

8

Ausgewählte produktspezifische Befehle

Was produktspezifische Befehle sind, wofür sie verwendet werden, und wie sie zu handhaben sind, wurde bereits in Kapitel »5.1 Produktspezifische Befehle« beschrieben.

Eine detaillierte Beschreibung aller produktspezifischer Befehle (über 1.300) würde den Rahmen dieses Buches sprengen. Sie ist außerdem auch nicht erforderlich, da einerseits die überwiegende Anzahl der Begriffe selbsterklärend ist und sie andererseits im Makrobefehlsmanager mit allen Parametern und zugehörigen Ausdrücken angezeigt werden. In dem OnLine-Makrohandbuch sind diese Befehle an der Groß-/Kleinschreibung zu erkennen. Für die Auswahl eines Befehls wählen Sie im Makrobefehlsmanager [*Befehlsart*, *WordPerfect-DE*]. Unter [*Befehle*] erscheinen in der angezeigten Liste zuerst die Systemvariablen (an dem vorangestellten Fragezeichen zu erkennen) und dann die Produktbefehle.

In diesem Kapitel erfolgt nur die Beschreibung einiger wichtiger Befehle, die in der Praxis häufig benötigt werden.

8.1 Display

Befehlsform:	Display(State:Status)
---------------------	------------------------------

Die Makro-Ausführung, d. h. der Ablauf des Makros, wird normalerweise am Bildschirm nicht angezeigt, Dialogfelder oder Menüs, die ausgewählt werden, sind darum nicht sichtbar. Unter bestimmten Umständen, besonders beim Testen von Makros (Fehlersuche) ist die Anzeige aber manchmal erforderlich. Über den Befehl *Display* können Sie die Anzeige ein- oder ausschalten. Ist die Anzeige eingeschaltet, wird die Makro-Ausführung merklich verlangsamt. Standardmäßig ist die Anzeige ausgeschaltet. In der Statuszeile wird während des Makroablaufes evtl. der Hinweis »Bitte warten« eingeblendet, wenn eine Eingabe erforderlich ist.

Wenn Sie während der Makro-Ausführung wieder Menüs, Dialogfelder bzw. Texte unterdrücken möchten, fügen Sie an der betreffenden Stelle *Display(State:On!)* ein. Jeder dieser Befehle hat so lange Gültigkeit, bis er durch das Gegenteil widerrufen wird. Schalten Sie die Anzeige aus, um besonders bei lang laufenden Makros die volle Verarbeitungsgeschwindigkeit zu erreichen. Dies gilt insbesondere, wenn Sie über Makroroutinen eine größere Tabelle mit Daten füllen.

Parameter

State:	Parameter zum Ein- oder Ausschalten der Anzeige:	
	On!	Anzeige wird eingeschaltet.
	Off!	Anzeige wird ausgeschaltet.
Beispiel:	Display	(State:On!)

Der Parametername *State:* ist ein Lesbarkeitswort, das Sie auch weglassen können: *Display(On!)*.

8.2 Edit (Bearbeiten)

Befehlsform:	Edit...()
---------------------	------------------

Über die Befehle *SelectMode...* können Sie diverse Bereiche eines Dokumentes selektieren (siehe *SelectMode..()*). Für die Weiterverarbeitung der selektierten Bereiche können Sie u.a. folgende Befehle verwenden:

EditCut	BearbeitenAusschneiden. Selektierter Text oder Grafik werden im Dokument gelöscht und in der Zwischenablage gespeichert. Der bisherige Inhalt der Zwischenablage geht dadurch verloren.
EditPaste	BearbeitenEinfügen. Der Inhalt der Zwischenablage wird im Dokument auf Cursorposition eingefügt.
EditCopy	BearbeitenKopieren. Selektierter Text oder Grafik werden in der Zwischenablage gespeichert. Der bisherige Inhalt der Zwischenablage geht dadurch verloren.

8.3 FileSave

Befehlsform:	FileSave(FileName;Exporttyp;Overwrite)
---------------------	-----------------------------------------------

Mit diesem Befehl können Sie den Inhalt des aktuellen Dokumentfensters als eine Datei speichern. Wurde das Dokument bisher noch nicht gespeichert, wird das übliche Dialogfeld zum Speichern von Dokumenten angezeigt. Nach der Eingabe eines Dateinamens durch den Benutzer wird die Makro-Ausführung fortgesetzt. Ist das Dokument bereits gespeichert, wird es erneut unter diesem Namen gespeichert.

Parameter

FileName:	Geben Sie hier einen Dateinamen ein, oder geben Sie den Namen einer Variablen an, die den Dateinamen enthält.	
ExportType:	Wenn Sie diesen Parameter verwenden, können Sie ein Dateiformat angeben, in dem das Dokument gespeichert werden soll. Die zur Verfügung stehenden Formate können Sie der betreffenden Feldliste des Makrobefehlsmanagers entnehmen.	
Overwrite:	Damit der Makroablauf (nicht) unterbrochen wird, können hier folgende Optionen gewählt werden:	
	Yes!	Dateien mit demselben Namen werden überschrieben.
	No!	Dateien mit demselben Namen werden nicht überschrieben.
	Prompt!	Bei Dateien mit demselben Namen wird eine Abfrage eingeblendet, ob die betreffende Datei überschrieben werden soll.

Beispiel

```
FileSave(FileName:"C:\WPTEXTE\BRIEF.TXT<<";
ExportType:WordPerfect_5.1!;
Overwrite:Yes!)
```

Die Datei BRIEF.TXT wird in dem Textformat der WordPerfect-Version 5.1 gespeichert. Ist bereits eine Datei mit demselben Namen vorhanden, wird diese ohne Rückfrage überschrieben.

8.4 Dlg

Befehlsform: ...Dlg

Befehle mit der Endung ...Dlg blenden das Dialogfenster des betreffenden Befehls ein. Die Makroausführung wird an dieser Stelle so lange unterbrochen, bis das Dialogfeld über eine der zur Verfügung stehenden Tasten geschlossen wird. Zum Erstellen eigener Dialogfelder können Sie die in Kapitel 6 beschriebenen *Dialog...*-Befehle verwenden.

8.5 MacroStatusPrompt

Befehlsform: **MacroStatusPrompt**(Status;Prompt)

Dieser Befehl zeigt eine Meldung in der Statuszeile so lange an, bis sie durch eine andere Meldung überschrieben (z. B. erneuter *MacroStatusPrompt*) bzw. ausgeblendet wird. Die Meldung wird nur dann angezeigt, wenn die Statuszeile eingeblendet ist. Stellen Sie ggf. im Makro sicher, daß dies der Fall ist. Verwenden Sie hierzu den produktspezifischen Befehl *StatusBarShow*(Yes!).

Parameter

Status: Geben Sie hier an, ob das Anzeigen der Meldung ein- oder ausgeschaltet werden soll:

Off! / On! Meldung aus- / einschalten.

Prompt: Hier können Sie eine Meldung eingeben, die maximal 128 Zeichen lang sein kann. Die Anzahl der Zeichen, die angezeigt werden, richtet sich nach der Größe des WordPerfect-Fensters. Wird *Off!* angegeben, wird keine Meldung angezeigt. Das hat denselben Effekt, als wäre der Befehl nicht ausgeführt worden.

8.6 Select...

Befehlsform: **Select...**()

Für das Markieren von Texten oder Teilen davon können Sie diverse *Select...*-Befehle verwenden. Eine Auflistung der einzelnen Befehle ist an dieser Stelle nicht erforderlich, da Sie diese aus der Befehle-Feldliste des Makrobefehlsmanagers ansehen können. Dort können Sie je nach Bedarf einen der benötigten Befehle auswählen. Bei der späteren Makro-Ausführung wird der Befehl in Abhängigkeit der jeweiligen Cursorposition ausgeführt. Der markierte Text kann dann mit den Befehlen *Edit...* u.a. angehängt, ausgeschnitten, eingefügt und kopiert werden (bezieht sich jeweils auf die Zwischenablage (siehe Abschnitt »8.2 Edit (Bearbeiten)«). Einen Teil der Befehle können Sie während der Makro-Aufzeichnung mit den üblichen Funktionstasten ausführen, die restlichen vielleicht benötigten Befehle müssen Sie über den Makrobefehlsmanager in das betreffende Makro einfügen.

8.7 PauseCommand

Befehlsform: `PauseCommand1(Token)`

Über diesen Befehl können Sie ein Makro an jeder beliebigen Stelle unterbrechen, um z.B. für eine bestimmte Zeit andere Arbeiten in WordPerfect (z.B. Texteingabe) oder einer anderen Anwendung auszuführen. Die Pause wird erst dann beendet und das Makro fortgesetzt, wenn der unter *Token* angegebene Befehl ausgeführt wird. Dies kann ein beliebiger Makrobefehl sein oder eine bestimmte Taste. Die Makropause kann auch durch Auswählen von [*Tools, Makro, Pause*] beendet werden.

Parameter

Token: Angabe eines beliebigen Makrobefehls.

Beispiel

Application	(A1;"WordPerfect<<Default;"DE<<)
MBefehl:=	Value(Indent)
PauseCommand	(Token:MBefehl)
Type	("Ende der Pause")

Nach der Ausführung des Befehls *PauseCommand* wird hier das Makro so lange unterbrochen, bis [Einrücken] gedrückt wird. Danach wird die Makroausführung fortgesetzt.

Weitere Hinweise siehe: *Pause, PauseKey, PauseDefinieren, Value*

8.8 PauseSet

Befehlsform: `PauseSet(Token)`

Über diesen Befehl können Sie eine unterbrochene Makroausführung wieder fortsetzen. Die Pause wird erst dann beendet und das Makro fortgesetzt, wenn der unter *Token* angegebene Befehl ausgeführt wird. Dies kann ein beliebiger Makrobefehl sein oder eine bestimmte Taste. Die Makropause kann auch durch Auswählen von [*Tools, Makro, Pause*] beendet werden.

Parameter

Token: Angabe eines beliebigen Makrobefehls.

Beispiel

Application	(A1;"WordPerfect<<Default;"DE<<)
MBefehl:=	Value(Indent)
PauseSet	(Token:MBefehl)
Pause	
Type	(>>Ende der Pause<<)

Nach der Ausführung des Befehls *Pause* wird hier das Makro so lange unterbrochen, bis [Eintrücken] gedrückt wird. Danach wird die Makroausführung fortgesetzt.

Weitere Hinweise siehe: *Pause*, *PauseBefehl*, *PauseTaste*, *Value*

8.9 PauseKey

Befehlsform: `PauseKey(Key;Character)`

Unterbrechung der Makro-Ausführung, um z.B. eine Eingabe über die Tastatur zu verlangen oder um eine gewollte Unterbrechung (Anzeige einer Meldung) durch den Anwender beenden zu lassen. In Verbindung mit diesem Befehl können Sie keine Ein- oder Ausgabeoperationen ausführen.

Um die Unterbrechung des Makros zu beenden, müssen Sie die Taste drücken, die Sie unter *Key* oder *Character* definiert haben (*Eingabe*, *Abbrechen*, *Exit* oder ein beliebiges Zeichen). Das Makro arbeitet dann mit dem Befehl weiter, der der Taste *PauseKey* folgt.

Dieser Befehl darf nicht mit dem Menüpunkt [*Tools*, *Makro*, *Pause*] verwechselt werden. Bei der Verwendung dieses Befehls kann die Pause nur wieder durch Deaktivieren des Menüpunkts beendet werden.

Parameter

Key: Zum Beenden der Pause können Sie unter *Key* einen der folgenden Parameter verwenden:

Any! Cancel!
Character! Close!
Enter!

Beim Drücken einer dieser Tasten wird die Makropause beendet. Wenn Sie *Character!* gewählt haben, müssen Sie unter dem Parameter *Character* das Zeichen eingeben, mit dem Sie die Pause beenden möchten.

Character: Wenn Sie die Pause nicht über eine der zuvor genannten Tasten beenden möchten, können Sie ein beliebiges Zeichen zum Beenden der Pause definieren. Voraussetzung ist, daß Sie unter *Key* den Parameter *Character* ausgewählt haben. Es ist nur ein Zeichen erlaubt. Haben Sie mehrere Zeichen eingegeben, wird nur das erste aus der Zeichenkette akzeptiert.

Beispiel: `PauseKey(Character!;"K<")`

In diesem Beispiel wird die Pause durch Drücken des Großbuchstabens K beendet (WordPerfect unterscheidet hier zwischen Groß- und Kleinschreibung).

PauseKey(Key:Enter!)

In diesem Beispiel wird die Pause durch Drücken von *Eingabe* beendet. Wenn Sie diese Taste auswählen, können Sie bei der Texteingabe während der Makropause nicht *Eingabe* zum Beenden einer Zeile verwenden!

Weitere Hinweise siehe: Pause in Kapitel »6 Makro-Programmierbefehle«

8.10 Type

Befehlsform: **Type**(Text)

Über diesen Befehl können Sie Text in das aktuelle Dokument einfügen. Das Einfügen des Textes erfolgt immer auf Cursorposition. Unter *Ausdruck* können eine Konstante oder eine Variable, die jeweils Text enthalten, angegeben werden. Konstanten müssen durch Hochkommata eingeschlossen werden. Enthält eine Variable numerische Daten, brauchen diese nicht mehr wie bisher über den Befehl *NumStr* in einen Zeichenausdruck konvertiert werden. Der Befehl *NumStr* kann (wenn erforderlich) ab dieser Version direkt mit dem Befehl *Type* verwendet werden.

Parameter

Text: Konstante oder Name einer Variablen, die Daten enthalten, die in das aktuelle Dokument eingefügt werden sollen. Werden innerhalb einer Konstanten Hochkommata verwendet, müssen diese doppelt angegeben werden. Es erfolgt eine Fehlermeldung, wenn die angegebene Variable nicht vorhanden ist, oder wenn bei Konstanten die Hochkommata vergessen wurden.

Beispiel

```
Type      ("Das ist ein Test")      //Verwendung einer Konstanten.
Type      (Text:"Das ist ein Test")      //Verwendung einer Konstanten.
```

Der Begriff *Text*: in der Klammer ist ein Lesbarkeitswort, das Sie auch weglassen können (siehe Beispiel).

```
Type      (Nachname)      //Verwendung einer Variablen.
Ergebnis:=      10 * 7
Type      (Ergebnis)      //Verwendung einer Variablen.
```

In diesem Beispiel wird das Produkt von $10 * 7$ ermittelt und in *Ergebnis* gespeichert. Der Befehl *Schreiben* fügt das Ergebnis an der Cursorposition in das Dokument ein.

```
Type      NumStr(Ergebnis)
```

Umwandeln eines Strings in einen numerischen Wert und Schreiben des Wertes.

8.11 Systemvariable

Befehlsform: *?Variablenname*

Systemvariablen sind von WordPerfect benutzte Speicherfelder, die Sie nur lesen können, d.h. Sie können diesen Variablen keine Daten zuordnen. Die Namen sind fest vorgegeben und können von Ihnen nicht verändert werden. Die Inhalte dieser Felder beziehen sich immer auf den aktuellen WordPerfect-Status zum Zeitpunkt der Abfrage. Welche Bedeutung sie haben, können Sie der Auflistung dem Makrobefehlsmanager oder der Hilfefunktion entnehmen. Sie zählen zu den produktspezifischen Befehlen und enthalten vor dem Variablennamen ein Fragezeichen. Wählen Sie im Makrobefehlsmanager [*Befehlsart, WordPerfect-DE*].

Beispiel:

PosLineBeg	()
SelectMode	(On!)
PosLineEnd	()
SuchText:=	?SelectedText

In diesem Beispiel wird der Cursor am Anfang einer Zeile platziert. Danach wird die Blockfunktion eingeschaltet und der Cursor zum Zeilenende gesteuert. Der markierte Text wird in der Systemvariablen *?SelectedText* gespeichert. Der letzte Befehl übernimmt den markierten Block in die benutzerdefinierte Variable *SuchText*.

8.12 TypeOver

Befehlsform: *TypeOver(Status)*

Bei der Texteingabe wird zwischen zwei Eingabearten unterschieden: Einfügen oder Überschreiben. Der Wechsel von der einen zur anderen Art erfolgt durch Drücken von Einfg. Welche Eingabeart gewählt wurde, wird im linken Teil der Statuszeile angezeigt, sofern in der Statuszeile das Feld *AllgemeinerStatus* ausgewählt wurde. Wird »Überschreiben« angezeigt, werden die eingetippten Zeichen immer links des Cursors sichtbar. Sofern bereits Text vorhanden ist, wird dieser durch die neue Eingabe überschrieben. Standardmäßig wird »Einfügen« vorgegeben.

Ist »Überschreiben« ausgeblendet, wird ab Cursorposition neuer Text eingefügt, d.h. bestehender Text wird nach rechts und, wenn die Zeile voll ist, in die nächste Zeile verschoben. Anstelle von »Überschreiben« wird dann in der Statuszeile der auf Cursorposition aktuelle Schrifttyp angezeigt. Beachten Sie bitte, daß aufgrund der Eingabeart die Funktionstaste [Rückschritt] (= Backspace) unterschiedlich arbeitet:

Überschreiben	Das Zeichen links vor dem Cursor wird gelöscht. Der rechts des Cursors angezeigte Text bleibt unverändert bestehen.
Einfügen	Das Zeichen links vor dem Cursor wird gelöscht. Der rechts des Cursors angezeigte Text wird beigerückt.

Beim Drücken von Einfg wird in jedem Fall die Eingabeart in Abhängigkeit der bisherigen Eingabeart gewechselt. Nach dem Drücken von Einfg wird darum

Einfügen in Überschreiben oder

Überschreiben in Einfügen

umgesetzt.

Wenn Sie nun die Funktion der Einfügen-Taste in einem Makro verwenden, können Komplikationen auftreten, da beim Drücken von Einfg die Eingabeart immer gewechselt wird. Hier kann dann z.B. fälschlicherweise auf »Einfügen« geschaltet werden, obwohl »Überschreiben« erforderlich ist. Mit dem hier beschriebenen Befehl können Sie darum einen bestimmten Eingabemodus erzwingen, wobei dieser nicht vom vorherigen Eingabemodus abhängig ist.

Parameter

Status: Geben Sie hier an, ob Sie *Überschreiben* ein- oder ausschalten wollen.

On! Überschreiben ist ausgeschaltet.

Off! Überschreiben ist eingeschaltet.