

Erstellen von Befundvorlagen

Vorlagenerstellung für Befunde

Erstellung einer Vorlage

Eine Vorlage für den Befundassistenten kann mit jedem Textverarbeitungsprogramm, welches „verborgene Zeichen“ und das Dateiformat „Rich Text Format“ (.rtf) unterstützt, erstellt werden. Am besten eignet sich Word für die Erstellung der Befundvorlagen.

Um eine Vorlage in die Formularauswahl von Winchim aufzunehmen, muss sie im Rich-Text-Format (.rtf) in das Winchim-Vorlagenverzeichnis gespeichert werden. Der Pfad für das Winchim-Vorlagenverzeichnis ist in der Datei „Winchim.ini“ (Konfigurationseinstellungen) unter dem Eintrag „VORLAGEN“ definiert. (Meist im Verzeichnis „C:\Winchim\Daten\Vorlagen“)

Es kann also jedes beliebige (Word-) Dokument als Vorlage verwendet werden, indem zuerst das Dokument mit Word geöffnet wird und anschließend unter „Datei/Speichern unter“ im Rich-Text-Format in das Winchim-Vorlagenverzeichnis gespeichert wird.

Anweisungen an Winchim, bestimmte Daten an einer Stelle einzufügen müssen als „Verborgene Zeichen“ definiert werden.

Eingabe von Anweisungen

Der Befundassistent durchsucht die Vorlage nach verborgenen Texten und verwendet jeden verborgenen Text als Anweisung zur Befunderstellung.

Grundsätzlich wird zwischen „Feldern“ und „Befehlen“ unterschieden. Ein „Befehl“ veranlasst den Befundassistenten eine bestimmte Aktion durchzuführen, während Feldern lediglich ein Wert zugewiesen werden kann.

Jeder Anweisung an den Befundassistenten muss mit einem Strichpunkt abgeschlossen werden. Wird kein Strichpunkt angegeben, so wird bei der Befunderstellung eine Fehlermeldung ausgegeben.

Um in Word eine Anweisung für den Befundassistenten zu erstellen gehen Sie also nach folgenden Schritten vor:

- 1) Schreiben Sie alle Anweisungen der Vorlage im Klartext und im gewünschten Format (Fett, Kursiv, Unterstrichen, andere Schriftart, etc) an die gewünschte Stelle.
- 2) Schließen Sie den Anweisung mit einem Strichpunkt ab.
- 3) Markieren Sie die gesamte Anweisung (inklusive Strichpunkt!)
- 4) Klicken Sie unter „Format/Zeichen“ die Eigenschaft „verborgen“ an. (oder Kurztasten „Strg+H“)

Wurden diese Schritte korrekt durchgeführt, so verschwindet die Anweisung, sobald die Anzeige der verborgene Zeichen deaktiviert ist. (¶)

Beispiele für Anweisungen: <ul style="list-style-type: none">• <i>Word</i>;• <i>Speichern(„Brief“)</i>;• <i>Auswahl(„Rauchfang“)</i>;	Beispiele für Felder: <ul style="list-style-type: none">• <i>Rauchfang..Durchmesser1</i>;• <i>Feuerstätte..Brennstoff</i>;
---	---

Felder

Bei den Feldern wird zwischen „Eingabefeldern“ und „Datenbankfeldern“ unterschieden.

Eingabefelder

Über Eingabefelder können temporäre Eingaben für den Befund gemacht werden, die jedoch in keinem Feld der Datenbank abgespeichert werden. Diese Eingaben können für den Befund verwendet werden, sind danach aber nicht mehr verfügbar.

Eingabefelder werden meist in Kombination mit den Eingabebefehlen (s.u.) verwendet.

Die Namen für Eingabefelder können dabei beliebig vergeben werden. Es ist dabei allerdings darauf zu achten, dass keine Namen verwendet werden, die mit einem Befehl verwechselt werden können.

Beispiel: *Text(„Datum“)*; - diese Anweisung ist nicht erlaubt, da das Wort „Datum“ bereits als Eingabebefehl verwendet wird. (siehe unten)

Datenbankfelder

Datenbankfelder besitzen einen Tabellennamen und einen Feldnamen, welche durch einen, zwei Punkte oder durch die Zeichen .# voneinander getrennt sind. (Beispiel: *Rauchfang..Bezeichnung*, *Rauchfang.Bezeichnung*, *Rauchfang.#Bezeichnung*).

Der Name des Datenbankfeldes kann auch aus Winchim durch Drücken der ALT-Taste ermittelt werden. (ALT-Taste drücken und dann mit der Maus über das gewünschte Eingabefeld fahren)

Die Schreibweise des Datenbankfeldes mit ein oder zwei Punkten als Trennzeichen beeinflusst die Ausgabe in den Befund. Es gelten dabei folgende Regeln:

Rauchfang..Durchmesser1:

Werden zwei Punkte verwendet, so wird für den Fall, dass das Feld noch keinen Wert hat der Wert „0“ ausgegeben

Rauchfang.Durchmesser1:

Bei einem Punkt wird für den Fall, dass das Feld keinen Wert hat das Zeichen „-“ ausgegeben.
Bei Verwendung des „Feld“-Befehles sollte nur ein Punkt als Trennzeichen verwendet werden, da ansonsten immer der Wert 0 als Vorgabewert geladen wird

Rauchfang.#Durchmesser1:

Bei Verwendung des Trennzeichens .# wird statt einem 0-Wert ein Leerzeichen ausgegeben.

*Rauchfang.*Durchmesser:*

Es werden im Normalfall alle Datensätze einer Tabelle durchlaufen. Der zuletzt angezeigte Datensatz wird auch beim nächsten Feld verwendet. Dies kann durch die Verwendung des Trennzeichens .* verhindert werden. (im Zusammenhang mit der Funktion „Suchen“ !)

Bei Verwendung des Trennzeichens .* wird immer der aktuelle Datensatz ausgegeben.

Verwendung des Feldes ohne Ausgabe

Wird ein Feld nur für Abfragen genützt, ohne dass der Feldwert in den Befund ausgegeben werden soll, so kann dem Feldwert der Buchstabe S (für Still) angefügt werden.

Beispiel: Um den Inhalt des Eingabefeldes „Querschnitt“ in einer Abfrage auszuwerten kann folgende Anweisung verwendet werden: *if(EQ(QuerschnittS, „rund“))*; hier wird der Inhalt des Feldes „Querschnitt mit dem Wert „rund“ verglichen. Beachte: Der Buchstabe S wird für Abfragen dem Feldwert angefügt.

Kommentare

Es können in allen Vorlagen Kommentare gesetzt werden. Diese dienen der leichteren Lesbarkeit der Vorlage und werden nicht in den fertigen Befund aufgenommen.

Kommentare werden gekennzeichnet, indem sie in geschwungene Klammern gesetzt werden.
Beispiel: { Das ist ein Kommentar };

Eingabebefehle:

Unter Eingabebefehlen werden solche Steuerbefehle verstanden, die den Befundassistenten veranlassen, ein Eingabefenster zu öffnen, indem der Anwender Daten für den Befund angeben kann. Diese Daten werden entweder in Datenbankfeldern oder in Eingabefeldern gespeichert, welche im Befund auch öfters verwendet werden können.

Alle Eingabebefehle können entweder *still* oder mit Ausgabe in den Befund verwendet werden. Wird einem Befehl der Buchstabe „S“ (für „Still“) angefügt, so wird der eingegebene Wert NICHT in den Befund geschrieben.

Hinweis: Eingabebefehle werden in den neueren Vorlagen kaum mehr verwendet, da sämtliche Felder, die zur Erstellung eines Befundes sind, bereits in der Datenbank erfasst sein sollten. Die Eingabe der Daten sollte bereits vorher (über AppChim oder über das Anlagefenster) erfolgt sein.

Format der Eingabebefehle:

Jedem Eingabebefehl kann eine Liste von Eingabeparametern angegeben werden. Das Format für alle Eingabebefehle ist identisch:

Befehl(„Feldname“, [„Vorgabewert“], [„Format“], [„Eingabetext“], [Verarbeitung]);

Dabei ist die Eingabe des Feldnamens verpflichtend, die Angabe eines Vorgabewertes, des Formats und des Eingabetextes sind optional. (Optionale Parameter werden in dieser Darstellung in eckige Klammern gesetzt.)

Den Parametern kommt dabei folgende Bedeutung zu:

Feldname: Gibt den Namen des Eingabefeldes an. Die eingegebenen Daten werden dem angegebenen Feldnamen zugeordnet. Bereits eingegebene Werte können dabei aus einer für diesen Feldnamen angelegten Auswahlliste gewählt werden. Sollte es sich bei der Eingabe um ein Textfeld handeln, so können für jeden Feldnamen spezielle Textbausteine angelegt werden. Es empfiehlt sich daher in verschiedenen Befundvorlagen für identische Angaben den gleichen Feldnamen zu verwenden.

Vorgabewert: Der Vorgabewert wird als Default-Einstellung vorgeschlagen. Sollte es sich beim Feld um ein Datenbankfeld handeln, so wird dieser Wert nur vorgeschlagen, falls sich noch kein anderer Wert in der Datenbank befindet.

Format: Unter Format wird das Ausgabeformat des eingegeben Wertes festgelegt. Die Formatangabe ist dabei je nach Befehl verschieden. Details zur Formatangabe entnehmen Sie der Befehlsbeschreibung.

Eingabetext: Hier kann eine nähere Beschreibung zum Eingabefeld erfolgen. Dieser Text wird als Hilfe bei der Eingabe des Wertes anstatt des Feldnamens angezeigt.

Verarbeitung: Hier werden Einzelheiten zur Verarbeitung des Befehls angegeben. Derzeit sind folgende Verarbeitungskennungen zugelassen:0...Normale Verarbeitung. (Bei vorhandenem Wert wird die Eingabe übergangen)1...Das Eingabefeld wird auch nach Drücken der Taste „Fertigstellen“ angezeigt.

Verfügbare Eingabebefehle:

Text / TextS:

Über diese Befehle kann beliebiger Text eingegeben werden. Es besteht die Möglichkeit mit „Strg+F11“ Textbausteine für den aktuellen Feldnamen anzulegen, die in späteren Befunden mit „F11“ wieder abgerufen werden können.

TextS hat die gleiche Funktion wie Text, jedoch erfolgt dabei keine Ausgabe in den Befund. Für diese Befehle wird keine spezielle Formatbezeichnung verwendet.

Zahl / ZahlS:

Es wird eine Zahl eingegeben.

Unter dem Parameter „**Format**“ kann die Anzahl der erwünschten Kommastellen angegeben werden.

Beispiel: Zahl(„Durchmesser Verbindungsrohr“, „“, 1); = Zahl mit einer Kommastelle

Liste / ListeS:

Es wird Text aus einer Auswahlliste eingegeben. Neuer Text wird dabei in die Auswahlliste aufgenommen.

JaNein/JaNeinS oder JN/JNS:

Es kann eine Entscheidung zwischen mehreren Möglichkeiten (Ja/Nein) vorgenommen werden.

Unter dem Parameter **Format** können die Eingabemöglichkeiten mit Schrägstrich getrennt eingegeben werden. Wird dabei einem Buchstaben das Zeichen & vorgesetzt, so kann dieses Element durch tippen des entsprechenden Buchstabens ausgewählt werden. Wird keine Formatangabe benutzt, so werden die Eingabeoptionen „&Ja/&Nein“ verwendet.

Beispiel: JaNein(„Anlage OK“, „Anlage OK“, „Anlage &OK / Anlage &nicht OK“);

Datum / DatumS:

Es wird ein Datum eingegeben. Wird unter dem Parameter „Vorgabe“ das Wort „heute“ angegeben, so wird das aktuelle Datum als Vorgabewert gesetzt.

Unter dem Parameter Format kann das Datumsformat eingegeben werden. Für eine Liste der möglichen Datumsformate siehe Tabelle 1.

Feld/FeldS oder FN/FNS:

Es wird ein Datenbankfeld eingegeben. Die Eingabeart richtet sich dabei nach dem jeweiligen Datenbankfeld oder kann im „**Format**“-Parameter angegeben werden.

Dafür sind folgende Werte zulässig:

„Liste“ Die Eingabe erfolgt als Combo-Feld

„JN“ Die Eingabe erfolgt als Auswahl zwischen Ja und Nein

„Text“ Die Eingabe erfolgt als Text-Feld

„Zahl“ Die Eingabe erfolgt als Zahl

1-5 Die Eingabe erfolgt als Zahl wobei die Anzahl der Nachkommastellen hier angegeben wird. (Siehe Befehl „Zahl“)

„A1/A2/A3/...“ Es kann zwischen den angegebenen Auswahlkriterien entschieden werden.

Auch der Formatparameter ist je nach Art des Datenbankfeldes unterschiedlich. Handelt es sich dabei um eine Zahl, so kann die Anzahl der Kommastellen hier angegeben werden.

Bei der Eingabe des Wertes wird der aktuelle Wert aus der Datenbank angezeigt. Bei Änderungen wird der neue Wert in die Datenbank eingetragen.

Dabei ergibt sich allerdings ein Unterschied, ob der Feldname vom Tabellennamen mit ein oder zwei Punkten getrennt geschrieben ist.

Beispiele:

Feld(„Feuerstätte..Nennleistung_Max“) Falls keine Nennleistung_Max angegeben ist, wird „0“ eingefügt, außerdem wird eine Eingabe des Wertes bei der Fertigstellung des Befunds übergangen.

Feld(„Feuerstätte.Nennleistung_Max“) Falls kein Wert angegeben ist wird „-“,“ eingefügt, allerdings wird bei der Fertigstellung des Befunds der richtige Wert abgefragt. Wird dieser Wert geändert, so ändert sich damit auch der Datenbankeintrag.

Bereits eingegebene Felder können in der Vorlage ausgewertet und verwendet werden.

Dazu muss einfach der Feldname **ohne Anführungszeichen** ausgeschrieben werden.

Wird dem Feldnamen ein **S** angehängt, so wird der Wert ohne eine Ausgabe in den Befund verwendet. (zB in einer if-Abfrage)

Beispiel: JaNein(„Empfehlungen“); (hier wird eingegeben, ob Empfehlungen folgen)if(EQ(EmpfehlungenS, „Nein“); SchreibeFeld(Feuerstätte..Notiz, „“); endif; Falls keine weiteren Empfehlungen angefügt werden, wird die Notiz der Feuerstätte gelöscht.

Besondere Bedeutung kommt dabei dem Datenbankfeld „StringFeld“ zu. Über dieses Datenbankfeld können beliebige Eingabefelder deklariert werden. Diese Felder bleiben dauerhaft beim jeweiligen Datensatz gespeichert und sind auch über die Datenbank abrufbar.

Ausgabebefehle

Ausgabebefehle werden verwendet, um Datenfelder, bzw. sonstige Felder in bestimmter Form auszugeben.

Now(„Datumsformat“);

Fügt das aktuelle Datum im angegebenen Datumsformat in den Befund ein.

Dieser Befehl erfordert im Unterschied zu den übrigen Eingabebefehl keine Anwendereingabe (da immer das aktuelle Tagesdatum verwendet wird) und gilt somit als Sonderfall unter den Eingabebefehlen.

Das Datumsformat bestimmt die Art der Datumsausgabe.

DateTimeToStr(Format, Feld);

Mit diesem Befehl kann ein Datum gemäß den Datumsoptionen (siehe Tabelle) ausgegeben werden.

Beispiel: DateTimeToStr(„dddd, dd.mmmm yyyy“, Termine.Datum); gibt den Kehrtermin in der Form „Montag, 03. März 2002“ aus.

ZeitAnzeige(Datum, VonZeit, BisZeit);

Es wird die Uhrzeit gemäß den Einstellungen in den Optionen für Wunschzeit und Fixzeit ausgegeben.

Beispiel: ZeitAnzeige(Termin.Datum, Termin.VonZeit, Termin.Biszeit);

Ausgabe(Ausdruck, [Format]) oder *AG(Ausdruck, [Format])*

Mit diesem Befehl kann ein beliebiger Ausdruck in den Befund geschrieben werden. Dabei kann es sich um eine Konstante oder auch um einen mathematischen Ausdruck handeln, der vor der Ausgabe berechnet wird.

Unter **Format** kann die Anzahl der Dezimalstellen angegeben werden, falls es sich bei dem Ausdruck um eine Zahl handelt.

Soll eine Währung ausgegeben werden, so muss folgende Formatangabe verwendet werden: „%m“.

Weitere Formatmöglichkeiten:

%e:

Bei der Formatangabe „%e xxx“ wird falls der Text nicht leer ist der Wert xxx als „Einheit“ mit angedruckt. *Beispiel: AG(Feuerstätte..Nennleistung_Max, „%e kW“);*

%p:

Gleich wie bei %e, allerdings wird der Formattext VOR dem Wert gesetzt, falls ein Wert vorhanden ist.

%s:

Bei der Formangabe „%s“ wird der das %s durch den Text ersetzt. *Beispiel: AG(Verfüger..Titelname, „Über die Feuerstätte verfügt %s“);*

%sx:

Gleich wie %s, allerdings wird nur dann Text ausgegeben, wenn das Feld nicht leer ist.

%u:

Es wird die VonZeit (gespeichert als Integer bis 2359) als Uhrzeit ausgegeben, sofern eine Zeit vorhanden ist. Der Wert %u wird dabei durch die Uhrzeit ersetzt.

zB: AG(Object.VonZeit, „ab %u Uhr“);

JaNein oder jn:

Wird der Ausdruck „JaNein“ angegeben, so werden anstatt des in der Datenbank gespeicherten „J“, bzw. „N“ die Wörter „Ja“ bzw. „Nein“ ausgegeben. Beim Wert X wird „Nicht erforderlich“ ausgegeben

io:

Gleich wie bei jn, allerdings wird hier für J „In Ordnung“ und für N „Nicht in Ordnung“ ausgegeben.

Beispiele:

Zahl(„Gasdruck“);

if(NE(GasdruckS,0));mbar = Ausgabe(mul(GasdruckS,100),2); Pa endif; Es wird der Gasdruck in der Einheit" mbar" eingegeben und falls der Wert ungleich 0 ist wird der Gasdruck in Pascal umgerechnet und mit 2 Dezimalstellen angezeigt.

Ausgabe(Rechnung..BruttoGerundet „%m“); Es wird der Rechnungsbetrag im aktuellen Währungsformat ausgegeben.

Ausgabe(Emission.KesselDicht, „JaNein“); Es wird ein Ja oder ein Nein ausgegeben.

AddText(Text1, Text2, Text3, ...);

Es werden mehrere Texte verknüpft.

Beispiel: AddText(„Überprüfung von Feuerstätte ", Feuerstätte..Bezeichnung, „der Partei ", Verfüg..Titelname);

Coalesce(Ausdruck1, Ausdruck2, Ausdruck3, ...);

Es wird der erste Ausdruck ausgegeben, der nicht leer ist.

Beispiel: Coalesce(StringfeldS(„Feuerstätte.Stringfeld“,,,„Anschluss“), Feuerstätte.#VB_Abgasstutzen); (Gibt zuerst Anschluss aus, wenn nicht vorhanden dann VB_Abgasstutzen)

Stringfeld(Datenbankfeld,[Vorgabe],[Format],„Feldname“);

Hier wird der Wert eines Stringfeldes (Teil eines Datenbankfeldes) ausgegeben ohne vorher abgefragt zu werden. Die Angabe des Datenbankfelds, bzw. der Feldname sind zwingend erforderlich.

Der Parameter **Vorgabe** gibt den Wert, der ausgegeben wird, wenn das Feld leer ist.

Format gibt das Datenformat aus (gemäß der Tabelle im Anhang bei Datumsformaten, bzw. Format-String für andere Werte).

Beispiel:

Stringfeld(„Objektstamm..Stringfeld“, „k.A.“, %s, „Baufirma“);

Hinweis: Wurde das Feld vorher im Report bereits verwendet, so kann es einfach durch den Feldnamen auch aufgerufen werden. (Hier einfach: *Baufirma*);

Strf(„Feldname“);

Wird bereits vorher das verwendete Datenbankfeld definiert, so genügt es, in weitere Folge nur mehr den Namen des Stringfeldes anzugeben.

Beispiel:

```
Stringfeld(„Objektstamm..Stringfeld", "k.A.", "%s", "Baufirma");  
Strf(„Baufirma");
```

GemeindeEinfügen; GemeindeEinfügen(„Feldnamen");

Bei Verwendung dieses Befehls wird automatisch die Adresse der für das aktuelle Objekt zuständigen Gemeinde in den Befund eingefügt.

Wird dabei der Befehl ohne zusätzliche Parameter verwendet (Standard), so wird die gesamte Adresse übergeben. (Vorausgesetzt, die Adresse wurde unter

„Allgemeine Daten/Orte" bereits angelegt) Im Parameter „Feldnamen" können nur einzelne Felder der Gemeindeadresse geladen werden.

Beispiel: GemeindeEinfügen(„Titelname"); Fügt nur den Namen der Gemeinde in den Befund ein.

SmoothTime: Format: SmoothTime(„Zeitfeld")

Mit diesem Befehl wird das Zeitfeld, das als Parameter angegeben werden muss auf ganze 2 Stunden gerundet. Im „Format"-Parameter kann optional die Anzahl der Stunden eingegeben werden, auf die eingeschränkt werden soll. (Voreinstellung: 2 Stunden)

DefineAlias(„Alias=Vollständige Bezeichnung");

Alias(„Alias");

Es kann ein Alias definiert werden, der dann im folgenden „Alias“-Befehl anstatt der Kurzbezeichnung die vollständige Bezeichnung ausgibt.

Beispiel:

```
DefineAlias(„1=Anlage ist inaktiv");
```

Alias(„1"); → Es wird „Anlage ist inaktiv“ anstatt der 1 gedruckt.

Steuerbefehle:

Über sogenannte „Steuerbefehle" werden dem Befundassistenten Informationen über die Speicherung des Befundes, bzw. den Ablauf der Befunderstellung mitgeteilt. Über diese Steuerbefehle erfolgt keine unmittelbare Eingabe von Werten für den Befund oder die Datenbank.

Auswahl: Format: Auswahl(„Tabellenname");

Mit diesem Befehl kann eine Auswahl der erwünschten Datensätze der angegebenen Tabelle vorgenommen werden. Es wird dabei eine Tabelle mit allen vorhandenen Datensätzen angezeigt und es müssen alle Datensätze markiert werden, die in den Befund aufgenommen werden sollen.

Wird nur ein Datensatz in den Befund aufgenommen, so muß dieser Datensatz in der Tabelle markiert werden. (siehe auch Befehl „Loop")

Beispiel: Auswahl(„Rauchfang");

Alle für das Objekt gespeicherten Rauchfänge werden in einer Liste angezeigt und es müssen jene Rauchfänge markiert werden, die in den Befund aufgenommen werden sollen.

CheckAnlage;

Dieser Befehl sollte als erster Befehl in der Vorlage stehen. Dadurch wird das kleine Anlagenfenster aufgerufen, in welchem der Anwender eine neue Anlage (neue Fänge, Feuerstätten, Messdatensätze und Mängel) eingeben kann.

NeueMessung: (derzeit nicht verfügbar)

Es wird ein neuer Messdatensatz angelegt. Dabei muss dem Befundassistenten in einem eigenen Eingabefenster das Messdatum, bzw. der Prüfer eingegeben werden.

NeuePartei: (derzeit nicht verfügbar)

Es wird der Name einer neuen Partei im Objekt angelegt. Der Partei kann dabei eine bereits angelegte Feuerstätte zugeordnet werden.

NeuerMangel: (derzeit nicht verfügbar)

Es wird ein neuer Mangel zum aktuellen Objekt angelegt. Der Befundassistent öffnet dabei ein eigenes Fenster, in dem die Mangeldaten eingegeben werden können.

SchreibeFeld(„Datenbankfeld“, Wert);

Mit SchreibeFeld kann ein Datenbankfeld beschrieben werden. Als Wert kann ein beliebiges anderes Datenbankfeld, ein Eingabefeld oder eine Konstante (in Anführungszeichen) übergeben werden.

Beispiel: Text(„Bemerkung“); SchreibeFeld(„Feuerstätte..Notiz“, Bemerkung); Es wird das Datenbankfeld „Feuerstätte..Notiz“ mit dem Inhalt des Eingabefeldes „Bemerkung“ beschrieben.

SchreibeMax / SchreibeMin(„Feld“, Datenbankfeld, [Format]);

Es wird der Maximalwert / Minimalwert ins Feld geschrieben. Formatangaben: „Datum“, „Int“, „Float“

Beispiel: SchreibeMax(„Frist“, Mangel..Warndatum, „Datum“);

Filter(„Filterfeld“, Filterwert); FilterOff(„Tabelle“);

Es kann eine Auswahl Datensätze auf ein bestimmtes Filterkriterium eingeschränkt werden. Dabei wird im Parameter „Filterfeld“ das zu filternde Feld angegeben. Es muss sich dabei immer um ein Datenbankfeld in der Form „Tabelle..Feldname“ handeln. Unter „Filterwert“ wird der zu filternde Wert angegeben. Es kann sich dabei um einen Kurztext oder auch um einen Feldwert einer anderen Tabelle handeln.

Es werden dann alle Datensätze aussortiert, die nicht dem Filterwert entsprechen. Eine Kaskadierung der Filterkriterien ist durch mehrere hintereinander geschriebene Befehle möglich. Die Filterung wird mit dem Befehl „FilterOFF“ wieder deaktiviert.

Beispiel: Filter(„Rauchfang..Bezeichnung“, Feuerstätte..Rauchfang); ... FilterOff(„Rauchfang“); Filtert in der Tabelle Rauchfang nur den an die Feuerstätte angeschlossenen Rauchfang heraus.

Suchen(„Suchfeld“, Suchwert);

Mit dem Befehl Suchen kann auf einen bestimmten Datensatz positioniert werden. Wurde vorher bereits ein anderer Datensatz verwendet, so sollte daraufhin das Trennzeichen .* verwendet werden um auch den gesuchten Datensatz anzuzeigen!

Dabei wird im Parameter „Suchfeld“ das zu suchende Feld angegeben. Es muß sich dabei immer um ein Datenbankfeld in der Form „Tabelle..Feldname“ handeln. Unter „Suchwert“ wird der zu suchende Wert angegeben. Es kann sich dabei um einen Kurztext oder auch um einen Feldwert einer anderen Tabelle handeln.

Der Befehl gibt den Ausdruck „TRUE“ zurück, falls ein entsprechender Datensatz gefunden wurde. Damit kann der Befehl auch als Bedingung in einer if-Abfrage verwendet werden.

Beispiel: Suchen(„Rauchfang..Bezeichnung“, Feuerstätte..Rauchfang); Sucht in der Tabelle Rauchfang nach der Bezeichnung, die in der Tabelle Feuerstätte unter dem Feldnamen Rauchfang angegeben ist.

Beispiel: if(Suchen(„Mangel..Beanstandung“, „J“)); Im Objekt ist ein Mangel! Endif;

Achtung: Der Befehl ist nur wirksam, wenn die betreffende Tabelle noch nicht an oberer Stelle im Befund verwendet wurde. Anstelle dieses Befehls sollte der Befehl „Filter“ verwendet werden.

Speichern([„Befundkennung“,ReferenzID, „Name“]);

Dieser Befehl speichert automatisch den Befund in der Befundtabelle der Winchim-Datenbank.

Unter „Befundkennung“ kann eine Kennung angegeben werden, die in der Befundtabelle angezeigt wird. „ReferenzID“ = ReferenzID, der zum Befund abgespeichert wird.

„Name“ = Name des Befunds

Word: Format: Word;

Mit diesem Befehl wird automatisch nach Fertigstellung des Befundes das entsprechende Textverarbeitungsprogramm geladen. (meist Word)

NeueNummer([„Objekt/Jahr/Monat“, [„Alle“]);

Durch diesen Befehl wird eine neue fortlaufende Nummer erzeugt und in den Befund geschrieben. Die Nummern werden dabei entsprechend den optionalen Parametern vergeben:

- „Objekt“ = Es wird die nächste Dokumentnummer des Objektes vergeben, dh. wurde für das aktuelle Objekt noch kein Befund, erstellt, so beginnt die Nummerierung mit 1.
- „Jahr“ = Die Nummerierung beginnt mit jedem Jahr neu, dh. der erste Befund des Jahres erhält die Nummer 1.
- „Monat“ = Die Nummerierung beginnt mit jedem Monat neu, dh. der erste Befund des Monats erhält die Nummer 1.
- „Alle“ = Es wird die Befundkennung (wird beim Befehl „Speichern“ angegeben) nicht in die Nummerierung mit einbezogen. Wird dieser Parameter nicht angegeben, so gibt es für jeden Befundtyp eine eigene Nummerierung.

Beispiel1: Speichern(„Eignung“); NeueNummer(„Jahr“); nächste fortlaufende Nummer für Eignungsbefunde

Beispiel2: NeueNummer(„Objekt“,„Alle“); nummeriert alle Dokumente des Objekts

Adresse([„Adresstypen“]);

Mithilfe dieses Befehls kann ein erstellter Befund an mehrere Adressen versendet werden. Es wird aus dem Befund ein Serienbrief erstellt, in dessen Steuerdatei die Adressen aus den unter „Adresstypen“ angegebenen Tabellen geschrieben werden.

Diesem Befehl muss der Befehl „Speichern“ vorausgehen, um aktiviert zu werden!

(Verwendung zB bei Mängelmeldung: Ergeht an Hausherrn und an die Gemeinde)

Folgende Adresstypen stehen zur Auswahl:

OBJEKT: Die Objektadresse wird in die Steuerdatei aufgenommen.

KUNDE: Die Kundenadresse wird in die Steuerdatei aufgenommen.

PARTEI: Die Adressen der Parteien werden in die Steuerdatei aufgenommen.

Um einen Befund als Serienbrief zu erstellen muss auch die Vorlage bereits ein Serienbrief sein. Die zugehörige Steuerdatei muss sich dabei im gleichen Verzeichnis befinden, wie die Vorlage. Sie kann ggf. auch schon Datensätze enthalten. (zB: Gemeinde, Gaswerk).

Bei dieser Steuerdatei muss es sich um eine „.txt-Datei“ handeln, in welcher die Felder durch TAB getrennt sind. Das Datensatzbegrenzungszeichen ist RETURN.

Die Adressfelder werden in folgender Reihenfolge in die Steuerdatei geschrieben:

Anrede, Name1, Name2, Name3, Strasse, Land, Plz, Ort, Tel1, Tel2, Fax, Email, Blz, KtoNr, Bank.

Dateiformat([„Dateierweiterung“]);

Wird in der Vorlage der Befehl „Dateiformat“ verwendet, so wird der erzeugte Befund anstatt im RTF-Format im angegebenen Format (zB. „.doc“) abgespeichert.

Um Ladezeit sowie Festplattenplatz zu sparen sollte dieser Befehl immer dann eingesetzt werden, wenn eine Speicherung im „.rtf“-Format zu viel Speicherplatz benötigt. Idealerweise testet man bei

der Erstellung eines Befundes zuerst aus, welches Dateiformat für den Befund am günstigsten ist. Werden im Befund keine Grafiken (Logo, etc) verwendet, so ist eine Speicherung im RTF-Format meist günstiger, der Befehl „Dateiformat“ braucht in diesem Fall **nicht** verwendet zu werden. Wird allerdings eine Grafik verwendet, so erhöht sich der Speicherbedarf im RTF-Format (bis zu mehreren MB), während im DOC-Format nur einige wenige kB benötigt werden. In diesem Fall sollte man die Anweisung **Dateiformat(„.doc“)**; unbedingt in die Vorlage schreiben!

Abfragebefehle:

Hier handelt es sich um Befehle, mit deren Hilfe Verzweigungen oder bedingte Übersetzung der Vorlage in den Befund möglich sind.

Es können damit auch mehrere Datensätze in eine Tabelle übernommen werden.

if(Bedingung); Teil A else; Teil B endif;

Falls die Bedingung erfüllt ist wird Teil A der Vorlage in den Befund übernommen, ansonsten Teil B. Unter Bedingung kann jede Anweisung eingesetzt werden, die als Ergebnis den Wert „TRUE“ oder „FALSE“ (=„Zutreffend“ oder „Nicht zutreffend“) zurückgibt.

Es können auch mehrere Bedingungen mit Hilfe der „Operatoren“ verknüpft werden.

Beispiel: if(EQ(Mangel..Beanstandung, „J“)); Dieser Mangel ist zu beanstanden! Else;Kein wesentlicher Mangel! Endif;

In der Bedingung EQ(Mangel..Beanstandung, „J“) wird der Wert des Datenbankfeldes Mangel..Beanstandung mit dem Buchstaben J verglichen. Bei Übereinstimmung wird der Satz „Dieser Mangel ist zu beanstanden!“ ausgegeben.

In dem Teilen A und B der Abfrage können wieder andere Anweisungen und Abfragen verwendet werden. Die Größe der Teile spielt dabei keine Rolle.

Loop(„Tabelle“, [Bedingung], [[-]Sortierfeld]); Teil-A endloop;

Mit diesem Befehl können mehrere Datensätze in den Befund aufgenommen werden. Auch Tabellen können mittels Loop und Endloop erzeugt werden. Der zwischen den Befehlen „Loop“ und „Endloop“ liegende Textabschnitt wird dabei für jeden gefundenen Datensatz in den Befund übernommen.

Der Parameter für die Tabelle ist zwingend, optional kann auch noch eine Bedingung, sowie ein Sortierfeld in die Schleife aufgenommen werden.

Der Parameter **Bedingung** gibt an, unter welcher Bedingung ein Datensatz in den Befund aufgenommen werden soll.

Der Parameter **Sortierfeld** gibt das Feld der Tabelle an, nach dem die Ausgabe sortiert werden soll.

Beispiel: Loop("Mangel", EQ(Mangel..Beanstandung, „J“), „-Nummer“);Mangel..Nummer;; Mangel..Zusatz;EndLoop;

Hier wird nur jene Mängel in den Befund ausgegeben, die das Feld „Beanstandung“ gesetzt haben. Bei der Ausgabe wird nach der Mangelnummer sortiert, beginnend mit der höchsten Nummer.

Bedingungen

Bedingungen werden in if-Abfragen und in Schleifen (loop) verwendet, um abhängig vom Ergebnis der Bedingung unterschiedliche Bearbeitung bei der Befunderstellung zu erzielen.

Folgende Bedingungen stehen dabei zur Verfügung:

LT(Arg1, Arg2)	„True“ wenn Arg1 < Arg2
LE(Arg1, Arg2)	„True“ wenn Arg1 <= Arg2
GT(Arg1, Arg2)	„True“ wenn Arg1 > Arg2
GE(Arg1, Arg2)	„True“ wenn Arg1 >= Arg2

EQ(Arg1, Arg2), „True“ wenn Arg1 = Arg2
 NE(Arg1, Arg2), „True“ wenn Arg1 <> Arg2

Alle Bedingungen können dabei mit sogenannten „Operatoren“ verknüpft werden.

Folgende Operatoren sind dabei möglich:

AND(Arg1, Arg2{, Arg3{, ...}}) „True“ wenn Arg1=Arg2=Arg3
 OR(Arg1, Arg2{, Arg3{, ...}}) „True“ wenn mindestens ein Argument „True“ ist
 XOR(Arg1, Arg2) „True“ wenn Arg1=True und Arg2=False oder umgekehrt
 NOT(Arg1) „True“ wenn Arg1=False

Mathematische Funktionen

Neben den oben erwähnten Befehlen sind auch noch einige mathematische Funktionen verfügbar:

Sum(Arg1, Arg2{, Arg3{, ...}}) =Arg1+Arg2+Arg3+...
 Diff(Arg1, Arg2{, Arg3{, ...}}), =Arg1-Arg2-Arg3...
 Mul(Arg1, Arg2{, Arg3{, ...}}), =Arg1*Arg2*Arg3...
 Div(Arg1, Arg2), =Arg1/Arg2
 Mod(Arg1, Arg2), =Rest von Arg1/Arg2
 Round(Arg1), =Auf ganze Zahl gerundeter Wert von Arg1
 Currency(Arg1) =Rundet eine Zahl auf eine Währung (2 Komma)
 Trunc(Arg1), =Abgerundeter Wert von Arg1
 Frac(Arg1), =Kommastellen von Arg1
 Percent(Arg1, Arg2), =(Arg1/Arg2*100%) wenn Arg2 ungleich 0
 ansonsten wird 0 ausgegeben

Beispiel für die Verwendung logischer und mathematischer Funktionen:
 Zahl(„Rußzahl1“); Zahl(„Rußzahl2“); Zahl(„Rußzahl3“); {Eingabe der drei Meßwerte}
 if(AND(NE(Rußzahl1S,“”),NE(Rußzahl2S,“”),NE(Rußzahl3S,“”))); {Wenn Werte vorh.}
 SchreibeFeld(„Emission.EM_Russzahl“,div(sum(Rußzahl1S,Rußzahl2S,Rußzahl3S),3)); =
 Emission.EM_Russzahl;
 endif;

Datumsformate

Datumsformate können mit der Anweisung „DateTimeToStr“ dargestellt werden.

Bezeichner	Ergebnis
C	Zeigt das Datum in dem mit der globalen Variable ShortDateFormat und die Zeit in dem mit der globalen Variable LongTimeFormat angegebenen Format an. Hat der Nachkommaanteil des Parameters DateTime den Wert Null, wird die Zeit nicht angezeigt.
D	Zeigt den Tag als Zahl ohne führende Null an (1-31).
Dd	Zeigt den Tag als Zahl mit führender Null an (1-31).
Ddd	Zeigt den Tag als Abkürzung an (Son – Sam) und verwendet dabei die von der globalen Variable ShortDayNames angegebenen Strings.
Dddd	Zeigt den Tag als vollständigen Namen an (Sonntag - Samstag) und verwendet dabei die von der globalen Variable LongDayNames angegebenen Strings.
Ddddd	Zeigt das Datum in dem von der globalen Variable ShortDateFormat angegebenen Format an.
Dddddd	Zeigt das Datum in dem von der globalen Variable LongDateFormat angegebenen Format an.
M	Zeigt den Monat als Zahl ohne führende Null an (1-12). Folgt der Bezeichner m direkt nach der Angabe h oder hh, wird statt des Monats die Minute angezeigt.

Mm	Zeigt den Monat als Zahl mit führender Null an (01-12). Folgt der Bezeichner mm direkt nach der Angabe h oder hh, wird statt des Monats die Minute angezeigt.
Mmm	Zeigt den Monat als Abkürzung an (Jan - Dez) und verwendet dabei die von der globalen Variable ShortMonthNames angegebenen Strings.
Mmmm	Zeigt den Monat als vollständigen Namen an (Januar - Dezember) und verwendet dabei die von der globalen Variable LongMonthNames angegebenen Strings.
Yy	Zeigt das Jahr als zweistellige Zahl an (00-99).
Yyyy	Zeigt das Jahr als vierstellige Zahl an (0000-9999).
H	Zeigt die Stunde ohne führende Null an (0-23).
Hh	Zeigt die Stunde mit führender Null an (00-23).
N	Zeigt die Minute ohne führende Null an (0-59).
Nn	Zeigt die Minute mit führender Null an (00-59).
S	Zeigt die Sekunde ohne führende Null an (0-59).
Ss	Zeigt die Sekunde mit führender Null an (00-59).
T	Zeigt die Zeit in dem von der globalen Variable ShortTimeFormat angegebenen Format an.
Tt	Zeigt die Zeit in dem von der globalen Variable LongTimeFormat angegebenen Format an.
Am/pm	Verwendet das 12-Stunden-Format für den führenden Bezeichner h oder hh und zeigt 'am' für Zeitangaben vor 12 Uhr mittags oder 'pm' für Zeitangaben nach 12 Uhr mittags an. Der Bezeichner am/pm kann in Groß-, Klein- oder gemischter Schreibweise angegeben werden und wird dementsprechend angezeigt.
a/p	Verwendet das 12-Stunden-Format für den führenden Bezeichner h oder hh und zeigt 'a' für Zeitangaben vor 12 Uhr mittags oder 'p' für Zeitangaben nach 12 Uhr mittags an. Der Bezeichner a/p kann in Groß-, Klein- oder gemischter Schreibweise angegeben werden und wird dementsprechend angezeigt.
Ampm	Verwendet das 12-Stunden-Format für den führenden Bezeichner h oder hh und zeigt für Zeitangaben vor 12 Uhr mittags den Inhalt der globalen Variablen TimeAMString oder für Zeitangaben nach 12 Uhr mittags den Inhalt der globalen Variable TimePMString an.
/	Verwendet das mit der globalen Variable DateSeparator angegebene Datumstrennzeichen.
:	Verwendet das mit der globalen Variablen TimeSeparator angegebene Zeittrennzeichen.
'xx'/'"xx"	In halbe oder ganze Anführungszeichen eingeschlossene Zeichen wirken sich nicht auf die Formatierung aus und werden wie eingegeben angezeigt.

Tabelle 1: Datumsformate

Die Formatbezeichner können beliebig in Groß- oder Kleinbuchstaben angegeben werden, da sich die Schreibweise nicht auf das Ergebnis der Formatierung auswirkt.

Wird im Parameter Format ein leerer String übergeben, werden die Datums- und Zeitwerte entsprechend dem Formatbezeichner 'c' formatiert.

Beispiel: dd.mmmm.yyyy gibt aus: 03. Februar 2000

Häufige Fehler

Es werden keine Leerzeichen erstellt, Texte stehen alle hintereinander

Beim Markieren der Anweisung, bzw. beim Unsichtbar-Setzen wird meist (oft auch von Word automatisch) das abschließende Zeilenvorschubszeichen (¶) auch als verborgen gesetzt. Dadurch wird es von Winchim ignoriert, dh. einfach übergangen. Als Resultat liefert dann die folgende Sequenz das angegebene Ergebnis:

Kunde..Anrede;
Kunde..Titelname;
Kunde..Adresse;
Kunde..PLZ; Kunde..Ort;

liefert das Ergebnis:
HerrMax MustermannHauptstraße 18010Graz

Lösung:

Das Zeilenvorschubszeichen markieren und wieder auf Sichtbar setzen durch „Strg+H“.