(8,9,12,9,7,10) = 55
Mean(a,b,...)	Arithmetic mean	mean(8,9,12,9,7,10) = 9.166666666666667
Meanq(a,b,...)	Quadratic mean	meanq(8,9,12,9,7,10) = 9.30053761886914
Meang(a,b,...)	Arithmetic mean	meang(8,9,12,9,7,10) = 9.03598945281812
Var(a,b,...)	Variance	var(1,2,3,4,5,6,7) = 4.666666666666667
Varp(a,b,...)	Variance pop.	varp(1,2,3,4,5,6,7) = 4
Stdev(a,b,...)	Standard deviation	Stdev(1,2,3,4,5,6,7) = 2.16024689946929
Stdevp(a,b,...)	Standard deviation pop.	Stdevp(1,2,3,4,5,6,7) = 2
Step(x,a)	Haveside's step function	Returns 1 if $x \geq a$, 0 otherwise

Hinweise: {{Calc berechnet Winkel Funktionen in RAD (Pi == 360 Grad). Wenn mit Grad gearbeitet werden soll, muss der Wert umgewandelt werden. Beispiel: {{Calc sin(deg(90)). Resultat: 1. Für das 400 Grad System: grad(100)

Wichtig: {{Calc behandelt den Dezimalpunkt etwas speziell. Die Eingabewerte für {{Calc müssen einen Dezimal Punkt (.) als Dezimalpunkt enthalten. Beispiel {{Calc 1.234 + 2.345}. Der Ausgabewert enthält den Dezimalpunkt wie er in den Regional-Einstellungen definiert ist. Beispiel Deutschland: 3,579. Wenn man möchte, dass auch der Ausgabewert immer einen Punkt als Dezimalpunkt ausgibt, muss der Befehl {{Calc- verwendet werden. Mit {{Calc- sind aber keine Text Ausgaben möglich wie z.B Datum, Zeit oder Währung.- Die Einstellung wird aus der Betriebssystem Einstellung entnommen.- Regeln: Wenn der Dezimal Punkt ein Punkt ist (.) dann ist der Argument Separator ein Komma (,).- Wenn der Dezimal Punkt ein Komma ist, dann ist der Separator ein Strichpunkt (;)

Format-Befehl: Für mehr Format-Info siehe:

Limits: Max. 200 Operationen/Funktionen, Max Argumente per Funktion: 50, Max verschachtelte Funktionen: 50.

Für mehr Informationen siehe (in English)- [http://msdn.microsoft.com/en-](http://msdn.microsoft.com/en-us/library/0c899ak8.aspx)

us/library/0c899ak8.aspx<http://www.kratronic.com/tools/recorder/manual/Calc.pdf> (Zahlen) (in Englisch)

<http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx> (Datum / Zeit) (in Englisch)

Einschränkung: Format benötigt eine Zahl vor dem Dezimalpunkt: Falsch: format(.234,"##.0"), Richtig: format(0.234,"##.0"). Um diese Null zu erhalten kann man eine Textvariable verwenden und am Anfang dieser Variable eine "0" hinzufügen.

1.10.3 Datei Import

Die drei Befehle: {{ImportFile, {{ImportSeparator und {{DoImport ermöglichen die Verwendung von Listen in einer Datei. Beispiel: Während der Nacht sollen 100 Grafik-Dateien ausgedruckt werden. Dazu wird zuerst mit dem DOS Befehl `dir /b > file.txt` eine Liste erstellt, welche alle Datei-Namen enthält. Mit Hilfe der Makrowiederholung und dem folgenden Befehl können nun alle 100 Grafik-Dateien geöffnet (und mit weiteren Befehlen ausgedruckt) werden: {{Start C:\GrafikProgramm.exe {{DoImport}}

Prinzip: Eine Zeile des ImportFile kann mehrere Teile (Felder) haben. Die Felder sind unterteilt mit dem {{ImportSeparator. Jedes {{DoImport während einer Makro-Wiederholung liest ein Feld. ({{DoImport 1: Erstes Feld, {{DoImport 2: Zweites Feld, usw.) Ein ImportFile kann mehrere Zeilen enthalten. Jede Makro-Wiederholung (Schleufe) arbeitet auf einer Zeile. (Wiederholung 1: Zeile eins, Wiederholung 2: Zeile zwei, usw.)

Befehls Beschreibung

`{{1}}{{ImportFile C:\DateiName.Txt}`. Dieser Befehl definiert welche Datei als Quelle verwendet werden soll. **Wichtig:** Bei Verwendung von Schleifen muss immer ein {{1} (oder eine andere Zahl) verwendet werden, damit die Datei nur einmal geladen wird. Natürlich kann man auch mehr als eine Import-Datei verwenden. Beispiel: `{{50}}{{ImportFile C:\DateiName2.Txt}`. Ab der 50. Wiederholung wird die zweite Datei als Quelle verwendet.

Einschränkung: Es kann immer nur eine Importdatei aufs Mal verwendet werden. Der Befehl {{DoImport} verwendet immer die Datei des zuletzt ausgeführten {{ImportFile.

Die Import-Datei muss folgendermassen aufgebaut sein:

- Teil_a1,Teil_a2,Teil_a3,...
- Teil_b1,Teil_b2,Teil_b3,...

Es können so viele Teile wie nötig verwendet werden. Normalerweise werden die einzelnen Teile mit einem Komma separiert. Der Separator kann mit dem Befehl {{ImportSeparator @}. Am Ende der Zeile ist die Verwendung des Separators freiwillig. (Er kann dort auch weggelassen werden.) **Wichtig:** Ein Tabulator Zeichen kann nicht als Separator verwendet werden. Lösung: Anderes Zeichen verwenden oder mittels Microsoft Word Befehl: 'Suchen und Ersetzen' den Tabulator in ein anderes Zeichen wandeln (z.B. §)

Wenn `{{DoImport}}` öfters auftritt als Felder vorhanden sind, wird kein Fehler ausgegeben, sondern leere Texte (""). Ein leerer Text kann mit der Konstante: `#emptyX5#` getestet werden. Beispiel: `{{If {{Var1}} = #emptyX5# ;{{Beep;}}` (Bei `#emptyX5#` ist Gross-Kleinschreibung relevant)

Ein einzelner Leerschlag kann mit der Konstante: `#spaceX5#` getestet werden. Beispiel: `{{If {{Var1}} = #spaceX5# ;{{Beep;}}` (Bei `#spaceX5#` ist Gross-Kleinschreibung relevant)

Der Befehl `{{DoImport}}` liest nun den ersten Teil aus der ersten Zeile aus. Ein zweites `{{DoImport}}` liest den zweiten Teil. Eine neue Makro-Wiederholung beginnt mit einer neuen Zeile.

Beim obigen Beispiel passiert folgendes:

Während Makro-Wiederholung 1 wird Teil_a1 beim ersten Erscheinen von `{{DoImport}}` gelesen. Das zweite `{{DoImport}}` in der gleichen Wiederholung liest Teil_a2... etc.

Während Makro-Wiederholung 2 wird Teil_b1 beim ersten Erscheinen von `{{DoImport}}` gelesen. Das zweite `{{DoImport}}` in der gleichen Wiederholung liest Teil_b2... etc.

Hinweis:

Wenn mit dem Befehl `{{DoImport}}` Spezialzeichen eingelesen werden (Plus-Zeichen (+), Dächrlein (^), Prozent-Zeichen (%), Tilde (~), Klammern (), Eckige Klammern ([]) und geschwungene Klammern ({})) können diese nicht einfach normal ausgegeben werden. (Diese Zeichen haben Spezialfunktionen und werden nicht direkt ausgegeben). Lösung: Diese Texte in das ClipBoard (`{{SetClipboard}`) speichern und mit CTRL v (STRG **ve**) ausgeben.

Zusätzlicher Befehl:

In der Version 4.3 wurde der Befehl `{{ImportNextLine}}` neu hinzugefügt. Dieser Befehl erlaubt es, ohne Makro-Wiederholung auf die nächste Zeile einer Import-Datei umzuschalten. Beispiel:

```
{{ImportFile c:\MeineListe.txt
{{Label Start
{{MessageBox {{DoImport}}
{{If {{ImportNextLine}} = {{True}} ;{{Goto Start;{{Goto Ende}}
{{Label Ende
```

Der Befehl `{{ImportFile}}` liest bereits die erste Zeile der Import-Datei. Damit kann der Befehl `{{MessageBox {{DoImport}}` auch bereits das erste Feld anzeigen. `{{ImportNextLine}}` schaltet dann zur nächsten Import-Datei Zeile um. (Falls es keine weitere Import-Datei Zeile mehr gibt, springt der Recorder zum Label Ende)

Wenn `{{ImportNextLine}}` ohne `{{If}}` angewendet wird, erscheint eine Fehlermeldung, wenn keine weiteren Zeilen in der Import-Datei mehr vorhanden sind.

Wenn `{{ImportNextLine}}` verwendet wird, verursacht eine Makro-Wiederholung kein Umschalten mehr auf die nächste ImportDatei-Zeile. Dadurch kann man `{{ImportNextLine}}` und Makro-Wiederholungen gemeinsam verwenden.

Eine Zeile anspringen: Mit einem Parameter nach dem Befehl `{{ImportNextLine 5}}` kann direkt die gewünschte Zeile angesprungen werden. (Hier Zeile 5). Beispiel:

```
{{SetVarStartZeile 5
{{ImportFile c:\MeineListe.txt
{{SetVarTemp {{ImportNextLine}} {{VarStartZeile}}
{{Label Start
{{MessageBox {{DoImport}}
{{If {{ImportNextLine}} = {{True}} ;{{Goto Start;{{Goto Ende}}
{{Label Ende
```

Hinweis: Nach dem Ausführen des Befehles `{{ImportFile}}` ist die entsprechende Datei blockiert. Sie kann nicht gelöscht werden und es kann auch nichts in diese Datei geschrieben werden. Erst wenn das Makro endet oder wenn mit `{{ImportFile}}` eine andere Datei gewählt wird, wird die erste Datei wieder frei.

1.11 Makros bearbeiten

Allgemein **Makrocode** Geplante Tasks

Nach dem Aufnehmen eines Makros ist es möglich dieses Makro zu bearbeiten. Um ein Makro zu bearbeiten muss ev. das Register (Tab): 'Makrocode' angeklickt werden. Es erscheint dann das Makro-Bearbeitungs-Fenster. Der Makro-Editor unterstützt folgende Schnellzugriffs-Tasten: CTRL C/X und CTRL V für kopieren/ausschneiden und einfügen von Text (STRG C/X, STRG V). CTRL SHIFT HOME/END kann benutzt werden um den ganzen Text ab der aktuellen Cursorposition zu markieren. (STRG UMSCHALT ANFANG/ENDE). CTRL A markiert den ganzen Text von Anfang bis Ende.

Folgende interne Schnellzugriffs-Tasten werden vom Makro-Code-Editor unterstützt: Suchen und ersetzen (CTRL/STRG f), nochmals suchen (F3). Mit F9 = {{, F11 = {{ und F12 = } ist eine schnellere Befehlseingabe möglich. CTRL z (STRG z) bewirken ein Rückgängig machen der letzten Änderung. (Undo).

Fehler-Such-Unterstützung: Wenn ein Makro einen Fehler hat, wird die fehlerhafte Zeile mit einem ==> im Makro Hauptfenster angezeigt. Nach dem Öffnen des Makro Code Fensters steht der Positionszeiger ebenfalls vor der fehlerhaften Zeile. Ein Makroabbruch (oder {{Stop} erzeugt auch einen ==>. Das bedeutet, dass man das Makro in einer kritischen Situation abbrechen kann und genau sieht, wo gestoppt wurde. Beim nächsten Speichern oder Ausführen des Makros wird der Fehler-Pfeil automatisch wieder entfernt.

Beim Bearbeiten von Makros müssen bestimmte Regeln eingehalten werden.

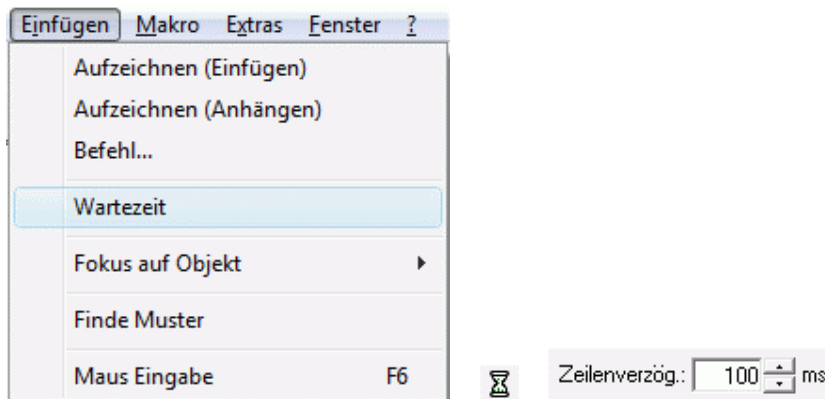
Das Plus Zeichen (+), das Dächlein (^), das Prozent Zeichen (%), das Tilde (~), Klammern (), eckige Klammern ([]) und geschwungene Klammern ({}) müssen mit geschwungenen Klammern eingefasst werden. Beispiel: {+}

Wenn CTRL (STRG), ALT und SHIFT (UMSCHALT) mit mehreren anderen Tasten kombiniert werden sollen, müssen diese Tasten in Klammern gesetzt werden. Beispiele: "+(ec)" = SHIFT (UMSCHALT) zusammen mit den beiden Tasten e und c. "+ec" = SHIFT (UMSCHALT) zusammen mit Taste e und nach dem Loslassen die Taste c. Tastenwiederholungen werden folgendermassen geschrieben: {LEFT 42} = Taste nach links 42 mal betätigen. {h 10} = Den Buchstaben h 10 mal schreiben.

Die Befehle {{1}, {{2}...{{n} müssen immer am Anfang einer Zeile platziert werden. Befehle welche eine Zahl (oder Text) liefern, können irgendwo in der Zeile stehen. Beispiele: {{Var1}, {{RandomNumber 1,10} etc. Befehle welche nichts liefern, müssen am Anfang der Zeile stehen (oder nach einem {{1}). Beispiele: {{SetVar1 10}, {{WriteLineToFile C:\Temp.txt, meinText}.

Hinweise: Zusätzlichen Anführungszeichen (") werden automatisch in Ersatz-Zeichen umgewandelt (å). Dieses Zeichen kann man umdefinieren. Dies ist allerdings nur in speziellen Fällen nötig. Das Ersatz-Zeichen muss in der verwendeten Sprach-Datei im Eintrag: QuotationMark=å geändert werden. Die Sprachdateien befinden sich im Recorder-Unterverzeichnis NLS.

1.11.1 Befehls-Verzögerungen



- Der Sanduhr-Knopf 'Wartezeit einfügen' (resp. Menü: 'Einfügen' / 'Wartezeit') erlaubt es Wartezeiten zwischen zwei Befehlen einzufügen. Das ist sehr nützlich bei Aktivitäten, bei denen der Recorder auf andere Programme warten muss. Beispiele: Warten bis Grafik-Konvertierung fertig ist, warten bis Netzwerkverbindung steht. Für eine genauere Synchronisation können die drei Befehle `{{WaitForScreenChange`, `{{WaitForFocus`, and `{{WaitForEndFocus` sowie `WaitForScreenChange` und `{{FindPattern` verwendet werden. Wartezeiten können auch mit Schnellzugriffs-Tasten eingefügt werden. Einzelheiten siehe am Ende des Kapitels: *Makro-Fenster und Tasten Eingabe*.

Die Zeilen-Verzögerung erlaubt es, generell zwischen jeder Zeile eine bestimmte Zeit zu warten. Dies ist besonders praktisch um ein Makro auszutesten. Wenn man z.B. diese Verzögerung auf 2000 ms (2s) setzt, kann man in aller Ruhe die Arbeit des Makros beobachten und bei Fehlern den Recorder rechtzeitig stoppen. (Mit dem Knopf 'Stoppen' oder der Abbruch-Schnellzugriffs-Taste: ALT-Pause/Break).

- ~~Hinweis: In Windows 98 und ME hat die Verzögerung eine Auflösung von 55 ms. Das bedeutet, dass eine Einstellung von 10 ms eine Verzögerung von 55 ms auslöst. (60 ms ergeben 110 ms). Die moderneren Betriebssysteme haben eine Auflösung von 1 ms.~~

1.11.2 Verzögertes Starten

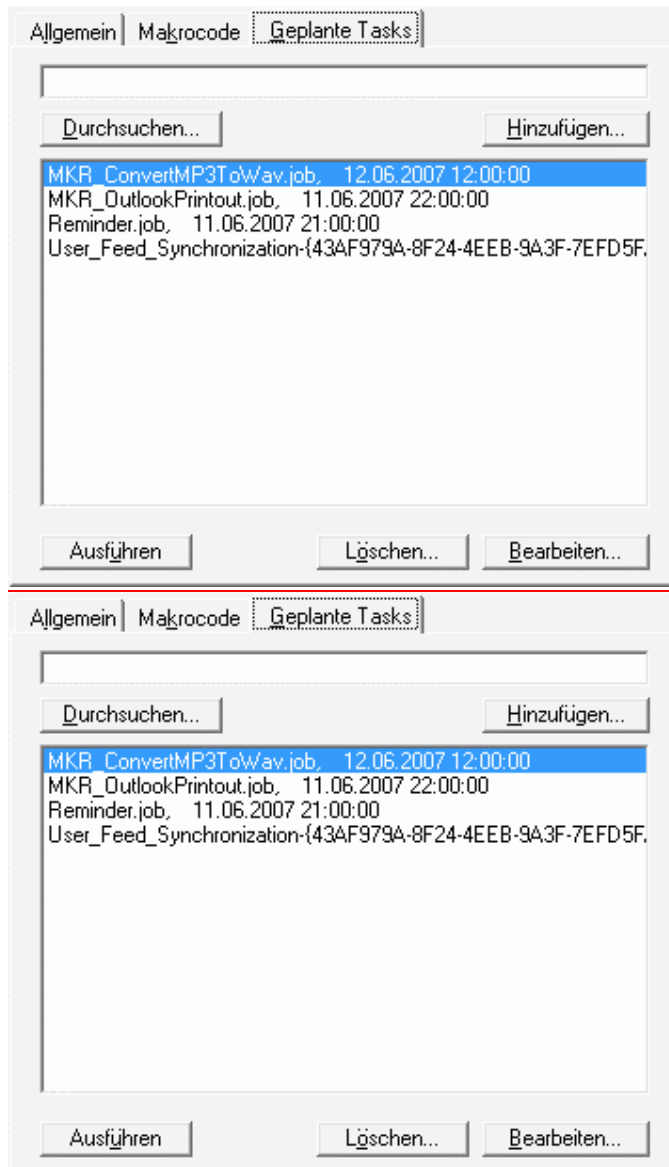


Der Schalter 'Wiedergabe starten' ermöglicht es, ein Makro erst später ablaufen zu lassen. Dies ist praktisch, wenn man z.B. einen grossen Druckauftrag erst in der Nacht laufen lassen will. Vorgehen: 'Wiedg. starten' aktivieren, Zeit einstellen und Knopf 'Abspielen' klicken.

Beispiel: +Tag = 2, Stunden = 03, Minuten = 15. Der Knopf 'Abspielen' wird vom Benutzer am Montag um 18:15 gedrückt. Das Makro startet am Mittwoch früh um 3:15. (Tage wechseln um Mitternacht). Hinweis: Der Wert Tage+ ist nicht der Wochentag, sondern die Anzahl Tage die der Recorder warten soll.

1.11.3 Verwendung von 'Geplanten Tasks' (Scheduler)

Wenn ein Makro periodisch gestartet werden muss, kann das Register 'Geplante Tasks' eingesetzt werden.



Hinweis: Bei Windows Vista, Windows 7 und neuer erscheint das mittlere Fenster (mit den Zeitschalteneinträgen) nicht mehr. Stattdessen wird beim anklicken der Knöpfe direkt das original Windows "Geplante Tasks" Fenster geöffnet. Dort kann dann der Makro-Task eingetragen, geändert, gestartet oder gelöscht werden.

Der Scheduler (Zeitschaltprogramm) verwendet die Windows-Funktion "Geplante Tasks". Darum erscheinen im Terminplan-Fenster auch Recorder-fremde Einträge. Um ein Makro regelmässig (oder auch nur einmal) automatisch laufen zu lassen, muss es in den 'Geplanten Tasks' eingetragen werden. Dazu wird zuerst mit 'Durchsuchen...' das Makro ausgewählt. Mit 'Hinzufügen...' öffnet sich das offizielle Windows "Geplante Tasks" - Fenster wo man die Zeit-Einstellungen machen kann. Im Terminplaner-Fenster wird nach der Eingabe die Zeit angezeigt, wenn das Makro das nächste mal ausgeführt wird. Der Knopf 'Ausführen' lässt ein sofortiges Ausführen zu.

Wenn die im Eintrag geplante Zeit erreicht ist, startet der Scheduler den 'Maus und Tastatur Recorder' und dieser startet dann das Makro.

Hinweise: Bei Windows 2000, XP und höher würde es **eigentlich erlaubendie Funktion** "Geplante Tasks" **-erlauben**, Einträge auch dann ablaufen zu lassen, wenn man nicht eingelogged ist, oder Tasks unter einem anderen Benutzer laufen zu lassen. Leider kann man dies bei den meisten Makros **nicht anwenden**. Grund dafür ist, dass keine Maus Klicks und Tasten-**DrückeSchläge** ausführbar sind, wenn keine Oberfläche vorhanden ist. (Geplante Tasks laufen im Hintergrund ohne eigene Oberfläche wenn

niemand eingeloggt ist, oder wenn die Tasks unter einem andern Benutzer laufen). Makros ohne Maus Klicks und Tasten- **DrückeSchläge** können im Hintergrund laufen gelassen werden.

Ein Makro welches durch einem wiederholenden Task gestartet wurde, benötigt am Ende ein {{Exit. Wenn der Recorder noch läuft, kann der nächste Task den Recorder nicht mehr starten.

Sicherheits-Aspekte: Wie oben erwähnt, darf man nicht ausloggen, wenn der Recorder z.B. um Mitternacht ein Makro ausführen muss. Auch ein Bildschirmschoner mit Passwort kann man nicht verwenden. (Der Bildschirmschoner lässt sich vom Recorder nicht öffnen. CTRL ALT DEL kann aus Sicherheitsgründen nur über die echte Tastatur eingegeben werden.) Das bedeutet, dass der Rechner die ganze Nacht offen ist für alle die Zugang zum PC haben.

Unter folgender Adresse kann ein spezialisierter Bildschirmschoner für die Verwendung mit dem Recorder heruntergeladen werden: www.kratronic.com/tools/recorder/sw/ScreenLock/ScreenLock.zip

Dieser Bildschirmschoner verhindert zuverlässig jeden Zugang zum PC, kann aber vom Recorder selber aufgeschlossen und nach dem Makroablauf wieder verschlossen werden. Kleiner Nachteil: Während dem Makro-Ablauf ist der PC offen. Dies kann aber nicht verhindert werden, da ja der Recorder einen Menschen an der Tastatur / Maus simuliert.

Wichtig: Bei Windows Clients die auf VMWare laufen, lässt sich ScreenLock aufbrechen. Daher ScreenLock nicht für Windows auf VMWare verwenden, wenn der Zugriffsschutz wichtig ist.

Windows XP/Vista/Win7: Windows XP, Vista und Windows 7 erlauben es Benutzer ohne Passwort aufzusetzen. Die Windows Funktion "Geplante Tasks" benötigt aber einen Benutzer mit Passwort. (Einträge für Benutzer ohne Passwort kann die Funktion "Geplante Tasks" nicht starten!)

Lösungsmöglichkeiten:

- Benutzer mit einem Passwort versehen. Nachteil: Beim Einloggen muss immer das Passwort angegeben werden.
- Einen anderen Scheduler verwenden. Nachteile: Das Planen von Tasks ist nicht mehr so einfach.

Empfehlung: Benutzer mit Passwort versehen und mittels dem Tool: "ALO" wieder ein automatisches Login erzwingen. Dieses kleine Tool kann von folgender Adresse heruntergeladen werden:

www.kratronic.com/tools/recorder/sw/ALO.exe

Probleme mit Geplanten Tasks: Auf bestimmten **XP-PCs** funktionieren die Geplanten Tasks nicht. Die Gründe können fehlende Zugriffsrechte oder Probleme in den Geplanten Task Komponenten sein. Während der Initialisierung liest der Recorder bestehende 'Geplante Tasks'. Wenn das Verzeichnis C:\Windows\Tasks einen Zugriffsschutz für den jeweiligen Benutzer hat, oder wenn die 'Geplante Tasks' - Programmteile einen Fehler haben, bricht der Recorder das Einlesen ab und zeigt im Register 'Geplante Tasks' eine Meldung.

Um die Zugriffsrechte zu ändern, siehe den Befehl **XCacls** in folgendem Microsoft Artikel:

<http://support.microsoft.com/kb/318754/technet/prodtechnol/windowsserver2003/de/library/ServerHelp/3be8b68f-f83c-409b-8b97-f026f75e17ca.mspx>. - (Bei der Verwendung dieses Befehles Vorsicht walten lassen, erst diese Aktivität mit dem IT abklären. **Beispiel: cacls c:\windows\tasks /e /g jeder:f**)

Für die Programmteile von 'Geplante Tasks' sind Patches von Microsoft erschienen: (Windows **2000- und XP**)

<http://support.microsoft.com/kb/812400/de>

<http://support.microsoft.com/kb/830924/de>

Weitere Möglichkeit: Starten des Maus und Tastatur Recorders als Administrator (Rechtsklick auf Icon und dann diese Option wählen)

Wenn trotz allem die Funktion 'Geplanten Tasks' im Recorder nicht arbeiten will, dann kann man die gewünschten Einträge auch direkt in der Windows Aufgabenplanung eintragen. (START / Programme / Zubehör / Systemprogramme / Aufgabenplanung) Für die Kommandozeile siehe Kapitel: 'Befehlszeilen Funktion'

1.11.4 Maus Eingaben

Der 'Maus und Tastatur Recorder' hat zwei unterschiedliche Modus um Maus Klicks aufzunehmen:

- Absolute Maus Aufzeichnung

- Relative Maus Aufzeichnung (Grund-Einstellung)

Absolute Maus Aufzeichnung: In diesem Modus werden alle Maus Klicks unabhängig von der Situation auf dem Bildschirm aufgezeichnet. Das heisst, ein Mausclick oben links wird immer oben links ausgeführt, unabhängig davon wo sich das zu bedienende Fenster befindet.

Relative Maus Aufzeichnung: In diesem Modus merkt sich der Recorder, wo im aktuellen Fenster der Maus Klick erfolgte. Beispiel: Wenn im Microsoft Word auf das Icon Abspeichern geklickt wird, wird bei der Wiedergabe des Klicks das Icon immer getroffen, unabgänglich davon wo sich das Word-Fenster auf dem Bildschirm befindet (Die Maus läuft dem Fenster 'hinterher'). Damit das möglich ist, wird auch der Fensternamen mit abgespeichert. Beim Abspielen schaltet der Recorder erst (wie mit einem **Switch**) auf das Fenster um und klickt dann die Maus. Das bedeutet, dass der Fensternamen beim Abspielen gleich sein muss, wie bei der Aufnahme. Das Kapitel 'Optionen' zeigt einen Weg um diese harte Anforderung abzuschwächen. (Relativer Fenster Filter). Dieses Kapitel zeigt auch, wie man zwischen den beiden Maus Modus umschaltet.

Hinweise: Einige Programme erhalten beim Aufnehmen den Fokus zu spät nach dem Maus Klick. Dadurch wird ein falscher (oder kein) Fensternamen aufgezeichnet. Wenn das passiert, wird ein Ton ausgegeben. Empfehlung: Immer zuerst mit ALT TAB auf das Ziel-Fenster wechseln und erst dann mit der Maus arbeiten.

Einige wenige Optionsfenster benötigen nicht den eigenen Namen, sondern den Namen des Programms zu dem sie gehören. Bei diesen Programmen muss das Makro manuell nachgebessert werden.

Siehe auch Befehl **Switch** für weitere Optionen. (Neue Umschalt-Technik)

Maus Befehl Aufbau: Der Maus Befehl ist folgendermassen aufgebaut: **(Maus Klicken)**

{{Mouse Taste, Klick, horizontale Pixel, vertikale Pixel, Klick-Schalter, Alt/Ctrl-Strg/Shift-Umschalt. (1 = linke Taste, 2 = mittlere Taste, 3 = rechte Taste. 1 = Einzelklick, 2 = Doppelklick.)

Beispiele: **{{Mouse 3, 2, 430, 283, 0,+^%** 3 = rechte Taste, 2 = Doppelklick, 430 = Pixel horizontal, 238 = Pixel vertikal, + = Shift-Umschalt, ^ = Ctrl-Strg, % = Alt

Mit Maus ziehen und fallen lassen:

{{DragMouse 1, 1, 197, 307, 0, 371, 297 1 = linke Taste, 1 = Einzelklick, 197 = Pixel horizontal (1. Koordinate), 307 = Pixel vertikal (1. Koordinate), 371 = Pixel horizontal (2. Koordinate), 297 = Pixel vertikal (2. Koordinate). ~~Ein DragMouse Eintrag entsteht, wenn mit der Maus 'Ziehen und fallen lassen' gemacht wird.~~

Relative-Mouse Befehlsaufbau: Beim Aufzeichnen von Mausclicks werden normalerweise Relative-Mouse Befehle aufgezeichnet. Beispiel: **{{RelativeMouse Microsoft Excel, 1, 1, 81, 198, 0.**

Wichtig: Wenn der Fensternamen ein Komma enthält (Beispiel **{{RelativeMouse Microsoft Excel - Erster,Zweiter,und Letzter.xls, 1, 2, 153, 302, 0**) dann verwendet der RelativeMouse Befehle falsche Koordinaten. In so einem Falle muss der Fensternamen gekürzt werden sodass er keine Kommas enthält. Beispiel: **{{RelativeMouse Microsoft Excel - Erster, 1, 2, 153, 302, 0.** Eine andere Möglichkeit besteht in der Verwendung eines Textspeichers. Beispiel **{{RelativeMouse {{StringFenster}, 1, 2, 153, 302, 0**

Hinweis: Führende und nachfolgende Leerschläge werden im **{{RelativeMouse** Befehl berücksichtigt. Beispiel: **: {{RelativeMouse Microsoft Excel, 1, 2, 153, 302, 0).** Die Leerschläge vor Microsoft Excel werden mitgenommen. Das heisst, wenn das wirkliche Fenster dort keine Leerschläge hat, wird das Fenster nicht gefunden. Das bedeutet, dass man auf die Leerschläge achten muss, oder dass man den Befehl **Trim** verwenden sollte.

SpezialSpecial Funktionen:

Die obigen Befehle werden beim Aufzeichnen von Maus-Aktivitäten automatisch erstellt. Durch das manuelle Ändern von einigen Parametern können die folgenden Spezial-Funktionen ausgeführt werden:

{{Mouse 1, 1, 430, 283, 1,+^%

Eine **1** an der Stelle der fetten Zahl: Nur Zeiger platzieren ohne Klick. Hinweis: Diese Funktion wirkt nur bei Einzelklicks. Doppelklicks werden trotz **1** normal ausgegeben.

{{DragMouse 1, 1, 197, 307, **1**, 371, 297, **50**

Eine fette **1** bedeutet: Mauszeiger langsam ziehen ohne klicken. Diese Funktion kann benutzt werden um den Weg der Maus anzuzeigen.

Eine fette **50** bedeutet: Ziehen mit halber Geschwindigkeit. (100 = normale Geschwindigkeit, 200 = doppelte Geschwindigkeit)

{{DragMouse 1, 1, 197, 307, **3**, 371, 297, **100**

Eine fette **3** bedeutet: Mauszeiger langsam ziehen mit klicken. Es gibt einige wenige Programme die das schnelle ziehen und fallenlassen (drag and drop) nicht mögen. Hier hilft diese Option. Die Fette 100 gibt wieder die Geschwindigkeit an. (100 = normale Geschwindigkeit)

{{Mouse 3, 2, 430, 283, **2** {{DragMouse 1, 1, 197, 307, **2**, 371, 297, **0**

Eine fette **2** bedeutet: Die erste Maus Koordinate wird "vergessen". {{Mouse klickt genau dort wo die Maus steht, {{DragMouse beginnt mit dem ziehen genau dort wo die Maus aktuell steht.

Hinweis: {{Mouse 1, 1, 430, 283, **1** (**2,3**) und {{Mouse 1, 1, 430, 283, **0** platzieren den Mauszeiger nicht immer an den gleichen Ort. Unterschied: Manchmal 1 Pixel. Grund: Die beiden Befehle bauen auf zwei unterschiedlichen Koordinaten Systemen auf. Das führt zu Rundungsfehlern.

~~Wenn ein Applikations-Fenster Name ein Komma enthält kann die Relative Maus Aufzeichnung nicht verwendet werden. In einem solchen Fall die Absolute Maus Aufzeichnung verwenden, oder nur den Teil des Applikations-Fensters ohne Komma verwenden.~~

Die Maus-Aufzeichnung unterscheidet automatisch zwischen Klicken und Ziehen. Standardmässig ist ein Ziehen von mehr als 3 Pixel ein Ziehen, ein Ziehen von weniger als 3 Pixel wird als Klick aufgezeichnet. Diese Grenze kann unter 'Extras' verändert werden (Fokus Toleranz).

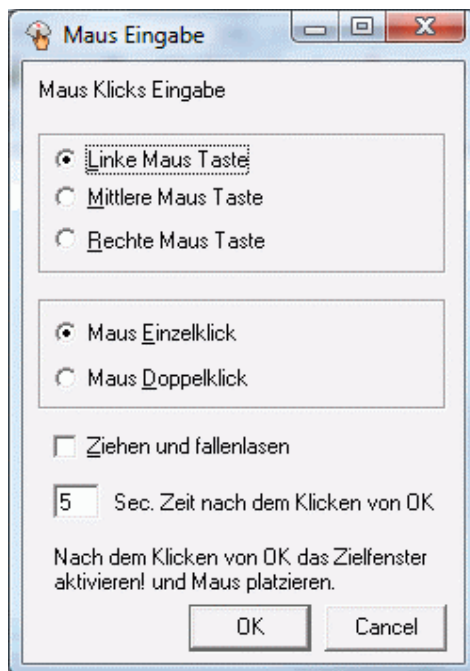
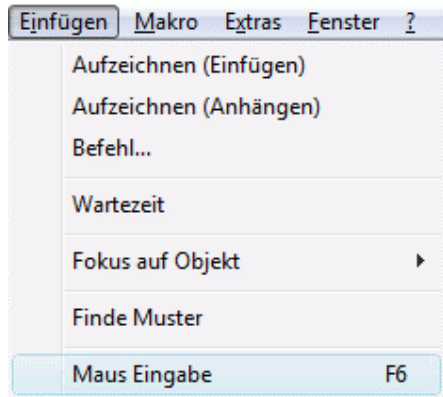
Der 'Maus und Tastatur Recorder' zeichnet nur die Befehle für die Maus-Klicks auf und nicht den ganzen Maus-Weg. Dies ergibt kleinere Makros, welche einfacher bearbeitet werden können.

Wenn ein Mauszeiger nur auf dem Bildschirm platziert werden (oder über den Bildschirm fahren) soll (ohne klicken), dann kann der Schalter: 'Maus-Klicks für Mausclicks und Mausziehen unterdrücken' im Menü: 'Extras' / 'Optionen' Register: 'Aufzeichnen' eingeschaltet werden. Dieser Schalter wirkt sich auf das Maus-positionieren und das Maus-ziehen aus. Beim Abspielen wird der Mauszeiger ohne klicken positioniert oder fährt langsam von der Start-Position zur End-Position. (ohne zu klicken).Die Geschwindigkeit des Mauszeigers (ziehen) kann nachträglich verändert werden zwischen 2 (2% der Original Geschwindigkeit) bis 10'000 (100 mal schneller als Original-Geschwindigkeit). Beispiel: {{DragMouse 1, 1, 82, 64, **1**, 584, 774, **50** = 50% der Original Geschwindigkeit.

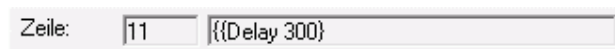
Maus-Unten-Zeit: Bei Spielen ist es hin und wieder nötig, die Maustaste für längere Zeit zu drücken. Der Befehl {{MouseDownTime erlaubt es diese Zeit zu setzen. Beispiel: {{MouseDownTime 3000. Jeder Mausclick hält nun die Maustaste für 3 Sekunden unten. {{MouseDownTime 0 stellt die Maus-unten Zeit wieder auf den Normalwert zurück.

Maus Aufzeichnung in Windows Vista/Win7:

Unter Windows Vista/Win7 ist ein Aufzeichnen von Tastendrücken und Mausclicks im Browser (z.B. Internet Explorer) nicht mehr möglich. (Massnahme gegen Viren und Trojaner welche Pishing anwenden wollen). Um trotzdem Mausclicks aufzeichnen zu können, wurde der Menu Befehl: 'Einfügen' / 'Maus Eingabe' (Schnelltaste F6) hinzugefügt.



1.11.5 *Status Information*



Am Fuss des 'Maus und Tastatur Recorders' befinden sich Status Informationen. Es sind dies ein Befehls-Zähler und eine Status Zeile. Der Befehlszähler zeigt die Befehlszeile, die Statuszeile zeigt alle 1 ½ Sekunden den gerade abgearbeiteten Befehl an.

1.11.6 *Befehlszeilen Funktion (Command Line)*

Der 'Maus und Tastatur Recorder' unterstützt die Verwendung von 'Befehlszeilen' Parametern. (Command Line) Die folgende Befehls-Zeile startet den Recorder, lädt das Makro 'MeinMakro' und führt es gleich aus: "C:\Programme\KraTronic\Recorder\RECORDER.EXE" "C:\My Documents\MeinMacro.krc".

Diese Funktion ermöglicht es, ein Makro direkt mit einer Verknüpfung (Ikon) zu starten.

Anwendungsmöglichkeiten: Ein Programm benötigt immer am Anfang spezielle Einstellungen wie Schriftart, Vorlageauswahl, etc. Das Programm wird nun über den 'Maus und Tastatur Recorder' (welcher per Icon gestartet wird) gestartet. Der Recorder macht dann jeweils automatisch die nötigen Einstellungen. Wie erstellt man ein Icon: Auf dem Desktop rechte Maustaste, NEU, VERKNUEPFUNG, die obige (oder irgendeine) Befehlszeile eingeben, Namen des Icons eingeben.

Falls der 'Maus und Tastatur Recorder' während dem Command-Line (Befehlszeile) Makro-Ablauf im Hintergrund bleiben soll, kann dies durch die Einstellung 'Minimieren' in der Verknüpfung (Ikon) erreicht werden.

Falls nur eine Command-Line ohne Icon verwendet werden soll (z.B. aus Batch Datei) dann kann der Command-Line eine Option angehängt werden, welche den Recorder von Anfang an im Hintergrund behält: `-MKRMinimize`. Beispiel: "C:\Programme\KraTronic\Recorder\RECORDER.EXE" "C:\Eigene Dateien\MeinMacro.krc" -MKRMinimize.

Falls der Maus und Tastatur Recorder völlig versteckt werden soll, kann die Command Line Option: `-MKRHide` verwendet werden. (Befehl unterscheidet Gross-Kleinschrift) Der Recorder erscheint dann nicht, auch nicht in der Taskliste. Beispiel: "C:\Programme\KraTronic\Recorder\RECORDER.EXE" "C:\Eigene Dateien\MeinMacro.krc" -MKRHide. Wichtig: Weil der Recorder völlig versteckt ist, kann man ihn auch nicht nach dem Ende des Makros schliessen. Daher sollte jedes Makro das mit der Option `-MKRHide` gestartet wird, am Ende den Befehl `{{Exit` enthalten. `{{MessageBox2` darf nicht verwendet werden (Wird versteckt dargestellt).

Wenn ein Makro geladen aber nicht gleich gestartet werden soll, kann die Option: `-NoRun` verwendet werden. Beispiel: "C:\Programme\KraTronic\Recorder\RECORDER.EXE" "C:\Eigene Dateien\MeinMacro.krc" -NoRun.

Im weiteren kann mit einem externen Terminplaner (Scheduler) zur programmierten Zeit oder periodisch ein Makro gestartet werden. (Beispiel: Auslösung eines Backup-Makros immer um Mitternacht). Dazu wird in den Terminplaner die 'Befehlszeile' eingetragen.

Befehlszeilen Variablen Parameter

Es ist möglich mit der Command-Line Zahlen- und Textspeicher (Variablen) zu übergeben. Die folgenden Parameter sind möglich: `--Var0`, `--Var1` ... `--Var9` und `--String0`, `--String1` ... `--String9`
Beispiel: "C:\Programme\KraTronic\Recorder\RECORDER.EXE" "C:\Eigene Dateien\MeinMacro.krc --Var1 4711 --String2 Mein Text".

Das automatisch gestartete Makro 'MeinMacro.krc' kann nun die Zahlen- und Textspeicher: `{{Var1` und `{{String2` benutzen.

Wichtig: Der Dateiname muss immer an erste Stelle in einer Befehlszeile stehen.

1.11.7 Starten eines Makros mit einer Schnellzugriffs-Taste (Hotkey, Shortcut)

Es gibt zwei Möglichkeiten um den 'Maus und Tastatur Recorder' mit einer Schnellzugriffs-Taste zu starten. In einem Falle wird eine Verknüpfung (Ikon) verwendet, welcher eine Schnellzugriffs-Taste zugewiesen wird. Im zweiten Falle wird eine interne Schnellzugriffs-Tasten Tabelle verwendet.

1. Möglichkeit:

Um den 'Maus und Tastatur Recorder' (oder ein Makro) mit einer Schnellzugriffs-Taste (z.B. F2 oder CTRL/STRG s) starten zu können, muss diese Schnellzugriffs-Taste in der Verknüpfung (Ikon) definiert werden. Vorgehen: Rechtsklick auf Verknüpfung, dann 'Eigenschaften', 'Verknüpfung' und 'Tastenkombination'. Hinweis: Diese Schnellzugriffs-Tasten wirken bloss, wenn die Verknüpfung auf dem Arbeitsplatz (Desktop) platziert ist.

2. Möglichkeit:

Im Menü 'Extras' / 'Optionen' / Register: 'Allgemein' befindet sich ein Knopf: 'Makro Schnellzugriffstasten zuweisen'. Mit dieser Funktion können Schnellzugriffstasten definiert werden. (Siehe Kapitel 'Optionen' für eine genaue Beschreibung des Eintrages von Schnellzugriffstasten). Um die Schnellzugriffstasten zu aktivieren muss der Schalter: 'Makro Schnellzugriffstasten verwenden' eingeschaltet werden. Hinweis: Die Schnellzugriffstasten der 2. Möglichkeit wirken nur wenn der 'Maus und Tastatur Recorder' läuft.

Hinweis: Die Internen Schnellzugriffstasten wirken nicht wenn der Internet Explorer den Fokus hat. (Microsoft Massnahme gegen Phishing)

1.11.8 *Erstellung einer Verknüpfung (Shortcut)*

Die zwei vorherigen Kapitel erwähnten die Verwendung einer Verknüpfung. Eine solche Verknüpfung wird folgendermassen hergestellt:

Mit der rechten Maustaste auf den Arbeitsplatz klicken und 'Neu' und dann 'Verknüpfung' wählen. Im Feld 'Befehlszeile' die in Kapitel 'Befehlszeile Funktion' erwähnte Zeile einsetzen. Nun kann ein passender Name für die neue Verknüpfung gewählt werden. Nach dem Beenden des Assistenten mit der rechten Maustaste auf die neue Verknüpfung klicken. Dann kann bei 'Eigenschaften', 'Verknüpfung' und 'Tastenkombination' eine Schnellzugriffstaste eingegeben werden. Bei Ausführen kann gewählt werden, ob der Recorder normal oder minimiert starten soll.

Log-Buch-Funktion

Die Log-Buch-Funktion ist ein spezieller Modus, welcher ein Journal aller in alle Programme eingegebener Texte (und Mausclicks) macht. Es werden alle Buchstaben und Zahlen abgespeichert. Mit der Option -RECORDMOUSE werden auch Maus Clicks, Positions-Zeiger-Verschiebungen und Funktions-Tasten aufgezeichnet. Während der Log-Buch Funktion ist der 'Maus und Tastatur Recorder' unsichtbar. Diese Funktion ist sehr praktisch um alle Aktivitäten auf dem PC aufzuzeichnen. Im Falle eines Netzunterbruches oder wenn z.B. Word abstürzt, sind alle eingegebenen Texte trotzdem noch im Log-Buch vorhanden.

Um die Log-Buch Funktion zu benutzen, muss der Recorder mit einer 'Befehlszeile' gestartet werden. Beispiel:-

```
C:\Programme\Recorder\Recorder.exe" D:\Daten\LogDatei-LOGX4567-BACKSPACE-RECORDMOUSE-EXIT: Mein73Ende88
```

Parameter:

DateiName

Der Datei Name muss den vollen Pfad zur Log-Buch-Datei enthalten. Es ist keine Datei-Erweiterung nötig. Der Recorder hängt automatisch ein .txt an den Dateinamen an. Im obigen Beispiel ist der Datei-Name: D:\Daten\LogDatei

-LOGX4567

Das ist der Schalter, welcher die Log-Buch-Funktion einschaltet. Dieser Schalter muss immer gleich hinter dem Dateinamen eingesetzt werden. (Zwischen Datei-Name und Schalter ein Leerzeichen setzen)

-RECORDMOUSE

Mit diesem Parameter in der 'Befehlszeile' werden auch alle Maus-Aktionen und die Funktionstasten mit aufgezeichnet.

-KEEPFILEOPEN

Dieser Parameter veranlasst den Recorder die Logging-Datei offen zu behalten solange der Recorder läuft. Ohne diese Option wird die Logging-Datei bei jedem ENTER geöffnet und wieder geschlossen.

~~Öffnen und Schliessen der Datei verbessert die Sicherheit für den gespeicherten Text. Der Nachteil dieses Vorgehens ist, dass bei grossen Datenmengen in der Logging-Datei der PC langsamer wird. Das bedeutet, dass bei längerem Aufzeichnen in die Logging-Datei diese Option eingeschaltet werden sollte.~~

~~-BACKSPACE~~

~~Wenn dieser Parameter in die 'Befehlszeile' eingefügt wird, löscht die Rückwärts-Lösch-Taste den letzten Buchstaben in der gleichen Zeile. Frühere Zeilen werden nicht beeinflusst. Wenn die Rückwärts-Lösch-Taste im Log-Buch nichts löschen soll, muss dieser Parameter einfach weggelassen werden.~~

~~Hinweis: Die normale Lösch-Taste wirkt nicht auf das Log-Buch. (auch nicht wenn -BACKSPACE- verwendet wird)~~

~~-EXIT:~~

~~Dieser Parameter erlaubt die Definition eines Abbruch-Wortes. Wenn dieses Wort eingetippt wird, fragt der Recorder ob die Log-Buch Session beendet werden soll. Nach der Antwort JA endet der Recorder. Es wird empfohlen, ein ungewöhnliches Exit-Wort zu verwenden. Beispiel: -EXIT: Mein73Ende88. Dadurch wird nicht zufällig ein Beenden des Log-Buches ausgelöst.~~

~~Hinweise: Das Exit-Wort darf nicht länger als 32 Zeichen lang sein. Wenn der Parameter -EXIT- weggelassen wird, kann der Recorder vom Benutzer nicht beendet werden. (Der Recorder wird dann beim Herunterfahren des PC's beendet)~~

~~Der Log-Buch-Datei Name wird folgendermassen gebildet:~~

~~Am Anfang steht der Name welcher in der 'Befehlszeile' definiert wurde. Beispiel: LogDatei. Dann werden Datum und Zeit hinzugefügt. Abgeschlossen wird der Datei Name mit einer Endung .txt. Ein Log-Buch-Datei Name könnte folgendermassen aussehen:~~

~~LogDatei_2000.12.31_21.32.54.txt.~~

~~Bei jedem Start einer Log-Buch Session wird ein neuer Datei Name erzeugt.~~

~~Hinweise:~~

~~Für den Start des Recorders im Log-Buch-Modus sollte eine Verknüpfung (Ikon) erstellt werden welche die obige 'Befehlszeile' enthält.~~

~~Die Log-Buch-Verknüpfung kann in das AutoStart Verzeichnis gelegt werden. Dadurch startet das Log-Buch jedes Mal wenn der PC gestartet wird.~~

~~Der Recorder speichert im Log-Buch-Modus jedes Mal wenn Enter (Return) gedrückt wird, die vorher eingegebenen Tasten in die Log-Buch-Datei ab. Während dem Herunterfahren des PC's (oder bei Exit) wird auch noch die letzte Zeile abgespeichert.~~

~~Wenn im Hintergrund die Log-Buch-Funktion läuft, kann es bei einem Recorder im Vordergrund **beim Aufzeichnen** zu Störungen kommen (Tastendrücke werden nicht aufgezeichnet). Das Makro-Abspielen arbeitet aber auch dann problemlos.~~

1.11.9 Fehler-Such Log-Buch

Der 'Maus und Tastatur Recorder' erlaubt es mit den vielen Befehlen wie z.B. `{{If`, `{{Goto`, etc. komplexe Makros zu schreiben. Um die Fehlersuche zu erleichtern, enthält der Recorder eine Fehler-Such Log-Buch Funktion. Diese Funktion kann im Menü 'Extras' / Optionen' / Register: Abspielen aktiviert werden. (Schalter: 'Wiedergabe Logbuch'). Während dem Makro-Ablauf wird nun jede Zeile dokumentiert. Dabei werden pro Zeile zwei bis fünf Log-Zeilen erzeugt:

1. Zeile: Datum und Zeit, Sekunden seit Makro Start
2. Zeile: Makro-Zeile
3. bis 5. Zeile: Aufgelöste Makrozeile, d.h. die Inhalte der Nummer und Text-Speicher werden angezeigt.

Mit dem Knopf: 'Logbuch öffnen' kann das Log-Buch geöffnet werden.

~~Hinweise: Die Lauf-Zeitanzzeige 'Sekunden seit Makro Start' benutzt die Sekunden seit Mitternacht. Das bedeutet, dass ein Makro, das über Mitternacht läuft nach Mitternacht eine falsche Lauf-Zeitanzzeige liefert.~~

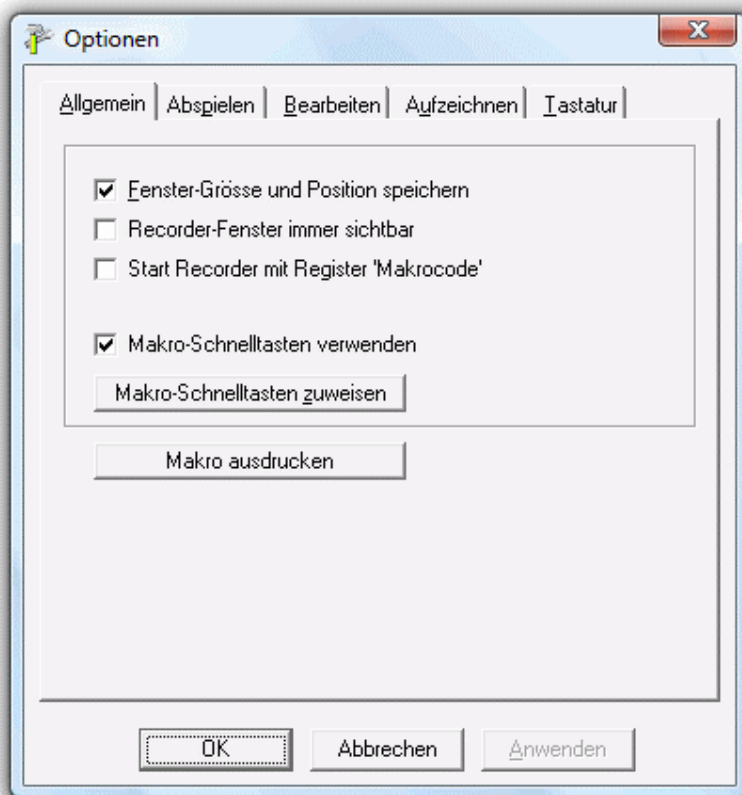
Siehe auch Befehl: `{{ResetLogFile}}`

1.11.10 *Debug Fenster*

Die Fehlersuche kann auch mit dem Debug Fenster (Entwanzungs-Fenster) unterstützt werden. Dieses wird mit dem Befehl `{{DebugOn}}` eingeschaltet. Bei jedem Lesen und Schreiben eines Speicherwertes wird der Name des Speichers, der Wert und der dazugehörige Befehl angezeigt. Die neuesten Einträge erscheinen immer zuoberst, die restlichen Einträge werden nach unten weggeschoben.

1.11.11 *Optionen*

Der 'Maus und Tastatur Recorder' enthält eine Anzahl von verschiedenen Optionen, welche es erlauben den Recorder an die eigene Arbeitsweise anzupassen. Die Optionen sind unter dem Menü 'Extras' erreichbar und sind in folgende 5 Register aufgeteilt: Allgemein, Abspielen, Bearbeiten, Aufzeichnen und SpracheTastatur.



Folgende Optionen stehen zur Verfügung:

Fenster Grösse und Position speichern:

Bei einem Neustart wird der 'Maus und Tastatur Recorder' an dem Platz und mit der Grösse, die er bei der letzten Anwendung innehatte, geöffnet. Mit diesem Schalter kann man dieses Verhalten abschalten.

Recorder Fenster immer sichtbar:

Dieser Schalter erlaubt es den Recorder auf der Oberfläche zu "fixieren". Dies ist besonders praktisch wenn man ein Makro testet und dabei sehen will, welche Befehle im Moment abgearbeitet werden.

Hinweis: Wenn vom Recorder eine MessageBox gestartet wird und diese unter dem Recorder erscheint, besteht keine Möglichkeit diese Box zu schliessen und man kann auch den Recorder nicht mehr beenden.. (In diesem Fall mit dem Taskmanager den Recorder abschliessen)

Start Recorder mit Register 'Macrocode':

Wenn man häufig neue Makros erstellt, ist diese Einstellung praktisch, da man nach dem Starten des Recorders direkt mit der Aufnahme beginnen kann (ohne erst das Register zu ändern)

Makro ausdrucken:

Dieser Knopf erlaubt es das aktuelle Makro auf dem Standard Drucker auszudrucken. (Listing)

Hinweis: Wenn das Makro auf einem anderen Drucker mit speziellem Format ausgedruckt werden soll, kann das Makro im Recorder mit CTRL/STRG a, dann CTRL/STRG c kopiert und in ein Microsoft Word Dokument mittels CTRL/STRG v eingefügt werden. In Word kann man dann einen andren Printer wählen oder andere Formate.

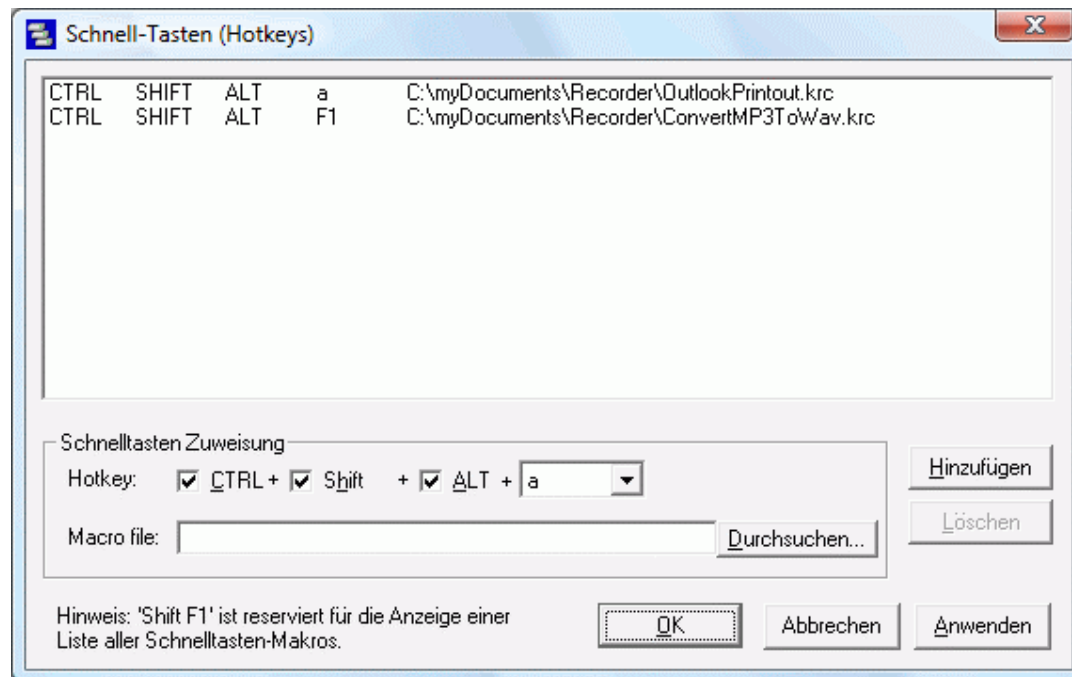
Makro-Schnellzugriffstasten ~~aktiviert~~ verwenden:

Dieser Schalter aktiviert die **Makro-Schnellzugriffstasten** (Hotkeys). Die Schnellzugriffstasten Diese erlauben es mit einer Tastenkombination ein Makro zu starten. Im Weiteren aktiviert dieser Schalter die Hotkeys für Stop, Pause, F11: {{ etc. Hinweis: Die Schnellzugriffstasten wirken nicht wenn der Internet Explorer den Fokus hat. (Microsoft Massnahme gegen Phishing)

Dieser Schalter hat einen kleinen Nebeneffekt. Die Sondertasten wie ^ und " verhalten sich anders. Normalerweise bewirkt der erste Tastendruck nichts, beim zweiten werden dann beide Tasten kombiniert. Beispiel: â, ë. Wenn der Schalter aktiviert ist (und der Recorder läuft), dann ergeben diese Tasten je zwei Zeichen, z.B. ^^, "" . Ist dies nicht erwünscht, muss der Schalter ausgeschaltet werden (oder der Recorder muss geschlossen werden)

Makro Schnellzugriffstasten zuweisen:

Nach dem Anklicken dieses Knopfes erscheint folgendes Fenster

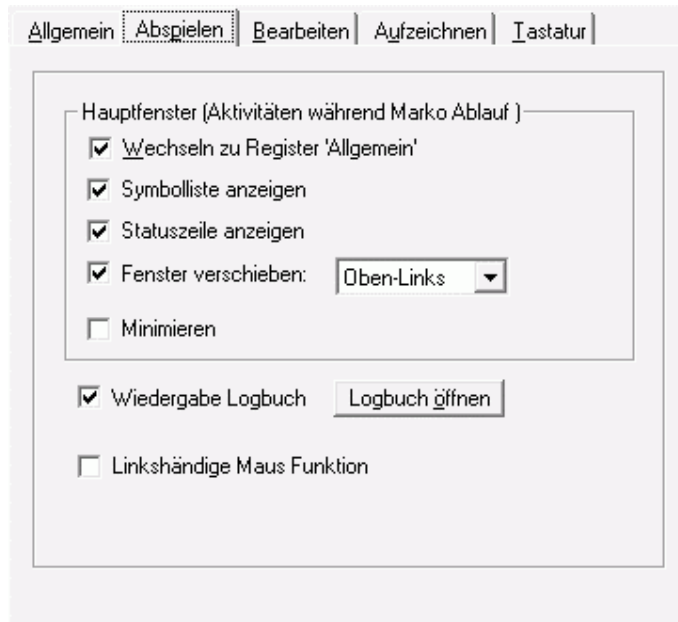
Neue Schnellzugriffs-Tasten hinzufügen:

Mittels 'Durchsuchen' ein Makro auswählen. Dann CTRL (entspricht: STRG), Shift (entspricht: Umschalt), Alt und Schnellzugriffs-Taste nach Bedarf einstellen und 'Hinzufügen' drücken.

Schnellzugriffs-Taste entfernen:

Eine Line auswählen. 'Entfernen' anklicken.

Einschränkung: Temporäre Makros können nicht in das Schnellzugriffs-Tasten Fenster eingefügt werden.



Wechseln zu Register Allgemein:

Während des Abspielens eines Makros wird vom Register Makrocode oder Terminplan zum Register Allgemein umgeschaltet. Nach Beendigung des Makros springt der Recorder wieder zum vorherigen Register zurück.

Symbolliste anzeigen:

Wenn dieser Schalter ausgeschaltet ist, wird während des Abspielens eines Makros die Symbolliste weggeschaltet.

Statuszeile anzeigen:

Wenn dieser Schalter ausgeschaltet ist, wird während des Abspielens eines Makros die Statuszeile weggeschaltet.

Fenster verschieben:

Während des Abspielens eines Makros wird das Recorder-Fenster in eine der Bildschirmecken verschoben.

Minimieren:

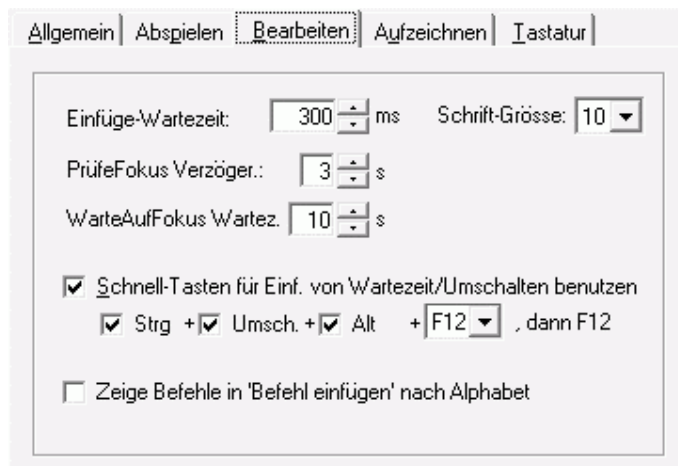
Während des Abspielens eines Makros wird das Recorder-Fenster minimiert (in die Taskliste)

Wiedergabe Logbuch:

Wenn dieser Schalter eingeschaltet ist wird während des Makro Abspielens ein 'Fehler-Such-Log-Buch' erstellt. Bei Verwendung von `{{ResetLogFile}}` wird ein zweites Logfile erstellt. Der Zugriff auf das alte Logfile erfolgt dann mit Log 2.

Linkshändige Maus Funktion:

Wenn die Windows Einstellungen für die Maus auf linkshändig eingestellt sind, muss auch hier diese Option aktiviert werden, damit der Recorder die richtigen Maus-Klicks ausgibt.

Einfüge-Wartezeit:

Dieser Wert wird verwendet, wenn mittels Menü: 'Einfügen' / 'Wartezeit' oder per Symbolliste (Sanduhr) ein {{Delay eingetragen wird.

Schriftgröße:

~~Dieser Wert beeinflusst die Schriftgröße im Makro-Editor~~

PrüfeFokus Verzögerung:

Dies ist die Zeit die verstreicht zwischen dem Drücken des Fokus Menü Befehls und dem Lesen des Fokus.

WarteAufFokus Wartezeit:

Vorgabewert für die maximale Zeit welche auf einen Fokus (oder EndeFokus) gewartet wird.

Schnellzugriffs-Tasten:

Hier können die diversen Schnellzugriffs-Tasten des 'Maus und Tastatur Recorders' an die eigenen Bedürfnisse angepasst werden. Die Grundeinstellung ist: CTRL-ALT-SHIFT-F12 (STRG-ALT-UMSCHALT-F12). Die Schnellzugriffs-Tasten können benutzt werden, um Verzögerungen und {{Switch Befehle einzugeben. Informationen über Verzögerungen sind in Kapitel 'Aufzeichnen von Makros' zu finden. Die {{Switch Schnellzugriffs-Taste arbeitet folgendermassen: Zuerst muss zum aktuellen Programm umgeschaltet werden, (ALT TAB) dann werden die folgenden Tasten gemeinsam gedrückt: CTRL-ALT-SHIFT-F12 (STRG-ALT-UMSCHALT-F12). Nach dem Loslassen dieser Tasten wird F12 gedrückt. Dadurch wird der {{Switch Befehl dem Makro hinzugefügt. Wenn die Schnellzugriffs-Tasten z.B. auf CTRL F3 (STRG F3) geändert werden sollen, muss für ein {{Switch folgendes getan werden: CTRL F3 (STRG F3) zusammen drücken und dann nach dem Loslassen F3.

Wichtig: Wenn die Schnellzugriffs-Tasten so gesetzt werden, dass nur noch eine Funktionstaste (F2) gedrückt werden muss, kann die Schnelltaste nicht mehr für das Einfügen eines {{Switch Befehls verwendet werden. Hinweis: Die Schnellzugriffstasten wirken nicht wenn der Internet Explorer den Fokus hat. (Microsoft Massnahme gegen Phishing)

Zeige Befehle nach Alphabet:

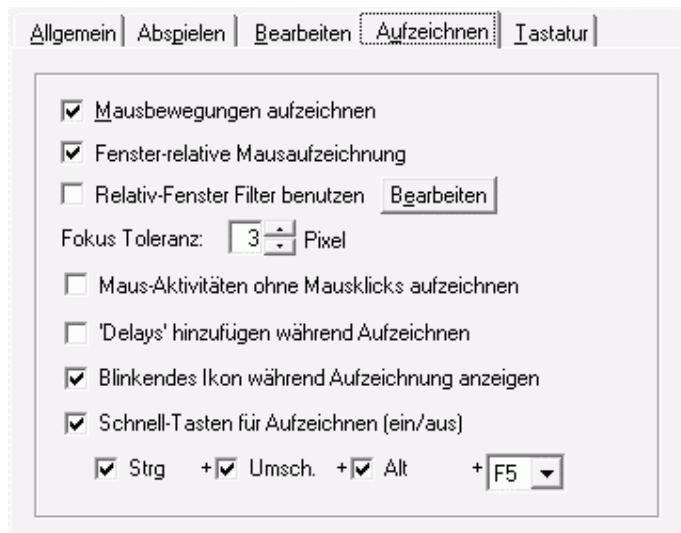
Das 'Befehl einfügen' Fenster zeigt die Befehle gruppiert. z.B sind {{Var und {{SetVar nebeneinander angeordnet. Einige Benutzer bevorzugen aber eine Anordnung nach Alphabet. Durch Einschalten dieser Option erscheinen die Befehle nach Alphabet geordnet.

Schriftgröße:

~~Dieser Wert beeinflusst die Schriftgröße im Makro-Editor.~~

Schriftart:

~~Dieser Wert beeinflusst die Schriftart im Makro-Editor.~~



Mausbewegungen aufzeichnen

Wenn nur Tasten-Klicks aufgezeichnet werden sollen, kann die Maus-Aufzeichnung hier ausgeschaltet werden.

Fenster-relative Mauseaufzeichnung:

Dieser Schalter erlaubt es zwischen absoluter und relativer Maus-Aufzeichnung umzuschalten. Für eine umfassende Information über absolute und relative Maus-Aufzeichnung siehe Kapitel: 'Maus Eingaben'.

Relativ-Fenster Filter benutzen:

Ein Nachteil der Benutzung der relativen Maus-Aufzeichnung ist die Notwendigkeit eines Fenstername. Dieser Fenstername wechselt oft. Oft enthält der Fenstername Dateinamen die bei jeder neuen Datei anders sind. Der relative Fenster Filter erlaubt es, nur die reinen Fenster-Namen (ohne den Datei-Namen) zu verwenden. Beispiel: 'Microsoft Word' anstelle von 'Dokument1 - Microsoft Word'.

(Relativ-Fenster Filter) Bearbeiten:

Um den relativen Fenster Filter benutzen zu können, muss eine Filter-Datei vorhanden sein. Der 'Maus und Tastatur Recorder' enthält eine solche Datei mit den gebräuchlichsten Fenstername. (MS Office, IE, Corel). Bei der ersten Installation des Recorders wird diese Datei installiert. Spätere Nachinstallationen verändern diese Datei nicht mehr.

Hinweis: Die neu installierte Datei enthält die Namen der englischen Programme. Ev. müssen diese an Deutsch angepasst werden.

Es ist nun möglich den Filter mit weiteren reinen Fenster-Namen zu erweitern. Beispiel: Bei ALT TAB erscheint: MindManager – [FerienMindMap.mmp]. In die Filter-Datei wird nur MindManager eingetragen. Unabhängig vom Dateinamen wird nun der MindManager bei relativen Maus Klicks und bei der Verwendung von (Schnellzugriffs-Taste) **Switch** immer gefunden.

Wenn ein kurzer Fenstername Teil eines längeren Fenstername ist, muss der lange Name vor dem kurzen in den Filter eingetragen werden. Beispiel: PdmExplorer muss vor dem Namen Explorer eingetragen werden.

Fokus Toleranz:

Dieser Wert in Pixel wird verwendet um die Toleranzen zum Finden des Fokus in den Befehl einzufügen. Im Weiteren definiert dieser Wert auch die Unterscheidung zwischen einem Maus Klick und dem Maus ziehen.

Maus Aktivitäten ohne Mausclicks aufzeichnen:

Wenn dieser Schalter eingeschaltet wird, wird die Maus nur noch positioniert (ohne Klick) oder beim Maus-ziehen gleitet die Maus langsam über den Bildschirm (ohne zu ziehen). Dieser Schalter wird bei einem Recorder Neustart automatisch wieder ausgeschaltet. (Siehe auch Kapitel 'Maus Eingaben')

'Delays' hinzufügen während Aufzeichnen:

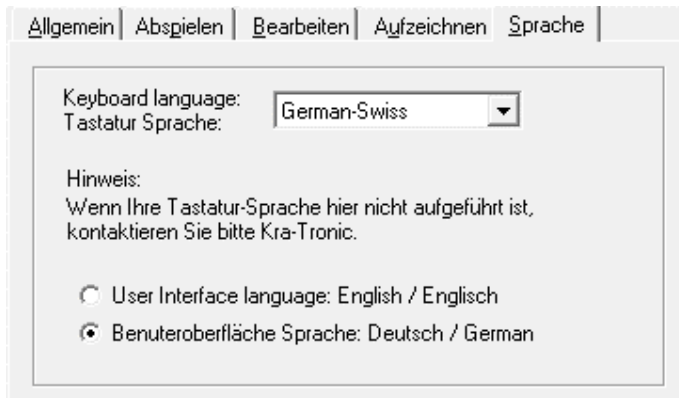
Diese Option wurde auf vielfachen Wunsch der Anwender hinzugefügt. Wenn dieser Schalter eingeschaltet ist, wird während des Aufzeichnens für jeden Befehl ein **Delay** (Wartezeit) eingefügt. Der Delay Befehl enthält genau die Zeit welche man brauchte den nächsten Befehl auszuführen.

Blinkendes Ikon während Aufzeichnung anzeigen:

Wenn man diese Option einschaltet erscheint während der Aufnahme ein kleines blinkendes Fenster, das immer auf der Oberfläche bleibt:

Schnellzugriffs-Tasten für Aufzeichnen (ein/aus):

Diese Option aktiviert Schnellzugriffs-Tasten für das Ein- und Ausschalten des Aufzeichnungs-Modes. Dies erlaubt es sehr einfach schnell das Aufzeichnen zu unterbrechen und nach einer kleinen Aenderung wieder aufzunehmen. *Hinweis:* Wenn das blinkende Ikon während der Aufnahme auch verwendet wird, springt der Fokus nach dem Einschalten per Schnellzugriffs-Taste auf das Ikon. Man muss dann erst mittels ALT TAB auf die Anwendung zurückspringen.

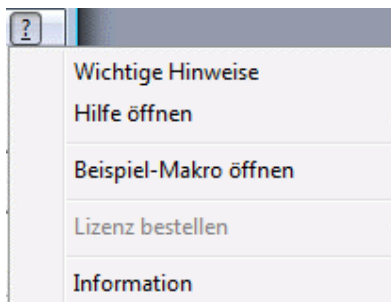
Tastatur Sprache:

Dieses Feld erlaubt es die Tastatur Sprache einzustellen. Der 'Maus und Tastatur Recorder' enthält Tastatur-Sprach-Dateien für 30 verschiedene Sprachen. Um Tasten-Klicks richtig aufnehmen zu können, muss die richtige Sprache eingestellt sein. (German-**Ing** = Deutschland, German-Swiss-**Ing** = Schweiz, German-Austria-**Ing** = Österreich)

Sprache der Benuteroberfläche:

Der Maus und Tastatur Recorder beinhaltet zwei Benuteroberflächen-Sprachen. (Deutsch und Englisch). Bei einem Wechsel der Sprache wird sofort auf die andere Sprache umgeschaltet.

1.11.12 *Hilfe-Menü*



Das Hilfe Menü enthält wichtige Informationen über den 'Maus und Tastatur Recorder'.

Wichtige Hinweise Informiert über grundlegende Vorsichtsmaßnahmen um nicht Probleme durch "wildgewordene" Makros zu erhalten.

Die Anleitung enthält dieses Manual / Hilfe.

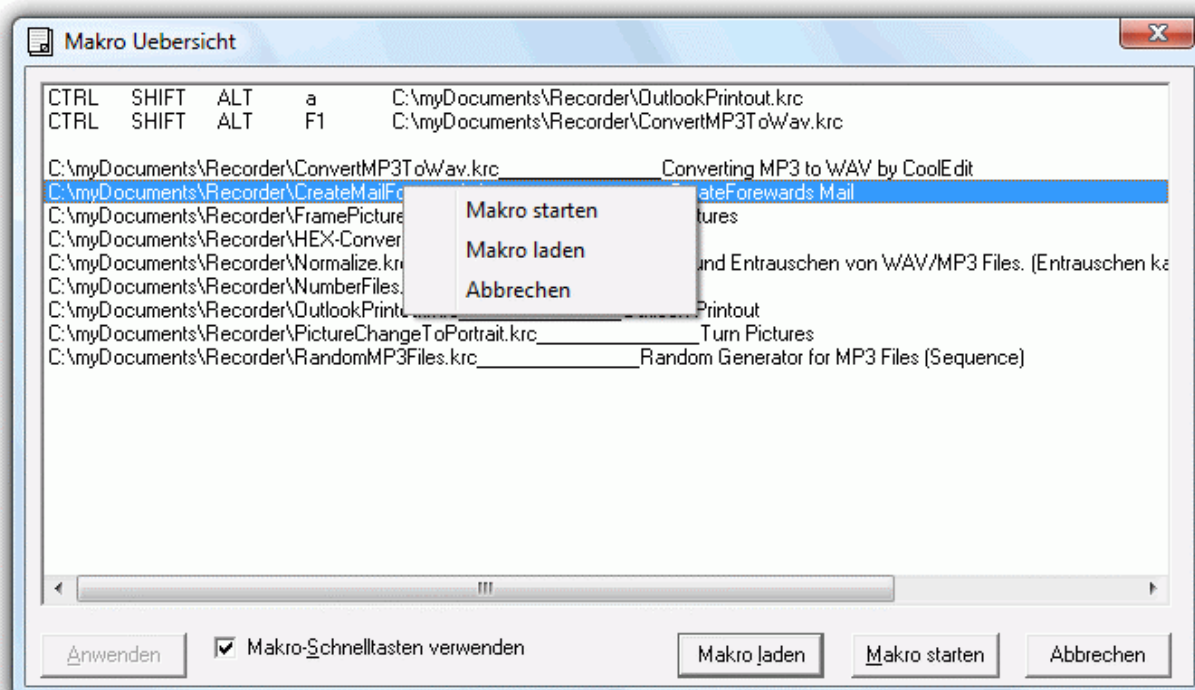
Das Beispiel-Makro ist ein ganz einfaches Makro das ein paar Funktionen des Recorders zeigt.

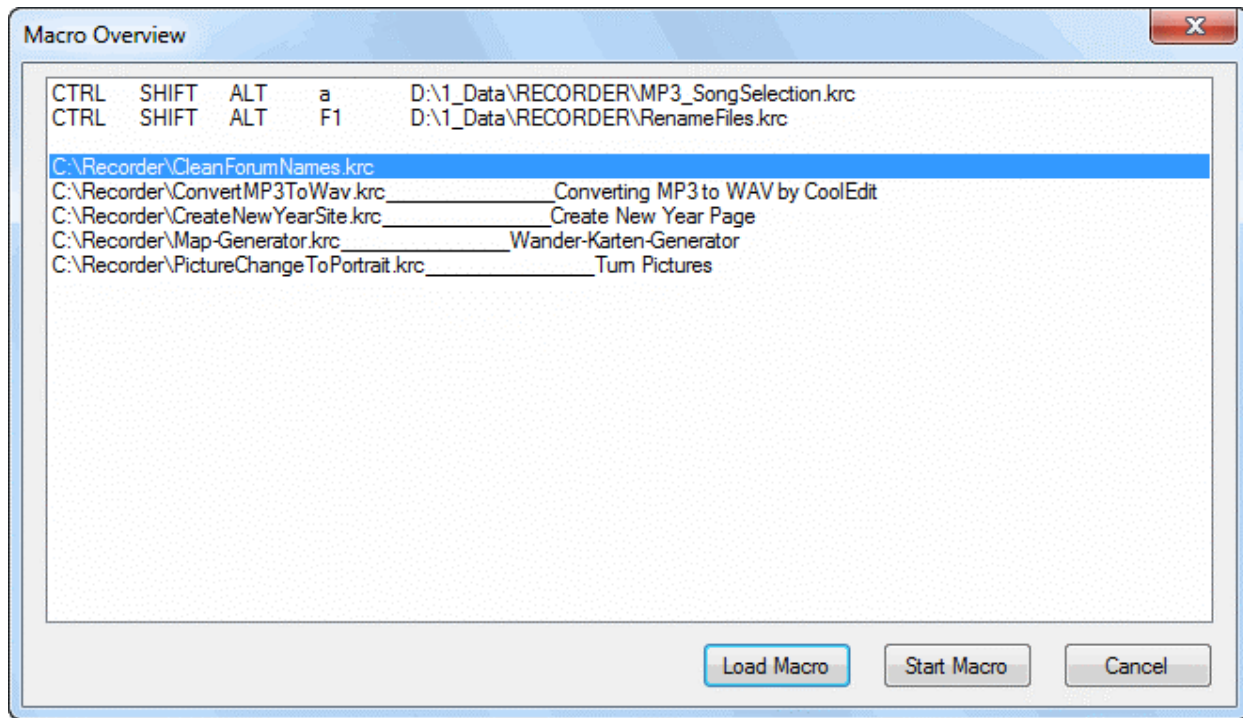
Bei einer Recorder Test-Version kann mit 'Lizenz bestellen' eine lizenzierte Version des Recorders bezogen werden.

Die Informationen enthalten unter anderem auch einen direkten Link zur 'Maus und Tastatur Recorder' Homepage.

1.11.13 *Macro List*

Die Spezial-Schnellaste 'Shift F1' (Umschalt F1) zeigt die folgende 'Makro Uebersicht' Liste:





Der erste Teil zeigt eine Liste aller Schnellzugriffs-Tasten Makros, der zweite Teil zeigt alle Makros aus dem Verzeichnis, das beim letzten Öffnen eines Makros verwendet wurde. Falls Makros eine Makro-Beschreibung haben, wird diese anschließend an den Makro-Namen angezeigt.

Ein ausgewähltes Makro kann direkt aus der Liste geladen oder **auch sogar** gestartet werden. **Ein Doppelklick auf ein Makro lädt dieses. Ein Rechts-Klick öffnet ein Menu, welches das Laden oder Starten des gewählten Makros erlaubt.**

Die Schaltfläche 'Anwenden' erlaubt es die Benutzung von Schnell-Tasten zu aktivieren oder deaktivieren. (Die Shift F1 Schnelltaste ist immer aktiv, solange sich der Recorder zuoberst auf dem Schirm befindet.)

1.12 Entwanzen (Debugging)

Das Aufnehmen, Verändern und Neuschreiben von Makros ist manchmal gar nicht so einfach wie es aussieht. Bestimmte Fehler sind eklig und schwierig zu finden. Die folgenden Hinweise enthalten einige Ideen wie man ein fehlerhaftes Makro zum Laufen bringt.

Befehlsaufbau:

Solange man Makros nur durch Aufzeichnen erstellt, ist der Befehlsaufbau nicht relevant (Der Recorder macht automatisch den richtigen Aufbau). Wenn man aber Makros ändert oder erweitert, dann wird der Befehlsaufbau wichtig. Das bedeutet: So oft wie möglich und soviel wie möglich aufzeichnen. (Recording). Sind weitere Befehle nötig, diese mit dem Menu: Einfügen | Befehl... eingeben. Das Einfügen-Fenster enthält Angaben über den Befehlsaufbau. Zudem wird ein kleines Help-Fenster angezeigt. Für kompliziertere Befehle wie z.B. `{{If` kann das Recorder Help (F1) konsultiert werden.

Makro stoppen:

Manchmal ist es nötig, ein Makro möglichst rasch zu stoppen. Der Recorder enthält dafür verschiedene Möglichkeiten:

- Stopp-Zeichen (das Quadrat oben rechts)
- Menu: Makro | Stoppen
- Schnelltaste (Hotkey): ALT Break/Pause (Notebook Fn ALT Break/Pause) (wirkt nicht wenn IE den Focus hat)

- Befehl `{{Stop}}` im Makro

Makro pausieren:

Mit der Schnelltaste (Hotkey): UMSCHALT Break/Pause (Notebook Fn UMSCHALT Break/Pause) kann ein Makro unterbrochen werden. Wenn man die gleiche Schnelltaste wieder drückt, läuft das Makro weiter. Unterbrechen bedeutet, das Makros steht still, alle Fokusse bleiben wie sie sind. Nun können ev. nötige Aenderungen vorgenommen und danach das Makro weitergeführt werden. Wichtig: Um das Makro erfolgreich weiterlaufen zu lassen, muss die Fokus-Situation die gleiche sein wie beim pausieren des Makros. Sonst kriegt man ein Durcheinander....

Hinweis: Bei einem laufenden Delay wird die Pause erst nach dem Ende des Dalays aktiviert.

Langsame Makro-Ausführung:

Wenn ein Makro fertig gestellt ist, sollte man es erst mal im 'Langsam-Modus' laufen lassen. Dieser Modus wird aktiviert indem man den Wert im Feld: Zeilenverzög. höherstellt. (Register: Allgemein). Wenn man diesen Wert auf z.B. 1000 stellt, wird jede Zeile um 1 Sekunde verzögert. Dadurch ist es möglich, den Makroablauf genau zu beobachten und wenn nötig schnell zu stoppen, falls etwas falsch läuft. Bei einem langen Makro will man vielleicht nur einen Teil beobachten. Dazu eignet sich der Befehl: `{{SetGenDelay}}`.

Wenn man beim Makro-Ablauf den Recorder sichtbar in einer Ecke platziert (auf Register 'Makrocode'), dann sieht man genau welcher Befehl im Augenblick abgearbeitet wird (die ersten zwei Buchstaben jeder Zeile werden blau)

Makro Ablauf-Geschwindigkeit:

Computer-Programme werden für menschliche Benutzer geschrieben. Programm Entwickler erwarten keinen Benutzer wie z.B. einen Maus und Tastatur Recorder, der die Programme unglaublich schnell bedienen kann. Darum testen die Entwickler ihre Programme auch nie mit so einem schnellen Benutzer. Wenn man ein Makro zu schnell laufen lässt, können die unterschiedlichsten Fehler im Programm auftreten. Das ist speziell so bei der Verwendung von Maus Klicks. (Tasten-Drücke werden vom Recorder verzögert, bis der vorherige Tasten-Druck von der Anwendung abgearbeitet wurde - ausser bei Verwendung von `{{SendTextSpecial}}`)

Empfehlung: Ueberall wo nötig `{{Delay}}` einfügen, oder das ganze Makro (oder ein Teil davon) mit 'Zeilenverzög.' oder mit `{{SetGenDelay}}` verlangsamen.

Logbuch:

Ein Logbuch zeichnet alle Aktivitäten eines Makros auf. Diese Funktion kann unter folgendem Menu aktiviert werden: Menü 'Extras' / Optionen' / Register: Abspielen. (Schalter: 'Wiedergabe Logbuch'). Während dem Makro-Ablauf wird jede Zeile dokumentiert. Dabei werden pro Zeile zwei bis fünf Log-Zeilen erzeugt:

1. Zeile: Datum und Zeit
2. Zeile: Makro-Zeile
3. bis 5. Zeile: Aufgelöste Makrozeile, d.h. die Inhalte der Nummer und Text-Speicher werden angezeigt.

Mit dem Logbuch kann man nachvollziehen welchen Weg das Makro genommen hat (bei `{{If}}`), welchen Wert welche Variable zu welcher Zeit hatte, wie Loop Filter (`{{<5}}`) wirkten, etc.

Debug-Fenster:

Die Fehlersuche kann auch mit dem Debug Fenster (Entwanzungs-Fenster) unterstützt werden. Dieses wird mit dem Befehl `{{DebugOn}}` eingeschaltet. Bei jedem Lesen und Schreiben eines Speicherwertes wird der Name des Speichers, der Wert und der dazugehörige Befehl angezeigt. Die neuesten Einträge erscheinen immer zuoberst, die restlichen Einträge werden nach unten weggeschoben.

1.13 Spezielle Funktionen: Automatisierter Software Test

Der 'Maus und Tastatur Recorder' enthält mehrere Befehle, welche den Recorder zum idealen Werkzeug für Software Tests machen. Wenn man Software oder auch Makros für Word / Excel schreibt, ist ein umfassender Test notwendig, welcher alle Funktionen der Software abdeckt. Dazu müssen oft die einzelnen Funktionen mit verschiedensten Eingaben und Daten geprüft werden. Solche Testes können recht gross und aufwendig werden. Schlimmer noch, nach jeder kleinen Änderung an der Software sollten alle Tests wieder durchgeführt werden. (Damit man sicher ist, dass die Änderung

auch wirklich keine anderen Funktionen 'zerstört' hat.) Für diese sich wiederholenden Tests ist der 'Maus und Tastatur Recorder' ein sehr gut geeignetes Werkzeug.

Folgende Befehle eignen sich besonders gut für automatisierten Software Tests:

```
{{CheckFocus, {{WaitForFocus, {{WaitForEndFocus, {{WaitForScreenChange, {{FindPattern,
{{GetPixel, {{Goto, {{Label, {{FindFile, {{Master, {{LoadMacro, {{WriteToFile, {{RandomNumber,
{{RandomString, {{If..Then..Else, {{MessageBox2}}
```

Stellen wir uns eine typische Software Test Situation vor. Der Recorder startet das zu testende Programm. (Annahme: Das Programm ist ein Text Editor). Wir wollen sicher sein, dass das Programm auch wirklich startet. Darum prüfen wir den aktuellen Fokus. Wenn der Fokus falsch ist, hat das Programm wohl nicht richtig aufgestartet. Abhängig von der Programmierung des Fokus Befehls bricht nun das Makro ab oder es verzweigt zu einer Marke. Wenn das Makro zu einer Marke verzweigt könnte dort weiterer Code durchlaufen werden, welcher eine SMS absetzt. Angenommen der Softwaretest dauert Stunden, dann können wir den PC alleine arbeiten lassen und werden, wenn der Recorder einen Fehler findet, per SMS alarmiert.

Nehmen wir weiter an, im Verlaufe des Tests haben wir einigen Text in den Editor geschrieben und möchten diesen nun abspeichern. Abhängig von der Situation auf der Festplatte (Datei existiert oder existiert nicht) werden unterschiedliche Fenster erscheinen. Mit `{{CheckFocus` kann eine solche Situation getestet und somit richtig reagiert werden. Das folgende Makro zeigt eine solche Lösung:

```
{{Start Notepad
Test
{ENTER}
%d {' Datei
u {' Speichern unter...
C:\RecTest\John.txt
%s {' Speichern
{{Delay 1000}
{{If {{CheckFocus 82, 128, 95, 23, 3, 3, 3, 3} = {{True} ;%};
%d {' Datei
e {' Beenden
```

Wenn das Makro das erste **Mal** läuft, existiert die Datei John.txt noch nicht auf der Festplatte. Das bedeutet `{{CheckFocus` liefert ein Falsch und es wird nichts weiter getan. (Kein Befehl nach dem zweiten Strichpunkt). Wenn John.txt bereits existiert, liefert `{{CheckFocus` ein Wahr und der Befehl `%j` (JA) wird ausgeführt.

1.13.1 Funktions Beschreibung

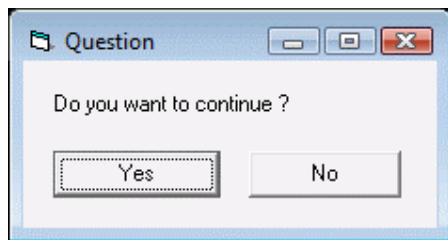
1.13.1.1 {{CheckFocus

Dieser Befehl prüft ob der Fokus während des Makroablaufes der gleiche ist, wie damals als er aufgezeichnet wurde. (Der Fokus hat derjenige Teil eines Fensters, auf den die Tastatur aktuell wirkt)

Ein `{{CheckFocus` wird eingefügt durch das anklicken des Menü: 'Einfügen' / 'Fokus auf Objekt' / 'Fokus Prüfen'. Nach dem Anklicken von 'Fokus Prüfen' muss innerhalb drei Sekunden zum dem zu prüfenden Fenster umgeschaltet werden. Diese Verzögerungszeit (3 sec) kann in 'Extras' / 'Optionen' / 'Bearbeiten' / 'PrüfeFocus Verzöger.' verändert werden.

Nach der Verzögerungszeit wird ein Befehl ähnlich dem folgenden eingefügt:

```
{{If {{CheckFocus 82, 288, 95, 23, 3, 3, 3, 3} = {{True}};{{ShowError}}. Die ersten vier Nummern
bedeuten: Distanz zum oberen Rand, zum linken Rand, Breite des Fokus, Höhe des Fokus. Beispiel:
```

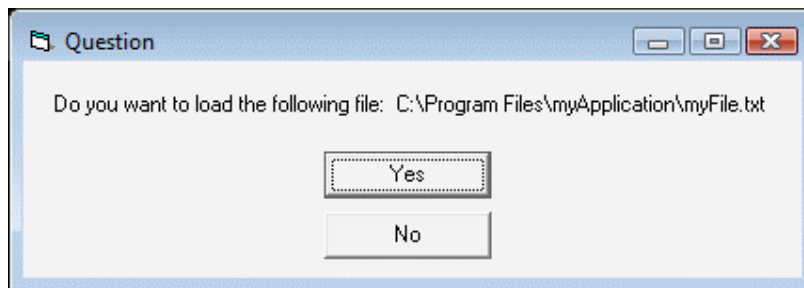


{{CheckFocus 63, 28, 81, 25. 'Yes' hat den Fokus. 63 Pixel befinden sich zwischen dem oberen Rand und dem Knopf, 28 Pixel zwischen dem linken Rand und dem Knopf. Der Knopf hat eine Grösse von 81 * 25 Pixel.

Das 'Question' - Fenster kann irgendwo auf dem Bildschirm platziert sein, es ergeben sich trotzdem immer die gleichen Zahlen. Das heisst auch, der Fokus von 'Yes' wird stets gefunden unabhängig vom Aufenthaltsort des 'Question' – Fensters.

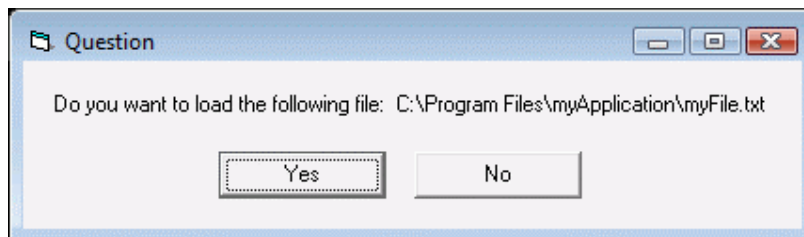
Einschränkung: Einzelne Teile auf dem Arbeitsplatz (z.B. Icons) können nicht mit {{CheckFocus geprüft werden. Sie haben immer die gleiche Nummer, unabhängig vom Ort wo sie sich befinden.

Manchmal ändert der Text im Fenster. Beispiel:



Abhängig von der Länge des Textes verschieben sich die Knöpfe horizontal. Es ist trotzdem möglich das 'Yes' zu prüfen: {{CheckFocus 71, -1, 49, 25. Das -1 bedeutet: Die Distanz zum linken Rand soll nicht geprüft werden. Eine -1 kann für alle vier Nummern verwendet werden.

Aber was machen wir wenn die Knöpfe nebeneinander angeordnet sind wie im nächsten Beispiel ?



Die Lösung heisst: Toleranzen. Für jede der vier Nummern kann eine eigene Toleranz definiert werden. Beispiel: {{CheckFocus 63, 105, 95, 23, 0, 20, 0, 0. Das bedeutet, dass der 'Yes'-Knopf sich um +- 20 Pixel verschieben kann, ohne dass ein Fehler ausgelöst wird. (100 Pixel sind ca. 3 cm auf einem 17" Bildschirm mit Auflösung 1024*768).

Manchmal haben die Fenster auf dem Bildschirm kleine Ungenauigkeiten. Das bedeutet, dass die Fenster manchmal um ein Pixel falsch gezeichnet werden (Rundungs-Fehler). Damit solche Windows-Fehler nicht zu Fehlern im Recorder führen, verwendet der Recorder standardmässig eine Toleranz von je drei Pixel. Die Pixel-Toleranz kann in 'Extras' / 'Optionen' / 'Aufzeichnen' / 'Fokus Toleranz' verändert werden.

Wenn {{CheckFocus den erwarteten Fokus nicht findet, wird eine '0' zurückgegeben, sonst eine '1'. Innerhalb eines {{If Befehles kann dies mit den Schlüssel-Wörtern {{True} und {{False} geprüft werden.

Im Wahr- und Falsch Teil des `{}If` Befehles kann das Schlüsselwort `{}ShowError` verwendet werden. Beispiel: `{}If {}CheckFocus 455, 20, 73, 17, 3, 3, 3, 3 = {}True;;{}ShowError`. Wenn der Fokus nicht gefunden wird, gibt der Recorder eine Fehlermeldung aus und zeigt mit einem `==>` die schuldige Zeile an. Ausserdem werden zusätzlich die Koordinaten des gefundenen Fokus in der LogBuch-Datei abgespeichert. (Wenn das Logbuch aktiviert ist.)

1.13.1.2 `{}WaitForFocus`, `{}WaitForEndFocus`

Ein `{}WaitForFocus` Befehl wird mittels des Menü: 'Einfügen' / 'Fokus auf Objekt' / 'Auf Fokus warten' eingefügt. Für `{}WaitForEndFocus` muss der Menü-Text: 'Auf Fokus-Ende warten' verwendet werden.

Der Befehl `{}WaitForFocus` arbeitet sehr ähnlich wie `{}CheckFocus`. Zusätzlich zu den vier Koordinaten wird noch eine Maximale Wartezeit eingefügt. Beispiel: `{}If {}WaitForFocus 63, 14, 95, 23, 10, 3, 3, 3, 3 = {}True;;{}ShowError`. Der Vorgabewert für die maximale Wartezeit kann 'Extras' / 'Optionen' / 'Bearbeiten' 'WarteAufFokus Wartezeit.' eingestellt werden. `{}WaitForFocus` wartet auf den Fokus während der Max. Wartezeit. Im obigen Beispiel sind dies 10 Sekunden. Wenn der Fokus innerhalb dieser 10 sec erscheint liefert `{}WaitForFocus` eine '1' (True). Wenn der Fokus nicht innerhalb 10 sec erscheint, gibt `{}WaitForFocus` eine '0' (False). `{}WaitForFocus` und `{}WaitForEndFocus` erlauben die Verwendung der oben erwähnten -1 Parameter und Pixel-Toleranzen.

`{}WaitForEndFocus` wartet darauf dass ein existierender Fokus verschwindet. Mit diesem Befehl ist es also möglich auf das Ende einer Operation zu warten.

Einschränkungen:

- Einige Programme enthalten nur ein Fokus Fenster, obwohl sie viele einzelne Icons anzeigen. (Spezielle Programmier-Technik). Bei diesen Programmen können die Fokus Befehle nicht angewendet werden..
- Die Fokus Befehle können auch nicht für Internet Browser (Internet Explorer) verwendet werden. Die Browser liefern immer die gleichen Koordinaten, unabhängig vom Inhalt. Lösungsmöglichkeiten `{}WaitForIEReady`, `{}WaitForScreenChange`, `{}FindPattern`. (Siehe auch Kapitel Makro Beispiele.)

WaitForFocus Beispiel.

Bitte folgende Schritte ausführen:

Den 'Maus und Tastatur Recorder' starten. Notepad von Hand starten (Start | Ausführen | Notepad.exe). Im Editor (Notepad) auf Datei | Öffnen klicken. Im neuen Fenster auf den Knopf Öffnen klicken. (Dadurch erhält dieser Knopf den Fokus. Ein gepunkteter Rahmen zeigt dies an.) Auf dem Recorder Menü: 'Einfügen' / 'Fokus auf Objekt' / 'Auf Fokus warten' klicken. Innert 3 Sekunden mit ALT Tabulator auf den Notepad umschalten. Der Recorder fragt nun nach einer Abbruch-Zeit. 100 eingeben und Ok drücken. Das folgende Makro wurde nun erstellt: `{}If {}WaitForFocus 322, 478, 75, 23, 100, 3, 3, 3, 3 = {}True;;{}ShowError`. Den Befehl `{}Beep` zum Makro hinzufügen. (Menü: 'Einfügen' / 'Befehl...'. `{}Beep`). Das Öffnen-Fenster des Editors (Notepad) schliessen. Nun das Makro abspielen. Nichts passiert, nur die Abbruch-Zeit zählt nach unten. Im Editor (Notepad) Datei | Öffnen klicken und dann auf den Knopf Öffnen drücken. In diesem Moment ist ein Ton zu hören (Beep) und das Makro macht weiter...

Das heisst, um diesen Befehl zu benutzen muss zuerst die richtige Fokus-Situation erstellt und aufgezeichnet werden. Während dem Makro Ablauf wartet der Recorder bis die richtige Fokus Situation erscheint oder bis die Abbruch-Zeit abgelaufen ist.

1.13.1.3 `{}MessageBox2`

Der Befehl `{}MessageBox` zeigt ein Fenster in der Mitte des Bildschirms. Während einem Software Test ist der Bildschirm oft gefüllt mit den Fenstern der zu prüfenden Software. Das MessageBox Fenster erscheint dann oft unter den anderen Fenstern. Die `{}MessageBox2` erscheint immer über dem

'Maus und Tastatur Recorder' (solange dieser sichtbar ist). Ansonsten ist die `{{MessageBox2}}` genau gleich zu verwenden wie die normale `{{MessageBox}}`.

Wichtig: Der `{{MessageBox2}}` Befehl darf nicht verwendet werden, wenn zuvor der Befehl `{{Minimize}}` ausgeführt wurde. Sonst kann das MessageBox-Fenster auf bestimmten Betriebssystemen nicht mehr geschlossen werden.

1.14 Tastatur Sprache

Mit dem 'Maus und Tastatur Recorder' werden 30 Sprach-Dateien mitgeliefert. Wenn eine Tastatur einer Sprache die nicht vorhanden ist, verwendet werden soll, muss eine Tastatur Sprach-Datei erstellt werden. Bitte kontaktieren Sie in diesem Falle Kra-Tronic Corp.

1.15 Makro Beispiele

Die Makro Beispiele teilen sich auf in Beispiele für einzelne Befehle (oder Befehlsgruppen) und Lösungen allgemeiner Aufgaben.

Makros mit dem Hinweis 'Lauffähiges Beispiel' können in den Recorder kopiert und auf allen Betriebssystemen und Sprachen laufen gelassen werden.

1.15.1 Beispiele für einzelne Befehle (oder Befehlsgruppen)

1.15.1.1 Kommentare

(Zusätzliche Befehle: `{{Start}}`, `{{Delay}}`)

(Lauffähiges Beispiel)

Dieses Beispiel zeigt den Unterschied von `{{'}}` und `{{"}}`

```

{{Start Notepad
{{Delay 1000}} {{' Milliseconds
Zeile1 {{' Mein Kommentar
Weiterer Text auf gleicher Zeile.
{ENTER}
Zeile2 {{" Mein Kommentar
Weiterer Text auf gleicher Zeile.
{ENTER 2}
Wie man sieht, lässt {{'}} den Leerschlag vor 'Weiterer Text...' stehen.

```

~~***{{SendTextNormal}}, {{SendTextSpecial}}***~~

~~(Zusätzliche Befehle: `{{Start}}`, `{{Delay}}`)~~

~~(Lauffähiges Beispiel)~~

~~Dieses Beispiel zeigt die Wirkung von `{{SendTextSpecial}}` bei einer DOS-Box.~~

```


{{Start CMD {{' DOS-Box starten
{{SendTextNormal}}
{{Delay 1000}} {{' Milliseconds
Text gesendet mit {{()SendTextNormal()}}
{{Delay 500}} {{' Milliseconds


```

```

{{SendTextSpecial}}
{{Delay 500}} {{' Milliseconds}}
Text gesendet mit {{(}SendTextSpezial{)}}
-----{{" Viele Leerschlage}}
Wie man sieht, wird nur bei 'SendTextSpecial' der Text richtig die DOS-Box geschrieben.
{{(}Je nach Betriebssystem funktioniert auch 'SendTextNormal' korrekt.{)}}

```

1.15.1.2 *Start, StartAndWait*

(Zusätzliche Befehle: `ExtDelay`)

(Lauffahiges Beispiel)

Hier wird der Notepad normal gestartet und nachher der Calculator mit `StartAndWait`. Erst wenn der Calculator geschlossen wird, lauft das Makro weiter.

```

{{Start Notepad}}
{{Delay 1000}} {{' Milliseconds}}
Dieser Notepad wurde normal gestartet: {{{}}Start Notepad}}
{ENTER}}
Es wird nun in 10 sec. ein Rechner gestartet. {{{}}StartAndWait Calc}}
{ENTER}}
Erst wenn der Rechner geschlossen wird, lauft das Makro weiter.
{ENTER}}
Also bitte den Rechner dann dann schliessen....
{ENTER 2}}
{{ExtDelay 10 {{' 10 sec. warten}}
{{StartAndWait calc}}
So, jetzt lauft es hier wieder weiter.....

```

1.15.1.3 *DOS*

(Zusätzliche Befehle: `ImportFile`, `ImportSeparator`, `DoImport`, `FlashBox`, `Goto`, `Label`, `DeleteFile`, `If`)

(Lauffahiges Beispiel)

Hier wird mittels DOS Box ([Eingabeaufforderung](#)) ein Dir gemacht und das Resultat in eine Textdatei gespeichert. Diese Textdatei wird dann mit Notepad geoffnet. Zusätzlich werden alle gelisteten Dateien der Reihe nach in einer FlashBox angezeigt. Am Schluss wird die Textdatei wieder geloscht (aufgeraumt).

```

{{DOS MKDir C:\DOS-Test}}
{{DOS Dir /B > C:\DOS-Test\DirList.txt}}
{{Delay 1000}} {{' Milliseconds}}
{{Start Notepad C:\DOS-Test\DirList.txt}}
{{Delay 1000}} {{' Milliseconds}}
{{Start Notepad C:\DirList.txt}}
{{Delay 1000}} {{' Milliseconds}}
{{ImportFile C:\DOS-Test\DirList.txt}}
{{ImportSeparator >}}

```

```

{{Label Start}}
{{FlashBox {{DoImport}} | 1 | Dir-List Anzeige | Weitermachen | Abbrechen}}
{{Delay 500}} {{' Milliseconds}}

```

```

{{If {{VarFlashBoxAnswerX55}} = {{True}} ;{{Goto Ende
{{If {{ImportNextLine}} = {{True}};{{Goto Start}};{{Goto Ende

```

```

{{Label Ende
{{Delay 1000}} {{' Milliseconds
{{ImportFile {{MacroPath}}{{MKRFileName}}

{{DeleteDir C:\DOS-Test | {{True}}
{{DeleteFile C:\DirList.txt

```

Hinweise:

- Dir /B produziert eine Liste mit nur Dateinamen (ohne Titel und Zusammenfassung)
- DOS Befehle funktionieren nur mit der 8.3 Dateiname Regel. Daher werden lange Dateinamen abgekürzt. z.B. KraTronicCorp.doc nach KRATRO~1.DOC
- DOS kann nicht benutzt werden für Verzeichnis Namen **länger als 8 Zeichen !**
- Die Datei DirList.txt kann nicht direkt auf C:\ abgelegt werden. Sonst kann sie nicht per Makro geöffnet werden. (Einschränkung Windows 7 und neuer)
- {{ImportSeparator > ist hier nötig, weil ein Dateiname ein Komma (Standart Separator) enthalten könnte
- {{ImportFile {{MacroPath}}{{MKRFileName}} ist nötig, damit die Datei DirList.txt gelöscht werden kann. (Ist durch {{ImportFile blockiert). Die Makro-Datei {{MKRFileName}} wird beim Beenden des Makros auch wieder freigegeben.

1.15.1.4 {{Switch, {{SwitchOrStart

Zusätzliche Befehle: {{DOS, {{Start

(NICHT überall lauffähiges Beispiel)

Dieses Beispiel zeigt verschiedene Umschalt-Möglichkeiten.

```


{{DOS MKDir C:\Swi-Test
{{Delay 300
{{DOS dir /B > C:\Swi-Test\DirList.txt
{{Delay 1000}} {{' Milliseconds
{{Start Notepad C:\Swi-Test\DirList.txt
{{Delay 1000}} {{' Milliseconds
{{Switch #MKRX5#
{{Switch Dir
{{Delay 300
{{SwitchOrStart DirList.txt | Notepad C:\Swi-Test\DirList.txt

{{Delay 300}} {{' Milliseconds
{{DeleteDir C:\Swi-Test | {{True}}
{{Delay 300
{{Switch <=DirList.txt
{{Delay 300
{{Switch <=<DirList.txt=>

{{Delay 300
{{Delay 1000}} {{' Milliseconds
{{DeleteFile C:\DirList.txt

```

Hinweise:

- {{Switch #MKRX5# schaltet auf den Recorder um ({{Switch Maus und Tastatur Recorder geht nicht)
- {{Switch Dir schaltet zum ersten gefundenen Fenster um das irgendwo im Titel Dir enthält.
- {{Switch <=<DirList.txt schaltet zum ersten gefundenen Fenster um das am Anfang ein DirList.txt enthält.

- ~~SwitchOrStart DirList.txt | Notepad C:\Swi-Test\DirList.txt macht ein Switch weil das Fenster DirList.txt gefunden wird.~~
 - ~~Switch <=> DirList.txt=>~~ gibt höchstwahrscheinlich eine Fehlermeldung, da im Titel allermeist nicht nur DirList.txt steht.
- ~~SwitchOrStart DirList.txt | Notepad C:\DirList.txt macht ein Switch weil das Fenster DirList.txt gefunden wird.~~

1.15.1.5 Loop

(Zusätzliche Befehle: {{1}}, {{}}, {{LeftLoop}}, {{RightLoop}}, {{Offset}}, {{SetLoop}}

(Lauffähiges Beispiel. Bedingung: Feld: Schleifen auf Register 'Allgemein' auf 10 stellen)

Dieses Beispiel zeigt diverse Befehle die mit Schleifen möglich sind.

```

{{1}}Start Notepad
{{1}}Delay 1000 {' Milliseconds
{{1}}Dies ist die erste Schleife
{{1}}ENTER
{{2}}Dies ist die zweite Schleife
{{2}}ENTER
Wir sind nun in Schleife: {{Loop}}
ENTER
{{>100}}Dies ist eine Schleife > 100
{{>100}}ENTER
{{3}}Dies ist die dritte Schleife
{{3}}ENTER
{{3}}Wir fahren nun mit dem Zeiger {{{LeftLoop 5 nach links und dann wieder {{{RightLoop 5 nach rechts.
{{3}}Delay 1000 {' Milliseconds
{{3}}LeftLoop 5
{{3}}Delay 1000 {' Milliseconds
{{3}}RightLoop 5
{{3}}Delay 1000 {' Milliseconds
{{3}}ENTER
{{4}}Jetzt setzen wir die Schleifen hoch auf 6
{{4}}ENTER
{{4}}SetLoop 6
{{8}}Nun setzen wir {{{Offset auf 100
{{8}}ENTER
{{8}}Offset 100
{{n}}Dies ist die letzte Schleife
{{n}}ENTER

```

1.15.1.6 WaitForKey

(Zusätzliche Befehle: {{MessageBox}}

(Lauffähiges Beispiel)

```

{{MessageBox Bitte nach dem Klicken auf OK eine Taste drücken{{CR}}(Keine Kombitasten wie ALT f)
{{WaitForKey}}
{{MessageBox Es wurde die Taste: '{{StringWaitForKeyX55}}' gedrückt.

```

1.15.1.7 *SetVar, Var, SetString, String*

(Zusätzliche Befehle: SetGlobVar, GlobVar, SetGlobString, GlobString, DivVar, MulVar, Goto, Label)

(Lauffähiges Beispiel)

Dieses Beispiel zeigt den Umgang mit Speichern. Auch die indirekte Speicherung wird gezeigt mittels Verwendung der Speicher für Arrays (Datenfelder)

```

{{Start Notepad
{{Delay 1000}} {' Milliseconds
Speichern der Zahl 133 in Speicher VarZahl
{ENTER}
{{SetVarZahl 133
Speichern der Zahl 266 in globalen Speicher VarZahlGlobal
{ENTER}
{{SetGlobVarZahl 266
Speichern des Text 'Mein Text 1' in Speicher StringText
{ENTER}
{{SetStringText Mein Text 1
Speichern des Text 'Mein Text Global' in Speicher StringTextGlobal
{ENTER 2}
{{SetGlobStringText Mein Text Global
Speicherinhalte anzeigen:
{ENTER}
VarZahl = {{VarZahl}
{ENTER}
GlobVarZahl = {{GlobVarZahl}
{ENTER}
StringText = {{StringText}
{ENTER}
GlobStringText = {{GlobStringText}
{ENTER 2}

```

Mit Zahlen rechnen und Texte kombinieren:

```

{ENTER}
Zu VarZahl 100 hinzuzählen und durch 5 teilen
{ENTER}
{{IncVarZahl 100
{{DivVarZahl 5
Zu GlobVarZahl 200 hinzuzählen und mit 3.5 multiplizieren
{ENTER}
{{IncGlobVarZahl 200
{{MulGlobVarZahl 3.5
Zu VarText den Text 'Zusätzlicher Text' hinzufügen
{ENTER 2}
{{SetStringText {{StringText} Zusatzlicher Text

```

Speicherinhalte anzeigen:

```

{ENTER}
VarZahl = {{VarZahl}
{ENTER}
GlobVarZahl = {{GlobVarZahl}
{ENTER}
StringText = {{StringText}
{ENTER 2}

```

Ein Array (Datenfeld) bilden und anzeigen:

```
{ENTER}
Es sollen 10 Zufallszahlen in ein Array abgelegt werden
{ENTER}
{{SetVarArrayZaehler 1
{{Label Start
{{SetVarArray{{VarArrayZaehler}} {{RandomNumber 1,100}}
{{IncVarArrayZaehler 1
{{If {{VarArrayZaehler} <= 10 ;{{Goto Start;

{{SetVarArrayZaehler 1
{{Label StartAnzeige
VarArray{{VarArrayZaehler}} = {{VarArray{{VarArrayZaehler}}
{ENTER}
{{IncVarArrayZaehler 1
{{If {{VarArrayZaehler} <= 10 ;{{Goto StartAnzeige;
```

Hinweis: Das Beispiel erklärt sich selber, wenn man es laufen lässt :-)

1.15.1.8 *Ask, WriteAsk*

(Zusätzliche Befehle: {{SetClipboard, {{Clipboard, {{RandomString)

(Lauffähiges Beispiel)

Dieses Beispiel zeigt den Umgang mit Benutzerabfragen und Zwischenspeicher.

```
{{AskFrage1 Wie heisst die Abkürzung der Recorders ? |MTR
{{AskFrage2 Wieviele US$ kostete die 1. Version des Recorders? |15
{{Start Notepad
{{Delay 1000} {{' Milliseconds
{{SetClipboard {{WriteAskFrage1}}
Die Abkürzung für den Recorder heisst:
^v {{' Microsoft Befehl für Einfügen (Paste)
{ENTER}
{{SetClipboard {{WriteAskFrage2}}
Die erste Recorder Version kostete: ^v $
{ENTER 2}
{{SetClipboard {{RandomString 32, !, ~}}
Zufallszeichen = ^v
```

Hinweis: Für Sonderzeichen (hier aus RandomString) muss zwingend das Clipboard verwendet werden. Einige Sonderzeichen haben spezielle Funktionen. z.B. % entspricht ALT. [Weitere Möglichkeit: Verwendung von {{SendTextSpecial+}}](#)

1.15.1.9 *SerialTime*

(Zusätzliche Befehle: {{Div, {{Mod, {{IncVar, {{DecVar, {{MessageBox)

(Lauffähiges Beispiel)

Dieses Beispiel zeigt die Umrechnung von SerialTime in Tage, Std. Min. und Sek.

```
{{SetVarStStart {{SerialTime}}
{{MessageBox Ein paar Sekunden warten um den Effekt zu sehen.
```

```

{{SetVarStEnd {{SerialTime}}
{{DecVarStEnd {{VarStStart}} {{' Wartezeit}}
{{IncVarStEnd 93720 {{' 1 Tag, 2 Std. und 2 Min. hinzuzählen}}
{{SetVarStDelta}} {{VarStEnd}}
{{SetVarStMin {{Div {{VarStDelta}} | 60}}
{{SetVarStSec {{Mod {{VarStDelta}} | 60}}
{{DecVarStDelta}} {{VarStSec}}
{{SetVarStHour {{Div {{VarStDelta}} | 3600}}
{{SetVarStMin {{Mod {{VarStDelta}} | 3600}}
{{DecVarStDelta}} {{VarStMin}}
{{SetVarStMin {{Div {{VarStMin}} | 60}}

```

```

{{SetVarStDay {{Div {{VarStDelta}} | 86400}}
{{SetVarStHour {{Mod {{VarStDelta}} | 86400}}
{{SetVarStHour {{Div {{VarStHour}} | 3600}}

```

```

{{MessageBox {{VarStDay}}-{{VarStHour}}:{{VarStMin}}:{{VarStSec}}

```

Anzeige: z.B 1-2.2.14 (1 Tag, 2 Std, 2 Min und die gewarteten 14 Sekunden.

Hinweis: Anstelle von {{IncVarStEnd 93720 könnte auch der neue Befehl {{AddSerialTime verwendet werden.

1.15.1.10 *{{ImportFile}, {{ImportSeparator}, {{DoImport}, {{ImportNextLine}}*

(Zusätzliche Befehle: {{If}, {{Goto}, {{Label})

(Lauffähiges Beispiel)

Dieses Beispiel zeigt die Verwendung von Import Dateien.

Bedingung: Der Teil: ImportText muss in eine Datei mit dem Namen ImportText.txt im

VerzeichnisLaufwerk C:\Temp abgespeichert werden. (Notepad öffnen und die 3 Zeilen hineinkopieren)

C:\Temp\ImportText.txt:

```

Zeile1_Teil1;Zeile1_Teil2;Zeile1_Teil3
Zeile2_Teil1;Zeile2_Teil2;Zeile2_Teil3;Zeile2_Teil4
Zeile3_Teil1;Zeile3_Teil2;Zeile3_Teil3

```

Makro:

```

{{ImportFile C:\Temp\ImportText.txt
{{ImportSeparator ;
{{Label Start
{{MessageBox {{DoImport}}
{{MessageBox {{DoImport}}
{{MessageBox {{DoImport}}
{{MessageBox {{DoImport}}
{{If {{ImportNextLine}} = {{True}};{{Goto Start;

```

Hinweise:

- Das {{1}} vor {{ImportFile}} ist nur nötig, wenn man Schleifen verwendet. Hier verwenden wir aber {{ImportNextLine}} und {{Goto}}
- Das 4. {{DoImport}} liefert bei Zeile 1 und 3 einen leeren Text... weil dort eben nix ist.
- Wir müssen {{ImportSeparator}} ; definieren, da wir nicht den standard Separator (,) verwenden.

1.15.1.11 *ScanDirectories*

(Zusätzliche Befehle: `[[CreateDir`, `[[CopyFile`, `[[RenameFile`, `[[DeleteFile`, `[[WriteLineToFile`, `[[StartAndWait`, `[[ImportFile`, `[[ImportSeparator`, `[[MessageBox`, `[[MessageBox2`)

(Lauffähiges Beispiel)

Dieses Beispiel zeigt die Verwendung von `ScanDirectories`. Dazu werden zwei Verzeichnisse und Dateien erstellt. Die Liste ~~von~~ der Dateien wird dann angezeigt, Dateien werden kopiert und umbenannt und dann wieder angezeigt. Zum Schluss wird gezeigt, welche Teile `[[ScanDirectories` abspeichert.

```
[[CreateDir C:\MTRTestDir77
[[CreateDir C:\MTRTestDir77\TestDir88
[[WriteLineToFile C:\MTRTestDir77\_Dieses Verzeichnis kann gelöscht werden.txt | Dieses
Verzeichnis kann gelöscht werden.
[[WriteLineToFile C:\MTRTestDir77\TestFile.txt | Dies ist eine Test-Datei
[[WriteLineToFile C:\MTRTestDir77\TestFileDelete.txt | Dies ist eine Test-Datei die bald gelöscht
wird.
[[WriteLineToFile C:\MTRTestDir77\TestDir88\TestFile2.Doc | Dies ist eine Test-Datei
[[ScanDirectories C:\MTRTestDir77 | *.* | [[True] | [[MacroPath]DirList.txt | [[False]
[[WriteLineToFile [[MacroPath]DirList.txt | [[CR]
[[WriteLineToFile [[MacroPath]DirList.txt | Dieses Datei enthält nun alle Dateien im Verzeichnis
C:\MTRTestDir77 (und Unterverzeichnisse)
[[WriteLineToFile [[MacroPath]DirList.txt | Bitte dieses Fenster schiessen. Dann fährt das Makro
weiter.
[[StartAndWait Notepad [[MacroPath]DirList.txt

[[CopyFile C:\MTRTestDir77\TestFile.txt | C:\MTRTestDir77\TestFileCopy.txt
[[RenameFile C:\MTRTestDir77\TestFileDelete.txt | C:\MTRTestDir77\TestDir88\TestFileCopy2.txt
[[' Dieser Befehl verschiebt die Datei und nennt sie um.
[[DeleteFile C:\MTRTestDir77\TestFileDelete.txt

[[ScanDirectories C:\MTRTestDir77 | *.* | [[True] | [[MacroPath]DirList.txt | [[False]
[[WriteLineToFile [[MacroPath]DirList.txt | [[CR]
[[WriteLineToFile [[MacroPath]DirList.txt | Dieses Datei enthält nun auch die
kopierten/verschobenen Dateien im Verzeichnis C:\MTRTestDir77 (und Unterverzeichnisse)
[[WriteLineToFile [[MacroPath]DirList.txt | Bitte dieses Fenster schiessen. Dann fährt das Makro
weiter.
[[StartAndWait Notepad [[MacroPath]DirList.txt

[[ScanDirectories C:\MTRTestDir77 | *.Doc | [[True] | [[MacroPath]DirList.txt | [[True]
[[ImportFile [[MacroPath]DirList.txt
[[ImportSeparator >
[[SetStringList Pfad, Filename und Erweiterung: [[DoImport]{{CR}}[[CR]
[[SetStringList {{StringList}Filename und Erweiterung: [[DoImport]{{CR}}[[CR]
[[SetStringList {{StringList}Filename: [[DoImport]{{CR}}[[CR]
[[SetStringList {{StringList}Erweiterung: [[DoImport]{{CR}}[[CR]
[[SetStringList {{StringList}Pfad: [[DoImport]{{CR}}[[CR]

[[MessageBox2 {{StringList}

[[ImportFile [[MacroPath]{{MKRFileName}
[[DeleteFile C:\MTRTestDir77\TestFile.txt
```

```
{{DeleteFile C:\MTRTestDir77\TestFileCopy.txt
{{DeleteFile C:\MTRTestDir77\TestDir88\TestFile2.Doc
{{DeleteFile C:\MTRTestDir77\TestDir88\TestFileCopy2.txt
{{DeleteFile {{MacroPath}}DirList.txt
```

```
{{DeleteDir C:\MTRTestDir77\TestDir88
{{DeleteDir C:\MTRTestDir77
```

{{MessageBox Auf einigen Betriebssystemen wird das Verzeichnis C:\MTRTestDir77 nicht gelöscht. {{CR}}In diesem Fall bitte von Hand löschen (Ev. erst den Recorder schliessen).

Hinweise:

- Einschränkung: Es ist nicht möglich eine Datei die erst kopiert wurde dann auch noch umzubenennen. Auch führt das Kopieren und Umbenennen zu Problemen beim Löschen der Verzeichnisse. (Fehler ist Betriebssystem abhängig)
- `{{ImportFile {{MacroPath}}{{MKRFileName}}` ist nötig, damit die Datei DirList.txt gelöscht werden kann. (Ist durch `{{ImportFile}}` blockiert). Die Makro-Datei `{{MKRFileName}}` wird beim Beenden des Makros auch wieder freigegeben.
- Man könnte auch alle Verzeichnisse und Dateien auf einmal löschen mit: `{{DeleteDir C:\MTRTestDir77 | {{True}}`

1.15.1.12 `{{Gosub}}`

(Zusätzliche Befehle: `{{MessageBox}`, `{{Goto}`, `{{Label}}`)

(Lauffähiges Beispiel)

In diesem Beispiel zeigen Message Boxen den Fortlauf des Makros. Man sieht schön, wie man in das Unterprogramm reinkommt, und ans richtige Ort zurückspringt.

```
{{Gosub SubRoutine_1
{{MessageBox Nach Aufruf SubRoutine_1
{{Gosub SubRoutine_2
{{MessageBox Nach Aufruf SubRoutine_2
```

```
{{Goto Ende
{{Label SubRoutine_1
{{MessageBox In SubRoutine_1
{{Return
```

```
{{Label SubRoutine_2
{{MessageBox In SubRoutine_2
{{Return
```

```
{{Label Ende
{{MessageBox Nach Label Ende
```

1.15.1.13 `{{Master}}`

(Zusätzliche Befehle: `{{MessageBox}`, `{{SetGlobVar}`, `{{MLoop}}`)

(NICHT lauffähiges Beispiel. Für Lauffähigkeit müssten Macro1 und Macro2 vorhanden sein)

```
{}{Master
{}{SetGlobVarCounter 1
{}{LoadMacro C:\MyMacros\Macro1.krc
{}{LoadMacro C:\MyMacros\Macro2.krc
{}{MessageBox MLoop
```

Hinweise:

- Die Globale Variable Counter kann in beiden geladenen Makros verwendet werden.
- MLoop zeigt die Schlaufen des Master Makros.

1.15.1.14 *{}{WaitForIReady*

(Zusätzliche Befehle: {}{WindowKey, {}{If, {}{NoFocus)

(Lauffähiges Beispiel)

In diesem Beispiel wird die Internet Seite <http://www.kratronic.com> geöffnet. Erst nachdem die Seite fertig geöffnet ist, ertönt ein Beep

```
{}{WindowsKey r
http://www.kratronic.com
{ENTER}
{}{Delay 600} {}{ ' Millisekunden
{}{If {}{WaitForIReady 30} = {}{True};;{}{ShowError}
{}{Beep
{}{NoFocus
```

Hinweise:

- {}{WindowsKey r öffnet das 'Ausführen'-Fenster.
- NoFocus behält den Recorder im Hintergrund, sodass man gleich auf der Internet Seite weiterarbeiten kann (ohne das Fenster umschalten zu müssen)

1.15.1.15 *{}{GetPixel*

(Zusätzliche Befehle: {}{MessageBox, {}{RGB)

(Lauffähiges Beispiel)

In diesem Beispiel werden die Pixelfarben der Koordinaten 10,10 und 20,20 angezeigt.

```
{}{Mouse 1, 1, 10, 10, 1
{}{SetStringText Pixel an Position 10,10: {}{GetPixel -1, -1} {}{CR}
{}{SetStringText {}{StringText}-Pixel an Position 20,20: {}{GetPixel 20, 20}{}{CR}
{}{SetStringText {}{StringText}-RGB von 128,100,255: {}{RGB 128,100,255}.
```

```
{}{MessageBox {}{StringText}
```

Hinweise:

- {}{Mouse 1, 1, 10, 10, 1: Eine 1 damit der Zeiger nur platziert aber nicht geklickt wird

1.15.1.16 *{}{CheckWindow*

(Zusätzliche Befehle: {{GetWindowPos, {{GetWindowName, {{GetCursorPos, {{MessageBox

(Lauffähiges Beispiel)

In diesem Beispiel wird geprüft ob das Maus und Tastatur Recorder Fenster da ist. Zusätzlich werden die Koordinaten des Fensters und der Maus angezeigt.

```

{{If {{CheckWindow #MKRX5#} = {{True} ;{{MessageBox MTR existiert;{{MessageBox MTR existiert
nicht
{{MessageBox Aktueller Fenstername: {{GetWindowName}
{{GetWindowPos #MKRX5#
{{SetStringText MTR Fenster Position (Pixel){{CR}}{{CR}
{{SetStringText {{StringText}Horizontal oben links: {{VarGetWinCoord_X}}{{CR}
{{SetStringText {{StringText}Vertikal oben links: {{VarGetWinCoord_Y}}{{CR}
{{SetStringText {{StringText}Horizontal unten rechts: {{VarGetWinCoord_X_End}}{{CR}
{{SetStringText {{StringText}Vertikal unten rechts: {{VarGetWinCoord_Y_End}
{{MessageBox {{StringText}

```

```

{{RelativeMouse #MKRX5#, 1, 1, 25, 25, 1

```

```

{{GetCursorPos
{{SetStringText Cursor (Zeiger) Position (Pixel){{CR}}{{CR}
{{SetStringText {{StringText}Horizontal: {{VarGetCursorCoord_X}}{{CR}
{{SetStringText {{StringText}Vertikal: {{VarGetCursorCoord_Y}}{{CR}
{{MessageBox {{StringText}

```

Hinweise:

- #MKRX5# ist ein symbolischer Name für den Maus und Tastatur Recorder.
- {{RelativeMouse #MKRX5#, 1, 1, 25, 25, 1 (Das 1 bedeutet, nur Maus positionieren, nicht klicken.)

1.15.1.17 *Sort*

(Zusätzliche Befehle: {{SetVar, {{Var,{{IncVar, {{String, {{Goto, {{Label

(Lauffähiges Beispiel)

In diesem Beispiel werden 10 Zufallszahlen von 1-5 erzeugt und in ein Datenfeld abgelegt. Das Feld wird dann sortiert und die doppelten Zahlen entfernt (Leere Stellen enthalten dann eine Null)

```

{{SetVarCounter 1
{{Label Start
{{SetVarArray{{VarCounter} {{RandomNumber 1,5}
{{IncVarCounter 1
{{If {{VarCounter} <= 10 ;{{Goto Start;

```

```

{{Sort VAR, Array, 1, 10, {{True}

```

```

{{SetStringText Sortierte Zahlen (Doppelte Zahlen entfernt){{CR}}{{CR}
{{SetStringText {{StringText} 1: {{VarArray1}}{{CR}
{{SetStringText {{StringText} 2: {{VarArray2}}{{CR}
{{SetStringText {{StringText} 3: {{VarArray3}}{{CR}
{{SetStringText {{StringText} 4: {{VarArray4}}{{CR}
{{SetStringText {{StringText} 5: {{VarArray5}}{{CR}
{{SetStringText {{StringText} 6: {{VarArray6}}{{CR}
{{SetStringText {{StringText} 7: {{VarArray7}}{{CR}
{{SetStringText {{StringText} 8: {{VarArray8}}{{CR}

```

```

{{SetStringText {{StringText} 9: {{VarArray9}}{{CR}}
{{SetStringText {{StringText} 10: {{VarArray10}}{{CR}}
{' Für das Füllen von StringText hätte man natürlich auch Label und Goto verwenden können..
{' Das hätte dann so ausgesehen:

```

```

{{SetStringText Sortierte Zahlen (Doppelte Zahlen entfernt)}}{{CR}}
{{SetVarCounter 1
{{Label Start2
{{SetStringText {{StringText} {{VarCounter}: {{VarArray{{VarCounter}}}}{{CR}}
{{IncVarCounter 1
{{If {{VarCounter} <= 10 ;{{Goto Start2;

{{MessageBox {{StringText}

```

1.15.1.18 *Mathematische Befehle*

(Befehle: {{IncVar, {{DecVar, {{MulVar, {{DivVar, {{Power {{Div, {{Mod)

(Lauffähiges Beispiel)

In diesem Beispiel werden alle mathematischen Befehle verwendet. [\(Ausnahme: Befehle in {{Calc}\)](#)

```

{{SetVarExample 10.5
{{IncVarExample 2.7
{{MessageBox 10.5 + 2.7 = {{VarExample}

```

```

{{SetVarExample 10.5
{{DecVarExample 2.7
{{MessageBox 10.5 - 2.7 = {{VarExample}

```

```

{{SetVarExample 10.5
{{MulVarExample 3.5
{{MessageBox 10.5 * 3.5 = {{VarExample}

```

```

{{SetVarExample 10.5
{{DivVarExample 2.4
{{MessageBox 10.5 / 2.4 = {{VarExample}

```

```

{{SetVarExample {{Power 5 | 2}
{{MessageBox 5 hoch 2 = {{VarExample}

```

```

{{SetVarExample {{Div 10 | 3}
{{MessageBox 10 div 3 = {{VarExample} (Ganzer Wert der Division)

```

```

{{SetVarExample {{Mod 10 | 3}
{{MessageBox 10 mod 3 = {{VarExample} (Rest der Division)

```

```

{{If {{IsNumeric 23f3} = {{True} ;{{MessageBox 23f3 ist eine Zahl;{{MessageBox 23f3 ist keine Zahl
{{If {{IsNumeric 23e3} = {{True} ;{{MessageBox 23e3 ist eine Zahl;{{MessageBox 23e3 ist keine Zahl

```

1.15.1.19 *Text-Befehle*

(Befehle: {{Len, {{LnStr, {{Mid, {{Left, {{Right, {{Ucase, {{Lcase, {{Asc, {{Chr, {{Replace)

(Lauffähiges Beispiel)

In diesem Beispiel werden alle Text- Befehle verwendet.

```

{{SetVarLen {{Len Maus und Tastatur Recorder}}
{{MessageBox 'Maus und Tastatur Recorder' hat {{VarLen}} Zeichen

{{SetVarLen {{InStr 1 | Maus und Tastatur Recorder | Tastatur}}
{{MessageBox Bei 'Maus und Tastatur Recorder' beginnt das Wort Tastatur an Stelle: {{VarLen}}

{{SetStringText {{Mid Maus und Tastatur Recorder | 6 | 3}}
{{MessageBox Bei 'Maus und Tastatur Recorder' heisst das Wort an Stelle 6 das 3 Zeichen lang ist:
{{StringText}}

{{SetStringText {{Left Maus und Tastatur Recorder | 4 }}
{{MessageBox Bei 'Maus und Tastatur Recorder' heissen die ersten 4 Zeichen: {{StringText}}

{{SetStringText {{Right Maus und Tastatur Recorder | 8 }}
{{MessageBox Bei 'Maus und Tastatur Recorder' heissen die letzten 8 Zeichen: {{StringText}}

{{SetStringText {{Ucase Maus und Tastatur Recorder}}
{{MessageBox 'Maus und Tastatur Recorder' in Grossbuchstaben: {{StringText}}

{{SetStringText {{Lcase Maus und Tastatur Recorder}}
{{MessageBox 'Maus und Tastatur Recorder' in Kleinbuchstaben: {{StringText}}

{{MessageBox Der ASCII Wert des Buchstabens: 'M': {{Asc M}}
{{MessageBox Der Character des ASCII Wertes: 80: {{Chr 80}}

{{SetStringText {{Replace Maus und Tastatur Recorder | Tastatur | Key | 1 | -1}}
{{MessageBox Wenn man Tastatur mit Key ersetzt erhält man: {{StringText}}

```

1.15.2 *OR Funktion* ||

(Lauffähiges Beispiel)

Dieses Beispiel zeigt die Benutzung von OR (||). Das Makro zeigt alle Zahlen zwischen 1 und 30 die durch 2 und 3 teilbar sind.

```

{{Start Notepad
{{Delay 500
Zahlen welche durch 2 oder 3 geteilt werden können (1-30)
{ENTER 2}
{{SetVarCounter} 0

{{Label Start
{{IncVarCounter} 1
{{If {{Mod {{VarCounter} | 2} = 0;{{SetVarEven {{True}};{{SetVarEven {{False}}
{{If {{Mod {{VarCounter} | 3} = 0;{{SetVarThree {{True}};{{SetVarThree {{False}}
{{If {{VarEven} || {{VarThree} ;;{{Goto NichtBenoetigteNummer
{{Switch Editor
{{VarCounter}
{ENTER}
{{Label NichtBenoetigteNummer
{{If {{VarCounter} < 30;{{Goto Start;

```

1.15.3 Beispiele für allgemeine Aufgaben

1.15.3.1 Beispiel für das Wiederholen eines Makros oder eines Teiles des Makros

(Lauffähiges Beispiel)

Mit dieser Technik ist es möglich ohne die Verwendung der Schleifen-Funktion ganze Makros oder Teile davon beliebig oft zu wiederholen. In diesem Beispiel werden die Befehle ... 10 mal wiederholt.

```

{{SetVarZaehler 0
{{Label Start
{{IncVarZaehler 1
{{MessageBox Wir sind in Schlaufe: {{VarZaehler}
{'...
{'...
{{If {{VarZaehler} < 10 ;{{Goto Start;

```

1.15.3.2 {{If Befehl mit mehreren Befehlen

(Lauffähiges Beispiel)

Ein {{If Befehl im Recorder ~~konnte in früheren Versionen~~ kann nur je einen Befehl enthalten für den Then-Teil (Wahr-Aktion) und den Else-Teil (Falsch-Aktion). Dieses Beispiel zeigt wie man trotzdem mehrere Befehle für jeden Teil verwenden kann

```

.....
{{SetString1 ok
{{If {{String1} = ok ;;{{Goto Else
{{MessageBox Then Befehl 1
{{MessageBox Then Befehl 2
{{MessageBox Then Befehl 3
{{Goto EndElse
{{Label Else
{{MessageBox Else Befehl 1
{{MessageBox Else Befehl 2
{{MessageBox Else Befehl 3
{{Label EndElse

{{SetString1 not-ok
{{If {{String1} = ok ;;{{Goto Else2
{{MessageBox Then Befehl 1
{{MessageBox Then Befehl 2
{{MessageBox Then Befehl 3
{{Goto EndElse2
{{Label Else2
{{MessageBox Else Befehl 1
{{MessageBox Else Befehl 2
{{MessageBox Else Befehl 3
{{Label EndElse2

```

Ab Version 7 geht das einfacher:

```

{{SetString1 ok
{{If {{String1} = ok ;{{MessageBox Then Befehl 1$ {{MessageBox Then Befehl 2$ {{MessageBox Then
Befehl 3; {{MessageBox Else Befehl 1$ {{MessageBox Else Befehl 2$ {{MessageBox Else Befehl 3
{{SetString1 not-ok

```

```
{{If {{String1} = ok ;{{MessageBox Then Befehl 1$}}{MessageBox Then Befehl 2$}}{MessageBox Then Befehl 3;}}{MessageBox Else Befehl 1$}}{MessageBox Else Befehl 2$}}{MessageBox Else Befehl 3
```

1.15.3.3 Warten auf das vollständige Herunterladen einer Internetseite in Firefox

(Lauffähiges Beispiel)

Hier warten wir auf die Farb-Änderung des Stop-Knopfes. Wenn der Knopf grau wird, ist die Seite voll heruntergeladen.

```

{{Switch Mozilla Firefox
% {" Diesen Leerschlag nicht entfernen. Wichtig für das Maximieren
x {" Maximier Fenster (Damit ist der Stopknopf immer am gleichen Ort und wir finden ihn schneller
^l {" Adress-Zeile öffnen
www.kratronic.com
{ENTER}
{{Label WaitAgain
{{Delay 300} {" Millisekunden
{{Mouse 1, 1, 135, 61, 1
{{If {{FindPattern 1, -1, -1, 200, 200, 0} = {{True}};{{Goto WaitAgain

```

Hinweis: Damit das Makro funktioniert muss man noch ein Pattern (15x15 Pixel) des grauen Stop-Knopfes aufzeichnen. (Pattern Nummer 1)

Das Makro maximiert den FireFox und startet die gewünschte Seite. Dann wartet das Makro bis der rote Stop-Knopf grau wird. Dann geht's weiter.

1.15.3.4 RandomString mit eigenen Buchstaben

Mit dem Befehl `{{RandomString` sind nicht alle Kombinationen möglich z.B. nur die Buchstaben von a bis z und die Nummern 0 bis 9. Das folgende Makro ermöglicht dies.

```

{{SetGenDelay 0
{{SetVarAllCharacters 36 {" Anzahl der zur Auswahl stehender Buchstaben
{{SetStringCharacters abcdefghijklmnopqrstuvwxyz1234567890 {" Gewünschte Buchstaben (36 Stk)
{{SetVarAllCharacters {{Len {{StringCharacters}} {" Anzahl der zur Auswahl stehender Buchstaben
{{SetStringRandomChars #emptyX5#
{{SetVarNoOfChars 10 {" Anzahl gewünschter Buchstaben
{{SetVarCounter 0
{{Label Start
{{IncVarCounter 1
{{Gosub GetOneChar
{{SetStringRandomChars {{StringRandomChars}}{{StringChar}
{{If {{VarCounter} < {{VarNoOfChars} ;{{Goto Start;
{{MessageBox {{StringRandomChars} {" Anzeige der Zufalls-Buchstaben
{{Goto Ende

{{Label GetOneChar
{{SetStringChar {{Mid {{StringCharacters} | {{RandomNumber 1,{{VarAllCharacters}} | 1}
{{Return

{{Label Ende

```

1.15.3.5 Verschiedene Schnellzugriffs-Tasten Sets verwenden

Die Schnellzugriffs-Tasten (Hotkeys) sind in ~~einer eine~~ Datei abgespeichert. Das heisst, es gibt eigentlich nur ein Set von Schnellzugriffs-Tasten. Mit einem Trick kann man sich aber mehrere Sets bauen.

Die Schnellzugriffs-Tasten liegen in der Datei: `{{AppDataPath}}\MacroHotKeys.txt`. Um zu sehen wo diese Datei auf dem eigenen PC liegt einfach `{{MessageBox {{AppDataPath}}` ausführen. Nun kann man eine abgeänderte Schnellzugriffs-Tasten Datei auf `MacroHotKeys1.txt` speichern.

Mit folgendem Marko kann man zwischen den Schnellzugriffs-Tasten Sets umschalten:

```
{{CopyFile {{AppDataPath}}\MacroHotKeys.txt | {{AppDataPath}}\MacroHotKeysOriginal.txt
{{CopyFile {{AppDataPath}}\MacroHotKeys1.txt | {{AppDataPath}}\MacroHotKeys.txt
{{ReadHotKeys}}
```

`{{ReadHotKeys}` ist nötig, um die geänderten Schnellzugriffs-Tasten wieder einzulesen.

Hinweis: Die Schnellzugriffstasten wirken nicht wenn der Internet Explorer den Fokus hat. (Microsoft Massnahme gegen Phishing)

1.15.3.6 Weitere Beispiele

Während der Installation wird die Datei 'RecTest\RecText.zip' in das gleiche Verzeichnis, in welches der Recorder installiert wurde, kopiert. Die Dateien im ZIP können in das Verzeichnis C:\RecTest extrahiert werden. C:\RecTest enthält dann 18 Makros, welche für den Test des 'Maus und Tastatur Recorders' verwendet wurden. Die Makros sind so geschrieben, dass sie auf allen Betriebssystemen und in allen Sprachen funktionieren sollten. Nach dem Starten des Makros: 'RTestMaster.krc' werden alle anderen Makros automatisch aufgerufen und ausgeführt. (Bedingung: Die Makros müssen sich in C:\RecTest befinden)

Ein anderes kleines Beispiel-Makro kann unter dem Menü: '?' und dort unter 'Beispiel Makro öffnen' geladen werden.

Zusätzliche Beispiele können auf der folgenden Internet-Seite gefunden werden:

<http://www.kratronic.com/tools-de/recorder/faq/faq.htm>

1.16 Hinweise / Einschränkungen

- Ab Version V7.0 ist der Schalter für Makro-Schnellstasten und Schnellstasten für Stop, Pause etc. kombiniert.. Dieser Schalter hat aber einen kleinen Nebeneffekt. Die Sondertasten wie ^ und " verhalten sich anders. Normalerweise bewirkt der erste Tastendruck nichts, beim zweiten werden dann beide Tasten kombiniert. Beispiel: â, ë. Wenn der Schalter aktiviert ist (und der Recorder läuft), dann ergeben diese Tasten je zwei Zeichen, z.B. ^^, "". Ist dies nicht erwünscht, muss der Schalter ausgeschaltet werden (oder der Recorder muss geschlossen werden). Auch beim Abspielen von Makros hat ein aktivierter Schalter einen Einfluss auf diese Zeichen.
- Beim deutschen Windows wird das Zeichen {^} als & ausgegeben... Dies ist ein bekannter Fehler im deutschen Windows :- (Workaround: Zwischenablage und CTRL v verwenden.
- Der Befehl {{Calc hat sich in V7.0 geändert. Einige Befehle sind neu hinzugekommen, andere wurden entfernt oder geändert. Falls solche Calc-Befehle verwendet wurden, müssen diese angepasst werden.
- Der Befehl `{{FindPattern}` wurde verbessert in Version 6.7. Die Mustersuche ist nun ca. 7-mal schneller als vorher. Zusätzlich kann nun der Aero-Glass-Modus aktiv sein während der Muster-Suche.
- Auch der Befehl `{WaitForScreenChange}` wurde verbessert in V6.7. Auch hier kann der Aero-Glass-Modus aktiv sein während diesem Befehl.

- Der Befehl: ReplaceBraces.exe wurde in Version 6.7 ersetzt durch {{BracesReplace. Falls dieser Befehl verwendet wurde, muss er entsprechend abgeändert werden.
 - Windows Vista/Win7:
In Windows Vista/Win7 wurden mehrere Veränderungen vorgenommen, welche einen Einfluss auf den Maus und Tastatur Recorder haben:
 - ~~Wenn der Windows Aero (Aero-Glass-Modus) Modus eingeschaltet ist, verlangsamen sich 'GetPixel' Befehle enorm. Das bedeutet, dass dann der Recorder Befehl: {{FindPattern (alte-Version) Minuten, ja sogar Stunden dauern können... Um trotzdem mit diesem Befehl arbeiten zu können, kann der Aero-Glass-Modus temporär mit den Befehlen: {{WindowsAeroOff}, {{WindowsAeroOn} aus, und wieder eingeschaltet werden.~~
 - Unter Windows Vista/Win7 haben Programme und Web-Formulare zum Teil Probleme mit der normalen Recorder Textausgabe. (Zeichen oder ganze Wörter werden verschluckt). In diesem Fall hilft der Befehl {{SendTextSpecial} weiter.
 - ~~Die Ausgabe von Text in Windows Vista/Win7 hat geändert. Vista wartet nach der Ausgabe eines Buchstabens teilweise nicht mehr bis die Applikation diesen abgeholt hat, bevor der nächste Buchstabe ausgegeben wird. Die Ausgabegeschwindigkeit wird nur noch durch die Zeilenverzögerung bestimmt. Daher müssen ev. notwendige Verzögerungen manuell eingebaut werden.~~
 - Unter Windows Vista/Win7 ist ein Aufzeichnen von Tastendrücken und Mausclicks im Browser (z.B. Internet Explorer) nicht mehr möglich. (Microsoft Massnahme gegen Viren und Trojaner welche Pishing anwenden wollen). Um trotzdem Mausclicks aufzeichnen zu können, wurde der Menu Befehl: 'Einfügen' / 'Maus Eingabe' hinzugefügt. Um Tastendrücke und Schnellzugriffs-Tasten eingeben zu können, kann man in ein anderes Fenster wechseln (z.B. Notepad) oder den Text im Recorder Makro Edit Fenster eingeben.
 - Die oben erwähnte Einschränkung (kein Aufzeichnen im IE) wirkt auch bei den Schnellzugriffs-Tasten. (Schnellzugriffs-Tasten arbeiten gleich wie das Aufzeichnen) Das bedeutet, dass Schnellzugriffs-Tasten im Browser nicht funktionieren.
 - Der Befehl {{SendTextSpecial} schaltet ja auf einen anderen Ausgabe Modus um. Dabei wird nicht darauf gewartet, dass die Applikation den Tastendruck abholt bevor der nächste gesendet wird. Seiteneffekt: Nach einem ^c muss man genügend lange warten, bis man auf das Clipboard zugreift, sonst kriegt man den vorherigen Inhalt des Clipboards.
 - UltraVNC kann Probleme machen mit dem {{Switch Befehl.
 - Der Recorder prüft bei jedem Befehl den Befehlsaufbau und gibt bei falschem Aufbau eine Fehlermeldung aus. Wenn allerdings mehrere Fehler in einer Zeile vorhanden sind, kann es passieren, dass der Recorder abstürzt :- (Laufzeitfehler) In diesem Falle in der Log-Buch-Datei nachsehen welches der vorletzte Befehl war und den fehlerhaften Befehl korrigieren.
 - Das Vergleichen von Text kopiert aus Excel Zellen ist ein wenig kompliziert. Ein kopierter Text aus einer Excel Zelle enthält immer ein CR und LF (Zeilenende und Zeilenvorschub) am Ende des Textes. Das bedeutet, dass der {{If Befehl folgendermassen aussehen muss: {{If {{Clipboard} = TEXT{{CR}}{LF};Wahr-Aktion;Falsch-Aktion. Wenn man Texte aus Excel weiterverwenden will muss man ev. diese CR und LF entfernen. Hier eine Möglichkeit:


```

{{Switch Microsoft Excel
^c
{{Delay 300} {{' Milliseconds
{{SetVarLen {{Len {{Clipboard}}
{{DecVarLen 2}
{{SetStringExcel {{Trim {{Left {{Clipboard} | {{VarLen}}}

```

Hinweis: Im neueren Office (z.B. Office 2003) werden manchmal Leerschläge zum Zelleninhalt hinzugefügt. Diese kann man mit dem zusätzlichen {{Trim entfernen.
 - In bestimmten Situationen sind die Reaktionen auf ein Makro nicht die gleichen wie wenn man diese Aktionen manuell macht. Der Grund für diese 'falschen Reaktionen' sind spezielle Programmieretechniken in bestimmten Programmen. Oft verwenden diese Programme 'schmutzige' Programm-Tricks. In solchen Situationen muss versucht werden einen anderen Weg zu finden um das gleiche Resultat zu erreichen.
- ~~Der 'Maus und Tastatur Recorder' arbeitet im Normal-Text Modus nicht in DOS Boxen ! Für DOS-Boxen den Spezial-Text Modus: {{SendTextSpecial verwenden.~~

~~Während dem Makroablauf wird 'NumLock' automatisch ausgeschaltet. Nach dem Ende des Makros wird wieder der Zustand vor dem Beginn des Makros hergestellt. Grund: Auf Windows 98/ME laufen Makros sehr langsam, wenn NumLock eingeschaltet ist. Wenn während des Makro-Ablaufes Nummern eingetippt werden müssen, kann NumLock mit {{NumLockOn und {{NumLockOff ein- und ausgeschaltet werden.~~

- Einige Programme reagieren äusserst eigenartig auf die rechte Maustaste. Die folgenden Programme erzeugen 3 bis 6 Klicks für einen rechten Maus Tasten Klick: [Aeltere Versionen von Microsoft Word, Excel, Power Point](#). Bei Verwendung der rechten Maustaste sollten eventuelle überzählige Mausklicks gelöscht werden. Dies kann mit dem Makro-Editor erfolgen.
- Das Programm Notepad hat ein eigenartiges Verhalten. Wird mittels dem Recorder das Notepad Fenster mit ALT F4 geschlossen, dann wird nicht nur Notepad beendet, sondern auch der 'Maus und Tastatur Recorder' ! Dies passiert nicht wenn man mehrere Notepads offen hat. Lösung: Datei | Beenden verwenden um den Notepad zu schliessen.

~~Der 'Maus und Tastatur Recorder' erinnert sich an die alte Position, wenn er neu gestartet wird. Diese Funktion kann im Menü: 'Extras' / 'Optionen' / 'Allgemein' / 'Fenster Grösse und Position speichern' ausgeschaltet werden.~~

- Alle Beispiele in dieser Anleitung zeigen eine schliessende geschwungene Klammer nach Ausgabe-Befehlen. Beispiel: `{{Var1}}` Diese Klammer ist absolut notwendig, wenn der Befehl als Parameter verwendet wird. Beispiel: `{{SetVar1}} {{Loop}}`. Hier muss `{{Loop}}` eine schliessende Klammer haben. Wenn `{{Loop}}` jedoch am Anfang der Zeile verwendet wird, dann kann diese schliessende Klammer weggelassen werden.

Beispiele:

```
{{Loop
{{SetVar1 {{Loop}}
```

- Wichtig: Befehle, welche nicht als Parameter verwendet werden und Text enthalten, dürfen keine geschwungene schliessende Klammer enthalten. Beispiele:

```
{{SetString1} MyText
{{Switch Document2 Microsoft Word
```

~~Speicher wie z.B. {{Var, {{String, {{Ask werden in Dateien abgespeichert. Das hat den Vorteil, dass die Speicher dauerhaft sind. Das bedeutet, dass der Inhalt der Speicher auch nach einem PC-Neustart noch zur Verfügung steht. Diese Technik hat aber auch einen Nachteil. Wenn man sehr viele Speicher verwendet (mehr als 500 bis 1000) werden die Makros langsamer. (Die Verlangsamung ist abhängig von der Prozessor und Festplatten Geschwindigkeit). Diese Einschränkung gilt nicht mehr für die Version V6.5 und höher. Hier werden die Speicher im RAM gehalten und erst am Ende des Makros in die Datei abgespeichert.~~

- Globale Speicher werden für alle Makros ~~auch bei Version 6.5 und höher~~ nicht im RAM sondern in einer gemeinsamen Datei gespeichert. Das bedeutet, dass die globale Datei mit der Zeit wachsen kann. Wenn die globale Datei mehr als 500 Einträge enthält, kann der Makroablauf verlangsamt werden. Wenn dies passiert, sollte die globale Speicherdatei (oder Teile davon) gelöscht werden. Der Speicherort der globalen Speicher-Datei: Global4107Kra.var hängt vom Betriebssystem ab. Am Besten mit der Such-Funktion die Datei suchen. Leider findet die Windows eigene Such-Funktion solche lokalen Dateien nicht :(Eine gute Alternative zur allgemein unbefriedigenden Windows Such Funktion ist das Programm: "Effective File Search" <http://www.sowsoft.com/search.htm> Eine andere Möglichkeit diese Datei zu finden ist der Befehl: `{{MessageBox {{AppDataPath}}`
- Makros, welche mit einer bestimmten Bildschirm-Auflösung oder einem bestimmten Betriebssystem aufgezeichnet wurden, laufen ev.oft nicht mehr richtig in anderen Auflösungen oder auf anderen Betriebssystemen. Diese Einschränkung ist nicht gültig, wenn im Makro keine Maus und Fokus-Befehle verwendet wurden.
- Bei neueren Betriebssystemen (Windows ~~98, ME, 2000~~, XP, Vista und höher) werden die Benutzer-Makros standardmässig in das Verzeichnis: 'Eigene Dateien \Recorder' gespeichert. Die Einstellungen des 'Maus und Tastatur Recorders' werden in ein Unterverzeichnis des Verzeichnis 'Dokumente und Einstellungen' abgelegt. Das bedeutet, wenn man nur die Makros sichern will (Datensicherung, Backup) dann reicht es das Verzeichnis 'Eigene Dateien' zu sichern. Will man auch die Einstellungen (Hotkeys, Windows Filter etc.) sichern, muss das

- Verzeichnis AppData (Verstecktes Verzeichnis) in Verzeichnis Benutzer'Dokumente und Einstellungen' gesichert werden.
- Der Maus und Tastatur Recorder verwendet für den Editor eine Komponente von Microsoft (RichTextBox). Frühere Versionen benutzen die Komponente "TextBox" die aber eine Beschränkung auf 32 KB hatte. Die neue RichTextBox hat ein sehr eigenartiges Verhalten unter Windows 2000. Wenn eine Zeile genau auf der Grenze von 2048 (oder 4096 oder 6144 etc.) Charakter endet, fügt RichTextBox automatisch ein CR (Carriage Return) hinzu. Der Maus und Tastatur Recorder hat dann Probleme das Makro zu parsen (die einzelnen Befehle zu trennen). Die Auswirkung sind **{{Goto** Befehle die einen existierenden **{{Label** nicht finden oder **{{Switch** Befehle die ein existierendes Fenster nicht finden. Es sind auch andere seltsame Effekte möglich. Um den Fehler zu beheben muss vor der magischen Grenze ein (oder mehrere) weitere Charakter eingefügt werden. Wenn der Recorder eine solche Situation findet, hängt er beim Laden oder Speichern vor der magischen Grenze einen Kommentar ein. (**{{Win2K_Err_Corr**) Dadurch ist die Grenze verschoben und der Fehler wirkt sich nicht mehr aus.

1.17 Installation

Der 'Maus und Tastatur Recorder' wird als ZIP-Datei geliefert. Diese ZIP-Datei muss erst dekomprimiert werden. Moderne Betriebssystemen wie Windows XP oder höher können selbstständig ZIP-Dateien entpacken. ~~Für ältere Betriebssystem findet sich ein gutes Werkzeug, um die ZIP-Datei zu dekomprimieren, unter folgender Adresse: <http://www.winzip.com>~~

Nach dem dekomprimieren der ZIP-Datei muss nur noch die Datei Setup.exe gestartet werden. Die folgenden Fenster führen dann durch die Installation. Wichtig: Für die Installation sind Administrator Rechte nötig. ~~(Ausnahme: Windows 98 and ME).~~

Abhängig vom Betriebssystem verlangt die Installation erst das Installieren von .NET Framework 2. ~~Dieses kann automatisch über das Internet erscheinen (Datei neuer... etc.) In diesen Fällen der Empfehlung, welche ebenfalls erscheint,~~ folgen.

Wenn der 'Maus und Tastatur Recorder' von einer anderen Person installiert wurde, dann fehlt eventuell der Recorder Eintrag im Start Menu. In diesem Fall muss manuell eine Verknüpfung erstellt werden. Dazu klickt man mit der rechten Maustaste auf das Programm C:\Programme\KraTronic\Recorder\Recorder.exe und wählt: Verknüpfung herstellen. Die erzeugte Verknüpfung kopiert man dann auf den Arbeitsplatz. (Wenn der Recorder nicht in C:\Programme installiert wurde, kann Recorder.exe mit der Windows Funktion: 'Suchen nach Dateien und Ordner' gefunden werden)

Hinweise:

~~Auf einigen PCs passiert das folgende:~~

~~Man beginnt den Maus und Tastatur Recorder zu installieren. Die Installation verlangt dann einen Neustart. Nach dem Neustart und dem erneuten Starten der Installation wird wieder ein Neustart verlangt.... und das immer wieder und wieder.~~

~~Es gibt verschiedene Gründe für dieses Fehlverhalten. Oft sind es Dateien welche durch andere laufende Programme blockiert sind und nicht erneuert werden können. Daher ist es wichtig vor dem Installieren alle Programme zu schliessen. (Auch jene in der rechten unteren Ecke des Bildschirms (System Try). Wenn dies nicht hilft, (weil man nicht alle Programme schliessen konnte) dann sollte man den Recorder im abgesicherten Windows Modus installieren. Den abgesicherten Modus erreicht man, wenn man am Anfang des Windows Starts F8 drückt.~~

~~Weitere Informationen finden sich auf der Microsoft Seite:-~~

~~<http://support.microsoft.com/default.aspx?scid=KB;DE;Q191096&>~~

Probleme mit dem Virens scanner:

Einige Virens scanner blockieren die Installation von neuen Programmen. Falls dies passiert, muss temporär während der Installation der Virens scanner ausgeschaltet werden. Nachher für den Betrieb des Recorders kann der Virens scanner natürlich wieder eingeschaltet werden.

1.17.1 *PC-unabhängige Installation*

Falls der Maus und Tastatur Recorder ohne Installation auf einem PC verwendet werden soll, kann das Tool: MojoPac (<http://www.mojopac.com/>) verwendet werden (Nur Windows XP, gratis) oder Prayaya V3 (<http://en.prayaya.com/index.html>) (Alle Windows Versionen, Shareware)

1.18 *Installation aktualisieren*

Es wird empfohlen einen bereits installierten 'Maus und Tastatur Recorder' erst zu deinstallieren (entfernen), bevor eine höhere Version installiert wird. Vorgehen: Klick Start, Einstellungen, Systemsteuerung, Software (oder 'Software hinzufügen/entfernen' oder 'Programme und Funktionen'). Wähle 'Recorder' und klicke 'Hinzufügen/Entfernen/Ändern' oder 'Deinstallieren/Ändern'. Wenn gleich vor dem deinstallieren mit dem Recorder gearbeitet wurde, wird empfohlen nach dem deinstallieren den PC neu zu starten (reboot).

~~Wenn schon mal eine Test-Version des Recorders installiert war, kann keine neuere Test-Version installiert werden. (Das Speichern der Makros funktioniert dann nicht). Mit einer lizenzierten Version ist das Abspeichern der Makros natürlich wieder möglich.~~

~~Wenn eine neuere Version des Recorders installiert wird (ohne vorher zu deinstallieren) fragt das Installations-Programm ob alte Dateien beibehalten werden sollen. Es muss mit 'Nein für alle' geantwortet werden.~~

Hinweis: Aufgezeichnete Makros und die Recorder Einstellungen werden nicht gelöscht, wenn der Recorder deinstalliert und neu installiert wird.

1.19 *Aktualisierungs- Information*

~~Version 7.0 wurde mit VB.Net 2005 entwickelt (Frühere Versionen mit VB 6.0). Dies hat einige Anederunge in der Benutzung des Recorders zur Folge. Falls bereits frühere Versionen benutzt wurden, kann man im "Delta Dokument" alle Aenderungen finden (markiert). Siehe FAQ (HFG) für den Link zum Dokument.~~

Version V6.5 unterstützt Zahlenspeicher-Dateien (.VAR) welche mit V5.3 oder älter erstellt wurden, nicht mehr. Wenn ein Makro von einer solchen alten Zahlenspeicher-Datei begleitet ist, wird diese alte .VAR Datei gelöscht. Vor dem Löschen wird die Datei mit der Endung _backup gesichert. Normalerweise beeinflusst das Löschen der VAR Datei das Makro in keiner Weise. Nur wenn Zahlenspeicher den Wert für das nächste Starten des Makros behalten müssen (z.B. Zähler) muss dieser Wert wieder neu gesetzt werden. Grund für diese Änderung war eine Verbesserung des Maus und Tastatur Recorders. Der Recorder speichert nun die Zahlenspeicher-Werte im Rechner RAM und schreibt sie erst am Ende des Makros in eine Datei. Dadurch laufen Makros schneller und der manchmal auftauchende Fehler: *VAR Datei konnte nicht gelöscht/umbenannt* werden kann nicht mehr vorkommen.

Ab V5.4 kann mit **Switch** nicht mehr direkt zum Recorder umgeschaltet werden. Will man den Recorder aktivieren, muss man den symbolischen Namen: #MKRX5# verwenden. Beispiel: **Switch #MKRX5#**. Dieser symbolische Name kann auch bei **SwitchOrStart** und **CheckWindow** angewendet werden.

Version V5.3 (und kleiner) enthielt einen kleinen Fehler beim Speichern von Text- und Zahlenspeichern. Die Speichernamen wurden in Gross-Kleinschrift gespeichert, aber beim Lesen spielte Gross-Kleinschrift keine Rolle mehr. Das führte dazu dass z.B. "varPeter" und "varpeter" zwei

mal gespeichert wurde. Beim Lesen wurde dann immer nur varPeter gelesen. (Der erste Eintrag) Das geänderte, korrekte Speichern der Speichernamen kann nur auf neue Makros angewendet werden. (Weil sonst noch mehr doppelte Einträge entstehen.) Will man auch bei bestehenden Makros diesen Fehler weg haben, kann man die VAR-Dateien löschen. **Achtung:** Wenn man Speicher verwendet, welche auch nach einem Beenden des Makros weiterlaufen, werden diese durch das Löschen der VAR Dateien zurückgesetzt. (z.B. ein Zähler der sicherstellt, dass bei jeder Verwendung des Makros ein anderer Dateiname verwendet wird.) Vorgehen: Menu: Datei | Oeffnen, dann den Dateityp auf "All Files" umstellen und alle Dateien mit der Endung .VAR löschen. Oeffnenen-Fenster schliessen und im Makrocode Fenster folgenden Befehl eingeben: `{{Start Explorer {{AppDataPath}}}`. Makro laufen lassen und im erscheinenden Fenster die beiden Dateien mit Endung .VAR löschen. (Temporary und Global)

In Version V5.0 ist die Verwendung von Text- und Zahlenspeicher ohne Nummer oder Name nicht mehr erlaubt. Z.B. `{{Var}}`. Solche Speicher müssen mit einer Nummer oder einem Namen ergänzt werden. Z.B. `{{Var1}}` oder `{{VarMeineZahl}}`

~~Version 4.5 enthält eine neue Lizenz Technik. Diese soll die freie Zirkulation von gehackten Lizenz-Kodes im Internet verhindern. Der Nachteil dieser Technik ist, dass es nicht möglich ist die Recorder Test Version zu installieren, danach zu de-installieren und wieder zu installieren. Nach diesem Vorgehen können Makros nicht mehr abgespeichert werden. (Mit einer lizenzierten Version ist das Abspeichern der Makros natürlich wieder möglich)~~

1.20 Ablehnungshinweis

Der 'Maus und Tastatur Recorder' wurde von uns eingehend auf Fehler getestet. Der Autor des 'Maus und Tastatur Recorder' kann nicht verantwortlich gemacht werden für Schäden, welche durch die Benutzung des 'Maus und Tastatur Recorders' auf dem PC des Benutzers auftreten könnten.