

7

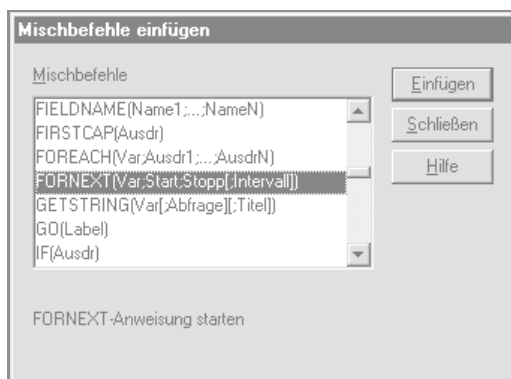
Mischbefehle

In Mischdateien können sog. Mischbefehle integriert werden, um umfangreiche Mischvorgänge automatisch ablaufen zu lassen, zu vereinfachen bzw. komfortabler zu gestalten. Die Handhabung und die Syntax dieser Mischbefehle sind mit den Makro-Programmierbefehlen weitestgehend identisch. Mischbefehle werden über die Funktion [*Tools, Mischen*] in eine Formular- und/oder in eine Datendatei eingefügt. Sind Variable erforderlich, müssen diese unmittelbar dem Befehl folgen. Sie müssen durch Klammern eingeschlossen sein. Eine Referenzliste der Mischbefehle finden Sie in Kapitel »14 Makro- und Mischbefehle«.

Halten Sie beim Einfügen von Mischbefehlen folgende Reihenfolge ein:

- ▲ Öffnen Sie die betreffende Daten- oder Formulardatei.
- ▲ Positionieren Sie den Cursor an der Stelle, an der ein Mischbefehl eingefügt werden soll.
- ▲ Blenden Sie die Mischen-Funktionsleiste ein, sofern Sie nicht angezeigt wird.
- ▲ Klicken Sie auf [Mischbefehle].
- ▲ Wählen Sie den gewünschten Mischbefehl aus.
- ▲ Ergänzen Sie ggf. die geforderten Variablen für den Mischbefehl.

Die beiden oft benötigten Befehle *Date* und *Keyboard* können Sie direkt aus der Funktionsleiste auswählen. Für die Anzeige aller weiteren Mischbefehle müssen Sie auf [*Mischbefehle*] klicken.



In dem eingeblendeten Dialogfeld werden die verfügbaren Mischbefehle in alphabetischer Reihenfolge aufgelistet. Positionieren Sie den Cursor mit den bekannten Möglichkeiten auf dem gewünschten Befehl. Klicken Sie zum Einfügen des Befehls auf [*Einfügen*], oder doppelklicken Sie auf dem Befehl. Befehle, die keine Parameter erfordern wie z.B. *Date*, *Break*, *Continue* usw., werden sofort auf Cursorposition eingefügt.

Für alle anderen Befehle wird ein Dialogfeld eingeblendet, in dem Sie die erforderlichen Parameter in den dafür vorgesehenen Textfeldern ergänzen müssen (siehe nachfolgende Beschreibung der einzelnen Befehle und deren Beispiele). Möchten Sie z.B. den Befehl *ForNext* einfügen, wird das abgebildete Dialogfeld angezeigt. Klicken Sie nach dem Ergänzen der Parameter zum Einfügen des Befehls auf [OK]. Das Dialogfeld der Mischbefehle bleibt für weitere Auswahlen geöffnet. Erst nach dem Klicken auf [*Schließen*] wird das Dialogfeld ausgeblendet.

Parameter, die in eckigen Klammern [] angezeigt werden, sind wahlfrei, d. h. sie können bei Bedarf auch weggelassen werden.

Versehentlich eingefügte Mischbefehle können Sie jederzeit im Dokument mit den üblichen Methoden löschen. Werden falsche Parameter eingegeben oder Parameter an falschen Positionen eingefügt, erfolgt beim Mischen eine Fehlermeldung. Sollte das Mischen überhaupt nicht anlaufen, müssen Sie alle Mischbefehle in den Daten- und Formulardateien überprüfen.

Achtung: Ab dieser Version werden die Mischbefehle in der Auswahlliste nur noch in englischer Sprache angezeigt. Befehle, die bisher in deutscher Sprache eingefügt wurden, werden weiterhin in deutsch eingefügt. Mischdateien mit deutschen Mischbefehlen sind lauffähig. Für Anwender früherer Versionen werden bei den nachfolgenden Beschreibungen zusätzlich die deutschen Befehle im rechten Bereich der Abschnitts-Überschriften angezeigt.

In den nachfolgenden Abschnitten werden nur die reinen Mischbefehle beschrieben. Die Beschreibung der Mischbefehle, die mit Makro-Programmierbefehlen identisch sind, finden Sie in Kapitel »6 Makro-Programmierbefehle«.

Beachten Sie bitte folgende Hinweise bei der Verwendung von Variablen, Parametern und Ausdrücken:

| Parameter | Bedeutung |
|---------------|---|
| Abfrage | Bedienungs-/ Fehlerhinweis, der in einem Dialogfeld angezeigt wird. |
| Ausdruck | Ausdruck, Befehl, String, Variable, Zahl oder eine Kombination aus diesen. |
| Dateiname | Eindeutiger Dateiname, evtl. mit Laufwerk und Pfad. |
| Feld | Eindeutiger Feldname oder Feldnummer einer Datendatei. |
| Label | Eindeutiger Name eines Labels. |
| Kommentar | Beschreibung oder Hinweise zu den Makro- oder Mischbefehlen (Dokumentation von Makros oder Mischvorgängen). |
| Makroname | Eindeutiger Makroname, evtl. mit Laufwerk und Pfad. |
| Meldung | Bedienungs-/ Fehlerhinweis, der in einem Dialogfeld angezeigt wird. |
| Titel | Titel eines Dialogfeldes. |
| Unterausdruck | Ausdruck, Befehl, String, Variable, Zahl oder eine Kombination aus diesen. |
| Variable | Eindeutiger Variablenname. |

Für die zuvor genannten Parameter dürfen keine reservierten Wörter (Mischbefehle oder Makro-Programmierbefehle) verwendet werden!

Da die Mischbefehle nicht zeichenweise eingegeben oder bearbeitet werden können wie Makro-Programmierbefehle, müssen Sie folgende Regeln beachten:

- ▲ Die Befehle können nicht von Hand eingegeben werden. Sie sind generell über die Mischen-Funktionsleiste auszuwählen.
- ▲ Wird ein Befehl oder eine zugehörige Klammer gelöscht (auch die Klammern werden als Mischbefehle eingefügt!), wird die komplette zugehörige Befehlsstruktur ebenfalls gelöscht. Nur die von Hand eingegebenen Variablen bleiben erhalten.

Beispiel

1. **Char** (*Abfrage*;Bitte »j« oder »n« eingeben;*Adressenverwaltung*)

Beim Löschen des Befehls oder einer der beiden Klammern wird der nicht kursiv gedruckte Befehlsteil gelöscht.

2. **If**(**Variable** (*Abfrage*)=n)

Diese Befehlsstruktur besteht aus zwei geschachtelten Befehlen (*If* und *Variable*). Die Löschung ist davon abhängig, welcher Befehl bzw. welche Klammer gelöscht wird. Beim Löschen des If-Befehls oder der letzten Klammer, die den IF-Befehl beendet, bleibt *Variable*(*Abfrage*)=*j* erhalten. Beim Löschen der zweitletzten Klammer oder des Variable-Befehls bleibt *If*(*Abfrage*=*j*) erhalten. Versehentliche Löschungen können Sie über [*Bearbeiten, Rückgängig machen*] wieder zurückholen.

3. If(Variable(Abfrage)=n)

Hier handelt es sich um einen geschachtelten Befehl, der aus den Befehlen *If* und *Variable* zusammengesetzt ist. Solche Befehle können über die Mischen-Funktionsleiste nicht in der angegebenen Reihenfolge in einem Arbeitsschritt eingegeben werden. Halten Sie bei geschachtelten Befehlen sinngemäß folgende Reihenfolge ein (bezogen auf das gezeigte Beispiel):

1. Nach der Auswahl des *If*-Befehls aus der Mischen-Funktionsleiste wird ein Dialogfeld zur Eingabe eines Ausdrucks angezeigt. Hier müßte jetzt der Befehl *Variable* eingefügt werden, was aber nicht möglich ist. Klicken Sie auf [OK], um den Befehl in das Dokument einzufügen. Er wird als **If()** übernommen.
2. Plazieren Sie den Cursor zwischen den beiden Klammern.
3. Wählen Sie aus der Mischen-Funktionsleiste den Befehl *Variable* aus. Geben Sie in dem angezeigten Dialogfeld den Namen der Variablen (hier: *Abfrage*) ein. Klicken Sie auf [OK]. Der Befehl im Dokument wird jetzt auf **If(Variable(Abfrage))** erweitert.
4. Plazieren Sie den Cursor zwischen den beiden letzten Klammern.
5. Geben Sie die Bedingung ein (hier: = j).

- ▲ Befehle und Parameter können zur besseren Übersicht nicht so einfach getrennt werden, wie das bei den Makro-Programmierbefehlen möglich ist, da das Befehlswort und die zugehörige Klammer-Auf einen Befehl darstellen. Eine Formatierung könnte daher anstelle von

AssignLocal(Multiplikand;5)

AssignLocal(Ergebnis;**Variable**(Multiplikand)*5)

Variable(Ergebnis)

wie folgt vorgenommen werden:

AssignLocal Multiplikand;5)

AssignLocal(Ergebnis;**Variable**(Multiplikand)*5)

Variable(Ergebnis)

7.1 Assign / AssignLocal

| | |
|---------------------|--|
| Befehlsform: | Assign (<i>Variable</i> ; <i>Ausdruck</i>) |
| | AssignLocal (<i>Variable</i> ; <i>Ausdruck</i>) |

Der in *Ausdruck* genannte Wert wird der lokalen Variablen *Variable* zugewiesen. *Ausdruck* darf nicht in Hochkommata eingeschlossen werden. Lokale Variablen werden nach dem Ende des Mischvorgangs bzw. beim Verlassen der Datei gelöscht. Label und Variablen dürfen nicht denselben Namen aufweisen.

Assign Über diesen Befehl wird eine *globale* Variable definiert. Die definierten Variablen stehen auch nach Beendigung eines Mischvorgangs weiterhin zur Verfügung.

AssignLocal Über diesen Befehl wird eine *lokale* Variable definiert. Die definierten Variablen stehen nur für den aktuellen Mischvorgang zur Verfügung und werden nach dessen Beendigung gelöscht.

Parameter

- Variable:** Beliebiger Variablenname. Die Verwendung eines reservierten Wortes (siehe Kapitel »5.11 Reservierte Wörter«) ist hier nicht erlaubt. Der Variablenname darf nicht in Hochkommata eingeschlossen sein. Es wird keine Unterscheidung zwischen Groß- und Kleinbuchstaben getroffen. Die Variablennamen *Eingabe*, *eingabe* oder *EiNgAbE* sind darum identisch.
- Ausdruck:** Angabe einer Variablen, fixer Text, numerische Werte oder eine Kombination aus diesen Angaben. Der Ausdruck darf nicht in Hochkommata eingeschlossen sein, da diese als Teil des Strings behandelt werden.

Beispiel

1. **AssignLocal**(Multiplikand;5)

AssignLocal(Ergebnis;**Variable**(Multiplikand)*5)

Variable(Ergebnis)

Der Variablen *Multiplikand* wird der Wert 5 zugewiesen. In der zweiten Anweisung wird der Inhalt der Variablen *Multiplikand* mit der numerischen Konstanten 5 multipliziert und in *Ergebnis* gespeichert. Über die letzte Anweisung wird das ermittelte Ergebnis im Dokument auf Cursorposition eingefügt.

2. **AssignLocal**(Drucktext;Das ist ein Test)

Der Variablen *Drucktext* wird der String *Das ist ein Test* zugewiesen. Der String darf nicht in Hochkommata eingeschlossen sein.

3. **AssignLocal**(Drucktext;)

Der Inhalt der Variablen sowie die Variable selbst werden gelöscht, d.h. die Variable ist nicht mehr verfügbar. Wird diese Variable mit dem Befehl *IfExists* geprüft, gilt sie als nicht vorhanden (Bedingung trifft nicht zu).

7.2 Caps

Befehlsform: **Caps**(*Ausdruck*)

Der erste Buchstabe jedes Wortes des Parameters *Ausdruck* wird in einen Großbuchstaben umgesetzt. Dieser Befehl kann nicht für sich alleine ausgeführt werden. Er tritt immer in Verbindung mit anderen Befehlen wie z. B. *Assign*, *If* usw. auf (siehe Kapitel »5.17 Änderungen gegenüber WPWin 5.1/5.2«).

Parameter

Ausdruck: Variable, die den Text enthält, dessen erster Buchstabe jeden Wortes in einen Großbuchstaben umgesetzt werden soll.

Beispiel

```
GetString(Text;Bitte Text eingeben;Test des Befehls CAPS)
Assign(Text;Caps(Variable(Text)))
```

Über den Befehl *GetString* wird ein beliebiger Text in die Variable *Text* eingegeben. Der erste Buchstabe jedes Wortes wird in einen Großbuchstaben umgewandelt.

Weitere Hinweise siehe: *FirstCap*, *ToLower*, *ToUpper*

7.3 ChainData

(VerkettenDaten)

Befehlsform: **ChainData**(*Dateiname*)

Die Verarbeitung wird mit der angegebenen Datendatei fortgeführt, wenn die bisherige Datendatei komplett abgearbeitet wurde. Verwenden Sie diesen Befehl, um eine große Datendatei für bestimmte Arbeitsschritte in mehrere kleine Dateien zu splitten. In jeder der kleinen Datendateien können Sie dann den Verkettungsbefehl einfügen, um die nächste Datei zuzuordnen.

Parameter

Dateiname: Dateiname der zu verarbeitenden Datendatei.

Beispiel: Siehe unter **ProcessOn** / **ProcessOff**

Weitere Hinweise siehe: *ProcessOn* / *ProcessOff*

7.4 ChainForm

(VerkettetenFormular)

Befehlsform: ChainForm(*Dateiname*)

Die Verarbeitung wird mit der angegebenen Formulardatei fortgeführt, wenn die bisherige Formulardatei komplett abgearbeitet wurde.

Parameter

Dateiname: Dateiname der zu verarbeitenden Formulardatei.

Beispiel:

1 ChainForm(VERTRAG.FRM)

Das Mischen wird mit der Formulardatei *VERTRAG.FRM* fortgesetzt.

2 ChainForm(Keyboard(Bitte neue Formulardatei eingeben))

Der Dateiname der Formulardatei wird beim Antreffen des Befehls *Chain-Form()* über die Tastatur eingegeben.

7.5 ChainMacro

(VerkettetenMacro)

Befehlsform: ChainMacro(*Makroname*)

Mit diesem Mischbefehl wird nach Beendigung eines Mischvorgangs (nicht bei Abbruch) automatisch das angegebene Makro gestartet. Dies ist dann sinnvoll, wenn Sie z.B. mehrere Textbausteine mischen und anschließend sofort einen bestimmten Drucker auswählen und den Text drucken möchten. Dieser Mischbefehl kann an einer beliebigen Stelle innerhalb der Formulardatei enthalten sein. Sind mehrere dieser Befehle definiert, wird nur der **letzte** ausgeführt.

Parameter

Makroname: Dateiname des auszuführenden Makros.

Beispiel


```
Document(Absender)
Field(Anrede)*
Field(Vorname) Field(Nachname)
Field(Straße)
Field(Wohnort)*
Sehr geehrteField(Anrede) Field(Nachname)*
Document(Einleit)
```

```
Document(Absatz1)
Document(Vertrag)
ChainMacro(Statistik)
```

Der Abstand zwischen den mit * gekennzeichneten Zeilen wird durch den Befehl [*Format, Drucksatz, Textposition*] erst beim Ausdrucken erzeugt. Nach dem Einfügen der Datenfelder werden die Textbausteine ergänzt. Anschließend wird durch das Makro *Statistik* eine vordefinierte Statistik geladen und entsprechend aufbereitet.

7.6 Char

Befehlsform: `Char(Variable[:Abfrage;Titel])`

In einem Dialogfeld wird der unter *Abfrage* angegebene Text angezeigt, z.B. ein Bedienungshinweis. Der Benutzer kann *ein* Zeichen eintippen, das in der zugeordneten Variablen gespeichert wird. Das Drücken von  ist nach dem Eintippen des Zeichens nicht erforderlich. Nach dem Eintippen des Zeichens wird das Dialogfeld wieder ausgeblendet. Verwenden Sie für die Eingabe *mehrerer* Zeichen den Befehl *GetString*.

Parameter

Variable: Das eingetippte Zeichen wird in dieser Variablen gespeichert.
Abfrage: Text, der über dem Eingabefeld angezeigt werden soll.
Titel: Titel des Dialogfelds.

Beispiel

`CHAR(JaNein;Verarbeitung beenden? (j oder n)?;Adressen erfassen)`

Abfrage, ob eine Verarbeitung beendet werden soll oder nicht. Das eingetippte Zeichen wird in der Variablen *JaNein* gespeichert. Nach der Eingabe muß geprüft werden, welches Zeichen eingetippt wurde, um das Mischen dementsprechend zu steuern.


Weitere Hinweise siehe: *GetString*


7.7 Codes

(Befehle)


Befehlsform: `Codes(Mischbefehle)`

Beim Eingeben von Mischbefehlen wird normalerweise das Format eines Dokuments geändert (z.B. durch Einfügen von Zeilenschaltungen am Zeilenende der Mischbefehle). Um das Format beizubehalten, müssen die Mischbefehle in Verbindung mit dem Befehl *Codes* eingegeben werden. Hierbei wird die Formatierung des Dokuments beibehalten, auch dann, wenn

Sie innerhalb dieses Bereichs  drücken. Beim Mischen werden dann die innerhalb der Klammern stehenden Befehle ausgeführt, wobei Text oder Formatierungen nicht berücksichtigt werden. Sollten trotzdem Text oder Formatierungen erforderlich sein, können diese über den Befehl *Einfügen* eingefügt werden. Sollten mehrere Befehle eingefügt werden, achten Sie bitte darauf, daß die letzte Klammer nicht vergessen wird. Fügen Sie darum den Befehl wie folgt ein:

- ▲ Wählen Sie den Befehl aus der Mischen-Funktionsleiste aus.
- ▲ Plazieren Sie nach dem Einfügen den Cursor zwischen den beiden Klammern, und drücken Sie für einen größeren Zwischenraum (zum Einfügen der erforderlichen Befehle) mehrmals .
- ▲ Plazieren Sie den Cursor zwischen den beiden Klammern am Beginn einer Leerzeile, und fügen Sie die erforderlichen Befehle ein (siehe Beispiel). Reicht der Platz nicht aus, fügen Sie weitere Leerzeilen ein.

Muß dieser Befehl einmal nachträglich eingefügt werden, um bereits vorhandene Befehle einzuschließen, verfahren Sie bitte wie folgt:

- ▲ Wählen Sie den Befehl aus der Mischen-Funktionsleiste aus.
- ▲ Plazieren Sie nach dem Einfügen den Cursor zwischen den beiden Klammern, und drücken Sie für einen größeren Zwischenraum (zum Einfügen der erforderlichen Befehle) mehrmals .
- ▲ Markieren Sie die betreffenden Befehle als Block, klicken Sie mit der rechten Maustaste, und wählen Sie [*Ausschneiden*].
- ▲ Plazieren Sie den Cursor zwischen den beiden Klammern von *Codes()* am Beginn einer Leerzeile. Klicken Sie mit der rechten Maustaste, und wählen Sie [*Einfügen*].

Parameter

Mischbefehle: Alle verfügbaren Mischbefehle können verwendet werden.

Beispiel

```

1      Codes(Comment: Beachten Sie die Klammer.)
2      Label(Anfang)
3      AssignLocal(Zeichen;0)
4      Char(Zeichen;Bitte ein Zeichen eingeben;Testen des Befehls CToN)
5      AssignLocal(Satznummer;CToN(Variable(Zeichen)) / 256)
6      AssignLocal(Zeichennummer;Cton(Variable(Zeichen)) % 256)
7      Insert(Zeichensatz:)
8      Variable(Satznummer)
9      Insert(, Zeichennummer:)
10     Variable(Zeichennummer)
11     Insert(Comment: Einfügen einer Zeilenschaltung.)
11     Insert(Comment: Einfügen einer Zeilenschaltung.)
12     Rewrite
13     Go(Anfang)
14     )Comment: Beachten Sie die Klammer.
```

In dem gezeigten Beispiel soll für ein eingegebenes Zeichen dessen Zeichensatz- und Zeichennummer ermittelt werden.


- 1 Mit den Befehlen innerhalb des Mischbefehls *Codes()* kann durch die Eingabe eines Zeichens dessen Zeichensatz- und Zeichennummer ermittelt werden. Die schließende Klammer für diesen Befehl ist die in Zeilennummer 14.
- 2 Beginn einer Schleife (wird so lange wiederholt, bis in dem Dialogfeld auf [Abbrechen] geklickt wird).
- 3 Der Inhalt der Variablen *Zeichen* wird auf 0 gesetzt.
- 4 Anzeige einer Dialogbox, Übernahme des eingetippten Zeichens in die Variable *Zeichen*.
- 5–6 Ermitteln der Zeichensatz- und Zeichennummer (siehe Befehl *CToN*).
- 7 Schreiben des Textes »*Zeichensatz:* ».
- 8 Schreiben der Variablen *Satznummer*.
- 9 Schreiben des Textes », *Zeichennummer:* ».
- 10 Schreiben der Variablen *Zeichennummer*.
- 11 Einfügen von zwei Zeilenschaltungen (siehe Mischbefehl *Insert*).
- 12 Anzeigen des bisher in das Dokument eingefügten Textes.
- 13 Rücksprung zum Schleifenanfang.
- 14 Ende des Mischbefehls *Codes* (Klammer ist wichtig und darf nicht vergessen werden!).

Weitere Hinweise siehe: Insert, NToC

7.8 Comment

(Kommentar)


Befehlsform: **Comment(Text)**

Text, der innerhalb der beiden Klammern steht, hat keinen Einfluß auf den Makroablauf. Dieser Text kann mehrere Zeilen lang sein. Nutzen Sie  oder [Einrücken], um evtl. Einrückungen vorzunehmen. Das Makro wird dadurch übersichtlicher. Makro- oder Mischbefehle, die sich innerhalb des Kommentars befinden, werden als Kommentar interpretiert, d.h. sie werden nicht ausgeführt. Kommentare dürfen sowohl in Daten- als auch in Formulardateien verwendet werden.

Nutzen Sie Kommentare zur

- ▲ Beschreibung komplizierter Mischvorgänge. Dies ist besonders dann wichtig, wenn die Mischdatei zu einem späteren Zeitpunkt durch eine andere Person als durch den Ersteller geändert werden muß. Die Einarbeitung in die Logik großer Mischdateien wird dadurch wesentlich erleichtert.

- ▲ vorübergehenden »Ausblendung« von Mischbefehlen oder Funktionen, um z.B. beim Testen eines Mischvorgangs bestimmte Mischroutinen vom Test auszuschließen (schnellerer Testdurchlauf).

Geben Sie die Kommentare am besten rechts der Befehle ein, die einer Erläuterung bedürfen. Bei einwandfrei verständlichen Befehlen können Sie Kommentare auch weglassen. Dadurch wird die Mischdatei übersichtlicher. Rücken Sie Kommentare durch Drücken von  ein.

Parameter

Text: Beliebiger Text.

7.9 CToN

Befehlsform: CToN(*Zeichen*)

Konvertieren des angegebenen Zeichens (Character **To** Number) in einen numerischen Wert. Mit Hilfe dieses Befehls können Sie den Zeichensatz und die Zeichennummer des angegebenen Zeichens innerhalb des betreffenden WordPerfect-Zeichensatzes ermitteln. WordPerfect bietet Ihnen 15 verschiedene Zeichensätze an. Der errechnete Wert besteht aus einer Dezimalzahl. Wenn Sie z.B. die Taste **a** drücken, wird der Wert 0,97 angezeigt, d.h. das kleine **a** ist das Zeichen mit der Nummer 97 im Zeichensatz 0.

Beispiel

```
1 AssignLocal(Zeichen;0)
2 Char(Zeichen;Bitte ein Zeichen eingeben;Testen des Befehls CToN)
3 AssignLocal(Satznummer;CToN(Variable(Zeichen)) / 256)
4 AssignLocal(Zeichennummer;CToN(Variable(Zeichen)) % 256)
5 Insert(Zeichensatz: )
6 Variable(Satznummer)
7 Insert(, Zeichennummer: )
8 Variable(Zeichennummer)
9 Rewrite
```

In dem gezeigten Beispiel soll für ein eingegebenes Zeichen dessen Zeichensatz- und Zeichennummer ermittelt werden.

- 1 Der Inhalt der Variablen *Zeichen* wird auf 0 gesetzt.
- 2 Anzeige einer Dialogbox, Übernahme des eingetippten Zeichens in die Variable *Zeichen*.
- 3 Ermitteln der Zeichensatznummer.
- 4 Ermitteln der Zeichennummer innerhalb eines Zeichensatzes.
- 5 Schreiben des Textes »Zeichensatz: ».
- 6 Schreiben des Inhalts der Variablen *Satznummer*.
- 7 Schreiben des Textes », Zeichennummer: ».

- 8 Schreiben des Inhalts der Variablen *Zeichennummer*.
- 9 Anzeigen des bisher in das Dokument eingefügten Textes. Beim Weglassen dieses Befehls wird der Text erst nach dem Beenden des Mischens angezeigt.

Sie finden ein Beispiel für diesen Befehl auf der CD-ROM unter dem Dateinamen CTON.FRM.

Weitere Hinweise siehe: NToC

7.10 Date

(Datum)

Befehlsform: Date

Anstelle dieses Mischbefehls wird während des Mischens das aktuelle Tagesdatum in dem vordefinierten Format (z.B. in Verbindung mit der Uhrzeit oder konstanten Texten) eingefügt. Achten Sie darauf, daß noch genügend Platz zum folgenden Datenfeld bzw. bis zum Zeilenende vorhanden ist, damit das Datum komplett übernommen werden kann, ohne Textverschiebungen zu verursachen. Dies sollten Sie besonders dann berücksichtigen, wenn Sie den Monatsnamen in Worten ausgewählt haben, denn für »September« wird wesentlich mehr Platz benötigt, als für »Mai«. Das aktuelle Datum wird natürlich nur dann korrekt übernommen, wenn beim Starten des Computers Datum und Uhrzeit ordnungsgemäß eingegeben bzw. überprüft wurden.

7.11 DisplayStop

Befehlsform: DisplayStop

Die Textanzeige wird ausgeblendet, das Mischen läuft im Hintergrund ab. Dies ist beim Mischen großer Datenbestände sinnvoll, um eine schnellere Verarbeitungszeit zu erreichen.

Weitere Hinweise siehe: Rewrite

7.12 Document

(Baustein)

Befehlsform: Document(Dateiname)

Der angegebene Textbaustein (WordPerfect-Dokument) wird an der Stelle eingefügt, an der dieser Befehl angegeben wurde. Der Baustein wird nur eingefügt, evtl. darin enthaltene Mischbefehle werden angezeigt, aber nicht ausgeführt.

Parameter

Dateiname: Name des aufzurufenden Dokuments, ggf. unter Angabe des Laufwerksbuchstabens und Ordnersnamens. Die Verwendung eines Variablennamens ist nicht erlaubt.

Beispiel

Document (C:\Einkauf\Vertrag\paragr-1)

7.13 EmbedMacro

(MakroEinbetten)

Befehlsform: EmbedMacro(Makro-Programmierbefehle)

Über diesen Befehl können Makro-Programmierbefehle in eine Mischdatei eingefügt werden. Beim erstmaligen Ausführen dieser Befehle werden sie kompiliert. Die Makro-Programmierbefehle müssen zwischen den beiden Klammern eingegeben werden.

Parameter

Makro-Programmierbefehle:

Beliebige Makro-Programmierbefehle.

Beispiel:

Die Handhabung dieses Befehls entspricht dem Mischbefehl *Codes()*. Entnehmen Sie darum bitte weitere Informationen der Beschreibung dieses Befehls (siehe oben).

Weitere Hinweise siehe: Codes

7.14 EndField

(EndeFeld)

Befehlsform: EndField

Dieser Mischbefehl beendet jedes Feld (= Feldende) einer Datendatei, wobei dieses Feld mehrere Zeilen lang sein kann. Zwischen dem Mischbefehl *EndField* und der unmittelbar nachfolgenden Zeilenschaltung [FNZ] dürfen keine Leerstellen vorhanden sein, sonst werden beim Mischen nach dem jeweiligen Feld zusätzliche Leerzeilen eingefügt. Adressen können Sie auch unformatiert eingeben, wobei Sie selbst auf dieselbe Anzahl Datenfelder je Datensatz achten müssen. In diesem Fall erhalten die Felder keine Feldnamen, sondern Sie werden fortlaufend, bei 1 beginnend, durchnummeriert. Die *EndField*- und *EndRecord*-Steuerzeichen müssen Sie hierbei ebenfalls selbst einfügen. Besser ist jedoch die Eingabe bei der Definition einer Datendatei über die Funktion *Blitzeintrag*. Hier werden *EndField* und *EndRecord* von WordPerfect automatisch gesetzt.

Beispiel

Herrn**EndField**

Willibald Hinterhuber**EndField**

Alpenrosenweg 27**EndField**

D 68161 Mannheim**EndRecord**

=====

Frau**EndField**

Eugenie Müller**EndField**

Rheinstraße 13**EndField**

D 64273 Darmstadt**EndRecord**

=====

Weitere Hinweise siehe: **EndRecord**, **Field**, **FieldName**

7.15 EndRecord

(EndeDatensatz)

Befehlsform: **EndRecord**

Jeder Datensatz einer Datendatei muß mit *EndRecord* abgeschlossen sein. Zusätzlich wird hinter diesem Befehl ein fester Seitenumbruch eingefügt, um die einzelnen Datensätze optisch besser abzugrenzen. Wenn *EndRecord* beim Mischen in einer Formulardatei auftritt, wird das Mischen beendet.

Beispiel: Siehe **EndField**.

Weitere Hinweise siehe: **EndField**, **Field**, **FieldName**

7.16 Field

(Feld)

Befehlsform: **Field(Feldname/-nummer)**

Wenn Sie während des Mischens Daten aus einer Datendatei übernehmen möchten, müssen Sie an den betreffenden Stellen Ihres Dokuments die Nummer oder den Namen des Feldes angeben, dessen Inhalt hier gedruckt werden soll. Vor dem Einfügen dieses Befehls sollten Sie alle Tabulatoren löschen und nur die setzen, die unbedingt erforderlich sind. Dies ist dann besonders wichtig, wenn Sie in einer Zeile mehrere Felder übernehmen möchten. Springen Sie

die Textstelle, an der das einzufügende Feld beginnt, mit der Tabulatortaste an. Ebenso sollten Sie weitere Felder innerhalb derselben Zeile sowie den Beginn fester Texte ebenfalls mit der Tabulatortaste anspringen.

Beispiel

Ihr Brief vom **Field(1)** und unsere Rechnung vom **Field(2)** haben

Tabulatoren

Feste Texte behalten dann ihre Position bei, wenn die einzufügenden Datenfelder nicht mehr Platz beanspruchen, als vorgesehen ist. Wenn Sie keine Tabulatoren benutzen, erfolgt eine Textverschiebung, die aber auch vom Benutzer mit voller Absicht gewollt sein kann!

Sollen mehrere Felder innerhalb einer Zeile direkt aneinander anschließen, müssen Sie, wenn erforderlich, auf den Zwischenraum zwischen den Mischbefehlen achten, sonst »kleben« die Felder direkt aneinander. Sind keine Tabulatoren gesetzt, bleiben zwischen den eingefügten Daten so viele Leerstellen frei, wie auch vor und nach den Mischbefehlen angegeben wurden:

Parameter

Feldname/-nummer:

Wurde mit Feldnamen gearbeitet, ist der jeweilige Feldname anzugeben. Sind keine Feldnamen vorhanden, ist die Feldnummer anzugeben. Hierbei werden alle Felder eines Datensatzes, beim ersten Feld beginnend, durchnummeriert.

Beispiel

Field (Anrede)

Field (Name)

Field (Straße)

Field (PlzOrt)

Weitere Hinweise siehe: EndField, EndRecord, FieldName

7.17 FieldName

(FeldNamen)

Befehlsform: **FieldName**(Name1;Name2;...)

Festlegung der Namen und der Reihenfolge von Datenfeldern in einer Datendatei. Felder einer Formulardatei können unter dem angegebenen Namen angesprochen werden, wenn sie zuvor in der Datendatei definiert wurden. Dieser Befehl muß in Datendateien an erster Stelle vor den Datensätzen stehen. Sind nach der Festlegung der Feldnamen und dem Erfassen von Daten mehr Felder vorhanden, als hier definiert wurden, werden zusätzliche Felder automatisch durchnummeriert.

Parameter

Namen: Namen der definierten Felder in der angegebenen Reihenfolge. Diese Reihenfolge wurde beim Definieren festgelegt und darf nachträglich nicht manuell geändert werden, da keine Änderung der Datensätze erfolgt! Sind Änderungen erforderlich, nehmen Sie diese über [*Blitzeintrag, Feldnamen*] vor. Die Namen können nach der Erstellung zur besseren Übersicht durch Einfügen von Zeilenschaltungen zeilenweise dargestellt werden.

Beispiel

FieldName (Anrede;Vorname;Nachname;Titel;Satzdefinition
Straße;Plz;Wohnort;Land)**EndRecord**

```
=====
HerrnEndField           Anrede
WillibaldEndField       Vorname
HinterhuberEndField     Nachname
ProfessorEndField       Titel
Alpenrosenweg 27EndField Straße
D 80335EndField         Plz
MünchenEndField         Wohnort
GermanyEndRecord       Land
=====
```

Weitere Hinweise siehe: Field, EndField, EndRecord

7.18 FirstCap

Befehlsform: FirstCap(*Ausdruck*)

Der erste Buchstabe des Parameters *Ausdruck* wird in einen Großbuchstaben umgesetzt. Dieser Befehl kann nicht für sich alleine ausgeführt werden (siehe Kapitel »5.17 Änderungen gegenüber WPWin 5.1/5.2«).

Parameter

Ausdruck: Variable die den Text enthält, dessen erster Buchstabe in einen Großbuchstaben umgesetzt werden soll.

Beispiel

```
GetString(Text;Bitte Text eingeben;Test des Befehls CAPS)
If(FirstCap(Variable(Text)) = Ende) Quit EndIf
Variable(Text)
```


Über den Befehl **GetString** wird ein beliebiger Text in die Variable **Text** eingegeben. Enthält die Variable **Text** den Inhalt **ende** oder **Ende**, wird das Mischen beendet, ansonsten wird der Inhalt der Variablen **Text** in das Dokument eingefügt.

Weitere Hinweise siehe: FirstCap, ToLower, ToUpper

7.19 IfBlank

Befehlsform: **IfBlank(Feldname/-nummer)**

Wenn das betreffende Feld keinen Inhalt hat, werden die Befehle ausgeführt, die auf *IfBlank* folgen. Hinter dem letzten auszuführenden Befehl muß ein *EndIf*-Befehl folgen! Die Verwendung von *Else* ist erlaubt. Beachten Sie bitte auch den Befehl *IfNotBlank*.

Parameter

Feldname/-nummer:

Wurde mit Feldnamen gearbeitet, ist der jeweilige Feldname anzugeben. Sind keine Feldnamen vorhanden, muß die Feldnummer angegeben werden. Hierbei werden alle Felder eines Datensatzes, beim ersten Feld mit 1 beginnend, durchnummeriert.

Dem nachfolgenden Beispiel liegt die folgende Datensatzdefinition zugrunde:

FieldName(Anrede;Vorname;Nachname;Titel;Straße;Plz;Wohnort;Land)EndRecord

```
=====
HerrnEndField           Anrede
WillibaldEndField       Vorname
HinterhuberEndField     Nachname
ProfessorEndField       Titel
Alpenrosenweg 27EndField Straße
D 80335EndField         Plz
MünchenEndField         Wohnort
GermanyEndRecord       Land
=====
```

Wenn mit *Feldnummern* gearbeitet werden soll, wäre hier Nachname Feld 3 und Wohnort Feld 7.

Die folgende Beschreibung bezieht sich auf das abgebildete Beispiel (siehe unten). Die Befehle in den gekennzeichneten Zeilen der Abbildung haben folgende Bedeutung:

1. Enthält das Feld *Titel* der Datendatei einen Inhalt (*IfNotBlank* trifft zu), wird dieser vor dem Vornamen gedruckt, ansonsten werden Vorname und Name am linken Rand gedruckt (ohne vorausgehende Leerstelle). Hierbei ist die Leerstelle vor dem *EndIf*-Befehl wichtig, da durch diese Titel und Name getrennt werden.

2. Enthält das Feld *Land* keine Daten (*IfBlank* trifft zu), wird *Deutschland* eingefügt.
3. Das Feld *Anrede* der Datendatei wird überprüft. Je nach Feldinhalt (Herrn oder Frau) wird die Anrede im Brief ergänzt:

Beispiel

Sehr geehrter Herr

Sehr geehrte Frau

Die Anreden wurden in diesem Beispiel auf *Herrn* und *Frau* beschränkt. Sie könnten natürlich noch auf *Fräulein*, *Firma* usw. erweitert werden. Der IF-Befehl wäre dann entsprechend abzuändern bzw. zu ergänzen.

Die restlichen Datenfelder werden wie üblich aus der Datendatei übernommen.

Computerladen Happy Bit47110 Nirgendwo, Date

Computerladen Happy Bit, Postfach 4711, D-68161 Nirgendwo

Field(Anrede)

IfNotBlank(Titel)Field(Titel) EndIfField(Vorname) Field(Nachname) 1.
Field(Straße)

Field(Plz) Field(Wohnort)

IfBlank(Land)Insert(Deutschland)EndIf 2.

Unser ZeichenIhr ZeichenIhr Brief vom

Sehr geehrteIf(Field(Anrede)=Herrn)Insert(r Herr) Else Insert(Frau)EndIfField(Name) 3.

Wir bedanken uns für Ihre Anfrage

Weitere Hinweise siehe: Insert, Else, Field, IfNotBlank

7.20 IfExists

Befehlsform: **IfExists(Variable)**

Dieser Befehl kontrolliert, ob die angegebene Variable einen Wert enthält bzw. ob sie definiert wurde. Ist ein Wert vorhanden, trifft die Bedingung zu, andernfalls nicht. Die Befehlsstruktur und die Programmierlogik entsprechen weitestgehend den Beschreibungen des *If*-Befehls. Auch hier muß das Ende der Struktur durch *EndIf* abgeschlossen werden. Die Verwendung von *Else* ist erlaubt. Eine Variable wird als vorhanden erkannt, wenn Sie zuvor definiert und ihr ein Wert zugewiesen wurde.

Assign(Name;) Diese Variable wurde zwar definiert, ihr wurde aber kein Wert zugewiesen. In diesem Fall trifft die *IfExists*-Bedingung nicht zu. Mit dieser Möglichkeit können Sie Variablen löschen.

Assign(Name;Meier) Diese Variablen wurde ein Wert zugewiesen, die Bedingung trifft zu.

Parameter

Variable Beliebige Variable.

Beispiel

```

1 Assign(DruckenText;)
2 Char(Abfrage;Bitte »j« oder »n« eingeben.;Befehl »IfExists« testen.)
3 If(Variable(Abfrage)=j)
   Assign(DruckenText;j)
EndIf
4 IfExists(DruckenText)
   Prompt(Variable ist vorhanden)
Else
   Prompt(Variable ist nicht vorhanden)
EndIf
5 Wait(30)
```

In diesem Beispiel wird in Abhängigkeit davon, ob die Variable *DruckenText* vorhanden ist oder nicht, einer der beiden Prompt-Befehle ausgeführt.

- 1 Löschen der Variablen, um sicherzustellen, daß sie nicht an anderer Stelle definiert wurde. Es darf kein Wert zugewiesen werden, auch keine Leerstelle!
- 2 Anzeige einer Meldung und Speichern des eingegebenen Zeichens in der Variablen *DruckenText*.
- 3 Wenn ein *j* eingegeben wurde, wird dieser Buchstabe der Variablen *DruckenText* zugewiesen.
- 4 Die Bedingung trifft nur dann zu, wenn der Assign-Befehl in Zeile 3 ausgeführt wurde.
- 5 Der Prompt-Befehl wird für 30/10 Sekunden am Bildschirm angezeigt.

Weitere Hinweise siehe: EndIf, If, Look

7.21 IfNotBlank

Befehlsform: **IfNotBlank**(Feldname/-nummer)

Wenn das betreffende Feld Daten enthält, werden die Befehle ausgeführt, die auf *IfNotBlank* folgen. Hinter dem letzten auszuführenden Befehl muß ein *EndIf*-Befehl folgen! Die Verwendung von *Else* ist erlaubt. Beachten Sie bitte auch den Befehl *IfNotBlank*.

Parameter

Feldname/-nummer: Wurde mit Feldnamen gearbeitet, ist der jeweilige Feldname anzugeben. Sind keine Feldnamen vorhanden, muß die Feldnummer angegeben werden. Hierbei werden alle Felder eines Datensatzes, beim ersten Feld mit 1 beginnend, durchnummeriert.

Beispiel: Siehe unter **IfBlank**.

Weitere Hinweise siehe: Insert, Else, Field, IfBlank

7.22 Insert

(Einfügen)

Befehlsform: **Insert**(Text)

Wenn Sie den Mischbefehl *Codes()* verwenden, können Sie über *Insert()* Texte oder Formatierungen in das Dokument einfügen (z.B. Zeilenschaltungen, Einrückungen, Tabstopps usw.).



Parameter

Text: Beliebiger Text oder Steuerfunktionen.

Beispiel

1. `Insert(Das ist ein Test)`
Der Text **Das ist ein Test** wird auf Cursorposition im Dokument eingefügt.
2. `Insert(Bitte folgende Vertragsbedingungen beachten)`
3. `Getstring(Text;Bitte Nachname eingeben;Adressen erfassen)`
`Insert(Variable(Text))`
Der über `Getstring` eingegebene Nachname wird über `Insert` auf Cursorposition im Dokument eingefügt.
4. `Insert([FNZ][FNZ] ist nur als Steuerzeichen sichtbar!`
`)`

Zum Einfügen von Zeilenschaltungen oder anderen Steuerfunktionen fügen Sie diesen Befehl ein, ohne in dem Dialogfeld einen Text einzugeben. Der Befehl wird im Dokument als *Insert()* eingefügt. Bringen Sie den Cursor zwischen die beiden Klammern, und wäh-

len Sie die gewünschte Funktion aus. In diesem Beispiel wurde zum Erzeugen einer Zeilenschaltung  gedrückt. Die schließende Klammer wird dadurch in die nächste Zeile verschoben. Wurde z.B. [Einrücken] oder  gewählt, würde die Klammern auf den nächsten Tabstop verschoben.



5. Einfügen des in Klammern angegebenen Textes.

Weitere Hinweise siehe: Codes

7.23 Keyboard

(Tastatur)

Befehlsform: **Keyboard**([Abfrage])

Dieser Befehl unterbricht das Mischen, um eine Eingabe von der Tastatur zu ermöglichen. Maximal 1.000 Zeichen werden angezeigt. Schließen Sie die Eingabe durch Drücken von  oder klicken auf [OK] ab. Der eingegebene Text wird in dem Dokument anstelle des Mischbefehls *Keyboard* eingefügt. Wird der Cursor in einem Feld positioniert, in dem momentan keine Eingabe erfolgen soll, drücken Sie  oder klicken Sie auf [OK]. Sie können das Mischen abbrechen, indem Sie in der angezeigten Funktionsleiste auf [Stopp] klicken.

Parameter

Abfrage: Beliebiger Text (max. 1.000 Zeichen). Der eingegebene Text wird in der Titelzeile des Dialogfelds angezeigt. Wird kein Text eingegeben, bleibt die Titelzeile leer.

Beispiel

Keyboard (Bitte Aktenzeichen eingeben.)

Weitere Hinweise siehe: Char, GetString

7.24 Look

Befehlsform: **Look**(Variable)

Mit diesem Befehl können Sie überprüfen, ob zuvor vom Benutzer ein Zeichen eingetippt wurde. Wenn ja, wird dieses Zeichen der angegebenen Variablen zugeordnet. Wurde kein Zeichen eingegeben, wird der bisherige Inhalt der Variablen gelöscht und das Mischen fortgesetzt. Eine Überprüfung durch *IfExists* verläuft daher negativ (Bedingung trifft nicht zu).

Parameter

Variable: Beliebige Variable.

Beispiel

```
1 Look(GedrückteTaste)
2 IfExists(Variable(GedrückteTaste))
   Prompt(Es wurde eine Taste gedrückt.)
   Else
   Prompt(Es wurde keine Taste gedrückt.)
   EnfiIf
```

In Zeile 1 wird über den Befehl *Look* geprüft, ob eine Taste gedrückt wurde. Wenn ja, wird die Taste der Variablen *GedrückteTaste* zugeordnet, ohne eine Ausführung zu veranlassen. Aufgrund des *IfExists*-Befehls kann jetzt gesteuert werden, welche Aktion ausgeführt werden soll (z.B. Mischen beenden, Wechseln einer Daten- oder Formulardatei usw.).

Weitere Hinweise siehe: If, Else, Field, IfExists

7.25 Mrgcmnd

(Mischbefehl)

Befehlsform: **Mrgcmnd**(Mischbefehle)

Der angegebene Mischbefehl wird in die erstellte Datei eingefügt. Dieser Befehl kann dann für weitere Mischvorgänge verwendet werden. Die Befehle werden lediglich eingefügt, aber nicht ausgeführt. Mit dem folgenden Beispiel (Ausschnitt einer Mischdatei) werden Adressen definiert, erfaßt und in die Datendatei ADRESSEN.DAT eingefügt:

```
Codes(
Mrgcmnd(FieldName(Anrede;Vorname;Nachname;Straße;Plz;Wohnort
)
EndRecord
:::::
Variable(Anrede)Mrgcmnd(EndField)
Variable(Vorname)Mrgcmnd(EndField)
Variable(Nachname)Mrgcmnd(EndField)
Variable(Straße)Mrgcmnd(EndField)
Variable(Plz)Mrgcmnd(EndField)
Variable(Wohnort)Mrgcmnd(EndRecord)
:::::
```

Damit die eingegebenen Adressen für späteres Mischen benutzt werden können, muß jedes Feld mit *EndField* und jeder Satz mit *EndRecord* enden. Diese beiden Befehle werden nach der Datenerfassung durch *Mrgcmnd* automatisch am Ende eines jeden Feldes bzw. jeden Satzes angehängt. Die Dateninhalte der einzelnen Variablen wurden in einem anderen Programmabschnitt eingegeben. Ein solcher Datensatz könnte nach der Eingabe wie folgt aussehen:

HerrnEndField
 KarlEndField
 MeierEndField
 Rheinstraße 15EndField
 65185EndField
 WiesbadenEndRecord

7.26 NestData

(VerschachtelnDaten)

Befehlsform: **NestData**(Dateiname)

Die angegebene Datendatei wird für die weitere Verarbeitung verwendet. Ist diese Datendatei komplett verarbeitet, wird wieder mit der vorherigen Datendatei weitergearbeitet. Datendateien können bis zu 10 Ebenen tief verschachtelt sein. Achten Sie darauf, daß Sie keine Endlosschleife erzeugen, indem Sie z.B. in einer verschachtelten Datei erneut die Datendatei verschachteln, von der aus die aktuelle Datendatei aufgerufen wurde. Fügen Sie sicherheits- halber einen *Cancel*-Befehl ein, um in solchen Fällen das Mischen abbrechen zu können.

Parameter

Dateiname: Dateiname der zu verschachtelnden Datendatei.

Beispiel

1 **NestData**(ADRESSEN2.DAT)

Das Mischen wird mit der Datendatei *ADRESSEN2.DAT* fortgesetzt.

2 **NestData**(**Keyboard**(Bitte neue Datendatei eingeben))

Der Dateiname der Datendatei wird beim Antreffen des Befehls *NestData()* aber die Tastatur eingegeben.

7.27 NestForm

(VerschachtelnFormular)

Befehlsform: **NestForm**(Dateiname)

Beim Auftreten dieses Befehls wird die Kontrolle an die angegebene Formulardatei übergeben. Ist diese Formulardatei komplett verarbeitet, geht die Ausführungskontrolle wieder an die aufrufende Formulardatei über. Hier wird mit dem Befehl fortgefahren, der dem Befehl *NestForm()* folgt. Formulardateien können bis zu 10 Ebenen tief verschachtelt sein. Achten Sie darauf, daß Sie keine Endlosschleife erzeugen, indem Sie z.B. in einer verschachtelten Datei

erneut die Formulardatei verschachteln, von der aus die aktuelle Formulardatei aufgerufen wurde. Fügen Sie sicherheitshalber einen *Cancel*-Befehl ein, um in solchen Fällen das Mischen abbrechen zu können.

Parameter

Dateiname: Dateiname der zu verschachtelnden Formulardatei.

Beispiel

1 **NestForm(VERTRAG.FRM)**

Das Mischen wird mit der Formulardatei *VERTRAG.FRM* fortgesetzt.

2 **ChainForm(Keyboard(Bitte neue Formulardatei eingeben))**

Der Dateiname der Formulardatei wird beim Antreffen des Befehls *ChainForm()* über die Tastatur eingegeben.

7.28 NestMacro

(VerschachtelnMakro)

Befehlsform: **NestMacro(Makroname)**

Beim Antreffen dieses Mischbefehls wird das angegebene Makro ausgeführt. Ist das Makro beendet, wird mit dem Befehl fortgefahren, der dem Befehl *NestMacro()* folgt.

Parameter

Makroname: Dateiname des zu verschachtelnden Makros.

Beispiel

1. **NestMacro(DRUCKEN.WCM)**

Das Makro *DRUCKEN.WCM* wird aufgerufen.

2. **ChainMacro(Keyboard(Bitte Name des auszuführenden Makros eingeben))**

Der Dateiname des Makros wird beim Antreffen des Befehls *ChainMacro()* über die Tastatur eingegeben.

7.29 NextRecord

(NächsterDatensatz)

Befehlsform: NextRecord

Durch diesen Befehl wird veranlaßt, den nächsten Datensatz einer bestimmten Datendatei zu lesen. Ist kein weiterer Satz mehr vorhanden, wird das Mischen beendet oder die Kontrolle an eine andere Datendatei abgegeben, sofern eine Verschachtelung von Datendateien vorgenommen wurde. Hier haben Sie die Möglichkeit, beim Mischen nur bestimmte Datensätze wie z.B. ausgewählte Firmenbranchen oder Postleitzahlengebiete zu selektieren. Ein Datensatz wird in diesem Fall nur dann gemischt, wenn eine bestimmte Bedingung zutrifft, ansonsten wird der nächste Datensatz gelesen.

Beispiel

```
Codes(
  While(Variable(Postleitzahl)<>64273)
    NextRecord
  EndWhile
  Call(AdressenDruck)
)
```

Hier wird ein Datensatz nur dann verarbeitet, wenn er die Postleitzahl 64273 enthält.

Weitere Hinweise siehe: PageOff, PageOn

7.30 NToC

Befehlsform: NToC(Zahl)

Konvertieren eines Tastenwertes (numerischer Wert) in das entsprechende Zeichenäquivalent. Beachten Sie bitte auch den Befehl CToN. Es wird kein *sinnvolles* Ergebnis angezeigt, wenn der Befehl bei einer Zahl verwendet wird, zu der es kein Zeichenäquivalent gibt.

Parameter

Zahl: Tastenwert, der in ein Zeichen umzurechnen ist. Nach der Ausführung dieses Befehls wird das ermittelte Zeichen auf Cursorposition eingefügt. In Verbindung mit *Assign* oder *AssignLocal* kann es auch in einer Variablen gespeichert werden (siehe Beispiel).

Beispiel

```
1 GetString(Eingabe;Bitte Tastenwert eingeben;Befehl "NToC" testen)
2 AssignLocal(Rechnen;Variable(Eingabe))
3 AssignLocal(Ausgabe;NToC(Variable(Eingabe)))
4 Insert(Ermitteltes Zeichen = )
```

```

5  Insert(Variable(Ausgabe))
6  AssignLocal(Zeichensatznummer;Variable(Rechnen) / 256)
7  Insert(, Zeichensatznummer = )
8  Insert(Variable(Zeichensatznummer)
9  AssignLocal(Zeichennummer;Variable(Rechnen) % 256)
10 Insert(, Zeichennummer = )
11 Insert(Variable(Zeichennummer)
12 Insert(
13 Rewrite

```

In dem gezeigten Beispiel soll für eine eingegebene Zahl das zugehörige Zeichen ermittelt werden.

- 1 Anzeige einer Dialogbox, Übernahme der eingetippten Zahl in die Variable *Eingabe*.
- 2 Speichern der Eingabe in der Variablen *Rechnen*, da nach der Ausführung von NToC der Inhalt von *Eingabe* verändert ist. Die Variable *Rechnen* wird benötigt, um die Zeichensatz- und die Zeichennummer zu errechnen.
- 3 Umrechnen der Zahl.
- 4 Text »Ermitteltes Zeichen = » schreiben.
- 5 Schreiben der Variablen *Ausgabe*.
- 6 Errechnen der Zeichensatznummer.
- 7 Schreiben des Textes », Zeichensatznummer: ».
- 8 Schreiben des Inhalts der Variablen *Zeichensatznummer*.
- 9 Errechnen der Zeichennummer.
- 10 Schreiben des Textes », Zeichennummer: ».
- 11 Schreiben des Inhalts der Variablen *Zeichennummer*.
- 12 Zeilenschaltung einfügen.
- 13 Bildschirm neu anzeigen.

Sie finden ein Beispiel für diesen Befehl auf der CD-ROM unter dem Dateinamen NTOC.FRM.

Weitere Hinweise siehe: CToN

7.31 PostNet

Befehlsform: PostNet(USA-Postleitzahl)

Für Postsendungen in die USA können Sie hier je nach Bedarf eine fünf-, neun- oder elfstellige Postleitzahl (numerisch) eingeben, die in einem Barcode dargestellt wird. Dieser besteht aus langen und kurzen vertikalen Strichen, die teilweise unterschiedlich dick sind. Er erscheint

erst beim Drucken. Es handelt sich um den POSTNET-Barcode des amerikanischen Postdienstes (ZIP-Code). Verwenden Sie diesen Code bei Postsendungen in die USA. Er dient in erster Linie dem automatischen Sortieren von Postsendungen. Die Zustellung kann dadurch vielleicht schneller erfolgen.

Parameter

USA-Postleitzahl: Fünf-, neun- oder elfstelliger Barcode des amerikanischen Postdienstes.

Beispiel: PostNet(Field(USA-Plz))

Beim Mischen wird hier die Postleitzahl des Feldes *USA-Plz* als Barcode gedruckt.

7.32 PageOff / PageOn

(FnsAus/FnsEin)

Befehlsform: PageOff
PageOn

PageOff unterdrückt einen festen Seitenumbruch, der nach jedem Formulardatei-Durchlauf in das Dokument eingefügt wird, z.B. beim Mischen von Datensätzen, um eine Liste zu erzeugen (siehe Beispiel). Dieser Befehl darf in der Formular- oder in der Datendatei enthalten sein. In Verbindung mit dem Mischbefehl *Print* wird das Drucken leerer Seiten verhindert.

PageOn (= Standardvorgabe) fügt nach jedem Formulardatei-Durchlauf einen festen Seitenumbruch in das Dokument ein:

| | | | | | |
|-------|--------|------------|----------|----------|---------|
| 0001 | Abel | Heinrich | 28.10.48 | 3.500,50 | Einkauf |
| ===== | | | | | |
| 0002 | Becker | Silke | 15.07.57 | 2.900,20 | Verkauf |
| ===== | | | | | |
| 0003 | Coch | Lieselotte | 29.02.60 | 3.900,70 | Verkauf |
| ===== | | | | | |
| 0004 | Dehmel | Karl | 10.04.52 | 2.800,90 | Lager |
| ===== | | | | | |
| 0005 | Maier | Manfred | 26.08.76 | 2.100,50 | Einkauf |
| ===== | | | | | |
| 0006 | Zeller | Iris | 14.11.83 | 3.500,00 | Einkauf |

Wenn Sie bei diesem Beispiel nicht mit dem Befehl *PageOff* arbeiten würden, würde je Seite nur eine Zeile erstellt. Die Namen würden also nicht untereinander erscheinen. Beim Verwenden von *PageOff* wird die Liste korrekt gedruckt

| Personal-Liste | | | | Seite 1 | |
|-----------------|----------|----------|--------------|-----------|-----------|
| Personal-Nummer | Nachname | Vorname | Geburtsdatum | Gehalt DM | Abteilung |
| 0001 | Abel | Heinrich | 28.10.48 | 3.500,50 | Einkauf |

| Personal-Liste | | | | Seite 1 | |
|----------------|--------|------------|----------|----------|---------|
| 0002 | Becker | Silke | 15.07.57 | 2.900,20 | Verkauf |
| 0003 | Coch | Lieselotte | 29.02.60 | 3.900,70 | Verkauf |
| 0004 | Dehmel | Karl | 10.04.52 | 2.800,90 | Lager |
| 0005 | Maier | Manfred | 26.08.76 | 2.100,50 | Einkauf |
| 0006 | Zeller | Iris | 14.11.83 | 3.500,00 | Einkauf |

Die Personaldaten wurden hier in der Datei »PERSONAL.DAT« (= Datendatei) gespeichert und die Mischbefehle in der Datei »PERSONAL.TXT«, die folgende Mischbefehle enthält:

| Personal-Liste | | | | Seite 1 | |
|-----------------|-----------|-----------|--------------|-----------|-----------|
| Personal-Nummer | Nachname | Vorname | Geburtsdatum | Gehalt DM | Abteilung |
| Field(PN) | Field(NN) | Field(VN) | Field(GD) | Field(GH) | Field(AB) |

Die Listenüberschrift wurde als Kopftext eingefügt, damit sie auf allen Seiten automatisch gedruckt wird.

7.33 Print(Drucken)

Befehlsform: Print

Der bis zu diesem Punkt gemischte Text wird zum Drucker übertragen und gedruckt. Der Befehl kann in einer Daten- oder in einer Formulardatei vorhanden sein. Sie sollten ihn u. a. dann verwenden, wenn beim Mischen vieler Datensätze der Speicherplatz Ihres Computers nicht ausreicht, um die gemischten Daten sofort zu drucken und nicht erst zu speichern. Beim Antreffen dieses Befehls wird der bis dahin gemischte Text zum Drucker gespoolet und anschließend aus dem Dokument gelöscht. Daten- und Formulardatei bleiben dabei unverändert.

7.34 ProcessOn / ProcessOff

(VerarbeitenEin/
VerarbeitenAus)

Befehlsform: **ProcessOff**
 ProcessOn

Diese Befehle werden in einer Datendatei verwendet, um zu gewährleisten, daß die durch die beiden Befehle eingeschlossenen Mischbefehle beim Mischen nicht als Daten interpretiert werden.

Beispiel

```
FieldName(Anrede;Name;Straße;Wohnort)
=====
ProcessOn ChainData(ADRESSEN2.DAT)ProcessOff
HerrnEndField
Karl MeierEndField
Rheinstraße 17EndField
64273 DarmstadtEndRecord
=====
::: weitere Datensätze :::
```

Nach Beendigung dieser Datendatei wird mit den Daten der Datei ADRESSEN2.DAT weitergearbeitet.

Weitere Hinweise siehe: ChainData

7.35 Prompt

Befehlsform: **Prompt(Meldung)**

Der unter *Meldung* eingegebene Text (Konstante oder Inhalt einer Variablen) wird in einem Dialogfeld angezeigt. Als Titel des Dialogfelds wird »Benutzerdefinierte Meldung« eingeblendet. Das Dialogfeld erscheint nur ganz kurz am Bildschirm und wird sofort wieder ausgeblendet. Verwenden Sie unmittelbar hinter dem Prompt-Befehl ggf. den Befehl *Wait*, damit die Meldung am Bildschirm stehen bleibt. Standardmäßig wird das Dialogfeld am unteren Fensterrand des WordPerfect-Fensters eingeblendet. Bei Bedarf können Sie es mit der Maus auf eine andere Position verschieben. Zu einer Zeit kann immer nur eine Meldung am Bildschirm angezeigt werden. Wird trotzdem ein weiterer Prompt-Befehl ausgeführt, wird dadurch die vorherige Meldung überschrieben bzw. ausgeblendet.

Parameter

Meldung: Beliebiger Text.

Beispiel

Prompt(Bitte Formular für die Endabrechnung in den Drucker einlegen.)

Weitere Hinweise siehe: Char, GetString, Look, StatusPrompt

7.36 Quit

Befehlsform: Quit

Beim Antreffen dieses Befehls in einer Formulardatei wird das Mischen beendet. Der eventuell noch verbleibende Rest der Formulardatei, der hinter dem Quit-Befehl definiert wurde, wird in das neu entstehende Mischdokument übernommen. Mischbefehle, die sich jedoch in diesem Bereich befinden, werden nicht ausgeführt. Parameterdaten, die in den auf *Quit* folgenden Befehlen vorhanden sind, werden als Text in das Mischdokument übernommen. Sollen weder die Parameter noch der Rest der Formulardatei übernommen werden, verwenden Sie anstelle von *Quit* den Befehl *Stop*. Wird *Quit* in einer Datendatei angetroffen, wird die Datendatei beendet und das Mischen abgebrochen. Die restliche Handhabung erfolgt wie oben beschrieben.

Weitere Hinweise siehe: Stop

7.37 RepeatRow

Befehlsform: RepeatRow

Verwenden Sie diesen Befehl, wenn von einer Datendatei Daten in eine Tabelle gemischt werden. Wird dieser Befehl in einer Reihe angetroffen, wird diese Reihe für jeden Datensatz der Datendatei wiederholt.

7.38 Rewrite

(Neuanzeige)

Befehlsform: Rewrite

Standardmäßig läuft das Mischen im Hintergrund ab, die gemischten Daten werden erst dann angezeigt, wenn das Mischen beendet ist. Wird dieser Befehl angetroffen, wird der bis zu diesem Zeitpunkt gemischte Text am Bildschirm angezeigt. Der Befehl kann in einer Daten- oder in einer Formulardatei vorhanden sein.

7.39 StatusPrompt

Befehlsform: **StatusPrompt**(Meldung)

Dieser Befehl zeigt eine Meldung in einem Dialogfeld so lange an, bis sie durch eine andere Meldung überschrieben wird (z. B. erneuter *StatusPrompt* oder *Keyboard* usw.).

Parameter

Meldung: Beliebiger Text.

Beispiel: **StatusPrompt** (Das Mischen der ersten Datendatei ist beendet.)

Weitere Hinweise siehe: Keyboard

7.40 StepOff / StepOn

Befehlsform: **StepOff / StepOn**

Das Mischen wird beim Antreffen von *StepOn* schrittweise ausgeführt. Nach der Ausführung einer Anweisung wird der Befehl angezeigt, der als nächster ausgeführt wird. Die Ausführung erfolgt jedoch erst dann, wenn eine beliebige Taste gedrückt wird. Der Befehl dient in erster Linie dem Anwender zur Beseitigung von Fehlern, die beim normalen Testen nicht sofort erkannt werden. Blenden Sie ggf. das Steuerzeichenfenster ein. Durch die paarweise Verwendung der Befehle *StepOn* und *StepOff* kann ein bestimmter Bereich eingegrenzt werden. Nach der Ausführung von *StepOn* wird in der Statuszeile die folgende Meldung eingeblendet:

Mischen im Einzelschrittmodus; weiter mit bel. Taste ...

Hinter den drei Punkten wird jeweils der nächste auszuführende Befehl angezeigt. Sollte die Statuszeile ausgeblendet sein, müssen Sie diese *vor* dem Mischen einblenden, sonst erfolgt keine Anzeige. Achten Sie darauf, daß die zu testenden Befehle durch den Mischbefehl *Codes()* eingeschlossen ist, sonst werden auch die Formatierbefehle wie z.B. Zeilenschaltungen in die Anzeige einbezogen, was jedoch in der Statuszeile nicht angezeigt wird. Der Bereich hinter den drei Punkten bleibt dann leer, was den Anschein erweckt, es würde nichts getan.

Weitere Hinweise siehe: Codes

7.41 Stop

Befehlsform: Stop

Durch diesen Befehl können Sie das Mischen beenden. Der Befehl kann in einer Daten- oder in einer Formulardatei definiert sein. Im Gegensatz zum Befehl *Quit* wird hier der Rest der Formulardatei nicht in das neu entstehende Mischdokument übernommen. Wurden Dateien verschachtelt, erfolgt kein Rücksprung in die übergeordnete Datei. Das Mischen wird auch hier in jedem Fall beendet. Dieser Befehl ist z.B. dann sinnvoll, wenn Sie von einer sehr großen Adressendatei nur die ersten 200 Adressen mischen möchten. Laden Sie hierzu diese betreffende Datendatei, und fügen Sie hinter dem 200. Satz den Befehl *Stop* ein, und starten Sie das Mischen. Versäumen Sie nicht, nach dem Mischen den Stop-Befehl wieder zu löschen. Dieser Befehl kann auch während der Texteingabe in Verbindung mit dem Mischbefehl *Keyboard* zum Beenden der Eingabe verwendet werden. Klicken Sie hierzu in der angezeigten Funktionsleiste auf [Stop].

Weitere Hinweise siehe: *Quit*

7.42 System

Befehlsform: System(*SysVar*)

Dieser Befehl ermittelt den aktuellen Wert der unter *SysVar* angeforderten Systemvariablen. Sie können sich hiermit den aktuellen WordPerfect-Status anzeigen lassen und mit diesem Wert weiterarbeiten. Die verfügbaren Systemvariablen sowie deren Namen werden nach der Auswahl des Befehls in dem Listenfeld [Variablenliste] angezeigt. Nach einem Doppelklick auf dem betreffenden Eintrag wird dieser in das Textfeld [Variable] übernommen.

Parameter

SysVar: Systemvariable, die aus der Liste der zur Verfügung stehenden Variablen ausgewählt werden kann.

Beispiel

1. If(System(Zeile)>20,0c)

Insert(

=====

)

EndIf

Wird beim Schreiben des neuen Mischdokuments ein Zeilenwert größer als 20 cm erreicht, wird ein fester Seitenumbruch eingefügt.

2. Insert(System(Zeile))

Die aktuelle Zeilenposition (z. B. 20,5c) wird im Dokument auf Cursorposition eingefügt.

7.43 SubstData

(ÜbergDaten)

Befehlsform: **SubstData**(Dateiname)

Mit diesem Befehl können Sie während des Mischens eine andere Datendatei zuordnen. Der Name der neuen Datei muß unter *Dateiname* eingegeben werden. Wird die angegebene Datei nicht gefunden, erfolgt eine Fehlermeldung und der Abbruch des Mischens. Verwenden Sie den Befehl *OnError*, um in diesem Fall eine entsprechende Meldung anzuzeigen oder um das Mischen ggf. mit einer anderen Datei fortzusetzen.

Parameter

Dateiname: Name der neuen Datendatei. Der Dateiname kann auch zum Zeitpunkt des Mischens über die Tastatur eingegeben werden (siehe Beispiel 2).

Beispiel

1. SubstData(ADRESSEN.DEU)
Das Mischen wird mit den Daten der Datei *ADRESSEN.DEU* fortgesetzt.
2. SubstData(Keyboard(Bitte neue Datendatei eingeben))
Der Dateiname der Datendatei wird beim Antreffen des Befehls *SubstData()* über die Tastatur eingegeben.

Weitere Hinweise siehe: OnError, NestData, ChainData

7.44 SubstForm

(ÜbergFormular)

Befehlsform: **SubstForm**(Dateiname)

Die Verarbeitung mit der aktuellen Formulardatei wird beendet und dafür mit der angegebenen Formulardatei fortgefahren. Die Ausführungskontrolle geht auf die neue Datei über. Es erfolgt keine Rückkehr mehr zu der bisherigen Formulardatei. Befehle, die in der bisher aktuellen Formulardatei nach dem Befehl *SubstForm()* folgen, werden nicht mehr abgearbeitet. Wird keine Datei angegeben oder wird die angegebene nicht gefunden, wird eine Fehlerbedingung gesetzt. Verwenden Sie den Befehl *OnError*, um in diesem Fall eine entsprechende Meldung anzuzeigen oder um das Mischen ggf. mit einer anderen Datei fortzusetzen.

Parameter

Dateiname: Name der neuen Formulardatei. Der Dateiname kann auch zum Zeitpunkt des Mischens über die Tastatur eingegeben werden (siehe Beispiel 2).

Beispiel

1. SubstForm(MONABSCH.94)
Das Mischen wird mit der Formulardatei *MONABSCH.94* fortgesetzt.
2. SubstForm(Keyboard(Bitte neue Formulardatei eingeben))
Der Dateiname der Formulardatei wird beim Antreffen des Befehls *SubstForm()* über die Tastatur eingegeben.

Weitere Hinweise siehe: ChainForm, NestForm

7.45 Variable

Befehlsform: **Variable**(*Variable*)

Variablen sind Speicherfelder, die Daten aufnehmen, speichern und wiedergeben können. Der Inhalt kann beliebig oft abgerufen oder verändert werden. Über diesen Befehl können Sie den Inhalt einer globalen oder lokalen Variablen abrufen. In Variablen können Sie die über Tastatur, Misch- oder Makrobefehle eingegebenen Daten oder Steuerzeichen speichern, um sie zu einem späteren Zeitpunkt zu verarbeiten (um z.B. aufgrund des Inhalts die Steuerung des Mischens zu beeinflussen oder um Unterroutrinen aufzurufen). Die Zuordnung eines Wertes zu einer Variablen können Sie über verschiedene Arten vornehmen. Der Wert darf nicht in Anführungszeichen stehen. Sollte es sich um eine Formel handeln, wird das Ergebnis ermittelt und in der Variablen gespeichert. Andernfalls wird der eingegebene (zugeordnete) Wert als Zeichenfolge behandelt. Mischroutinen bleiben dadurch voll variabel, weil anstelle von fixen Werten immer mit den aktuellen Variableninhalten gearbeitet werden kann. Sie können diese Variablen als Parameter, als Teil eines Parameters oder als selbständigen Befehl verwenden.

Parameter

Variable: Beliebiger Variablenname. Die Verwendung eines reservierten Wortes (siehe Kapitel »5.11 Reservierte Wörter«) ist hier nicht erlaubt. Der Variablenname darf nicht in Hochkommata eingeschlossen sein. Es wird keine Unterscheidung zwischen Groß- und Kleinbuchstaben getroffen. Die Variablennamen *Eingabe*, *eingabe* oder *EiNgAbE* sind darum identisch. Wurde für eine lokale und eine globale Variable derselbe Name vergeben, wird über diesen Befehl nur die *lokale* Variable angesprochen. Auf die *globale* Variable kann in diesem Fall nicht zugegriffen werden.

Beispiel

- 1 Variable(Name)
Der Inhalt der Variablen *Name* wird auf Cursorposition eingefügt.
- 2 Variable(Anrede)Mrgcmdnd(EndField)
Der Inhalt der Variablen *Anrede* wird auf Cursorposition eingefügt und durch den Mischbefehl *EndField* abgeschlossen.

- ```
3 If(Variable(Abfrage)=j)
 Assign(DruckenText;j)
 EndIf
 Wenn die Variable Abfrage den Inhalt j enthält, trifft die Bedingung zu.
4 While(Variable(Postleitzahl)<>64273)
 NextRecord
 EndWhile
 Überlesen aller Datensätze, die in der Variablen Postleitzahl einen anderen
 Wert als 64273 enthalten.
5 If(FirstCap(Variable(Text)) = Ende) Quit EndIf
 Enthält die Variable Text den Inhalt ende wird der erste Buchstabe in einen
 Großbuchstaben umgewandelt.
```

Weitere Hinweise siehe: *Assign, AssignLocal*