

ANLEITUNG

Version 2026

Die Informationen in diesem Handbuch haben keinen Vertragscharakter und können ohne vorherige Ankündigung geändert werden.

Die in diesem Handbuch beschriebene Software wird im Rahmen einer Lizenzvereinbarung verkauft. Die Software darf nur gemäß den Bedingungen der Vereinbarung verwendet, kopiert oder reproduziert werden.

Dieses Handbuch darf ohne schriftliche Genehmigung von TEKLYNX Corporation SAS weder ganz noch teilweise in irgendeiner Form, egal mit welchen Mitteln oder zu welchen Zwecken, außer zur eigenen Verwendung des Käufers kopiert, reproduziert oder übertragen werden.

©2026 TEKLYNX Corporation SAS,

Alle Rechte vorbehalten.

Inhaltsverzeichnis

Informationen zu diesem Handbuch 6

Typografische Konventionen 6

Informationen zu Ihrem Produkt 6

Verbindung zu Datenbanken 7

Einige Hinweise 7

Installation der ODBCDatenquelle 8

Importieren der Daten 8

Erstellung von variablen Objekten 9

Active Query Builder 10

Erste Schritte 10

Hinzufügen von Objekten zu einer Abfrage 11

Bearbeiten von Objekteigenschaften 12

Verknüpfen von Tabellen 12

Sortieren von Ausgabefeldern 13

Definieren von Kriterien 14

Das Gitter Abfrageergebnis 15

Database Manager 16

Die Schnittstelle Datenbankstruktur 16

Eine Datenbank in der Liste der Verbindungen auswählen 16

Eine Tabelle in einer Datenbank auswählen 16

Die Schnittstelle Datenbankstruktur des Database Manager muss geöffnet sein. 17

Tabelle in der aktiven Datenbank löschen 17

Zeigen / Die Daten der aktiven Tabelle ausblenden 17

Ein Feld als Schlüsselfeld definieren 18

Den Inhaltstyp eines Felds definieren 18

Die maximale Größe eines Feldes bestimmen ... 18
Ein leeres Feld autorisieren ... 19
Die Schnittstelle Datenbank bearbeiten ... 19
Einen Datensatz in Funktion des Inhalts eines Feldes suchen ... 19
Alle identischen Datensätze auswählen ... 20
Identischen Datensatz auswählen ... 20
Neuen Datensatz erstellen ... 20
Einen Datensatz duplizieren ... 20
Einen Datensatz ändern ... 20
Eine Datensätze löschen ... 20
Die Schnittstelle Datenbankabfragen ... 21
Abfrage hinzufügen ... 21
Ein oder mehrere Felder auswählen / abwählen ... 21
Die Reihenfolge der ausgewählten Felder ändern ... 21
Einen Filter mit Hilfe der vorbestimmten Daten erstellen ... 22
Einen logischen Operatoren auf mehrere Filter anwenden ... 22
Die Liste der Filter sortieren ... 22
Einen Filter löschen ... 22
Einen Filter in der SQL-Sprache ändern ... 23
Die Schnittstelle Drucken ... 23
Das zu druckende Dokument auswählen ... 23
Ein vorhandenes Etikettenmodell auswählen ... 24
Den zu verwendenden Drucker auswählen ... 24
Formel ... 25
Die Formel-Datenquelle ... 25
Die Funktionen ... 25
Die Berechnungsfunktionen der Prüfzeichen ... 26
Die Konvertierungsfunktionen ... 28

ATA 2000-Konvertierungsfunktionen ... 35
Die Datums- und Uhrzeitfunktionen ... 37
Die logischen Funktionen ... 45
Die mathematischen Funktionen ... 46
Die Funktion in den Texten ... 49
Festlegen der Eigenschaften der Formeldatenquellen ... 56
Praktische Übung : Berechnung des spezifischen "Modulo" ... 56
Zur Berechnung des Gewichts ... 57
Um das Ergebnis zur Berechnung des Gewichts hinzuzufügen: ... 57
Um das Prüfzeichen zu berechnen: ... 58
Zur Berechnung des zu codierenden Parameters: ... 58
Zur Erstellung des Strichcode ... 58
Netzwerkpaket ... 59
Funktionsbeschreibung ... 59
Installationsvorgang ... 59
Vorbemerkung ... 59
Beschreibung von Network License Manager ... 60
Installieren von Network License Manager ... 60
Konfiguration ... 61
Konfigurieren des Benutzer-Manager ... 61
Starten des License Service ... 62
Starten von Service Controller ... 62
Installieren der Software auf den Stationen ... 62
Installieren der Software auf einer Station ... 62

Informationen zu diesem Handbuch

Typografische Konventionen

In diesem Handbuch wird mittels der folgenden Konventionen zwischen verschiedenen Informationsarten unterschieden:

- Begriffe, die von der Benutzeroberfläche selber stammen, wie z. B. Befehle, werden **fett** dargestellt.
- Tasten werden in Kapitälchen angezeigt. Beispiel: Drücken Sie die Taste ALT.
- Nummerierte Listen weisen darauf hin, dass ein Verfahren folgt.
- Wenn die Konjunktion – oder – neben einem Absatz erscheint, bedeutet dies, dass es für die Ausführung einer bestimmten Aufgabe noch ein anderes mögliches Verfahren gibt.
- Wenn ein Menübefehl Untermenüs enthält, werden der Menüname, gefolgt vom auszuwählenden Befehl, in Fettdruck angezeigt. So bedeutet z. B. Gehen Sie zu **Datei > Open**, dass Sie zunächst das Menü **Datei** und anschließend den Befehl **Open** auswählen müssen.

Informationen zu Ihrem Produkt

Einige der in diesem Handbuch beschriebenen Funktionen stehen in Ihrem Produkt möglicherweise nicht zur Verfügung.

Eine vollständige Liste der genauen Funktionen, die in Ihrer Software zur Verfügung stehen, finden Sie in dem mit dem Produkt gelieferten Datenblatt.

Verbindung zu Datenbanken

Einige Hinweise

In diesem Kapitel wird gezeigt, wie leistungsfähig Ihre Software ist. Hier wird ein Etikett (der Behälter) mit einer Datenbank (dem Inhalt) verknüpft. Hierfür werden verschiedene **ODBC**-Verbindungen (**Open Data Base Connectivity**) und **OLE DB** (**Object Linking and Embedding Database**) verwendet.

Über die Datenbank können Daten gespeichert werden. Alle Daten werden in Form von **Tabellen** mit zwei Dimensionen gespeichert, die als **Relation** bezeichnet werden. Jede Zeile der Tabelle wird als **Datensatz** bezeichnet. Mit einem Datensatz wird ein Objekt verwaltet, dessen Eigenschaften in den verschiedenen Spalten der Tabelle in Form von **Feldern** enthalten sind.

Eine Datenbank kann mehrere Tabellen enthalten. Um die verschiedenen Tabellen einer Datenbank miteinander zu verbinden, können **Verknüpfungen** (Links) verwendet werden. Weiter unten in diesem Kapitel wird anhand eines konkreten Beispiels gezeigt, wie Verknüpfungen erstellt werden.

ODBC

Es handelt sich um einen Verbindungsstandard, über den der Zugriff auf Datenbanken möglich ist. Über ODBC kann eine Anwendung einfach verbunden werden, wie z. B. die Etikettierungssoftware mit einer bestimmten Anzahl von Datenbanken.

Die Software stellt mehrere ODBC-Treiber bereit, so dass ein Zugriff auf die aktuellsten Datenbanken möglich ist.

Unten werden einige Treiber aufgeführt:

- Microsoft Access Driver (*.mdb)
- Microsoft Excel Driver (*.xls)
- Microsoft FoxPro Driver (*.dbf)
- ...

OLE DB

Es handelt sich um einen Verbindungsstandard, über den der Zugriff auf alle Datenbankstandards und auf die in den Messaging-Systemen gespeicherten Daten möglich ist.

Installation der ODBCDatenquelle

Im ersten Beispiel wird eine Datenquelle installiert, die sich auf eine Access-Datenbank bezieht.

Zum Installieren der Datenquelle:

1. Wählen Sie **Datenquelle > Datenbank > Erstellen / Abfrage bearbeiten**.
   Das Dialogfenster Definition einer ODBC-Abfrage wird angezeigt.
2. Klicken Sie auf  rechts von der Liste Wählen Sie eine Datenbank aus.

   Das Dialogfenster für den Administrator der ODBCDatenquelle wird angezeigt.

   ODBC ist eine Komponente, die völlig unabhängig von Ihrer Etikettierungssoftware ist. Sie können den Administrator der ODBC-Datenquelle auch über die Windows-Systemanzeige aufrufen Wählen Sie Start Einstellungen Systemsteuerung und klicken Sie auf folgendes Symbol: ODBC.
3. Klicken Sie über das Register DSN-Benutzer auf Hinzufügen... , wählen Sie den Treiber Microsoft Access Driver (*.mdb) aus und klicken Sie anschließend auf Beenden.
   Das Dialogfenster ODBC-Installation für Microsoft Access wird angezeigt.
4. Geben Sie einen Namen in das Feld Name der Datenquelle ein, z. B. Classical music source und fügen Sie gegebenenfalls eine kurze Beschreibung hinzu.

   Es wurde bereits beschrieben, dass eine Datenquelle mit einer Datenbank verbunden ist. Im folgenden Schritt wird die für unser Beispiel verwendete Datenbank ausgewählt, d. h. Volmusic.mdb.

Importieren der Daten

Nach Installation der Datenquelle können die in der Datenbank gespeicherten Daten importiert werden, so dass sie in das Etikett eingefügt werden können.

Zum Importieren der Daten:

1. Wenn das Dialogfenster **Definition einer ODBC-Abfrage** nicht angezeigt wird **Zusammenführen Erstellen/Abfrage bearbeiten**.
2. Wählen Sie über die Liste **Wählen Sie eine Datenquelle** aus Classical music source aus und wählen Sie anschließend die **Tabelle** volumes aus.

3. Klicken Sie auf **Alles auswählen** und bestätigen Sie.

   Über diese Aktion können automatisch alle **Datenbankvariablen** über die Felder der Datenbank erstellt werden.

   Die Variablen werden unter der Verzweigung **Datenbank** des Registers Datenquellen.
4. Das Dokument speichern.

   Die Anwendung fragt, ob Sie die Abfrage speichern möchten.
5. Klicken Sie auf Ja, wählen Sie .InstallDir\Samples\Forms\Tutorial aus und geben Sie Volmusic.csq als Name der Abfrage ein.

Erstellung von variablen Objekten

Nun müssen die so erstellten Variablen einfach einem Textobjekt (Text, Strichcode usw.) zugeordnet werden, damit sie in das Etikett eingefügt werden können. In diesem Beispiel werden variable Objekte vom Typ Text erstellt.

1. Wählen Sie die erstellte Variablen in der Datenbank Niederlassung in den Datenquellen aufgelistet, dann per Drag & Drop in den Arbeitsbereich.
2. Wählen Sie Text, in dem Kontextmenü.

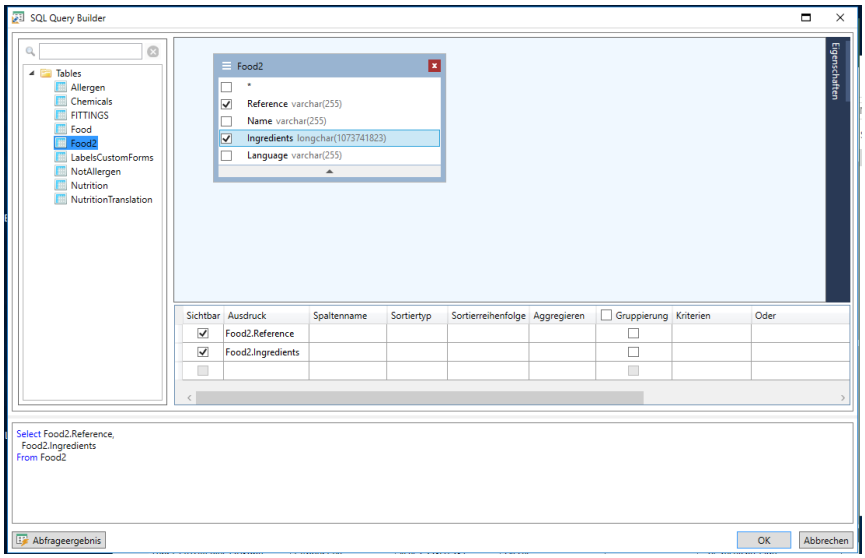
Active Query Builder

Erste Schritte

Die nachfolgende Abbildung zeigt die Active Query Builder-Oberfläche nach dem Start.

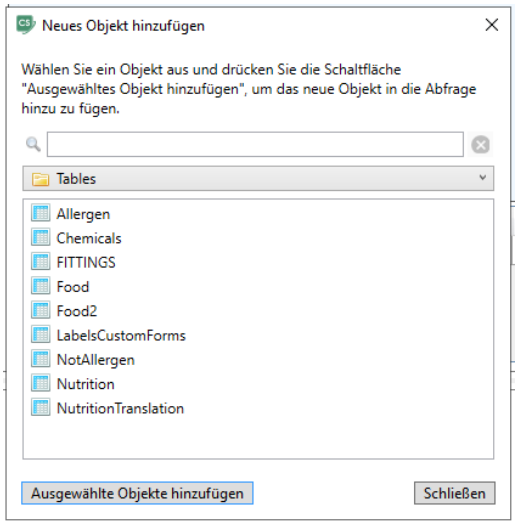
Das Hauptfenster gliedert sich in die folgenden Bereiche:

- Der **Bereich für die Abfrageerstellung** ist der Hauptbereich, in dem die Darstellung der Abfrage angezeigt wird. In diesem Bereich können Objekte und abgeleitete Tabellen der Quelldatenbank definiert werden. Ferner können Sie Verknüpfungen zwischen diesen Elementen erstellen und Eigenschaften von Tabellen und Verknüpfungen konfigurieren.
- Der **Spaltenbereich** befindet sich unterhalb des Bereichs für die Abfrageerstellung. In diesem Bereich werden alle erforderlichen Vorgänge für Abfrageausgabespalten und -ausdrücke vorgenommen. Sie können Feldaliasnamen, Sortierungen, Gruppierungen und Kriterien definieren.
- Auf der linken Seite befindet sich der **Bereich Tabellenansicht.** Hier können Sie Ihre Abfrage durchsuchen und schnell einen beliebigen Teil davon finden. Verwenden Sie das **Suchfeld** am oberen Rand des Bereichs, um Ihre Suche zu spezifizieren.
- Über die Seitensteuerung oberhalb des Bereichs für die Abfrageerstellung können Sie zwischen der Hauptabfrage und Unterabfragen wechseln.
- Das kleine Quadrat mit dem Buchstaben Q oben rechts im Bereich für die Abfrageerstellung ist das Steuerelement zur Handhabung von Union-Unterabfragen. Über dieses Steuerelement können neue Union-Unterabfragen hinzugefügt und die erforderlichen Operationen mit diese Abfragen ausgeführt werden.



Hinzufügen von Objekten zu einer Abfrage

Zum Hinzufügen von Objekten zu einer Abfrage klicken Sie mit der rechten Maustaste auf den Bereich für die Abfrageerstellung und wählen in der Dropdown-Liste den Eintrag **Objekt hinzufügen**.



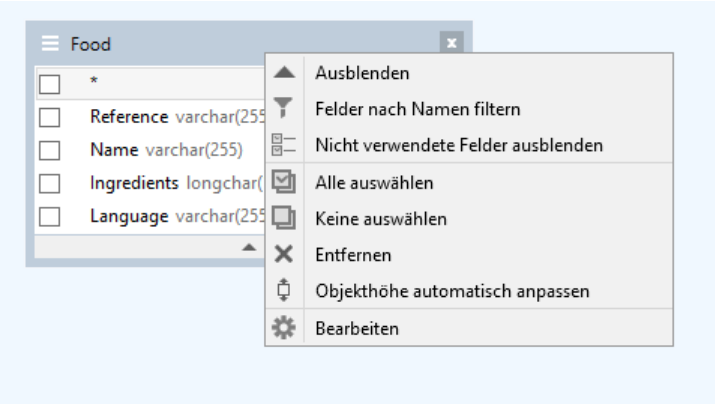
Im Fenster "Neues Objekt hinzufügen" können Sie mehrere Objekte auf einmal hinzufügen. Tabellenelemente werden im Kombinationsfeld aufgelistet. Der Filter **Alle Objekte anzeigen** zeigt alle Objekte an, die zur Auswahl stehen. Das **Suchfeld** am oberen Rand des Fensters wird verwendet, um nach einem benötigten Objekt zu suchen. Sie können ein oder mehrere Objekte auswählen, indem Sie die **STRG-Taste** gedrückt halten und dann die Schaltfläche Ausgewählte Objekte hinzufügen drücken, um diese **Objekte zur Abfrage hinzuzufügen**. Sie können diesen Vorgang mehrmals wiederholen. Nachdem

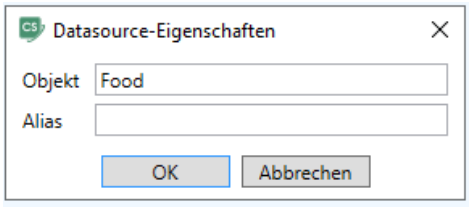
Sie das Hinzufügen von Objekten beendet haben, klicken Sie auf die Schaltfläche **Schließen** um dieses Fenster auszublenden.

Um ein Objekt aus der Abfrage zu entfernen, klicken Sie auf die Schaltfläche **Schließen** in der Kopfzeile des Objekts.

Bearbeiten von Objekteigenschaften

Sie können die Eigenschaften eines Objekts, das zur Abfrage hinzugefügt wurde, ändern. Klicken Sie dazu mit der rechten Maustaste auf das Objekt und wählen Sie im Dropdown-Menü den Eintrag **Editieren...** oder doppelklicken Sie auf die Titelleiste des Objekts.



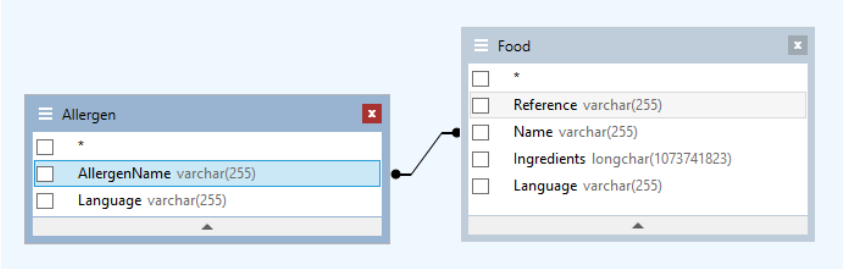


Das Dialogfeld **Datenquellen Eigenschaften** variiert möglicherweise auf unterschiedlichen Servern, die Aliaseigenschaft ist jedoch bei allen Datenbankservern identisch.

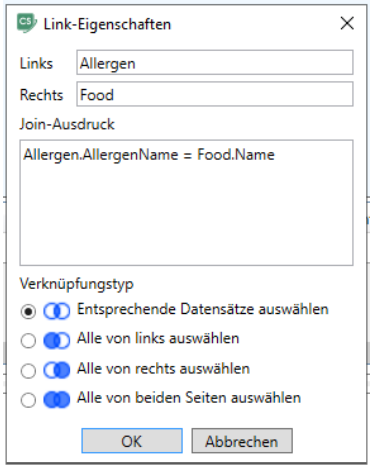
Verknüpfen von Tabellen

Um eine Verknüpfung zwischen zwei Objekten zu erstellen, wählen Sie das Feld aus, das mit dem Feld eines anderen Objekts verknüpft werden soll, und ziehen dieses Feld auf das entsprechende Feld des

anderen Objekts. Nach Abschluss dieses Vorgangs wird die Verbindung der verknüpften Felder durch eine Linie dargestellt.



Der standardmäßig erstellte Verknüpfungstyp lautet INNER JOIN, sodass ausschließlich übereinstimmende Datensätze beider Tabellen im resultierenden Dataset enthalten sind. Um andere Typen von Verknüpfungen zu definieren, klicken Sie mit der rechten Maustaste auf die Verknüpfung und wählen im Dropdown-Menü den Eintrag **Bearbeiten...**. Alternativ doppelklicken Sie auf die Verknüpfung, um das Dialogfeld **Link Eigenschaften** zu öffnen. In diesem Dialogfeld können Sie den Verknüpfungstyp und andere Verknüpfungseigenschaften festlegen.



Um eine Verknüpfung zwischen Objekten zu entfernen, klicken Sie mit der rechten Maustaste auf die Verknüpfungslinie und wählen Sie im Dropdown-Menü die Option **Entfernen** aus, oder wählen Sie die Verknüpfungslinie aus und drücken Sie die Schaltfläche **Löschen**.

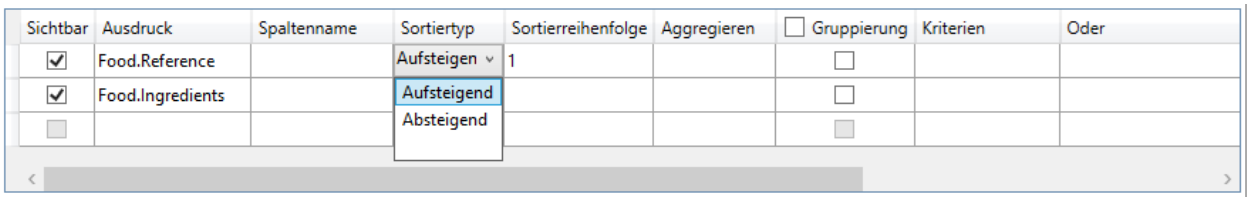
Sortieren von Ausgabefeldern

Um die Sortierung von Ausgabeabfragefeldern zu aktivieren, verwenden Sie im **Spaltenbereich** die Spalten **Sortierungstyp** und **Sortierreihenfolge**.

Über die Spalte **Sortierungstyp** kann die Sortierungsart für die Felder angegeben werden – zur Auswahl stehen **Aufsteigend** oder **Absteigend**.

Wenn mehrere Felder sortiert werden, ermöglicht die Spalte **Sortierrungsreihenfolge** das Festlegen einer Reihenfolge, mit der die Felder sortiert werden.

Um die Sortierung nach einem Feld zu deaktivieren, löschen Sie die Spalte **Sortierungstyp** für dieses Feld.



Definieren von Kriterien

Zum Definieren von Kriterien für den im **Spaltenbereich** aufgeführten Ausdruck verwenden Sie die Spalte **Kriterien**.

Lassen Sie den Ausdruck selbst aus, wenn Sie das Kritierium schreiben. Wenn in der Abfrage das folgende Kriterium enthalten sein soll

> **WHERE** (field >= 10) **AND** (field <= 20)
> schreiben Sie
> >= 10 **AND** <= 20 in die Spalte **Kriterien**.

Sie können auch die **Auswahlliste** in der Spalte **Kriterien** verwenden, um gängige Ausdrücke und Operatoren in das Eingabefeld einzufügen. Für die Erstellung erweiterter Kriterien verwenden Sie die Schaltfläche **Ausdruckseditor** in der Spalte **Kriterien**. Das Dialogfeld **Ausdruckseditor** wird angezeigt.

Über die Spalten **Oder...** können für einen einzelnen Ausdruck mehrere Kriterien angegeben werden. Diese Kriterien werden in der Abfrage über den Operator **OR** verkettet.

Das Gitter Abfrageergebnis

Um auf das Gitter **Abfrageergebnis** zuzugreifen, klicken Sie auf die Schaltfläche  über das Dialogfeld **Definition einer Abfrage,** in der Symbolleiste **Navigation in Datenbank Zusammenfügen** oder über das Menü **Datenquelle > Datenbanken**.

Dieses Gitter ermöglicht es, das Ergebnis einer Abfrage oder Suche eines bestimmten Begriffs und all seiner Exemplare anzuzeigen und die entsprechenden Etiketten zu drucken.

Hinweis:Der "**<Binary data>**"-Tag wird im Abfrageergebnisdialog für Datenbankfelder angezeigt, die den Datentyp BLOB enthalten (z. B. Bilddateien).

Das Gitter **Ergebnis einer Abfrage** umfasst:

- **Suchfunktionen**

  Suchfeld, über das das Feld eingegeben werden kann, in dem die Suche ausgeführt wird

  Suchwert, über das der gesuchte Wert eingegeben werden kann

  Suche des Werts an jeder beliebigen Stelle im Feld oder am Anfang des Felds .

- **der Funktionen zum Verschieben in den Suchergebnissen**

  Erster Datensatz 

  Vorheriger Datensatz 

  Nächster Datensatz 

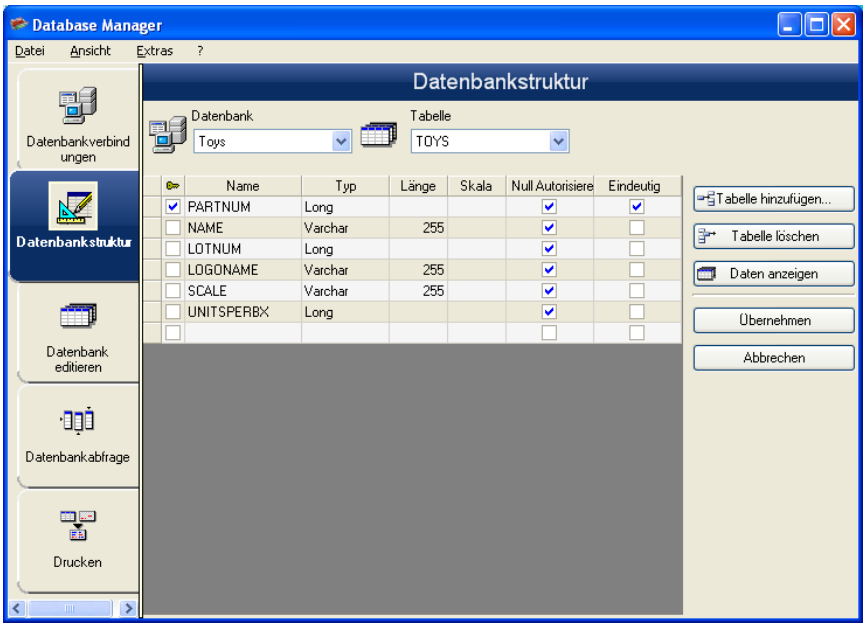
  Letzter Datensatz 

- **Das Ergebnisgitter**
  Zeigt die Exemplare des Abfrageergebnisses an.

- **Anforderung**
  Wiederausführung der Abfrage und Update der Daten.

Database Manager

Die Schnittstelle Datenbankstruktur



Schnittstelle, über die die Struktur der Datenbankdatei verwaltet wird: Hinzufügen / Ändern / Löschen von Tabellen, Feldern...

Eine Datenbank in der Liste der Verbindungen auswählen

Die Schnittstelle Datenbankstruktur des Database Manager muss geöffnet sein.

1. In der Dropdown-Liste **Datenbank** anklicken.
2. Auf den gewünschten Wert klicken



Eine Tabelle in einer Datenbank auswählen

Die SchnittstelleDatenbankstruktur des Database Manager muss geöffnet sein.

1. In der Dropdown-Liste **Tabelle** anklicken.
2. Auf den gewünschten Wert klicken

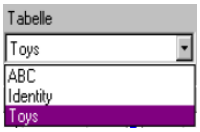
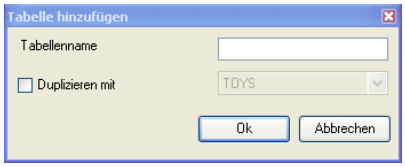


Tabelle in der aktiven Datenbank hinzufügen

Die Schnittstelle Datenbankstruktur des Database Manager muss geöffnet sein.

1. Auf die Schaltfläche **Tabelle hinzufügen** klicken.
2. Den Namen der neuen Tabelle eingeben.
3. Auf die Schaltfläche **OK** klicken



Sie können auch die Tabellenstruktur in eine bereits vorhandene Tabelle in der ausgewählten Datenbank kopieren. Dazu:

1. Das Kästchen **Duplizieren mit** abhaken.
2. In die Dropdown-Liste klicken.
3. Auf den gewünschten Wert klicken.
4. Auf die Schaltfläche **OK** klicken.

Tabelle in der aktiven Datenbank löschen

Die Schnittstelle Datenbankstruktur des Database Manager muss geöffnet sein.

1. In der Dropdown-Liste **Tabelle** anklicken.
2. Auf den gewünschten Wert klicken.
3. Auf die Schaltfläche **Tabelle löschen** klicken.

Zeigen / Die Daten der aktiven Tabelle ausblenden

Die Schnittstelle Datenbankstruktur des Database Manager muss geöffnet sein.

1. Auf die Schaltfläche **Daten anzeigen** klicken.

Ein Feld als Schlüsselfeld definieren

Die Schnittstelle Datenbankstruktur des Database Manager muss geöffnet sein.

1. Das Kästchen im gewünschten Feld abhaken.

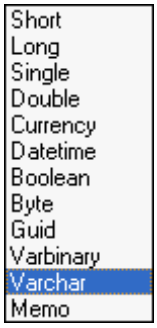


2. Auf die Schaltfläche **Übernehmen** klicken.

Den Inhaltstyp eines Felds definieren

Die Schnittstelle Datenbankstruktur des Database Manager muss geöffnet sein.

1. Das Kästchen **Typ** des gewünschten Felds anklicken.
2. Auf die Schaltfläche der Dropdown-Liste klicken.
3. Auf den gewünschten Wert klicken.



4. Auf die Schaltfläche **Übernehmen** klicken.

Die maximale Größe eines Feldes bestimmen

Die Schnittstelle Datenbankstruktur des Database Manager muss geöffnet sein.

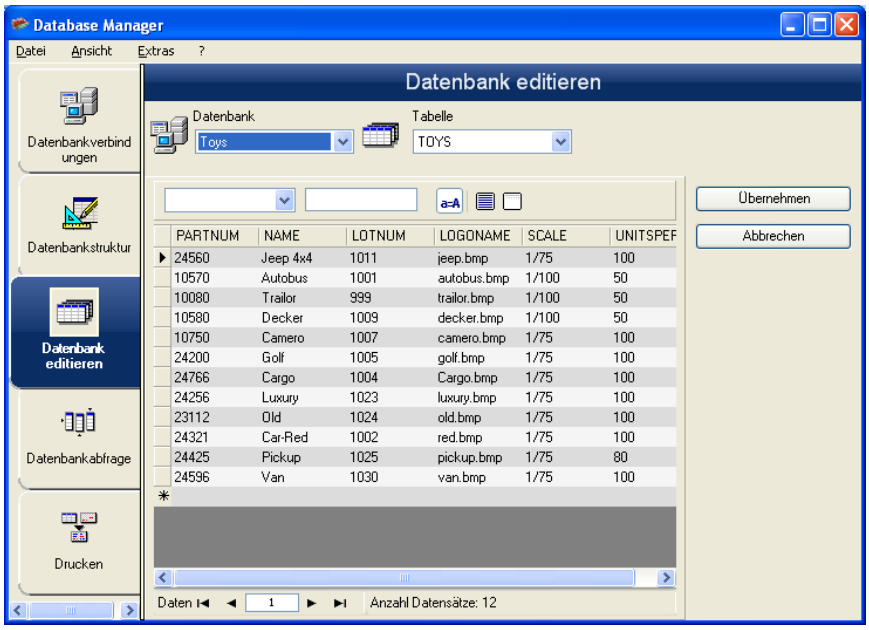
1. Das Kästchen **Länge** des gewünschten Felds anklicken.
2. Den gewünschten Wert eingeben.
3. Auf die Schaltfläche **Übernehmen** klicken.

Ein leeres Feld autorisieren

Die SchnittstelleDatenbankstruktur des Database Manager muss geöffnet sein.

1. Das Kästchen **Null erlauben** des gewünschten Felds abhaken.
2. Auf die Schaltfläche **Übernehmen** klicken.

Die Schnittstelle Datenbank bearbeiten



Schnittstelle, über die der Inhalt der Datenbankdatei verwaltet wird: Hinzufügen / Ändern / Löschen von Daten.

Diese Handlungen sind in Funktion des Datenbanktyps. Somit können die Datensätze der Exceldateien nicht verändert werden.

Einen Datensatz in Funktion des Inhalts eines Feldes suchen

1. Auf die Schaltfläche der Dropdown-Liste klicken.
2. Auf den gewünschten Wert klicken.
3. Auf das Eingabefeld klicken.
4. Den gewünschten Wert in das Eingabefeld eingeben.

Alle identischen Datensätze auswählen

1. Auf die Schaltfläche der Dropdown-Liste klicken.
2. Auf den gewünschten Wert klicken.
3. Auf das Eingabefeld klicken.
4. Den gewünschten Wert in das Eingabefeld eingeben.
5. Auf Schaltfläche **Alles auswählen** klicken Anmerkung.

Identischen Datensatz auswählen

1. Zum Auswählen eines Datensatzes: Erste, Zurück, Nächste, Weiter, auf Schaltfeld 1, 2, 3 oder 4 der Suchfunktion klicken.



Neuen Datensatz erstellen

1. In ein Feld der Eingabezeile klicken, die durch ein Sternzeichen gekennzeichnet ist.
2. Die gewünschten Werte in die entsprechenden Felder eingeben.
3. Auf die Schaltfläche **Übernehmen** klicken.

Einen Datensatz duplizieren

1. Auf eine Zeile oder Zeilenreihe klicken oder doppelklicken, um die ausgewählten Datensätze zu markieren.
2. Mit Rechtsklick das Kontextmenü aufrufen.
3. Markierte(n) Datensätze (-satz) duplizieren wählen.

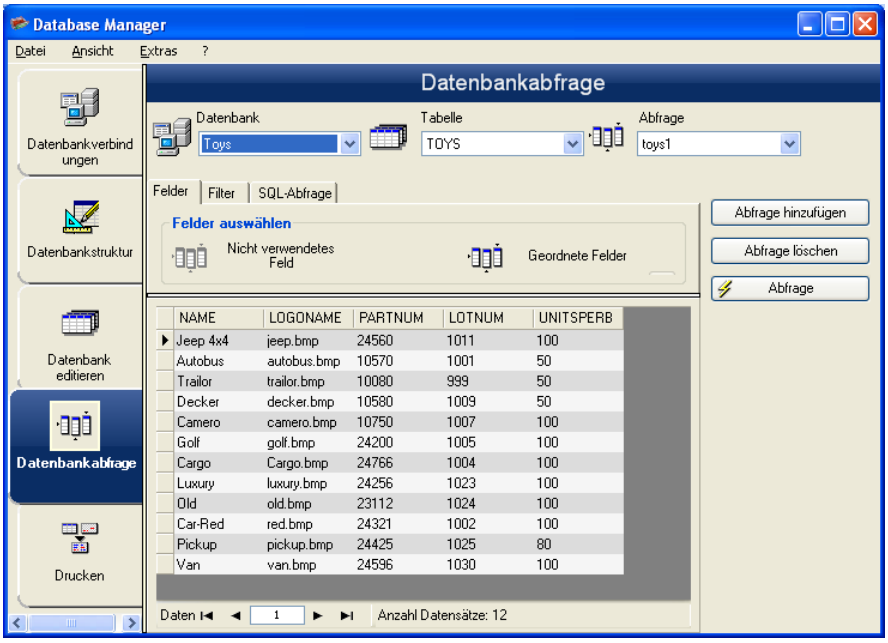
Einen Datensatz ändern

1. Auf den zu ändernden Wert klicken.
2. Den gewünschten Wert eingeben.
3. Auf die Schaltfläche **Übernehmen** klicken.

Eine Datensätze löschen

1. Auf das Kästchen der Datensatzmarke des gewünschten Felds klicken.
2. Auf das Kästchen der Datensatzmarke des gewünschten Felds rechtsklicken.
3. Auf das Kontextmenü **Ausgewählten Datensatz löschen** klicken.

Die Schnittstelle Datenbankabfragen



Schnittstelle, wo verschiedene Filter erstellt oder eingesetzt werden.

Abfrage hinzufügen

1. Auf die Schaltfläche **Abfrage hinzufügen** klicken.
2. Einen Namen für die Abfrage eingeben.
3. Auf die Schaltfläche **OK** klicken

Ein oder mehrere Felder auswählen / abwählen

Die Karteikarte **Felder** der Schnittstelle Datenbankabfragen des Database Manager muss geöffnet sein.

1. Um ein oder mehrere Felder aus- oder abzuwählen, auf die Schaltfläche 1, 2, 3 oder 4 klicken.



2. Auf die Schaltfläche **Abfrage** klicken.

Die Reihenfolge der ausgewählten Felder ändern

Die Karteikarte **Felder** der Schnittstelle Datenbankabfragen des Database Manager muss geöffnet sein.

1. Im Fenster **Sortierte Felder** auf das gewünschte Feld klicken.

2. Bis zum Erhalten des gewünschten Werts auf die Schaltfläche **Pfeil auf** oder **Pfeil ab** klicken.
3. Auf die Schaltfläche **Abfrage** klicken.

Einen Filter mit Hilfe der vorbestimmten Daten erstellen

1. Auf die Schaltfläche **Zur Zeile hinzufügen** klicken.
2. Auf das Feld **Feld** klicken.
3. Auf die Schaltfläche der Dropdown-Liste klicken.
4. Auf den gewünschten Wert klicken.
5. Auf das Feld **Operator** klicken.
6. Auf die Schaltfläche der Dropdown-Liste klicken.
7. Auf den gewünschten Wert klicken.
8. Auf das Feld **Wert** klicken.
9. Den gewünschten Wert eingeben.
10. Auf die Schaltfläche **Abfrage** klicken.

Einen logischen Operatoren auf mehrere Filter anwenden

Die Karteikarte ”r;Filter" der Schnittstelle Abfragen der Datenbank des Database Manager muss geöffnet sein. Mehrere Filter müssen erstellt werden.

1. Auf das Feld **Logical** klicken.
2. Auf die Schaltfläche der Dropdown-Liste klicken.
3. Auf den gewünschten Wert klicken.
4. Auf die Schaltfläche **Abfrage** klicken, um die Änderungen anzuwenden und anzuzeigen.

Die Liste der Filter sortieren

Die Karteikarte **Filter** der Schnittstelle Abfragen der Datenbankdes Database Manager muss geöffnet sein. Mehrere Filter müssen erstellt werden.

1. Auf das Kästchen der Marke des gewünschten Felds klicken.
2. Bis zum Erhalten des gewünschten Werts auf die Schaltfläche **Pfeil auf** oder **Pfeil ab** klicken.

Einen Filter löschen

Die Karteikarte **Filter** der Schnittstelle Abfragen der Datenbankdes Database Manager muss geöffnet sein. Mindestens ein Filter muss erstellt werden.

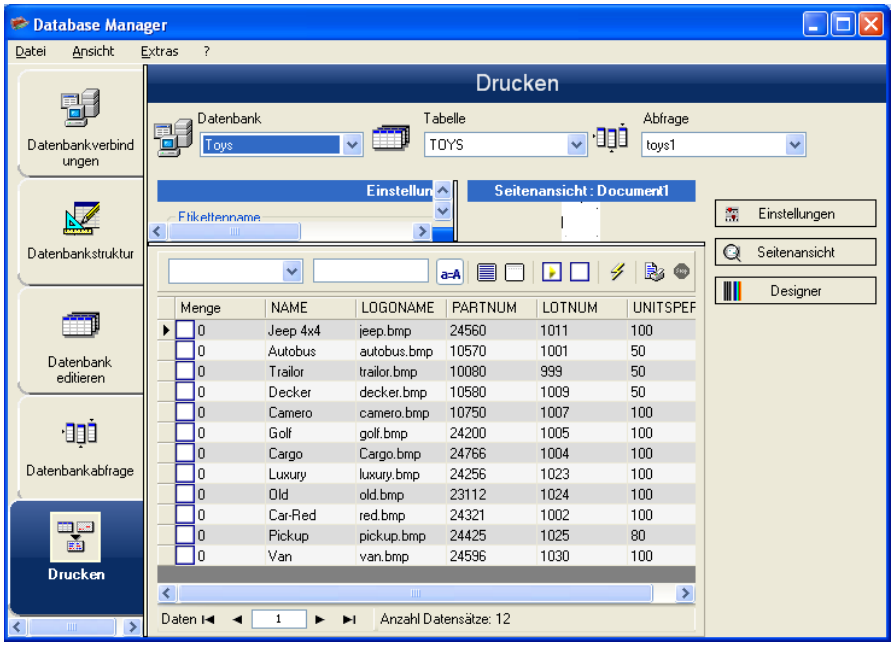
1. Auf das Kästchen der Marke des gewünschten Felds klicken.
2. Auf die Schaltfläche **Zeile löschen** klicken.

Einen Filter in der SQL-Sprache ändern

Die Karteikarte "SQL-Abfrage" der Schnittstelle Abfragen der Datenbankdes Database Manager muss geöffnet sein. Mindestens ein Filter muss erstellt werden.

1. Das Kästchen **Die Abfrage in der SQL-Sprache ändern** abhaken, um die Abfrage in der SQL-Sprache zu aktivieren und sie manuell ändern zu können.
2. Auf die Schaltfläche **Abfrage** klicken.

Die Schnittstelle Drucken



Über diese Schnittstelle werden die Druckdatei, der dedizierte Drucker und die verschiedenen Druckparameter ausgewählt, bevor der Druck aktiviert wird.

Das zu druckende Dokument auswählen

1. Auf die Schaltfläche **Assistent für Etikettenerstellung** klicken.
2. Die Anleitungen des Assistenten befolgen.

Anmerkung: Die Erstellung eines Etiketts in Verbindung mit der Datenbank ermöglicht die genaue Definition der für die Einrichtung der verschiedenen Felder der Datenbank erforderlichen Elemente.

Ein vorhandenes Etikettenmodell auswählen

1. Auf die Schaltfläche **Ein vorhandenes Dokument** öffnen klicken.
2. Eine .lab-Datei auswählen.
3. Auf die Schaltfläche **OK** klicken.

Anmerkung: Die Radiobuttons **Feld** in den Optionsgruppen **Etikettenname** und **Druckername** ermöglichen die Auswahl des gewünschten Etiketts oder Druckers, wenn diese in einem Feld der aktiven Datenbank definiert sind.

Den zu verwendenden Drucker auswählen

Die Schnittstelle Drucken des Database Manager muss geöffnet sein. Ein Etikettenmodell muss ausgewählt werden.

1. Auf die Schaltfläche **Einen Drucker hinzufügen** oder löschen klicken
2. Den gewünschten Drucker auswählen.
3. Auf die Schaltfläche **Bestätigen** klicken.

Anmerkung: Der zuletzt verwendete Drucker ist standardmäßig ausgewählt.

Formel

Die Formel-Datenquelle

Die Datenquelle **Formel** enthält eine Liste der von Ihnen erstellten Datenquellen. Diese Datenquellen werden von Kombinationen aus Operatoren, Konstanten, Datenquellen, Steuervariablen, Formeln und Funktionen gespeist. Die Daten können nummerisch oder alphanumerisch sein.

Um eine Berechnung in einem Dokument durchzuführen, müssen Sie zuerst eine Datenquelle des Typs Formel erstellen.

Diese Datenquelle verfügt über ein spezielles Dialogfeld, in dem Sie die erforderliche(n) Funktion(en) für eine bestimmte Formel definieren können.

Die Funktionen

Hinweis: Die Regionaleinstellungen des Windows-Systems können die Ausgabe einiger Formeln beeinflussen.

Die Funktionen sind vordefinierte Formeln, die Berechnungen mit speziellen Werten ausführen, die als Argumente bezeichnet werden und eine bestimmte Ordnung haben, die als Syntax bezeichnet wird.

Die Funktionen werden für die Rückgabe eines Werts vom Typ numerisch, Zeichenfolge oder logisch verwendet, der das Ergebnis einer Berechnung oder Operation ist.

Es gibt es Gruppen von Funktionen in der Definition der Formeln:

- Funktionen für die Berechnung der Prüfzeichen (check digits)
- Konvertierungsfunktionen
- Datums- und Uhrzeitfunktionen
- Logische Funktionen
- Mathematische Funktionen
- Funktionen bei den Zeichenfolgen

Die Berechnungsfunktionen der Prüfzeichen

Hinweis: Die Regionaleinstellungen des Windows-Systems können die Ausgabe einiger Formeln beeinflussen.

addmodulo10 (*«Zeichenfolge»*): gibt den ganzen Wert mit dem Prüfzeichen Modulo 10 am Ende zurück.

addmodulo10_212(*«Zeichenfolge»*): gibt ein Prüfzeichen Modulo 10_212 (eine Variante des Standard-Prüfzeichens Modulo 10) zurück und schließt die aktuelle Textfolge sowie die Prüfsumme am Ende der Zeichenfolge ein.

addmodulo43(*«Zeichenfolge»*): gibt ein Prüfzeichen Modulo 43 zurück und schließt die aktuelle Textfolge sowie die Prüfsumme am Ende der Zeichenfolge ein.

bimodulo11 (*«Zeichenfolge»*): gibt 2 Prüfzeichen Modulo 11 (Code 11) zurück.

canadacustomscd (*«Zeichenfolge»*): gibt das Prüfzeichen des Canada Customs Standard zurück.

Check103(«string»): Diese Funktion berechnet Mod103.

check128 (*«Zeichenfolge»*): gibt das Prüfzeichen des Code 128 und EAN 128 zurück.

checkPZN(«string»): Berechnet die Prüfziffer für den PZN-Barcode. Der PZN-Barcode basiert auf Code 39, der in Deutschland für pharmazeutische Produkte verwendet wird.

CheckSum2Mod47(«string»): Diese Funktion berechnet Mod47.

CheckSumMod10(«string»): Diese Funktion berechnet Mod10.

CheckSumMod10_Codabar(«string»): Diese Funktion berechnet die spezifische Mod10-Prüfsumme für Codabar.

CheckSumMod10_MSI(«string»): Diese Funktion berechnet die Mod10-Prüfsumme für einen MSI-Barcode.

CheckSumMod11_3Suisse(«string»):Diese Funktion berechnet die Mod11-Prüfsumme für 3Suisses.

CheckSumMod34(«string»): Diese Funktion berechnet Mod34.

checkupce (*«Zeichenfolge»*): gibt das Prüfzeichen für UPC-E (6 Zeichen obligatorisch) zurück.

CRC-16(«string») Gibt den CRC-16-Wert zurück.

Beispiel :
CRC16("123456789") = 47933
"0x" & hex(CRC16("123456789"), 4) = 0xBB3D

modulo10 (*«Zeichenfolge»*): gibt ein Prüfzeichen von Modulo 10 des Code 2/5 verschachtelt, EAN13, EAN 8, EAN 128 K-Mart, VPCA zurück.

modulo10_212(«*Zeichenfolge*») gibt eine Variante des Prüfzeichens Modulo 10 zurück

modulo10IBM(«*Zeichenfolge*»): gibt ein Prüfzeichen IBM Modulo 10 zurück (eine Variante des Standard-Prüfzeichens Modulo 10)

modulo10UPS(«*Zeichenfolge*»): gibt ein Prüfzeichen UPS Modulo 10 zurück (eine Variante des Standard-Prüfzeichens Modulo 10)

modulo11 (*«Zeichenfolge»*): gibt ein Prüfzeichen Modulo 11 (Code 11) zurück.

modulo11IBM(«*Zeichenfolge*»): gibt das Prüfzeichen Modulo 11 IBM (eine Variante des Modulo 11-Standardzeichens)

modulo24 (*«Zeichenfolge»*): gibt ein Prüfzeichen Modulo 24 (Code PSA) zurück.

modulo32 (*«Zeichenfolge»*): gibt ein Prüfzeichen Modulo 32 (pharmazeutische Branche Italien) zurück.

modulo43 (*«Zeichenfolge»*): gibt ein Prüfzeichen Modulo 43 (Code 39) zurück.

modulo47 (*«Zeichenfolge»*): gibt zwei Prüfzeichen Modulo 47 (Code 93) zurück.

plessey (*«Zeichenfolge»*): gibt zwei Prüfzeichen (Code Plessey) zurück.

pricecd (*«Zeichenfolge»*): gibt 4 Prüfzeichen UPC Random Price zurück.

pricecd5 (*«Zeichenfolge»*): gibt 5 Prüfzeichen UPC Random Price zurück.

StringToExt39(«string»): Diese Funktion bereitet eine Zeichenfolge zur Verwendung im Extended 39-Barcode vor.

UPSCheckDigit (*«Zeichenfolge»*): Gibt UPS-Prüfzeichen zurück. Die Eingabe für die Methode ist eine Zeichenfolge. Bei dieser Methode bleibt der Rest der Verfolgungsnummer Ihnen überlassen. Eine Sequenz aus 15 Zeichen wird benötigt, mit deren Hilfe die Prüfziffer berechnet wird.

1. Die ersten beiden Zeichen müssen "1Z" sein.
2. Die nächsten 6 Zeichen füllen wir mit unserer UPS-Kontonummer "XXXXXX".
3. Mit den nächsten 2 Zeichen bezeichnen wir den Servicetyp:

- "01" für Luftfracht am nächsten Tag.
- "02" für Luftfracht am übernächsten Tag.
- "03" für Beförderung zu Land.

4. Die nächsten 5 Zeichen sind unsere Rechnungsnummer (unsere Rechnungen belegen 6 Stellen; wir lassen die erste Stelle aus, z. B. würden aus der Rechnung 123456 die Zeichen 23456 resultieren).
5. Die nächsten 2 Ziffern stehen für die Paketnummer, mit Nullen aufgefüllt. Paket 1 ist z. B., "01", 2 ist "02".
6. Das letzte Zeichen ist die Prüfziffer.

Hinweis: Die oben beschriebene Sequenz ergibt 17 Zeichen, wobei nur 15 zum Berechnen der Prüfziffer benötigt werden. Verwerfen Sie hierzu den Anteil "1Z" und verwenden Sie in der Methode nur die letzten 15 Zeichen.

Die Konvertierungsfunktionen

Hinweis: Die Regionaleinstellungen des Windows-Systems können die Ausgabe einiger Formeln beeinflussen.

AI253 (*«Zeichenfolge»*): Spezifische Funktion zum Vorbereiten einer Zeichenfolge für Anwendungs-ID 253

AI8003 (*«Zeichenfolge»*): Spezifische Funktion zum Vorbereiten einer Zeichenfolge für Anwendungs-ID 8003

Ascii («Zeichenfolge») : gibt den ASCII-Code des ersten Zeichens des Zeichenfolgenarguments zurück

Beispiel **:**
ascii("A") = 65

char («Wert»):

- Für Werte von 128 bis 255: Gibt das Zeichen zurück, das dem Ganzzahlargument in der ASCII-Tabelle entspricht.
- Für alle negativen Werte und Werte von 0 bis 127 und über 255: Gibt das Unicode-Zeichen zurück, das dem Ganzzahlargument entspricht.

Beispiel :
char (65) = "A"

Hinweis: Die Anwendung unterstützt erweiterte ASCII-Zeichen.

CodabarData(«data», «start», «stop»): Passt Daten für CodabarData an.

Code128CData(«Zeichenfolge»): Passt Daten für Code 128 C an.

Currencytoeuro («Wert»): konvertiert die aktuelle Währung in Euro.

DatabarData(«data», «min», «max», «hasComposite»): Passt Daten für DatabarData an.

DBCSToUnicolde(«string»,«codepage»): Der Formel werden zwei Parameter übergeben - die zu konvertierende Daten-"Zeichenfolge" und die verwendete Zeichensatztabelle "Zeichensatztabelle". Der Zeichensatztabellenparameter wird durch den Sprachnamen (Thai, Japanese, ChineseGBK, Korean, ChineseBig5, European, Eastern, Cyrillic, Greek, Turkish, Hebrew, Arabic, Baltic, Vietnamese, UTF-8, ACP) oder einen numerischen Wert repräsentiert (siehe Dokumentation zu Zeichensatztabellenkennungen im Internet).

Beispiel:
DBCSToUnicode(unicodetoDBCS("傍傍傍傍", "UTF-8"), "UTF-8") =傍傍傍傍

Diese Funktion wird für Daten benötigt, die aus Nicht-Unicode-Dateien oder -Datenquellen (z. B. Datenbank) stammen.

Hinweis: Diese Funktion ist das Gegenstück zur Methode UnicodeToDBCS.

Dollar («Wert»): Ändert ein Feld, um einen Preis darzustellen.

Beispiel:
Wenn mit PRICE bezeichnetes Feld einen Wert 199 anzeigt, dann
Dollar(PRICE) ergibt $199,00.

Ean128Data («Zeichenfolge_1»,«Zeichenfolge_2»): Adjust "string_1" data for EAN 128 barcode according to template "string_2"

Eurotocurrency («Wert»): konvertiert den Euro in die aktuelle Währung.

Anmerkung: Die Anzahl der anzuzeigenden Dezimalstellen muss über das Register Ausgabe der Variablen Formel bestimmt werden, indem das Kontrollkästchen Dezimalstellen anzeigen abgehakt wird. Andererseits entspricht der verwendete Umrechnungskurs dem im Register Andere des Dialogfensters Optionen definierten Kurs.

FileToData(«fileName», «errorData», «maxSize»): Gibt die aus der Datei gelesenen Daten zurück.

fixed («wert» , «Dezimalzahl», «keine_Trennung») gibt eine formatierte Zeichenfolge mit einer Anzahl Dezimalstellen zurück, die der *Dezimalzahl* entspricht und mit oder ohne Tausendertrennzeichen.

Beispiele:
fixed (1234.5678, 3, TRUE) = "1234.568"
fixed (1234.5678, 3, FALSE) = "1,234.567"

Anmerkung: Das anzuzeigende Tausendertrennzeichen muss über das Register Ausgabe der Variablen Formel bestimmt werden, indem das Kontrollkästchen Dezimalstellen anzeigen abgehakt wird.

FormatDate(«date», «dateFormat», «localeID»): Konvertiert «date» in eine Zeichenfolge, die «dateFormat» entsprechend formatiert ist.

Rückgabewert	String	Ein Datumswert im Zeichenfolgenformat, formatiert nach einem Format.
date	String	Datumswert im Zeichenfolgenformat.
dateFormat	String / Number	Format des Basisdatumswerts. - Numerische Werte codieren vordefinierte Formate. (Beispiel: 1 dd/mm, 2 dd mmmm usw.). Mögliche Werte sind hier verfügbar. - 0 oder negative Werte bedeuten, dass das aktuelle Systemdatumsformat

		verwendet wird - Zeichenfolgenwerte codieren das Format direkt. (Beispiel: "yyyy/mm/dd")
LocaleID	Number	Der Parameter **localeID** legt die zum Anzeigen des Datenwerts verwendete Ländereinstellung fest. **localeID** ist optional (Standard: 0, Systemländereinstellung wird verwendet). Mögliche Werte (Zahl) finden Sie unter http://msdn.microsoft.com/en-us/goglobal/bb964664.aspx.

FormatMoney(«amount», «use separation characters(T/F)», «price characters», «force leading zaro (T/F)», «location for price character», «# of decimal places»)

Diese Funktion ermöglicht das Hinzufügen von Währungssymbolen, Punkte, vorangestellten Nullen sowie die Verwendung von Trennzeichen

Beispiele:

FormatMoney("123456.234", "T","$","F",0,2) = $123,456.23

FormatMoney("1234","F","$","T",8,2) = $1234.00

FormatMoney("0.234","F","$","F",0,2) = $.23

FormatMoney("0.234","F","$","T",0,2) = $0.23

Hinweis: Die Stelle des Währungszeichens ist für die Positionierung des Währungszeichens verantwortlich. Zischen dem Währungszeichen und dem Betrag wird ein Leerzeichen hinzugefügt. Dieser Wert wird ignoriert, wenn er kürzer als dies sich ergebende Länge des Ergebnisstrings ist.

FormatNumber(number): Diese Funktion ermöglicht die Formatierung eines numerischen Feldes. Bei Verwendung eines Rautezeichens (#) erfolgt die Anzeige nur bei einem vorhandenen Wert, bei Verwendung von Null (0) wird das Feld immer angezeigt.

Beispiel:

FormatNumber(123.45, "US$ #,###,###.00") = US$ 123.45

FormatNumber(123.45, "US$ 0,000,000.00") = US$ 0,000,123.45

FormatNumber(.45, "#,##0.00") = 0.45

FormatNumber(.45, "#,###.00") = 45

FormatNumber(7188302335, "(###) ###-####") = (718) 830-2335

FormatNumber(123.45, "00.00") = 23.45

FormatNumber(123.567, "###,##0.00") = 123.57

GetEnv(«name»): Gibt den Wert «name» (Zeichenfolge) der Umgebungsvariablen zurück.

Beispiel:

GetEnv("OS") = " Windows_NT".

Hinweis: Diese Funktion ist mit benutzerdefinierten Umgebungsvariablen nutzbar.

GS1AIData(«AIName», «AIData», «isLastAI»): Passt Daten für GS1 AI an.

GS1HRData(«AIName», «AIData», «isLastAI»): Adjusts data for GS1 Human readable

GS1Norm(«string»): Adjusts data for GS1 Norm.

int («wert») gibt die größte der ganzen Zahlen zurück, die kleiner oder gleich dem Argument *wert* sind.

Beispiele:

int (-5.863) = -6

int (5.863) = 5

LmChar («ganzzahl») : Gibt das Zeichen, das dem Integerargument in der ANSI-Tabelle entspricht, zurück. Diese Tabelle hängt nicht vom Lokale System ab.
Der Wert ist zwischen 0 und 255.

MaxiCodeData(«Mode»,«PostalCode»,«CountryCode»,«ClassOfService»,«TrackingNumber»,«UpsShipperNumber»,«JulianDate»,«ShipmentID»,«PackageNumber»,«TotalPackages»,«PackageWeight»,«AdressValidation»,«ShipToAdress»,«ShipToCity»,«ShipToState»): Erstellt Maxicode-Daten aus eingegebenem «string».

text («wert», «format») gibt das Argument *wert* zurück, geschrieben auf die durch das Argument *format* festgelegte Art und Weise, die dem Format entspricht, das bei der Erklärung der Variablen "Formular" beschrieben wurde.

Beispiele:

text(012.345, "######,(###)") = 12,(345)

text(1012.345, "######,(###)") = 1012,(345)

trunc («wert») gibt den gesamten Teil des Arguments *wert* an.

Beispiel :

trunc (123.45) = 123

UnicodetoDBCS(«Data», «Codepage»): Die Formel erfordert zwei Parameter – die zu konvertierenden Daten und die verwendete Codepage. Die Codepage-Parameter können irgendeinen der folgenden als Wert annehmen:

- Thai
- Japanese
- Chinese GBK
- Korean
- Chinese Big5
- European eastern
- Greek
- Turkish
- Hebrew
- Arabic
- Baltic
- Vietnamese

Anmerkung: Die Formel berücksichtigt Groß-/Kleinschrift nicht.

value («zeichenfolge») : gibt den numerischen Wert einer Zeichenfolge zurück.

Beispiel:
wert("123") = 123
value("16:48:00")-value("12:00:00") gleich "16:48:00"-"12:00:00" gleich 0,00, wobei die Seriennummer gleich 4 Stunden und 48 Minuten ist.

Anmerkung: Es ist normalerweise nicht erforderlich, die Funktion wert in einer Formel zu verwenden, weil die Anwendung den Text gegebenenfalls automatisch in Zahlen konvertiert.

ValueEx(«zeichenfolge»): Wandelt «zeichenfolge» in numerischen Wert um.

VoiceCode(«string», «string», «string»): Ein VoiceCode ist eine vierstellige Zahl, die aus den PTI-Werten GTIN, Lot und optional Date berechnet wird.

Rückgabewert	string	Berechnet einen VoiceCode-Wert.
GTIN	string	Aus 14 Ziffern bestehende GTIN-Nummer. Nur Ziffern sind zulässig. Bei anderen Werten als Ziffern wird eine ehlermeldung ausgegeben. Falls die Datenlänge 14 Ziffern

		überschreitet, werden die linken 14 Ziffern akzeptiert. Bei weniger als 14 Ziffern wird der Wert links mit der entsprechenden Anzahl von '0' aufgefüllt.
LotNumber	string	Bis zu 20 alphanumerische Symboldaten. Falls die Datenlänge 20 Ziffern überschreitet, werden die linken 20 Ziffern akzeptiert.
Date	string	Dies ist ein optionaler Parameter. Diese sechs Ziffern stellen das Datum im Format JJMMTT dar.

Diese Berechnung wird folgendermaßen durchgeführt:

1. Berechnen Sie PlainText.

    a. PlainText ist die 14-stellige GTIN, die von Lot-Code und Date (sofern vorhanden) angehängt wird.

    b. Beziehen Sie Präfixe bzw. Klammern von Anwendungsbezeichnern nicht ein.

    c. Es gibt keine Leerräume zwischen den Feldern GTIN, Lot und Date.

    d. Das Datum wird, sofern vorhanden, im Format JJMMTT mit Nullen aufgefüllt sowie ohne '/' dargestellt.

2. Berechnen Sie das ANSI CRC-16 Hash der PlainText-ASCII-Bytes unter Verwendung des standardmäßigen ANSI CRC-16 Hash mit dem Polynom X16 + X15 + X2 + 1.
   Siehe CRC16-Funktion.
3. Berechnen Sie den VoiceCode aus dem Hash, indem Sie die 4 niederwertigsten Ziffern in Dezimalform verwenden (Hash mod 10000).

Beispiel:

GTIN = (01) 10850510002011
Lot = (10) 46587443HG234
PlainText = 10850510002011146587443HG234
CRC-16 Hash = 26359
VoiceCode = 6359
Large Digits = 59
Small Digits = 63

Beispiele:

VoiceCode("10850510002011", "46587443HG234") = 6359
VoiceCode("65457886676767", "2", "100126") = 5836

ATA 2000-Konvertierungsfunktionen

Hinweis: Die Regionaleinstellungen des Windows-Systems können die Ausgabe einiger Formeln beeinflussen.

Bin2Hex(«Wert», «Pad»)

Konvertiert einen binären Wert in einen hexadezimalen Ausgabewert. Der 'Pad'-Parameter wird verwendet, um die Länge der gesamten Ausgabe zu definieren, indem links vom Ergebnis 0 Zeichen hinzugefügt werden.

Beispiel:
Bin2Hex("0010100111101001101",0) ergibt 14F4D
Bin2Hex ("0010100111101001101", 0) ergibt 00014F4D

CRC16_CCITT («String»)

berechnet die CRC16 CCITT-Prüfsumme.
Dies wird in der ATA2000 Regulation für TOC für die CRC-Berechnung verwendet.

Beispiel:
CRC16_CCITT ("A") ergibt B915
CRC16_CCITT ("AB") ergibt 4B74
CRC16_CCITT ("ABC") ergibt F508

CRC32_CCITT («String»)

berechnet die CRC32 CCITT-Prüfsumme.
Dies wird in der ATA2000 Regulation für TOC für die CRC-Berechnung verwendet.

Beispiel:
CRC32_CCITT ("A") ergibt D3D99E8B
CRC32_CCITT ("AB") ergibt 30694C07
CRC32_CCITT ("123456789") ergibt CBF43926

Dec2Bin («Wert», «Pad»)

wandelt einen Dezimalwert in einen binären Ausgabewert um. Der 'Pad'-Parameter wird verwendet, um die Länge der gesamten Ausgabe zu definieren, indem links vom Ergebnis 0 Zeichen hinzugefügt werden.

Beispiel:
Dec2Bin("5", 0) ergibt 101
Dec2Bin("5", 4) ergibt 0101

Dec2Hex («Wert», «Pad»)

wandelt einen Dezimalwert in einen hexadezimalen Ausgabewert um. Der 'Pad'-Parameter wird verwendet, um die Länge der gesamten Ausgabe zu definieren, indem links vom Ergebnis 0 Zeichen hinzugefügt werden.

Beispiel:
Dec2Hex (510, 0) ergibt 1FE
Dec2Hex (510, 4) ergibt 01FE

String2Hex(«Wert», «Pad»)

Konvertiert einen String-Wert in einen hexadezimalen Ausgabewert. Der 'Pad'-Parameter wird verwendet, um die Länge der gesamten Ausgabe zu definieren, indem rechts vom Ergebnis 0-Zeichen hinzugefügt werden.
Bei RFTAG werden die Daten sehr oft im Hexadezimalformat mit einer Wort-Ausrichtung (2Bytes) codiert.

Beispiel:
String2Hex ("A", 0) ergibt 41 (**A** ist AINSI Code **65** in Dezimal, wobei **41** in Hexadezimal ist)
String2Hex("ABC", 2) ergibt 414243
String2Hex ("ABC", 4) ergibt 41424300

hier ist ein Beispiel für einen RFID-ATA2000 Geburtsdatensatz:

String2Hex("MFR S0671*SEQ M37GXB92*PNO PQ7VZ4*PDT CONTROLLER*ICC 456789*UIC 2*DMF 20160701*UNT KG*HAZ UN0003*ESD 1*LOT 123456*CNT FR*PMLPML123456789", 2) ergibt
4D46522053303637312A534551204D333747584239322A504E4F20505137565A342A50445420434F4E54524F4C4C45522A49

String6BitsEncoding(«Wert», «Pad»)

Konvertiert einen String-Wert in einen 6-Bit ASCII-Kodierungs- Ausgabewert.

Der 'Pad'-Parameter wird verwendet, um die Länge der gesamten Ausgabe zu definieren, indem rechts vom Ergebnis 0-Zeichen hinzugefügt werden.
Bei ATA2000 können Daten zu Kompressionszwecken in 6bits (statt 8bits) kodiert werden.

Beispiel:
String6BitsEncoding ("11", 0) ergibt C710

VDAString6BitsEncoding(«Zeichenkette»)

Konvertiert einen Zeichenketten-Wert in einen 6-Bit-ASCII-Ausgabewert unter Verwendung der Anforderungen des deutschen Verbands der Automobilindustrie (VDA).
In ATA2000 können Daten zu Komprimierungszwecken in 6 Bit (statt 8 Bit) kodiert werden. Die für die Konvertierung verwendete Tabelle ist hier verfügbar. Weitere Einzelheiten finden Sie in den Spezifikationen VDA5500 und VDA4994.

Beispiel:
VDAString6BitsEncoding("11") ergibt C71860820820

Die Datums- und Uhrzeitfunktionen

Die unten aufgeführte Datumsvariable stellt das Datum und die Uhrzeit des Systems dar und nicht die durch das Programm definierte Datumsvariable.

Hinweis: Die Regionaleinstellungen des Windows-Systems können die Ausgabe einiger Formeln beeinflussen.

BestBefore(«date» , «dateFormat» , «offset» , «offsetUnit» , «changeMonth» , «outputFormat», «localeID»)

Ermöglicht die Berechnung eines Mindesthaltbarkeitsdatums basierend auf dem Tastatureingabewert (nicht basierend auf dem aktuellen Datum).

Rückgabewert	Zeichenfolge	Ein Datumswert im Zeichenfolgenformat, berechnet durch ein Basisdatum und einen Zeitabstand.
date	Zeichenfolge	Basisdatumswert im Zeichenfolgenformat
dateFormat	Zeichenfolge / nummer	Format des Basisdatumswerts. - Numerische Werte codieren vordefinierte Formate. (Beispiel: 1 dd/mm, 2 dd mmmm usw.). Mögliche Werte sind hier verfügbar. - 0 oder negative Werte bedeuten, dass das aktuelle Systemdatumsformat verwendet wird - Zeichenfolgenwerte codieren das Format direkt. (Beispiel: "yyyy/mm/dd")
offset	nummer	Anzahl von Datumseinheiten, die zum Basisdatum hinzugefügt werden sollen.
offsetUnit	Zeichenfolge / nummer	Bedeutung des offset-Parameters. - **1** oder "**d**" oder "**D**" steht für Tage - **2** oder "**m**" oder "**M**" steht für Monate - **3** oder "**y**" oder "**Y**" steht für Jahre
changeMonth	nummer	Optional (Standard: True) - **0** bedeutet **False** - **1** bedeutet **True** Wenn es das berechnete Datum nicht gibt (z.B. der 30. Februar) **True** gibt den ersten Tag des nachfolgenden Monats (also den 1. März) zurück **False** gibt den letzten Tag innerhalb des aktuellen Monats (also den 28. Februar) zurück
outputFormat	Zeichenfolge / nummer	Optional (Standard: wie dateFormat) Über diesen Parameter wird das Ausgabeformat des berechneten Datums angegeben. Mögliche Werte sind hier verfügbar.
localeID	nummer	Optional (Standard: 0, Systemländereinstellung wird verwendet) Dieser Parameter legt die zum Anzeigen des Datenwerts verwendete Ländereinstellung fest.

Beispiel:

BestBefore («*date*» , «dateFormat» , «offset» , «offsetUnit» , «changeMonth» , «outputFormat»)

BestBefore("14/06/2012" ,"dd/mm/yyyy","18","m", 1, "dd/mm/yyyy") will return 14/12/2013.

Beispiel: Formula with a date data source.

Create a date variable name date0 with the date of the day. 26/06/2012 for this example with dd/mm/yy format.

BestBefore(date0 ,"dd/mm/yy","18","m", 1, "dd/mm/yyyy") will return 14/12/2013.

CFIA_Month(«datum»):

Gibt den von der Canadian Food Inspection Agency (CFIA) verwendeten Monatsnamen zurück: JA, FE, MR, AL, MA, JN, JL, AU, SE, OC, NO, DE.

Beispiel:

CFIA_Month ("03/01/2021") gibt MR zurück

CFIA_Month (today()) gibt den Monatsnamen basierend auf dem aktuellen Datum zurück, wenn heute der 01/01/2021 ist, wird JA zurückgegeben

CFIA_Month ("06/01") gibt JN zurück

Hinweis: Das Format des Arguments "Datum" hängt von den für das System geltenden Umgebungseinstellungen ab.

DateOffset(«date» , «offset», «offsetUnit» , «changeMonth»)

Rückgabewert	datum	Fügt ein Datumsintervall zum angegebenen Basisdatum hinzu.
date	datum	Basisdatumswert.
offset	nummer	Anzahl von Datumseinheiten, die zum Basisdatum hinzugefügt werden sollen.
offsetUnit	Zeichenfolge / nummer	Bedeutung des offset-Parameters. - **1** oder "**d**" oder "**D**" steht für Tage - **2** oder "**m**" oder "**M**" steht für Monate - **3** oder "**y**" oder "**Y**" steht für Jahre
changeMonth	Zeichenfolge / nummer	Optional (Standard: True) - **0** bedeutet **False** - **1** bedeutet **True** Wenn es das berechnete Datum nicht gibt (z.B. der 30. Februar) **True** gibt den ersten Tag des nachfolgenden Monats (also den 1. März) zurück

		False gibt den letzten Tag innerhalb des aktuellen Monats (also den 28. Februar) zurück

Beispiel:
DateOffset("26/06/2012",1,"d",1) will return 27/06/2012
-OR-
DateOffset(today(),1,"D", 0)

Beispiel: Formula with a date data source.
Create a date variable name Date0 with the date of the day. 26/06/2012 for this example with dd/mm/yy format. You should ensure that date variable is in default system format (otherwise you can use DateValue() function to get the date value from the date variable).
DateOffset(Date0,1,"D", 0) will return 27/06/2012

DateValue(«formattedDate» , «format»)

Rückgabewert	datum	Gibt einen Datumswert des formattedDate-Parameters zurück.
formattedDate	Zeichenfolge	Ein Textformatdatum, das in einen Datumswert konviertiert werden soll.
format	Zeichenfolge	Optional (Standard: das Systemdatumsformat) Bei Angabe die direkte Formatzeichenfolge. Beispiel: "dd/mm/yy"

Beispiel:
DateValue(22062012,"ddmmyyyy") will return 22/06/2012.

day («datum») gibt den Tag des Monats des Arguments *datum* zurück.

FiscalDate(«fiscalStartDate», «outputFormat»)

Ermöglicht die Berechnung des Datums für den Beginn Ihres Geschäftsjahres (yyyy.mm.dd).

Beispiele (aktuelles Datum ist: 2009.11.24):
FiscalDate("2009.01.01", 1) = 09
FiscalDate("2009.01.01", "yyyy") = 2009
FiscalDate("2009.01.01", 3) = 47
FiscalDate("2009.01.01", "day") = 328

hour («datum») gibt die Uhrzeit des Arguments *datum* zurück.

minute («datum») gibt die Minuten des Arguments *datum* zurück.

month («datum») gibt den Monat des Arguments *datum* zurück.

now () gibt das aktuelle Datum und die Uhrzeit zurück.

second («datum»): Gibt die Sekunde des Datumarguments an.

shiftcode(«items»)

Der Shift-Code wird als Datenquelle für ein Etikettenfeld verwendet, das entsprechend der aktuellen Zeitangabe fortlaufend geändert werden muss. Ein Shift stellt eine Zeitspanne dar, die definiert und benannt ist.

Rückgabewert	Zeichenfolge (string)	Gibt den aktuellen Shift-Codewert zurück.
Argumente	Zeichenfolge (string)	Shift-Codeargumente in folgendem Format: StartStd:StartMin-StoppStd:StoppMin-Wert|…| StartStd:StartMin-StoppStd:StoppMin-Wert

Shifts können beispielsweise wie folgt definiert sein:

- 7:00 bis 15:00 = Tag
- 15:00 bis 23:00 = Abend
- 23:00 bis 7:00 = Nacht

Hinweise:
1. Das 24-Stunden-Format wird zur Definition von Zeitwerten verwendet. 0:00 ist Mitternacht; 12:00 ist Mittag.
2. Wenn bei der Zeiteingabe Überlappungen auftreten, wird der erste zulässige Wert zurückgegeben.
3. Wenn die aktuelle Zeit keinem Shift zugeordnet werden kann, wird eine leere Zeichenfolge zurückgegeben.

Beispiel (aktuelle Zeit ist 16:00):
shiftcode("7:00-15:00-Tag|15:00-23:00-Abend|23:00-7:00-Nacht") ="Abend"

SpecificDateFormat("[date format]", "+/[offset data][date interval]")

Mit der SpecificDateFormat-Funktion können Sie einen Datumsstempel anpassen, indem Sie das Datum um ein bestimmtes Intervall versetzen und dieses angepasste Datum in einem Formelausdruck verwenden. Die SpecificDateFormat-Funktion versetzt das aktuelle Datum auf Basis der Systemuhr entweder mithilfe des Standarddatumsformats oder eines anderen Formats Ihrer Wahl.

- [date format] Gibt das zu verwendende Datumsformat an. Dieser Parameter muss in Klammern gesetzt werden. Beachten Sie die ***Tabelle*** gültiger Datumsformate.

Datumseinstellung	Ausgabe	Beschreibung
D und d	1	Datum des Monats (ohne führende Null)
DD	01	Datum des Monats in 2 Zeichen (führende Null)
dd	1	Datum des Monats in 2 Zeichen (führendes Leerzeichen)
DDD	224	Tage seit Jahresbeginn (führende Null)
ddd	224	Tage seit Jahresbeginn (führendes Leerzeichen)
DDDDD und ddddd	33482	Tage seit 01.01.1900(5 bis 9 Zeichen verfügbar)
W and w	1	Tag der Woche (Son=1, Sam=7)
WW	34	Wochen seit Jahresbeginn (führende Null)
ww	34	Wochen seit Jahresbeginn (führendes Leerzeichen)
WWW und www	783	Wochen seit 01.01.1900 (letzte 3 Stellen)
WWWW und wwww	4783	Wochen seit 01.01.1900 (4 bis 9 Zeichen verfügbar)
WWWWWWWWW	000004783	Wochen seit 01.01.1900 (führende Null)
wwwwwwwww	4783	Wochen seit 01.01.1900 (führendes Leerzeichen)
M und m	9	Monatszahl (ohne führende Null)
MM	09	Monatszahl in 2 Zeichen (führende Null)
mm	9	Monatszahl in 2 Zeichen (führendes Leerzeichen)
MMM	009	Monatszahl in 3 Zeichen (führende Null)
mmm	9	Monatszahl in 3 Zeichen (führendes Leerzeichen)
MMMM und mmmm	1101	Monate seit 01.01.1900
MMMMM	01101	Monate seit 1900 in 5 Zeichen (führende Null)
mmmmm	1101	Monate seit 1900 in 5 Zeichen (führendes Leerzeichen)
MMMMMMMMM	000001101	Monate seit 1900 in 9 Zeichen (führende Null)
mmmmmmmmm	1101	Monate seit 1900 in 9 Zeichen (führendes Leerzeichen)
YY und yy	91	Jahr in 2 Stellen
Y und y	1991	ganzes Jahr
YYY und yyy	991	letzte 3 Stellen des Jahres (3 bis 9 Zeichen verfügbar)
YYYYYYYYY	000001991	ganzes Jahr (führende Null)
yyyyyyyyy	1991	ganzes Jahr (führendes Leerzeichen)

- +/[offset data] +/[Versatzdaten] Dieser Wert gibt an, in welchem Umfang das Datum versetzt wird. Dem Intervall muss entweder ein Plus (+)- oder Minus ()-Zeichen vorangestellt werden, je nachdem, ob Sie vom aktuellen Datum aus vor oder zurück gehen möchten.
- [Datumsintervall] Das Datumsintervall muss entweder d für Tage, w für Wochen, m für Monate oder y für Jahre sein.

Bei Verwendung in einem Ausdruck muss die SpecificDateFormat-Funktion mithilfe folgender Parameter geschrieben werden:

Die folgende Tabelle zeigt Beispiele der Verwendung der SpecificDateFormat-Funktion in Ausdrücken in verschiedenen Formaten. (Hinweis: In diesen Beispielen wird als aktuelles Datum der 29. Juli 2013 vorausgesetzt.)

SpecificDateFormat("mm dd yy" , "+1y") = 7 29 14 (1 Jahr nach dem aktuellen Datum)
SpecificDateFormat("ww/m/yyyy" , "30w) = 1/12/2012 (30 Wochen vor dem aktuellen Datum)
SpecificDateFormat("ddddd / wwww /mmmm" , "") = 41484/5928/1363 (Tage, Wochen und Monate seit 1900)

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"time interval +/offset time")

TMit der TimeOffset-Funktion können Sie einen Zeitstempel anpassen, indem Sie die Uhrzeit um ein bestimmtes Intervall versetzen und diese angepasste Uhrzeit in einem Formelausdruck verwenden. Die TimeOffset-Funktion versetzt die aktuelle Uhrzeit auf Basis der Systemuhr mithilfe des Standardzeitformats.

Bei Verwendung in einem Ausdruck muss die TimeOffset-Funktion mithilfe folgender Parameter geschrieben werden:

- Sie können nur das Standardformat "mm/dd/yyyy hh:nn:ss" verwenden. Zur Eingabe eines anderen Formats müssen Sie TimeOffset der Funktion DateValue übergeben und diese wiederum der Funktion FormatDate, wo Sie ein anderes Zeitformat angeben können.
- "+/ Versatzzeit" Dieser Wert gibt an, in welchem Umfang die Uhrzeit versetzt wird. Dem Versatz muss entweder ein Plus (+)- oder Minus ()-Zeichen vorangestellt werden, je nachdem, ob Sie von der aktuellen Uhrzeit aus vor oder zurück gehen möchten.
- "Zeitintervall" Das Zeitintervall muss entweder in s für Sekunden, m für Minuten oder h für Stunden angegeben werden.

Die folgende Tabelle zeigt Beispiele der Verwendung der TimeOffset-Funktion in Ausdrücken in verschiedenen Formaten. (Hinweis: In diesen Beispielen wird als aktuelle Uhrzeit 4:12:10 PM (16.12 Uhr und 10 Sekunden) am 30.07.13 vorausgesetzt.)

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"h+5") = 07/30/2013 21:12:10 (5 Stunden nach der aktuellen Uhrzeit)
FormatDate(DateValue(TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"), "n+10"), "mm/dd/yyyy hh:nn:ss"), "hhnn") = 16-22 (10 Minuten nach der aktuellen Uhrzeit)

today () gibt das aktuelle Datum zurück.

Week («datum») gibt die Nummer der Woche des Arguments *datum* zurück.

weekday («datum») gibt den Wochentag des Arguments *datum* zurück.

Anmerkung: Sonntag wird als erster Tag der Woche betrachtet.

Beispiel:
d. h. "Dienstag, 20. November 1999" ist das aktuelle Datum bei einem weekday(now()) = 3

WeekISO8601(«Date», «DateFormat»)

Ermöglicht Ihnen das Erstellen einer Formel, die ISO 8601 unterstützt. ISO 8601 ist ein internationaler Standard für den Austausch datums- und uhrzeitbezogener Daten.

Rückgabewert	datum	Gibt die Woche des angegebenen Datums zurück.
Date	Zeichenfolge	Ein Datum im Textformat, das in einen Datumswert konvertiert werden kann.
DateFormat	Zeichenfolge	Format des Basisdatumswerts. - Zahlen kodieren vordefinierte Formate. (z. B. 1 tt/mm/jj, 2 tt/mm/jjjj usw.) - 0 oder negative Werte stehen für die Verwendung des aktuellen Systemdatumsformats - Zeichenfolgenwert kodiert das Format direkt. (z. B. "jjjj/mm/tt")

Beispiel:

WeekISO8601(" 03/01/2010 ",0) = 53
WeekISO8601(" 02/01/2011 "," dd/mm/yyyy ") = 52
WeekISO8601(" 01/01/2011 ", " dd/mm/yyyy ") = 52

year («datum») gibt das Jahr des Arguments Datum zurück.

Beispiel:
minute (now ()) gibt die Minuten der aktuellen Uhrzeit zurück.
year (today ()) gibt das Jahr des aktuellen Datums zurück.

Die logischen Funktionen

Die logischen Funktionen ermöglichen es, zu prüfen, ob eine oder mehrere Bedingungen erfüllt werden oder nicht.

Hinweis: Die Regionaleinstellungen des Windows-Systems können die Ausgabe einiger Formeln beeinflussen.

Anmerkung: TRUE entspricht 1, FALSE entspricht 0

and («ausdruck_1», «ausdruck_2») gibt TRUE zurück, wenn die beiden Argumente wahr sind, und FALSE, wenn mindestens ein Argument falsch ist. Die Argumente müssen mit logischen Werten bewertet werden.

Beispiel:
and(exact("zeichenfolge","zeichenfolge"),exact("zeichenfolge","zeichenfolge")) = 0
and(exact("zeichenfolge","zeichenfolge"),exact("zeichenfolge","zeichenfolge")) = 1

exact («zeichenfolge_1», «zeichenfolge_2») gibt TRUE zurück, wenn die beiden Zeichenfolgen identisch sind. Ansonsten wird FALSE zurückgegeben. Diese Funktion unterscheidet zwischen Groß- und Kleinschreibung .

Beispiel:
exact("software","software") = 1
exact("software","software") = 0

if («ausdruck», «wert_wenn_wahr», «wert_wenn_falsch») gibt das Argument *wert_wenn_wahr* zurück, wenn *ausdruck* wahr ist und das Argument *wert_wenn_falsch*, wenn *ausdruck* falsch ist.

Beispiel:
if(exact("zeichenfolge", "zeichenfolge"), "wahr", "falsch")) = falsch
if(exact("zeichenfolge", "zeichenfolge"), "wahr", "falsch")) = wahr

not («logisch») gibt das Gegenteil des Arguments *logisch* zurück.

Beispiel: not(exact("zeichenfolge", "zeichenfolge")) = 1
not(exact("zeichenfolge", "zeichenfolge")) = 0
not(False) = 1 oder nicht(O) = 1
not(True) = 0 oder nicht(1) =0
not(1+1=2) = 0

or («ausdruck_1», «ausdruck_2») gibt TRUE zurück, wenn eines der beiden Argumente wahr ist und FALSE, wenn beide Argumente falsch sind. Die Argumente müssen mit logischen Werten bewertet werden.

Beispiel:
or(exact("zeichenfolge", "zeichenfolge"),exact("zeichenfolge", "zeichenfolge")) = 0
or(exact("zeichenfolge", "zeichenfolge"),exact("zeichenfolge", "zeichenfolge")) = 1
or(true,true) = 1 oder or(1,1) = 1
or(true,false) = 1 oder or(1,0) = 1
or(false,false)= 0 oder or(0,0) = 0

Die mathematischen Funktionen

Hinweis: Die Regionaleinstellungen des Windows-Systems können die Ausgabe einiger Formeln beeinflussen.

Abs(data): Diese Funktion gibt den absoluten (positiven) Datenwert zurück. Ermöglicht die Verwendung von Buchstaben nach Zahlen.

Beispiele
Abs(-5) = 5
Abs(5) = 5

base10tobaseX(«zeichenfolge_1»,«zeichenfolge_2») konvertiert *zeichenfolge_2* von Base 10 in Base *zeichenfolge_1*

Beispiele

Wenn das Feld namens Base 16 die Zeichenfolge 0123456789ABCDEF enthält

BASE10TOBASEX(Base16, 12) ergibt C

BASE10TOBASEX(Base16,10) ergibt A

BASE10TOBASEX("012345","9") ergibt 13

Note: This formula cannot accept negative decimal numbers for "string_2" parameter.

baseXtobase10(«zeichenfolge_1»,«zeichenfolge_2») konvertiert *zeichenfolge_2* von Base *zeichenfolge_1* in Base 10

Beispiele

Wenn das Feld namens Base 16 die Zeichenfolge 0123456789ABCDEF enthält

BASEXTOBASE10(Base16, "E") ergibt 14

BASEXTOBASE10(Base16,10) ergibt A

BASEXTOBASE10("012345","9") ergibt 13

Ceil(data): Diese Funktion rundet Daten auf die nächste Ganzzahl auf. Ermöglicht die Verwendung von Buchstaben nach Zahlen.

Beispiele

Ceil(3.234) = 4

Ceil(7.328) = 8

Decimals(data1, data2): Diese Funktion verwendet data2 Dezimalstellen in data1. Ermöglicht die Verwendung von Buchstaben nach Zahlen.

Beispiele

Decimals(4, 2) = 4.00

Decimals(3.524, 1) = 3.5

eval_add(«string»,«string»): Gibt die Summe der Parameter zurück.

Beispiele

eval_add(5,5)=10

eval_div(«string»,«string»):Gibt das Ergebnis der Division der Parameter zurück.

Beispiele

eval_div(20,2)=10

eval_mult(«string»,«string»): Gibt das Ergebnis der Multiplikation der Parameter zurück.

Beispiele

eval_mult(5,2)=10

eval_sub(«string»,«string»): Gibt das Ergebnis der Subtraktion der Parameter zurück.

Beispiele

eval_sub(20,10)=10

Floor(data): Diese Funktion rundet Daten auf die nächste Ganzzahl auf. Ermöglicht die Verwendung von Buchstaben nach Zahlen.

Beispiele

Floor(3.234)= 3

Floor(7.328)= 7

hex(wert_1,wert_2) konvertiert den Dezimalwert *wert_1* in Hexadezimalformat mit Gesamtbreite *wert_*

Note: This formula cannot accept negative decimal numbers for "val_1" parameter.

int («wert») gibt die größte der ganzen Zahlen zurück, die kleiner oder gleich dem Argument *wert* sind.

Beispiele:

int (-5.863) = -6

int (5.863) = 5

max(data1, data2): Diese Funktion zeigt den großen Wert an. Ermöglicht die Verwendung von Buchstaben nach Zahlen.

Beispiele:

Max(5, 12) = 12

min(data1, data2): Diese Funktion zeigt den niedrigsten Wert an. Ermöglicht die Verwendung von Buchstaben nach Zahlen.

Beispiele:

Min(5, 12) = 5

mod («wert_1», «wert_2») gibt den Rest der Division des Arguments *wert_1* durch das Argument *wert_2* zurück. Das Ergebnis hat dasselbe Vorzeichen wie der Teiler.

Beispiele:

mod (7,2) = 1

mod (-7,2) = 1

mod (7,-2) = -1

mod (-7,-2) = -1

quotient («wert_1», «wert_2») gibt das gesamte Ergebnis der Division des Arguments *wert_1* durch das Argument *wert_2* zurück.

round («wert_1», «wert_2») gibt das Argument *wert_1* zurück, das auf die Anzahl der Ziffern gerundet ist, die durch das Argument *wert_2* angegeben wird.

- Wenn *wert_2* höher als 0 ist, wird *wert_1* auf die angegebene Anzahl Dezimalstellen gerundet.
- Wenn *wert_2* gleich 0 ist, wird *wert_1* auf die nächste ganze Zahl gerundet.
- Wenn *wert_2* unter 0 liegt, wird *wert_1* links vom Komma gerundet.

Beispiele:

round (4.25,1) = 4,3

round (1.449, 1) = 1,4

round (42.6,-1) = 40

-

Die Funktion in den Texten

Hinweis: Die Regionaleinstellungen des Windows-Systems können die Ausgabe einiger Formeln beeinflussen.

Eine Zeichenfolge kann mit einer Tabelle assimiliert werden, bei der jedes Feld ein Zeichen enthält. Sie wird durch ihre Länge (gesamte Anzahl Zeichen der Zeichenfolge, einschließlich Leerzeichen) definiert). Die **Position** eines Zeichens in einer Zeichenfolge entspricht dem Platz in der Tabelle, d. h. dem ersten Zeichen der Position 1. Beispiel: Position 3 entspricht dem dritten Zeichen der Zeichenfolge.

cyclebasex ( ) ermöglicht das Zählen in irgendeinem Zählsystem einer Datenbank. Das Nummerierungssystem muss innerhalb des verbundenen Ausdrucks bestimmt sein. Der Ausgangswert, der Wert jedes Inkrements und die Anzahl Kopien jeder Nummer müssen ebenfalls angegeben werden. Alle diese Werte können mit anderen Feldern des Etiketts verbunden sein, aber die Namen der Felder dürfen nicht zwischen Anführungszeichen stehen.

Beispiel:
Wenn ein Base16 bezeichnetes Feld die Zeichenfolge 0123456789ABCDEF enthält, dann ;
cyclebasex(base16, « 8 », 1 ,1) = 8,9,A,B,C
cyclebasex(base16, « F », -1,1) = F,E,D,C,B,A 9,8,7
cyclebasex(base16, « B0 », 1,1) = B0, B1, B2
cyclebasex(«012345 », « 4 »,1,2) = 4,4,5,5,10,10,11,11

cyclechar () erstellt eine personalisierte Zeichenfolge für einen vollständigen Zyklus.

Beispiel:
cyclechar(« A », « C ») = A B C A B C A B C
cyclechar(« A », « C », 1,2) = A A B B C C A A B B

cyclenumber ( ) ermöglicht die Einstellung Ihrer eigenen Zahlenfolge, anders als die normale Reihenfolge von Zahlen oder Buchstaben (0,1,2… oder A,B,C…).

Beispiel:
cyclenumber(1,3) erzeugt Etiketten in der Reihenfolge 1 2 3 1 2 3 1 2 3...
cyclenumber(1,3,1,2) erzeugt Etiketten in der Reihenfolge 1 1 2 2 3 3 1 1 2 2 3 3...

cyclestring ( ) ermöglicht das Erstellen einer Wort- oder Zeichengruppe beim Durchlauf eines vollständigen Zyklus wie ein Inkrementfeld. Die gesamte Zeichenfolge muss zwischen Anführungszeichen stehen (" ") und jedes Wort oder jeder Zeichengruppe muss durch ein Semikolon von den anderen getrennt sein (;)

Beispiel:
cyclestring(« Mon ; Tue ; Wed ; Thu ; Fri ; Sat ; Sun ») = Mon Tue Wed Thu Fri Sat Sun
Das folgende Beispiel bezieht sich auf den Druck von Etiketten, für die alle Buchstaben des Alphabets außer O und I verwendet werden.
cyclestring(« A;B;C;D;E;F;G;H;J;K;L;M;N;P;Q;R;ST;U;V;W;X;Y;Z »)

exact («zeichenfolge_1», «zeichenfolge_2») gibt TRUE zurück, wenn die beiden Zeichenfolgen identisch sind. Ansonsten wird FALSE zurückgegeben.

Beispiel:

exact("software","software") = 1

exact("software","software") = 0

extract («Zeichenfolge», «Trennzeichen», «Position») gibt die Teilzeichenfolge aus der «Zeichenfolge» mit Daten, die durch das «Trennzeichen» getrennt sind, an der angegebenen «Position» zurück.

Beispiel:

Extract("1;2;3;4", ";", 3) = 3

find («zeichenfolge», «schlüssel», «anfang») gibt die Position des ersten Auftretens des Arguments *schlüssel* im Argument *zeichenfolge* an. Die Suche im Argument *zeichenfolge* beginnt ab der durch das Argument *anfang* gelieferten Position (*anfang* >= 1). Die Funktion sendet Null zurück, wenn kein Exemplar des Arguments *schlüssel* gefunden wird. Groß- und Kleinbuchstaben werden unterschieden.

Beispiel:

find("Peter McPeepert","P",1) = 1

find("Peter McPeepert","P",1) = 12

left («zeichenfolge», «anz_zeich») gibt die Zeichenfolge zurück, die aus dem Argument *zeichenfolge* extrahiert wurde. Diese Zeichenfolge beginnt bei Position 1 des Arguments *zeichenfolge* und hat dieselbe Länge wie das Argument *anz_zeich*.

Beispiel:

left("Peter McPeepert",1) = P

left("Peter McPeepert ",5) = Peter

len («zeichenfolge») gibt die Länge des Arguments *zeichenfolge* zurück. Die Leerzeichen werden als Zeichen gezählt.

Beispiel:

len("Paris, New York") = 15

len("") = 0

len(" ") = 1

lower («zeichenfolge») konvertiert alle Großbuchstaben einer Zeichenfolge in Kleinbuchstaben.

Beispiel:
lower("Paris, New York") = paris, new york

LTrim(«string»): Diese Funktion entfernt automatisch vor- oder nachgestellte Leerzeichen links von den Daten.

Beispiel:
LTrim(" No."): No

mid («zeichenfolge», «anfang», «anz_zeich») gibt die Zeichenfolge zurück, die aus dem Argument *zeichenfolge* extrahiert wurde. Diese Zeichenfolge beginnt an der Position, die dem Wert des Arguments *anfang* (*anfang* >=1) entspricht und hat dieselbe Länge wie das Argument *nb_car*.

Beispiel:
mid("Paris, New York",8,8) = New York

Output («Variable»): gibt den formatierten Wert der Variablen zurück.

Beispiele:
entweder der Zähler0, dessen Präfix "nummer :" ist. Die Formel zähler0&output(zähler0) hat das Ergebnis:
1 nummer : 001

Anmerkung: Die Formatierung der Variablen wird über das Register Ausgabe definiert.

pad («string», «length», «char») fügt Zeichen links im Feld hinzu, um der gesamten Eingabe eine vorgegebene Länge zuzuteilen. Jegliches Zeichen kann als Füllzeichen gewählt werden.

Beispiel:
Weist ein Feld mit dem Namen GREETING einen Wert HELLO auf, dann …
pad(GREETING,8) = 000HELLO
pad(5,3) = 005
pad(”Nine”,6,a) = aaNine

replace («zeichenfolge», «anfang, «nb_car», «neue_zeichenfolge») gibt das umgewandelte Argument *zeichenfolge* zurück. Eine Anzahl (gleich dem Argument *anz_zeich*) Zeichen ab der durch das Argument definierten Position *anfang* wurde durch das Argument *neue_zeichenfolge* ersetzt.

Beispiel:

replace("Paris, New York",8,8,"Singapour") = Paris, Singapur

replaceString(«Zeichenfolge», «alte Zeichenfolge», «neue Zeichenfolge») ersetzt alle Vorkommen von «alte Zeichenfolge» in der «Zeichenfolge» durch den für «neue Zeichenfolge» angegebenen Wert.

Beispiel:

ReplaceString("abc12def12", "12", "") = abcdef

rept («zeichenfolge», «anz_zeich») gibt die Zeichenfolge zurück, in der das Argument *zeichenfolge* eine Anzahl von Malen wiederholt wird, die dem Argument *anz_zeich* entspricht.

Beispiel:

rept("Ah Paris ! ",2) = Ah Paris ! Ah Paris !

right («zeichenfolge», «anz_zeich») gibt die Zeichenfolge zurück, die aus den letzten Zeichen des Arguments *zeichenfolge* besteht und eine Länge hat, die dem Argument *anz_zeich* entspricht.

Beispiel:

right("Bestellschein",8) = Bestellung

RTrim («string»): Diese Funktion entfernt automatisch vor- oder nachgestellte Leerzeichen rechts von den Daten.

Beispiel:

RTrim("Part ") :Part

search («zeichenfolge», «schlüssel», «anfang») gibt die Position des ersten Exemplars des Arguments *schlüssel* im Argument *zeichenfolge* zurück. Die Suche beginnt ab der durch das Argument *anfang* definierten Position (*anfang* >= 1). Die Funktion sendet Null zurück, wenn kein Exemplar des Arguments *schlüssel* gefunden wird.

Beispiel:

search("Bestellschein","Bestellung,1) = 8

search("Bestellschein","c",1) = 8

Sort(«Liste», «Trennzeichen», «Richtung»): Gibt eine sortierte Liste von Elementen zurück. Die Sortierung erfolgt durch den Vergleich von Zeichenketten. Liste ist ein String-Argument, das mehrere Elemente mit einem Trennzeichen wie Komma oder Semikolon enthält. Richtung ist eine optionale ganze

Zahl, die die Richtung der Sortierung angibt: 0 = aufsteigend, 1 = absteigend. Wenn die Richtung nicht angegeben wird, ist die Standardeinstellung aufsteigend.

Beispiele:
Sort(“3;5;6;1”, “;”) = “1;3;5;6”
Sort("one;two;three", ";", 1) =“two;three;one”

StrAfter(«data»,«start after», «length»): Diese Funktion führt zu einer Zeichenfolge, die genau die angegebene Länge nach einem festgelegten Startzeichen aufweist.

Beispiel:
StrAfter("1234-5678", '-', 3)= Verwendet die 3 Zeichen nach dem Bindestrich (567)
StrAfter("1234-5678", '-')= Verwendet alle Zeichen nach dem Bindestrich (5678)

StrBefore(«data»,«start before», «length»): Diese Funktion führt zu einer Zeichenfolge, die genau die angegebene Länge vor einem festgelegten Startzeichen aufweist.

Beispiel:
StrBefore("1234-5678", '-', 2)= Verwendet die 2 Zeichen vor dem Bindestrich (34)
StrBefore("1234-5678", '-')= Verwendet alle Zeichen vor dem Bindestrich (1234)

SuppressBlankRows («string»): Gibt eine Zeichenfolge mit übersprungenen leeren Zeilen zurück. Es ermöglicht Ihnen, ein Objekt zu erstellen und die gewünschten Felder zu platzieren und diejenigen, die leer sind, zu unterdrücken.

Beispiel:
SuppressBlankRows ({Var0} & char(10) & {var1} & Char (10) & {Var2}) (Var0, Var1 und Var2 sind Variablen, Char(10) ist "\n" Symbole, die eine neue Zeile bedeuten)

Datenquellen/Wert	Formel	zeigt
Var0 = "Doris" Var1 = "Bull Run Ranch" Var2 = "Aurora"	SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2})	Doris Bull Run Ranch Aurora
Var0 = "Craig" Var1 = "" Var2 = "Chicago"	SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2})	Craig Chicago
Var0 = "Craig" Var1 = "" Var2 = "Chicago"	{Var0} & char(10) & {Var1} & char(10) & {Var2}	Craig Chicago

SuppressEmpty(«Liste», «Trennzeichen»): Löscht leere Einträge aus einer Liste von Elementen. Liste ist ein String-Argument, das mehrere Elemente mit einem Trennzeichen wie Komma oder Semikolon enthält.

Beispiel:
SuppressEmpty(“3;;;1”, “;”) = “3;1”

trim(«zeichenfolge») gibt das umgewandelte Argument *zeichenfolge* zurück. Alle am Anfang oder am Ende der Zeichenkette angetroffenen Leerzeichen werden gelöscht. Die Anzahl Leerzeichen zwischen zwei Wörtern wird auf eins reduziert.

Beispiel:
trim(" Bestellschein ") = Bestellschein

trima(«zeichenfolge») gibt das umgewandelte Argument *zeichenfolge* zurück. Alle vorhandenen Leeräume werden gelöscht.

Beispiel:
trimall("Paris / New York / Rom") = Paris/NewYork/Rom

Unique(«Liste», «Trennzeichen», «Vergleichszeichen»): Löscht doppelte Einträge aus einer Liste von Elementen. Liste ist ein String-Argument, das mehrere Elemente mit einem Trennzeichen wie Komma oder Semikolon enthält. Das optionale Vergleichszeichen gibt an, ob beim Vergleich die Groß- und Kleinschreibung beachtet wird oder nicht: 0 = Groß- und Kleinschreibung beachten, 1 = Groß- und Kleinschreibung ignorieren. Wenn das Vergleichszeichen nicht angegeben wird, ist die Standardeinstellung Groß- und Kleinschreibung beachten.

Beispiele:
Unique(“3;5;3;1”, “;”) = “3;5;1”
Unique("one;ONE;two", ";", 1) = “one;two”

upper («zeichenfolge») gibt das Argument *zeichenfolge* in Großbuchstaben umgesetzt zurück.

Beispiel:
upper("Bestellschein") = BESTELLSCHEIN

ztrim («value») : löscht die Nullen links des numerischen Wertes. Das Feld muss numerisch sein.

Beispiel:
Zeigt ein Feld mit dem Name WEIGHT den Wert 000200 an, dann
ztrim(weight) = 200

Festlegen der Eigenschaften der Formeldatenquellen

1. Geben Sie die Formel direkt in das Bearbeitungsfeld ein.

   -oder-

   Wählen Sie die gewünschten Elemente aus, und klicken Sie dann auf **Einfügen**.

2. Klicken Sie auf '**OK**'.

Tipp Sie können ein Element einfügen, indem Sie darauf doppelklicken.

Hinweis: Wenn eine in der Formel verwendete Variable einen Namen hat, in dem das Zeichen &+-*/<>=^%,!\ oder enthalten ist, muss dieses Zeichen von geschweiften Klammern {} umgeben werden.

Hinweis: Wenn der Wert nicht richtig ist, befolgen Sie die Anweisungen am Bildschirm, und nehmen Sie die erforderlichen Änderungen vor. Wenn der Wert nicht richtig ist, befolgen Sie die Anweisungen am Bildschirm, und nehmen Sie die erforderlichen Änderungen vor.

Praktische Übung : Berechnung des spezifischen "Modulo"

In Ihrer Software sind Modulo-Berechnungsfunktionen integriert, die es ermöglichen, ein Prüfzeichen in die Strichcodedaten einzufügen.

In manchen Fällen ist die Berechnung des Prüfzeichens jedoch spezifisch, so dass die in die Software integrierten Funktionen Ihre Anforderungen nicht erfüllen. Sie müssen die Funktion in diesem Fall selbst erstellen.

In dieser Sequenz wird gezeigt, dass es möglich ist, mit Hilfe von Formeln jede beliebige Berechnung von Prüfzeichen durchzuführen.

Im hier gezeigten Beispiel wird die Berechnung eines Prüfzeichens für einen Strichcode **2/5 interleaved** veranschaulicht.

Berechnungsmethod für ein Prüfzeichen

Bei dieser Methode wird das erste Zeichen des Werts mit 1, das zweite mit 2, das dritte mit 1 multipliziert usw.

- Öffnen Sie das Etikett mit dem Titel CHECK_01.lab im Ordner InstallDir\Samples\Forms\Tutorial.

Die Berechnung des Prüfzeichens erfolgt anhand des Parameters Postennummer.

Zur Berechnung des Gewichts

1. Klicken Sie auf die Verzweigung Formel des Registers Datenquellen des Dokument-Browsers und fügen Sie eine Formel hinzu.
2. Geben Sie die folgende Formel ein :
   mid(POSTENNUMMER,1,1)&
   mid(POSTENNUMMER,2,1)*2 &
   mid(POSTENNUMMER,3,1)&
   mid(POSTENNUMMER,4,1)*2 &
   mid(POSTENNUMMER,5,1)

   Das Ergebnis ist 2120103; tatsächlich wird folgendes erzielt:
   2*1 = 2
   6*2 = 12
   0*1 = 0
   5*2 = 10
   3*1 = 3

   Die Verkettung der Ergebnisse ergibt 2120103.
3. Ordnen Sie der Formel den Namen WEIGHTED zu.

Um das Ergebnis zur Berechnung des Gewichts hinzuzufügen:

Im folgenden Schritt werden die Ziffern addiert, die sich aus der vorherigen Formel ergeben:

1. Erstellen Sie eine zweite Formel, der Sie den Titel SUM zuordnen.
2. Geben Sie im Eingabefeld folgenden Ausdruck ein:
   mid(weighted,1,1)+mid(weighted,2,1)+
   mid(weighted,3,1)+mid(weighted,4,1)+

mid(weighted,5,1)+mid(weighted,6,1)+ mid(weighted,7,1)

Das Ergebnis ist 9; tatsächlich wird folgendes erzielt:
2+1+2+0+1+0+3= 9.

Um das Prüfzeichen zu berechnen:

Der Wert des Prüfzeichens wird mit Hilfe des vorherigen Ergebnisses berechnet.

1. Erstellen Sie eine dritte Formel, der Sie den Titel CHECK DIGIT zuordnen.
2. Geben Sie im Eingabefeld folgenden Ausdruck ein:
   if((SUM % 10)>0,10-SUM % 10,0)

   Das Ergebnis ist 1; tatsächlich wird folgendes erzielt:
   SUM % 10 = 9 (% = modulo).
   Da 9 höher als 0 ist, wird die Subtraktion ausgeführt: 10- 9 = 1.

Anmerkung: Wenn das Ergebnis gleich 0 wäre, hätte das Prüfzeichen den Wert 0.

Zur Berechnung des zu codierenden Parameters:

Bei der Erstellung des Strichcodes muss der zu codierende Parameter enthalten sein, d. h. der Wert der Variablen LOTNUMBER verkettet mit dem Wert des Prüfzeichens (CHECK DIGIT).

1. Erstellen Sie eine vierte Formel, die Sie DATA nennen.
2. Geben Sie im Eingabefeld folgenden Ausdruck ein:
   LOT NUMBER & CHECK DIGIT.

Zur Erstellung des Strichcode

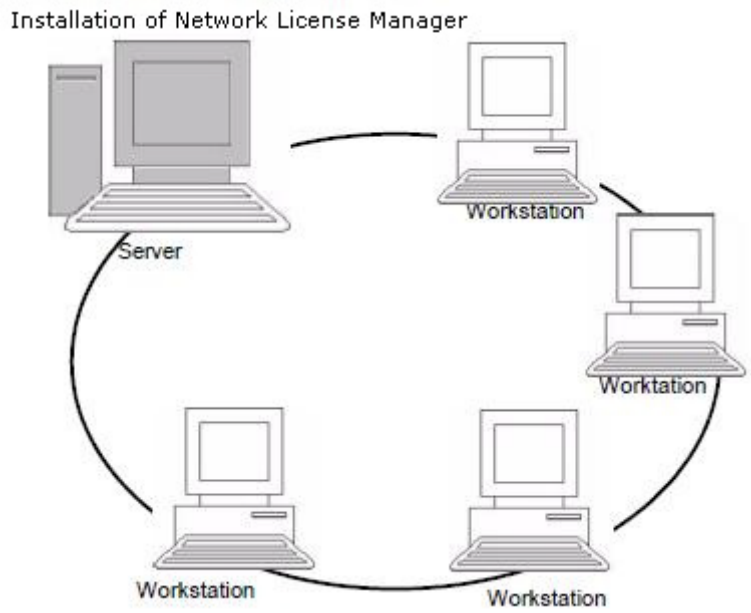
1. Klicken Sie auf das Werkzeug Strichcode in der Werkzeugleiste für Erstellung, wählen Sie anschließend die Formel DATA aus und ziehen Sie diese auf das Etikett.
2. Wählen Sie den Strichcode aus und wählen Sie über das Kontextmenü Objekteigenschaften aus. Das Dialogfenster Strichcode wird angezeigt.
3. Konfigurieren Sie das Dialogfenster entsprechend Abbildung 21.
   Symbologie: 2/5 Interleaved
   Prüfziffer: Keine

Netzwerkpaket

Funktionsbeschreibung

Das Netzwerkpaket (Mehrbenutzer) ermöglicht es Ihnen, den Zugriff auf Etikettier-Softwarelizenzen über das Netzwerk zu steuern. Wenn Sie dieses Dienstprogramm verwenden, können mehrere Benutzer gleichzeitig auf die Label-Design-Software von jedem beliebigen Standort im Netzwerk zugreifen.

Um eine Netzwerk-/Mehrplatzversion Ihrer Etikettier-Software zu verwenden, müssen Sie die **Network License Manager** auf dem Server oder der Station installieren, die als Server fungieren soll. Installieren Sie anschließend die Etikettier-Software auf allen Stationen.



Installationsvorgang

Vorbemerkung

Vor der Installation der Software muss der Netzwerkadministrator zunächst für die Benutzergruppe die Struktur des Netzwerks festlegen. Dies umfasst im Einzelnen:

- Festlegen des Lizenzservers auf dem die **Network License Manager** und der Dongle installiert werden.
- Festlegen der Stationen, d. h. der Client-Rechner, auf denen die Etikettier-Software zur Verfügung stehen soll.

Beschreibung von Network License Manager

Network License Manager unterstützen die Verwendung der Netzwerkkonfiguration Ihrer Etikettier-Software.

Network License Manager umfasst:

- Den Network License Manager (License Service)
- **Den Network Settings Wizard**: Der **Network Settings Wizard** ist ein Assistent, der Sie bei der Festlegung der Netzwerkkonfiguration unterstützt.
- **Benutzer-Manager**: Der **Benutzer-Manager** wird zusammen mit den **Network License Manager** installiert. Er ermöglicht die Beschränkung der Zugriffsrechte auf die Etikettier-Software im Netzwerk.

Installieren von Network License Manager

Bevor Sie die Etikettier-Software auf allen Stationen installieren, auf denen sie laufen soll, müssen Sie die **Network License Manager** auf dem Server installieren, um die Netzwerkeinstellungen vorzunehmen.

Installieren der Network License Manager auf dem Server

1. Legen Sie die DVD für die Installation in das zugehörige Laufwerk ein. Das Installationsfenster wird geöffnet.
   Falls die DVD nicht automatisch ausgeführt wird:

   Wählen Sie **Start > Ausführen** und geben Sie den Buchstaben des DVD-Laufwerks sowie die Datei index.hta ein (z. B.: D:\index.hta).
2. **License Service** und der **Benutzer-Manager** gehören. Klicken Sie anschließend auf die Schaltfläche Installieren.
3. Befolgen Sie die Anleitungen auf dem Bildschirm
4. Wenn Sie die Einstellungen der Netzwerkkonfiguration bearbeiten möchten, starten Sie auf dem Server den **Network Settings Wizard**. Standardmäßig werden die Einstellungen spezifisch an die einzelnen Stationen angepasst, wenn Sie die Konfiguration nicht ändern.

Konfiguration

Alle notwendigen Werkzeuge zur Konfiguration der Netzwerkversion sind in der **Network License Manager Toolbar** verfügbar, auf die über die Windows-Taskleiste (Systray) zugegriffen werden kann.

Doppelklicken Sie in der Windows-Taskleiste auf das Symbol , um die **Network Toolbar** anzuzeigen.

Der **Network Settings Wizard** unterstützt Sie bei der Einrichtung Ihrer Netzwerkversion.

1. Klicken Sie zum Starten des Network Settings Wizard auf das Symbol .
2. Legen Sie in Schritt 1 des Assistenten einen Einstellmodus fest: **generisch, benutzerspezifisch** oder **stationsspezifisch**.

    - **Generisch**: Für sämtliche Benutzer auf allen Stationen gelten dieselben Einstellungen. (Benutzer.ini).
    - **Benutzerspezifisch**: Jeder Benutzer kann auf jeder beliebigen Station seine spezifischen Einstellungen aufrufen. (Benutzername.ini).
    - **Stationsspezifisch**: Für jede Station gelten eigene Einstellungen (Station.ini)

3. Wo diese Einstellungen gespeichert werden sollen. Sollen die betreffenden Einstellungen zwischen den verschiedenen Stationen freigegeben werden, geben Sie einen Pfad im Netzwerk ein, der von allen Stationen aus zugänglich ist. (Beispiel:TKDongle).
4. Wo die globalen Daten (Variablen, Listen, Druckprotokoll usw.) gespeichert werden sollen. Stellen Sie sicher, dass alle Benutzer über die richtigen Berechtigungen

Konfigurieren des Benutzer-Manager

Wenn Sie die Netzwerk-Zugriffsrechte der Benutzer der Etikettier-Software definieren möchten, müssen Sie dies bereits bei der Installation vornehmen (ziehen Sie die Hilfe des Benutzer-Manager zu Rate).

Klicken Sie auf das Symbol **Benutzer-Manager** in der Symbolleiste Netzwerk.

Starten des License Service

Bevor Sie die Etikettier-Software auf allen Stationen installieren, müssen Sie sicherstellen, dass der License Service gestartet wurde.

Der Lizenzdienst wird als Dienst installiert. Dieser Dienst, der als **TkxWebLicenseServer** gekennzeichnet ist, wird automatisch aktiviert, wenn der Server gestartet wird.

Starten von Service Controller

Klicken Sie auf das Symbol  auf der Symbolleiste **Network License Maanger**

- oder -

Doppelklicken Sie auf die Datei TkxWebLicenseServerController.exe.

- oder -

Rechtsklick auf das **Netzwerkleisten-**Symbol auf der Windows-Taskleiste und **Lizenzdienststeuerung** (License Service Controller) auswählen**.**

Installieren der Software auf den Stationen

Die Etikettier-Software muss auf allen Stationen installiert werden, auf denen sie verwendet werden soll.

Installieren der Software auf einer Station

1. Legen Sie die DVD für die Installation in das zugehörige Laufwerk ein.
   Das Installationsfenster wird geöffnet.
   Falls die DVD nicht automatisch ausgeführt wird:
   Wählen Sie **Start > Ausführen** und geben Sie den Laufwerkbuchstaben des DVD-Laufwerks sowie die Datei INDEX.HTA ein (z. B.: D:\index.hta).
2. Wählen Sie das Produkt, das installiert werden soll, und klicken Sie auf die Schaltfläche Installieren. Folgen Sie den Hinweisen auf dem Bildschirm.
3. Starten Sie die Etikettier-Software. Wählen Sie im Menü Extras die Option **Netzwerkverwaltung** aus.
   - oder -

Sie wählen vom Windows Startmenü aus in der Etikettensoftwaregruppe die Verknüpfung **Netzwerkadministration**.

4. Aktivieren Sie **Einsatz im Netz**.
5. Wählen Sie den **Netwerklizenztyp**.

   - Die **Lizenz für den gemeinsamen Ordner** nutzt die Dateifreigabefunktion von Windows zur Kommunikation zwischen der Software und dem Lizenzmanager.

   - **Web-Lizenz** verwendet http/https communication zwischen der Software und dem License Manager.
6. Geben Sie den **Server-Port** an, wenn Sie den **Web-Lizenztyp** gewählt haben.
7. Klicken Sie auf **Bearbeiten**, um den Server, auf dem der **License Service** und der Dongle installiert sind, auszuwählen.
   - oder -
   Klicken Sie auf **Suchen**, um eine automatische Suche nach dem Server durchzuführen, auf dem der **License Service** installiert ist.

   Wenn die Netzwerkeinstellungen bereits konfiguriert wurden, wird eine Meldung mit der Frage, ob Sie die aktuelle Netzwerkkonfiguration verwenden möchten, angezeigt.
8. Zum Bearbeiten oder Konfigurieren der Netzwerkeinstellungen klicken Sie auf die Schaltfläche **Network Settings Wizard**.
9. Klicken Sie auf **OK**.
10. Starten Sie das Programm neu.



France
+33 (0) 562 601 080

Germany
+49 (0) 2103 2526 0

Singapore
+65 6908 0960

United States
+1 (414) 837 4800

Copyright 2026 TEKLYNX Corporation SAS. All rights reserved. LABEL MATRIX, LABELVIEW, CODESOFT, LABEL ARCHIVE, SENTINEL, PRINT MODULE, BACKTRACK, TEKLYNX CENTRAL, TEKLYNX, and Barcode Better are trademarks or registered trademarks of TEKLYNX Corporation SAS or its affiliated companies. All other brands and product names are trademarks and/or copyrights of their respective owners.

www.teklynx.com