

TUTORIAL

Versão 2026

As informações contidas neste guia não são de natureza contratual e podem estar sujeitas à mudanças sem aviso prévio.

O software descrito neste guia é vendido sob um acordo de licença. O software pode ser usado, copiado ou reproduzido apenas em conformidade com os termos do acordo.

Nenhuma parte deste guia pode ser copiada, reproduzida ou transmitida por qualquer forma ou por qualquer meio ou para qualquer outra finalidade que não uso próprio do comprador sem a permissão por escrito da TEKLYNX Corporation .

© 2026 TEKLYNX Corporation SAS

Todos os direitos reservados.

Índice

Índice 3

Sobre este manual 6

Convenções tipográficas 6

Sobre seu produto 6

Conectando a uma base de dados 7

Visão Geral 7

Instalando uma fonte de dados ODBC 8

Importando dados 8

Criando objetos da variável 9

Active Query Builder 10

Active Query Builder 10

Iniciar 10

Adicionando um objeto na consulta 11

Editando propriedades do objeto 12

Ligar tabelas 12

Classificando campos de saída 13

Definindo critérios 14

Definindo uma consulta parametrizada 14

A grade de resultado da Consulta 15

Database Manager 17

Janela Editar Base de Dados 17

Selecionar registros de acordo com seu conteúdo 17

Selecionar todos os registros idênticos 17

Selecionar um registro idêntico 18

Criar um novo registro 18

Modificar um registro ... 18
Excluir um registro ... 18
Janela de Consulta da Base de Dados ... 19
Adicionar uma consulta ... 19
Selecionar/desselecionar um ou mais campos ... 19
Modificar a ordem dos campos selecionados ... 19
Criar um filtro usando dados pré-definidos ... 20
Aplicar um operador lógico em vários filtros ... 20
Remover um filtro ... 20
Modificar um filtro no SQL ... 20
A janela Imprimir ... 21
Selecionar um documento para ser impresso ... 21
Selecionar um modelo de etiqueta existente ... 21
Selecionar um documento de um campo ... 22
Selecionar uma impressora ... 22
Fórmulas ... 23
A fonte de dados da Fórmula ... 23
Sobre funções ... 23
Operadores ... 24
Verificar funções de cálculo do caractere ... 25
Funções de conversão ... 27
Funções de conversão do ATA 2000 ... 34
Funções de data e hora ... 36
Funções lógicas ... 43
Funções matemáticas ... 45
Funções de texto ... 48
Definindo as propriedades de uma fonte de dados da Fórmula ... 54
Exercício: Criando um módulo específico ... 55

Para calcular o peso 55
Para adicionar o resultado do cálculo de peso: 56
Para calcular o dígito de verificação: 56
Para calcular os dados a serem codificados: 56
Para criar o código de barras: 57
Instalando a versão de rede 58
Descrição 58
Procedimento de Instalação da Rede 58
Configuração da rede 58
Descrição do Gerenciador de Rede 59
Instalando os Utilitários de Usuário e Rede no servidor 59
Configuração 60
Iniciando o Serviço de Licença 61
Para iniciar o Controlador do Serviço de Licença 61
Instalando o software nas estações de trabalho 61
Para instalar o software em uma estação de trabalho 61

Sobre este manual

Convenções tipográficas

Este manual distingue entre diferentes tipos de informações usando as seguintes convenções:

- Termos tirados da interface propriamente, como comandos, aparecem em **negrito**.
- Os nomes de tecla aparecem todos em letra maiúscula. Por exemplo: "Pressione a tecla SHIFT."
- As listas numeradas indicam que existe um procedimento para seguir.
- Quando a conjunção -ou- aparece próxima a um parágrafo, quer dizer que existe a opção de outro procedimento para executar uma determinada tarefa.
- Quando um comando do menu tem submenus, o nome do menu seguido pelo comando para selecionar aparece em negrito. Deste modo, "Vá até **Ficheiro > Abrir**" significa selecionar o menu **Ficheiro**, então o comando **Abrir**.

Sobre seu produto

Algumas das funções descritas neste manual não podem estar disponíveis no seu produto.

Para a lista completa de recursos específicos disponíveis no seu software, consulte a folha de especificação fornecida com o produto.

Conectando a uma base de dados

Visão Geral

Neste capítulo vamos vincular uma etiqueta (o recipiente) com uma base de dados (o conteúdo). Para fazer isto, usaremos as conexões ODBC (Open DataBase Connectivity) ou OLE DB.

As bases de dados permitem a você armazenar dados. Todos os dados são organizados em tabelas bidimensionais nas quais são denominados de relacionamento. Cada linha em uma tabela é denominada de registro. O propósito de um registro é gerenciar um objeto, as propriedades do qual são organizadas através de diferentes colunas da tabela na forma de campos.

Uma base de dados pode conter várias tabelas. Para vincular as diferentes tabelas dentro de uma base de dados fornecida, usamos ligações. Um exemplo concreto posteriormente neste capítulo demonstrará como as ligações são criadas.

ODBC

As fontes de dados ODBC tornam possível acessar dados que pertencem a uma grande variedade de sistemas de gerenciamento da base de dados. O ODBC torna fácil vincular um aplicativo como seu software de desenho da etiqueta com um determinado número de bases de dados. O software vem com vários drivers ODBC. Estes permitem a você acessar os tipos mais comuns de bases de dados.

Alguns dos drivers mais comuns estão listados abaixo:

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

OLE DB

O OLE DB é um conjunto de interfaces que fornece acesso para todas as bases, não importando o tipo, formato ou local. Fornece componentes como interfaces de acesso, drivers de consulta, e assim por diante. Estes componentes são denominados "provedores".

O exemplo abaixo descreve um processo de conexão quando uma base de dados não está conectado ao seu software.

Instalando uma fonte de dados ODBC

O processo descrito abaixo usa o modo de criação direto. Se você desejar, é possível usar o assistente ao selecionar **Assistente** no menu de contexto.

Conectando à base de dados TKTraining.mdb

1. Selecione **Ferramentas > Administrador ODBC**.
2. Clique na guia **Fonte de Dados do Sistema** (DSN), então clique em **Adicionar**.

Nota: Você pode definir fontes de dados com Nomes das Fontes de Dados do sistema (DSNs). Estas fontes de dados são únicas para um computador específico, mas não para um usuário específico. Qualquer usuário com os direitos necessários pode acessar um DSN do sistema.

3. Selecione **Microsoft Access Driver** então clique em **Finalizar**.
4. Digite "TK Training Level 2" no campo **nome da Fonte de Dados**.
5. Clique em **Selecionar** e selecione a base de dados TKTraining.mdb, que pode ser localizada no diretório InstallDir\Samples\Forms\Tutorial.
6. Clique no botão **Opções**. Marque a opção **Somente Leitura**. Esta opção torna possível abrir a base de dados ao mesmo tempo que seu software de etiquetagem sem causar quaisquer problemas de leitura/escrita.
7. Clique em **OK** na caixa de diálogo **Configurar ODBC Microsoft Excel**.

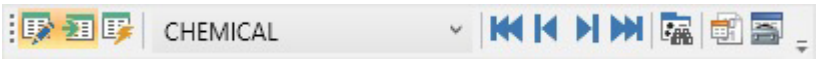
Importando dados

Após a base de dados estar conectada ao seu software, você tem que conectá-la ao seu documento.

1. Abra a etiqueta PRODUCT_WS3.
2. Selecione **Fontes de Dados > Base de Dados > Criar/Modificar Consulta**.
3. Selecione TK Training Level 2 na lista **Selecionar Fonte de Dados**.
4. Selecione "Fruits" na lista **Selecionar Tabela**.
   Os campos da base de dados aparecem na lista **Selecionar Campos**.
5. Selecione os campos "ProdName", "Origin", "Weight" e "Reference".
6. Clique no botão . Ele permite que os registros selecionados sejam classificados em ordem alfabética ou numérica, crescente ou descendente.

7. Selecione "Reference" como **Chave de Classificação** e "Ascending" como **Ordem de Classificação**.
8. Guarde a consulta no diretório InstallDir\Samples\Forms\Tutorial\PRODUCT_WS4_ODBC.CSQ.
9. Clique em **OK**.
   As variáveis são automaticamente criadas e são listadas na ramificação da **Base de Dados** e na visualização das **Fontes de Dados**.

Para visualizar ou imprimir os diferentes valores que seu objeto pode utilizar, use a barra de navegação. Você também pode imprimir da janela **Resultado da Consulta**.



Criando objetos da variável

1. Selecione as variáveis criadas, listadas na lista **Base de Dados** na visualização das **Fontes de Dados**, então arraste e solte na área de trabalho.
2. Selecione **Texto** no menu de contexto.

Active Query Builder

Active Query Builder

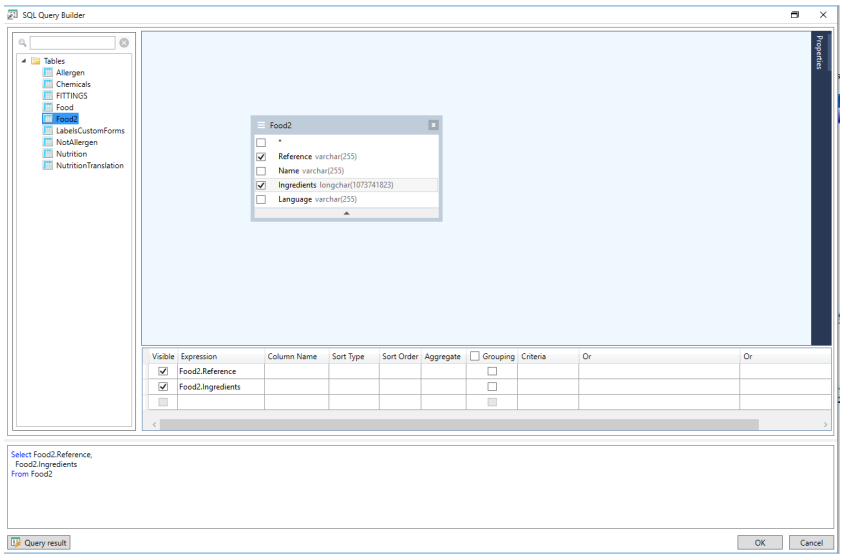
O Active Query Builder é um componente do construtor de consulta visual que lhe permite construir consultas SQL complexas usando uma interface de construção de consulta visual intuitiva.

Para trabalhar com Active Query Builder, você deve ter conhecimento básico de conceitos do SQL. O Active Query Builder ajudará você a escrever o código SQL correto ocultando detalhes técnicos, mas você deve ter uma compreensão dos princípios de SQL para alcançar os resultados desejados.

Iniciar

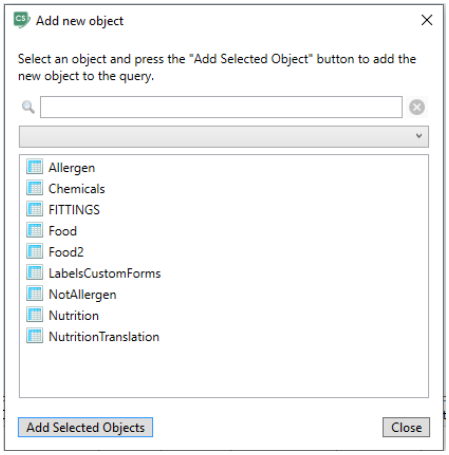
A janela principal Ativar Construtor de Consulta é dividida nas seguintes partes:

- A **área de Construção da Consulta** é a área principal onde a representação visual da consulta será exibida. Esta área permite a você definir objetos da base de dados de origem e tabelas derivadas, definir links entre eles e configurar propriedades de tabelas e links.
- O **painel Colunas** está localizado abaixo da área de Construção da Consulta. É utilizado para executar todas as operações necessárias com expressões e colunas de saída da consulta. Aqui você pode definir nomes alternativos do campo, classificar e agrupar, e definir critérios.
- A **área de visualização da Tabela** está localizada à esquerda. Aqui você pode navegar pela sua consulta e localizar rapidamente qualquer parte dela. Use o campo **Pesquisar** na parte superior do painel para especificar sua pesquisa.
- O controle de página acima da área de Construção da Consulta permitirá a você trocar entre a consulta principal e subconsultas.
- A área pequena no canto da área de Construção da Consulta marcada pela letra **Q** é a união do controle de manipulação da subconsulta. Aqui você pode adicionar novas subconsultas da união e executar todas as operações necessárias.



Adicionando um objeto na consulta

Para adicionar um objeto na consulta, clique com o botão direito na Área de Construção da Consulta e selecione o item **Adicionar Objeto** do menu suspenso.

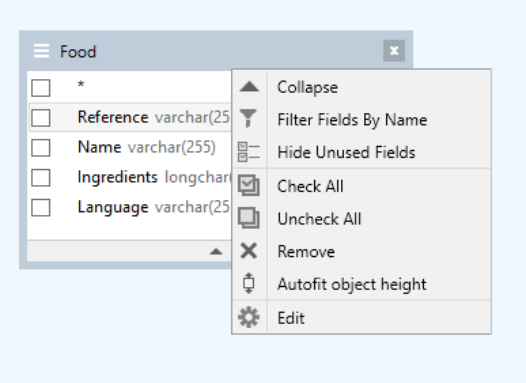


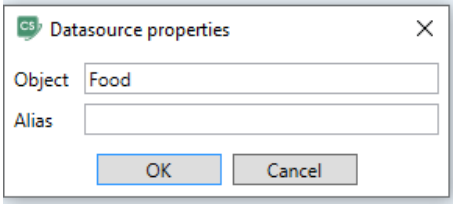
A janela **Adicionar Novo Objeto** permite adicionar vários objetos de uma vez. Os elementos da tabela são listados na caixa de combinação. O filtro **Mostrar todos os objetos** mostrará todos os objetos disponíveis para a seleção. O **campo de pesquisa** na parte superior da janela é usado para procurar um objeto necessário. Você pode selecionar um ou vários objetos segurando a **tecla CTRL** e então pressionando o botão **Adicionar Objetos Selecionados** para adicionar estes objetos na consulta. Você pode repetir esta operação várias vezes. Após terminar de adicionar objetos, clique no botão **Fechar** para ocultar esta janela.

Para remover um objeto da consulta, clique no botão **Fechar** no cabeçalho do objeto

Editando propriedades do objeto

Você pode mudar as propriedades de cada objeto adicionado à consulta ao clicar com o botão direito no objeto e selecionar a opção **Editar** do menu suspenso or simplesmente clicar duas vezes no cabeçalho do objeto.

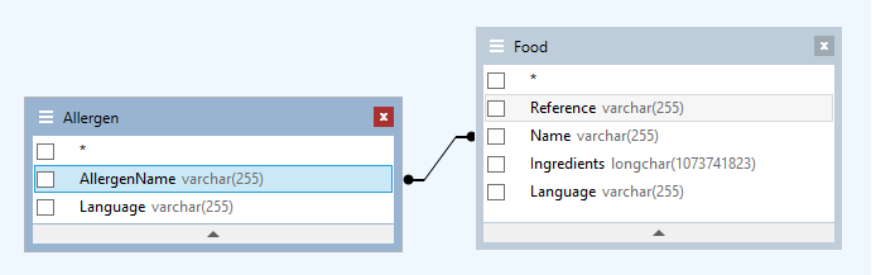




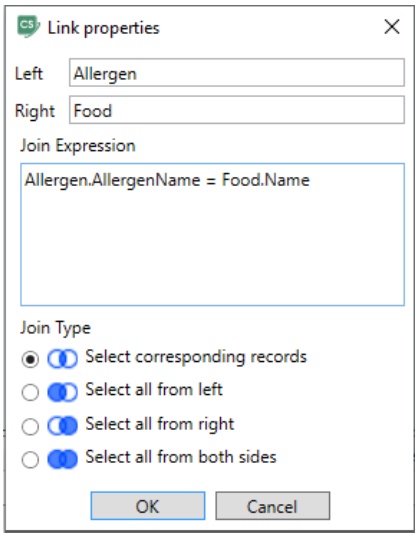
O diálogo **Propriedades da Fonte de Dados** pode variar de servidor para servidor, mas a propriedade **Nome Alternativo** é a mesma para todos os servidores da base de dados.

Ligar tabelas

Para criar um link entre dois objetos (isto é, juntá-los), você deve selecionar o campo para vincular aos objetos e arrastá-lo para o campo correspondente do outro objeto. Após arrastar o campo para o outro objeto, uma linha conectando os campos vinculados aparecerá.



O tipo de ligação pré-definido é INNER JOIN. Isto significa que somente registros correspondentes em ambas as tabelas serão incluídos no conjunto de dados resultante. Para definir outros tipos de ligações, clique com o botão direito no link e selecione a opção **Editar** no menu suspenso ou clique duas vezes sobre ele para abrir a janela **Propriedades do Link**. Esta janela permite a você definir o tipo de ligação e outras propriedades do link.



Para remover um link entre objetos, clique com o botão direito na linha do link e selecione a opção **Remover** no menu suspenso ou selecione a linha do link e pressione o botão **Excluir**.

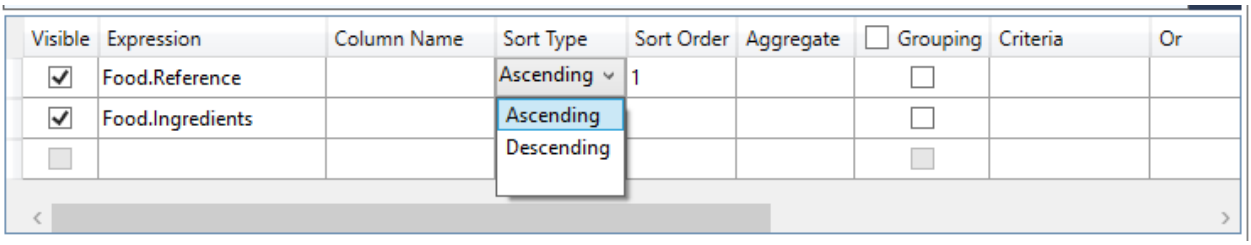
Classificando campos de saída

Para classificar os campos de consulta de saída, use as colunas **Tipo de Classificação** e **Ordem de Classificação** no **Painel de Colunas**.

A coluna **Tipo de Classificação** permite a você especificar o modo no qual os campos serão classificados, em ordem **Crescente** ou **Decrescente**.

A coluna **Ordem de Classificação** permite a você definir a ordem na qual os campos serão classificados, se mais de um campo for usado para classificação..

Para desativar a classificação por um campo, limpe a coluna **Tipo de Classificação** do campo.



Definindo critérios

Para definir critérios da expressão listada no **Painel de Colunas**, use a coluna **Critérios**.

Nesta coluna, escreva o critério omitindo a expressão propriamente. Por exemplo, para obter o seguinte critério na sua consulta:

> **WHERE** (field >= 10) **AND** (field <= 20)
> você deve escrever
> >= 10 **AND** <= 20 na coluna **Critérios**.

Você também pode usar a **lista suspensa** na coluna **Critérios** para colar expressões e operadores comuns no campo de edição. Para criação de critérios avançados use o botão **Editor de expressão** na coluna**Critérios**. A caixa de diálogo **Editor de expressão** é exibida.

Você pode especificar vários critérios para uma única expressão usando as colunas **Ou**. Estes critérios serão concatenados na consulta usando o operador **OR**.

Definindo uma consulta parametrizada

O **Construtor de Consulta** permite a você criar uma consulta parametrizada onde o valor do parâmetro é mantido em uma variável.

Nota: Você deve ter criado uma variável antecipadamente.

1. Arraste e solte a tabela na qual sua consulta será executada.
2. Selecione os campos para os quais o critério ou critérios serão aplicados.
3. Na coluna **Critérios** ou no campo de edição do formato SQL, especifique a variável a ser usada como o objeto de um critério de pesquisa.

Exemplo: Para procurar pelo valor da variável Var0 que você criou antecipadamente:

- **No SQL:**

```
SELECT [Table].*
FROM [Table]
WHERE [Table].Champ = APPLICATION.DOCUMENT.Var0
```

- **Coluna de critério**
  = APPLICATION.DOCUMENT.Var0

4. Clique no botão **Resultado da consulta** para exibir o resultado de sua consulta.

A grade de resultado da Consulta

Para acessar a grade de **resultado da Consulta**, clique no botão  na caixa de diálogo Definindo uma consulta, na barra de ferramentas **Unir Banco de Dados** do navegador ou via menu **Fonte de Dados > Base de Dados > Visualizar os dados de resultado da consulta** .

Esta grade permite que o resultado de uma consulta seja exibido e permite a você procurar por um termo em particular e todas as suas ocorrências, e imprimir as etiquetas correspondentes.

Nota: A tag **"<dados Binários>"** é mostrada no diálogo de resultado da Consulta para campos do banco de dados que contêm o **tipo de dados LOB** (por exemplo, arquivos de imagens).

A grade de **resultado da Consulta** contém:

- **Procurar funções** :

  Campo **Pesquisar**, que permite a você inserir o campo que deseja pesquisar.

  Dados para pesquisar, permite a você inserir o valor para pesquisar.

  Procure pelo valor em qualquer lugar no campo ou no começo do campo .

- **Funções de navegação para procurar registros de resultado da consulta:**

  Primeiro registro 

  Registro anterior 

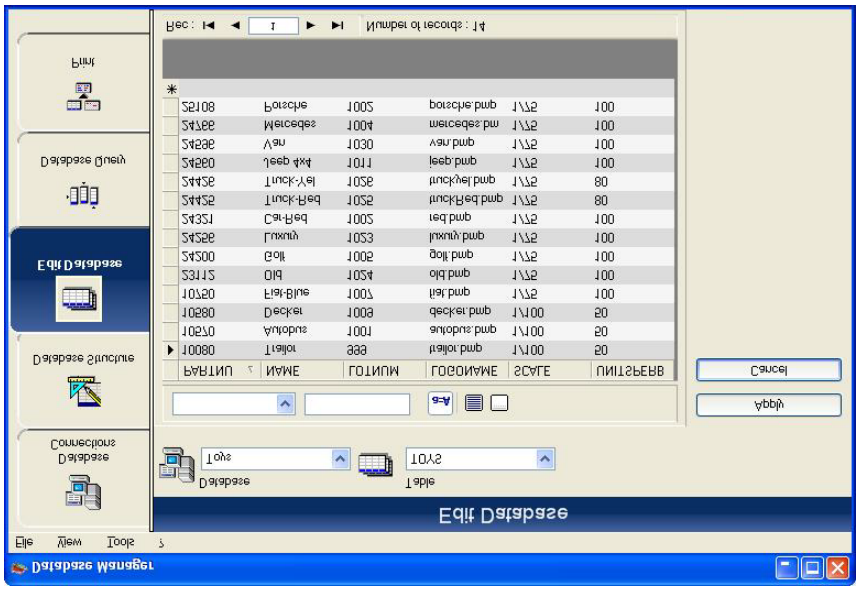
  Próximo registro 

  Último registro 

- **A grade de resultados**
  Exibe os resultados da pesquisa.
- **Consultar Novamente**
  Consulta novamente e atualiza a grade.

Database Manager

Janela Editar Base de Dados



A janela Editar Base de Dados é usada para gerenciar o conteúdo do ficheiro da base de dados. Nesta janela, você pode adicionar, modificar ou excluir os dados.

Estas ações dependem do tipo da base de dados. Os registros de ficheiro do Excel não podem ser modificados.

Selecionar registros de acordo com seu conteúdo

Use o conteúdo de um campo para localizar um registro.

1. Clique no botão da lista suspensa da tabela.
2. Clique nos dados exigidos.
3. Clique no campo **Entrada de Dados**.
4. Digite o valor exigido no campo **Entrada de Dados**.

Selecionar todos os registros idênticos

Pelo menos um registro deve ter sido localizado.

1. Clique no botão da lista suspensa.
2. Clique nos dados exigidos.
3. Clique no campo **Entrada de Dados**.
4. Insira os dados exigidos no campo **Entrada de Dados**.
5. Clique no botão **Selecionar Tudo** ().

Selecionar um registro idêntico

Pelo menos um registro deve ter sido localizado. Deve haver vários registros idênticos no campo **Pesquisar**.

Para selecionar um registro, use a ferramenta de pesquisa: 

Criar um novo registro

1. Clique em um campo na linha marcada com um asterisco.
2. Digite os valores exigidos nos campos correspondentes.
3. Clique em **Aplicar**.

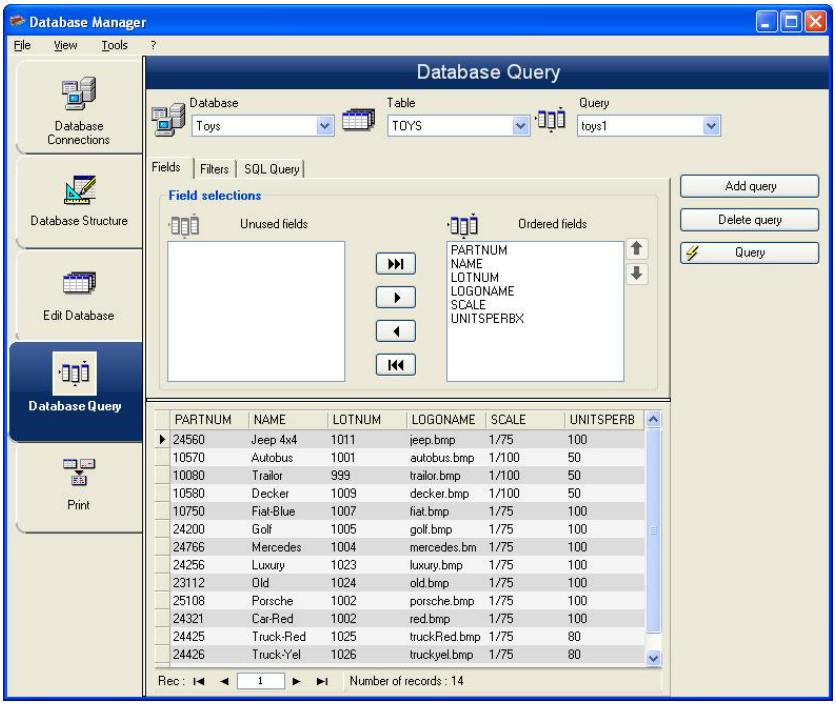
Modificar um registro

1. Clique nos dados que você deseja modificar.
2. Insira os dados exigidos.
3. Clique em **Aplicar**.

Excluir um registro

1. Clique no cursor da base de dados do campo exigido.
2. Clique com o botão direito no cursor da base de dados do campo exigido.
3. Clique em **Excluir Registro**.

Janela de Consulta da Base de Dados



A janela de Consulta da Base de Dados é usada para criar e aplicar vários filtros.

Adicionar uma consulta

1. Clique em **Adicionar consulta** na guia **Campos**.
2. Digite um nome para a consulta.
3. Clique em **OK**.

Selecionar/desselecionar um ou mais campos

1. Use  para selecionar os campos exigidos.
2. Clique em **Consultar** para atualizar a pré-visualização da base de dados.

Modificar a ordem dos campos selecionados

1. Clique no campo exigido na janela **Campos Ordenados**.
2. Clique na seta para cima ou para baixo para reordenar os campos.
3. Clique em **Consultar** para atualizar a pré-visualização da base de dados.

Criar um filtro usando dados pré-definidos

1. Selecione a guia **Filtros**.
2. Clique no botão **Adicionar Seta** ().
3. Em **Campo**, selecione na lista suspensa os dados exigidos.
4. No campo **Operador**, selecione o operador desejado da lista suspensa.
5. No campo **Valor**, digite o valor exigido.
6. Clique em **Consulta** para visualizar o resultado.

Aplicar um operador lógico em vários filtros

1. Clique no botão **Adicionar Seta** ().
2. No campo **Lógica**, selecione a lógica desejada (AND ou OR) da lista suspensa.
3. Crie um filtro (como descrito acima).
4. Clique em **Consulta** para visualizar o resultado.

Remover um filtro

Nota: Pelo menos um filtro deve existir.

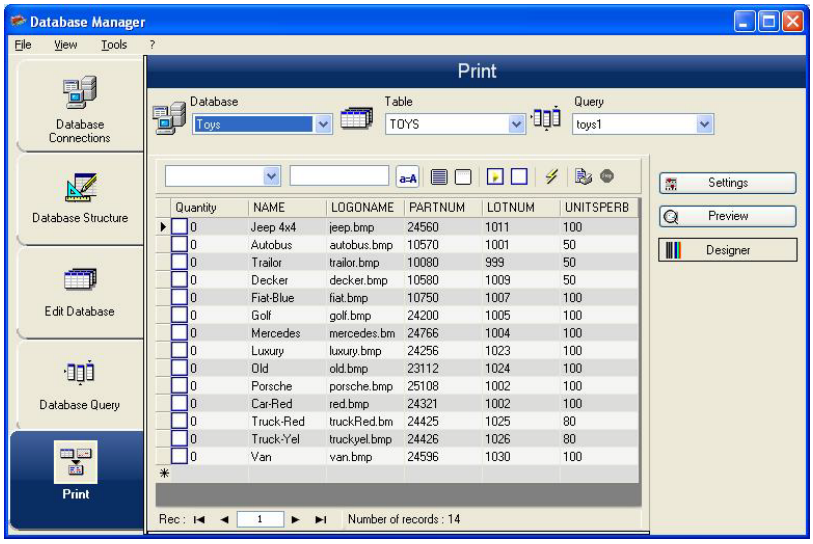
1. Clique no cursor da base de dados do campo exigido.
2. Clique no botão **Eliminar** da linha ().

Modificar um filtro no SQL

Nota: Pelo menos um filtro deve existir.

1. Selecione a guia **Consulta SQL**.
2. Marque **Modificar a consulta na linguagem SQL** para ativar a Consulta SQL e realizar mudanças manuais.
3. Clique em **Consulta** para visualizar o resultado.

A janela Imprimir



A janela Imprimir é usada para selecionar arquivos para imprimir, atribuir impressoras e definir vários parâmetros antes de iniciar a impressão.

Selecionar um documento para ser impresso

1. Selecione um documento de um ficheiro

2. Marque a caixa de seleção **Ficheiro** no grupo de nome **Etiqueta**.

   - ou -

1. Clique no botão **Assistente de Criação de Etiqueta** ( ).
2. Siga as instruções na tela.

Nota: A criação de uma etiqueta em relação à base de dados permite a você definir exatamente quais elementos são exigidos para posicionar cada campo da base de dados.

Selecionar um modelo de etiqueta existente

1. Clique no botão **Abrir um documento existente** ( ).
2. Selecione o ficheiro .lab .lbl.
3. Clique em **OK**.

Nota: Os botões de rádio do Campo nos grupos de opções do Nome da Etiqueta e Nome da Impressora permitem a você escolher a etiqueta ou impressora exigida, quando estas opções são definidas em um dos campos da base de dados ativa.

Selecionar um documento de um campo

Se a sua base de dados tiver o nome da etiqueta a ser usada para o trabalho de impressão em um dos campos, você poderá definir este campo como o local onde o Gerenciador da Base de Dados selecionará o ficheiro .lbl . Por exemplo, considere a seguinte base de dados:

Ref	Designação	Qt	Código	Labname
6574	Ref1	1	9876546321	Label1.lab
6354	Ref2	2	1236478855	Label2.lab
6987	Ref3	3	6987456321	Label1.lab
3684	Ref4	4	3698745632	Label3.lab

1. Marque a caixa **Campo** no grupo **Nome da Etiqueta**.
2. Selecione o campo exigido.

Selecionar uma impressora

1. Clique no botão **Adicionar** ou **Remover uma impressora** ().
2. Selecione a impressora exigida.

Fórmulas

A fonte de dados da Fórmula

A fonte de dados da Fórmula contém uma lista de fontes de dados.

Estas fontes de dados são preenchidas por combinações de operadores, constantes, fontes de dados, variáveis de controle, fórmulas e funções. Os dados podem ser numéricos ou alfanuméricos.

Para executar um cálculo dentro de um documento, você deve primeiro criar uma fonte de dados da **Fórmula**.

Esta fonte de dados tem uma caixa de diálogo específica permitindo a você definir a função exigida para uma determinada fórmula.

Sobre funções

Nota: As definições regionais do sistema Windows podem afetar o resultado de algumas fórmulas.

As funções são fórmulas pré-definidas que executam cálculos usando valores denominados parâmetros, adicionados em uma determinada ordem, denominada sintaxe.

As funções são usadas para retornar um valor numérico, string alfanumérica ou lógica, que é o resultado de um cálculo ou uma operação.

Há grupos de funções na definição da fórmula:

- Verificar funções de cálculo do caractere
- Funções de conversão
- Funções de conversão do ATA 2000
- Funções de data e hora
- Funções de lógica
- Funções de matemática
- Funções da string de Texto/Caractere

Operadores

Os operadores são símbolos matemáticos indicando uma operação que será executada. Há diferentes tipos de operadores: aritmética, comparativa, concatenação e lógica.

O programa contém operadores de matemática, comparativo, concatenação e lógica.

Operadores aritméticos

Operador	Usado
*	Multiplica dois números.
+	Adiciona dois números juntos.
-	Subtrai um número de outro, ou atribui um valor negativo para um operador.
/	Divide um número por outro.
^	Eleva um número a uma potência.
%	Módulo.

Nota: Algumas fórmulas não dão erro quando normalmente seria de esperar que isso ocorresse. Alguns exemplos são: 1/0, 0/0, 0.0/0.0, 1/000 e 0^(-2). Tal é intencional e visa evitar o bloqueio do processo de impressão. O utilizador precisa de definir o comportamento desejado, adicionando controlos extra, se necessário.

Operadores comparativos

Operador	Significando
<	Menor que
<=	Menor que ou igual a
>	Maior que
>=	Maior que ou igual a
=	Igual a
<>	Diferente de

Operador de concatenação

Usado para combinar duas strings.

Operador	Significando
&	Concatenação de duas strings.

Operador lógico

(Veja também Funções Lógicas)

Operador	Significando
!	Não lógico

Verificar funções de cálculo do caractere

Nota: As definições regionais do sistema Windows podem afetar o resultado de algumas fórmulas.

addmodulo10 («string»): Retorna um valor inteiro com o caractere de verificação Modulo 10 adicionado ao final.

addmodulo10_212(«string»): Retorna um caractere de verificação modulo 10_212 (uma variável do caractere de verificação modulo 10 padrão) e inclui a string de texto atual e a soma de verificação no final da string.

addmodulo43(«string»): Retorna um caractere de verificação modulo 43 e inclui a string de texto atual e a soma de verificação no final da string.

bimodulo 11 («string»): Retorna os dois caracteres de verificação modulo 11 (Code 11).

canadacustomscd («string»): Retorna o caractere de verificação do Padrão da Alfândega do Canadá.

Check103(«string»): Calcula a soma de verificação Mod103.

check 128 («string»): Retorna o caractere de verificação Code 128 e EAN-128.

Check PZN(«string»): Calcula o dígito de verificação para código de barras PZN. O código de barras PZN é baseado no Code 39 usado para produtos farmacêuticos na Alemanha.

CheckSum2Mod47(«string»): Calcula a soma de verificação Mod47.

CheckSumMod10(«string»): Calcula a soma de verificação Mod10.

CheckSumMod10_Codabar(«string»): Calcula a soma de verificação Mod10 específica para Codabar.

CheckSumMod10_MSI(«string»): Calcula a soma de verificação Mod10 para códigos de barra MSI.

CheckSumMod11_3Suisse(«string»): Calcula a soma de verificação Mod10 para 3Suisses.

CheckSumMod34(«string»): Calcula a soma de verificação Mod34.

checkupce («string»): Retorna o caractere de verificação UPC-E (6 caracteres obrigatórios).

CRC16(«string»): Retorna um valor CRC-16.

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

modulo 10 («string»): Retorna um caractere de verificação modulo 10 do Code 2 de 5 interlaçado, EAN-13, EAN-8, EAN-128 K-Mart, VPCA.

modulo10_212(«string»): Retorna uma variável do caractere de verificação modulo 10.

modulo10IBM(«string»): Retorna um caractere de verificação modulo 10 de IBM (uma variável do caractere de verificação modulo 10 padrão).

modulo10UPS(«string»): Retorna um caractere de verificação modulo 10 de UPS (uma variável do caractere de verificação modulo 10 padrão).

modulo 11 («string»): Retorna um caractere de verificação modulo 11 (code 11).

modulo11IBM(«string»): Retorna um caractere de verificação modulo 11 de IBM (uma variável do caractere de verificação modulo 11 padrão).

modulo16(string): Retorna um dígito de verificação modulo 16

modulo 24 («string»): Retorna um caractere de verificação modulo 24 (code PSA).

modulo 32 («string»): Retorna um caractere de verificação modulo 32 (indústria farmacêutica Italiana).

modulo 43 («string»): Retorna um caractere de verificação modulo 43 (code 39).

modulo 47 («string»): Retorna dois caracteres de verificação modulo 47 (Code 93).

Plessey («string»): Fornece dois caracteres de verificação (Código Plessey).

pricecd («string»): Retorna 4 caracteres de verificação do UPC Random Price.

pricecd5 («string»): Retorna 5 caracteres de verificação do UPC Random Price.

StringToExt39(«string»): Prepara a string para ser usada no código de barras Extended 39.

UPSCheckDigit («string»): Retorna caracteres de verificação do UPS. A entrada para o método é uma string. Este método deixa o resto do número de rastreamento para você. Utiliza uma sequência de 15 caracteres, e calcula o dígito de verificação usando esta sequência.

1. Os primeiros dois caracteres devem ser "1Z".
2. Os próximos 6 caracteres preencheremos com nosso número de conta "XXXXXX" do UPS.
3. Os próximos 2 caracteres denotam o tipo de serviço:
    - "01" para entregas do Próximo Dia por via Aérea.
    - "02" para entregas do Segundo Dia por via Aérea.
    - "03" para entregas por via Terrestre.
4. Os próximos 5 caracteres é nosso número de fatura (nossas faturas têm 6 dígitos; retiramos o primeiro dígito, por exemplo, a fatura 123456 teria os caracteres 23456).
5. Os próximos 2 dígitos é o número do pacote, zero preenchido. Ex: o pacote 1 será "01", 2 será "02".
6. O caractere final e último é o dígito de verificação.

Nota: A sequência descrita acima fornece 17 caracteres, onde conforme necessário, somente 15 são necessários para calcular o dígito de verificação. Para fazer isto, retire a parte "1Z", e use somente os últimos 15 caracteres no método.

Funções de conversão

Nota: As definições regionais do sistema Windows podem afetar o resultado de algumas fórmulas.

AI253 («string»): Função específica para preparar a string para identificador 253 do aplicativo.

AI8003 («string»): Função específica para preparar a string para identificador 8003 do aplicativo.

ASCII («string»): Retorna o código ASCII do primeiro caractere no parâmetro da string.

Exemplo:
ascii("A") = 65

char («integer»):

- Para valores de 128 até 255: Fornece o caractere correspondente ao parâmetro integer na tabela ASCII.
- Para todos os valores negativos e valores de 0 até 127 e maiores que 255: Retorna o caractere Unicode correspondente ao parâmetro *integer*.

Exemplo:
char (65) = "A"

CodabarData(«data», «start», «stop»): Ajusta dados para CodabarData.

Code128CData(«string»): Ajusta dados para Code 128 C.

Currencytoeuro («value»): Converte a moeda atual nacional em Euros.

DatabarData(«data», «min», «max», «hasComposite»): Ajusta dados para DatabarData.

DBCSToUnicode(«string»,«codepage»): A fórmula utiliza dois parâmetros - os dados «*string*» a serem convertidos, e a página do código «*codepage*» usada. O parâmetro Codepage é representado pelo nome do idioma (Tailandês, Japonês, ChineseGBK, Coreano, ChineseBig5, Europeu, Oriental, Cirílico, Grego, Turco, Hebreu, Árabe, Báltico, Vietnamita, UTF-8, ACP) ou valor numérico (documentação de identificadores da página de código da pesquisa na Internet).

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

Esta função é necessária para dados que vêm do ficheiro diferente de Unicode ou fontes de dados (ex: Base de Dados).

Nota: Esta função tem ação reversa ao método UnicodeToDBCS.

dollar («value»): Modifica um campo para representar um preço.

Exemplo:
Se um campo denominado PRICE exibir um valor de 199, então:
dollar(PRICE) exibirá $199.00.

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

Eurotocurrency («value»): Converte o Euro na moeda nacional atual.

Nota: O número de decimais a serem exibidas deve ser definido na guia **Saída** na fonte de dados da **Fórmula** ao selecionar o campo Exibir de decimais. A taxa de conversão usada é a definida na guia **Outros** da caixa de diálogo Opções.

FileToData(«fileName», «errorData», «maxSize»): Retorna os dados lidos a partir do ficheiro.

fixed («value» , «num_decimals», «non_sep»): Retorna uma string de caracteres formatados com várias decimais igual ao parâmetro *num_decimals* e com ou sem um separador de milhares.

Exemplos:
fixo (1234.5678, 3, TRUE) = "1234.568"
fixo (1234.5678, 3, FALSE) = "1,234.568"

Nota: O separador de milhares a ser exibido deve ser definido na guia **Saída** da fonte de dados da Fórmula ao selecionar a caixa **Mostrar Decimais**.

FormatDate(«date», «dateFormat», «localeID»): Converte «date» em uma string formatada de acordo com «dateFormat».

Valor de retorno	String	um valor de data em formato de string, formatado de acordo com um formato.
data	String	Valor de data em formato de string.
dateFormat	String / Número	Formato do valor de data base. - Os números codificam formatos pré-definidos. (Por exemplo, 1"dd/mm", 2 "dd mmmm" e assim por diante). Os possíveis valores estão disponíveis aqui. - 0 ou valores negativos significam o uso do formato de data do sistema atual. - O valor da string codifica o formato diretamente. (Por exemplo, "yyyy/mm/dd")

LocaleID	Número	**O parâmetro localeID especifica o local usado para mostrar o valor de data. localeID é opcional (Padrão: 0, o local de sistema será usado). Possíveis valores(número) pode ser encontrado em http://msdn.microsoft.com/en-us/goglobal/bb964664.aspx.**

FormatMoney(«amount», «use separation characters(T/F)», «price characters», «force leading zero (T/F)», «location for price character», «# of decimal places»)
Permite a você adicionar símbolo da moeda atual, pontos, forçar zero à esquerda, usar caracteres de separação

Exemplo:
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

Nota: O **local para caractere de preço** é responsável pela posição do caractere de preço, espaços são adicionados entre o caractere de preço e a quantidade. Este valor é ignorado se for menor que o comprimento da string resultante.

FormatNumber(number): Esta função permite a você formatar um campo numérico, no qual o sinal (#) significa exibir somente se tiver um valor e o zero (0) significar para sempre exibir.

Exemplos:
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»): Retorna o valor «name» (string) da variável de ambiente.

Exemplo:
GetEnv("OS") = " Windows_NT".

Nota: Esta função trabalha com variáveis de ambiente definidas pelo usuário

GS1AIData(«AIName», «AIData», «isLastAI»): Ajusta dados para GS1 AI.

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

GS1Form(«string»): Adjusts data for GS1 Norm.

int («*value*») Retorna o maior valor inteiro menor ou igual ao parâmetro *value*.

Exemplos:
int (-5.863) = -6
int (5.863) = 5

LmChar («integer») : Retorna o caractere correspondente ao parâmetro integer na tabela ANSI. Esta tabela não depende do sistema local.
O valor está entre 0 e 255.

MaxiCodeData(«Mode»,«PostalCode»,«CountryCode»,«ClassOfService»,«TrackingNumber»,«UpsShipperNumber»,«JulianDate»,«ShipmentID»,«PackageNumber»,«TotalPackages»,«PackageWeight»,«AdressValidation»,«ShipToAdress»,«ShipToCity»,«ShipToState»): Cria dados do Maxicode a partir da entrada «string».

text («*value*», «*format*») Retorna o parâmetro *value* escrito no formato imposto pelo parâmetro *format*, que corresponde ao formato descrito na explicação da variável do Formulário.

Exemplos:
text(012.345, "######,(###)") = 12,(345)
text(1012.345, "######,(###)") = 1012,(345)

trunc («*value*») Retorna o inteiro que é parte do parâmetro *value*.

Exemplo:
trunc (123.45) = 123

UnicodetoDBCS(«*string_1*», «*string_2*») : A fórmula utiliza dois parâmetros - os dados «*string_1*» a serem convertidos e a página do código «*string_2*» usada. O parâmetro Codepage pode ter qualquer dos seguintes como um valor:

- Tailandês
- Japonês

- Chinês GBK
- Coreano
- Chinês Big5
- Oriental
- Grego
- Turco
- Hebreu
- Árabe
- Báltico
- Vietnamita

Nota: O parâmetro Codepage diferencia letras maiúsculas e letras minúsculas.

value («*string*») Retorna o valor numérico de uma string do caractere.

Exemplo:
value("123") = 123
value("0.00:48:00")-value("12:00:00") igual "16:48:00"-"12:00:00" igual 0.00, o número de série por 4 horas e 48 minutos.

Nota: Normalmente não é necessário usar a função do valor em uma fórmula, pois o aplicativo converte automaticamente o texto em números, se necessário.

ValueEx(«string»): Converte «string» em um valor numérico.

VoiceCode(«string», «string», «string»): Um VoiceCode é um número de 4 dígitos calculado usando o GTIN, Lot e Date opcional de um PTI.

Valor de retorno	string	Calcula um valor do VoiceCode.
GTIN	string	Número do GTIN de 14 dígitos. Somente dígitos são permitidos. O erro será retornado no caso do valor não ser um dígito. Os 14 dígitos da parte à esquerda são aceitos no caso do comprimento de dados ser maior que 14. '0' são adicionados à esquerda no caso do comprimento de dados ser menor que 14 dígitos.
LotNumber	string	Até 20 dados de símbolos alfanuméricos. Os 20 símbolos da parte à esquerda são aceitos no caso do comprimento de dados ser maior que 20 dígitos.

Date	string	Este é um parâmetro opcional. Dados de 6 dígitos que representam a data no formato YYMMDD. A fórmula retorna um erro no caso de comprimento errado, valor diferente do dígito ou data errada.

Este cálculo é executado como segue:

1. Calcule o PlainText.

    a. PlainText é o GTIN de 14 dígitos anexado pelo Lot Code e Date (onde presente).

    b. Não inclua os prefixos de identificador do aplicativo ou parênteses.

    c. Não há nenhum espaço entre os campos GTIN, Lot e Date.

    d. Date se presente é representado como YYMMDD com compactação de zero e sem os caracteres '/ '.

2. Calcule ANSI CRC-16 Hash de bytes do PlainText ASCII usando o ANSI CRC-16 Hash padrão como o polinômio de X16 + X15 + X2 + 1

    Consulte a função CRC-16.
3. Calcule o VoiceCode pelo Hash utilizando os 4 dígitos menos significativos na formulário de decimal (Hash mod 10000).

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

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

Funções de conversão do ATA 2000

Nota: As definições regionais do sistema Windows podem afetar o resultado de algumas fórmulas.

Bin2Hex(«value», «pad»)

Converte um valor binário em um valor hexadecimal de saída. O parâmetro 'pad' é utilizado para definir o comprimento total de saída adicionando 0 caracteres à esquerda do resultado.

Exemplo:
Bin2Hex("0010100111101001101",0) retorna 14F4D
Bin2Hex("0010100111101001101",0) retorna 00014F4D

CRC16_CCITT(«string»)

Calcula a soma de verificação do CRC16 CCITT.
Isto é utilizado no regulamento de ATA2000 para TOC no cálculo de CRC.

Exemplo:
CRC16_CCITT ("A") retorna B915
CRC16_CCITT ("AB") retorna 4B74
CRC16_CCITT ("ABC") retorna F508

CRC32_CCITT(«string»)

Calcula a soma de verificação do CRC32 CCITT.
Isto é utilizado no regulamento de ATA2000 para TOC no cálculo de CRC.

Exemplo:
CRC32_CCITT ("A") retorna D3D99E8B
CRC32_CCITT ("AB") retorna 30694C07
CRC32_CCITT ("123456789") retorna CBF43926

Dec2Bin(«value», «pad»)

Converte um valor decimal em um valor binário de saída. O parâmetro 'pad' é utilizado para definir o comprimento total de saída adicionando 0 caracteres à esquerda do resultado.

Exemplo:
Dec2Bin("5", 0) retorna 101
Dec2Bin("5", 4) retorna 0101

Dec2Hex(«value», «pad»)

Converte um valor decimal em um valor hexadecimal de saída. O parâmetro 'pad' é utilizado para definir o comprimento total de saída adicionando 0 caracteres à esquerda do resultado.

Exemplo:
Dec2Hex(510, 0) retorna 1FE
Dec2Hex(510, 4) retorna 01FE

String2Hex(«value», «pad»)

Converte um valor de string em um valor hexadecimal de saída. O parâmetro 'pad' é utilizado para definir o alinhamento em Bytes de saída adicionando 0 caracteres à direita do resultado.
No RFTAG, os dados geralmente são codificados em hexadecimal com um alinhamento de palavras (2Bytes).

Exemplo:
String2Hex("A", 0) retorna 41 (**A** é AINSI code **65** no decimal que é **41** no hexadecimal)
String2Hex("ABC", 2) retorna 414243
String2Hex("ABC", 4) retorna 41424300

Aqui está uma amostra para um registro do RFID ATA2000 Birth:

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) retorna
4D46522053303637312A534551204D333747584239322A504E4F20505137565A342A50445420434F4E54524F4C4C45522A49

String6BitsEncoding(«value», «pad»)

Converte um valor de string em um valor de saída de codificação ASCII de 6-bits.
O parâmetro 'pad' é utilizado para definir o alinhamento em Bytes de saída adicionando 0 caracteres à direita do resultado.
No ATA2000, os dados podem ser codificados em 6-bits (ao invés de 8-bits) para fins de compressão.

Exemplo:
String6BitsEncoding("11", 0) retorna C710

VDAString6BitsEncoding(«cadeia»)

Converte um valor em cadeia para uma codificação ASCII 6 bits utilizando os requisitos da Associação da Indústria Automóvel Alemã (VDA).

Em ATA2000, os dados podem ser codificados em 6 bits (em vez de 8 bits) para fins de compressão. A tabela utilizada para conversão está disponível aqui. Para mais detalhes, consulte as especificações VDA5500 e VDA4994.

Exemplo:
VDAString6BitsEncoding("11") devolve C71860820820

Funções de data e hora

A variável de data representa a data e hora do sistema, e não a variável de data definida pelo programa:

Nota: As definições regionais do sistema Windows podem afetar o resultado de algumas fórmulas.

Nota: 30/12/1899 = data de referência do seu aplicativo. Se você desejar que a fórmula retorne a data atual, a solução é calcular o número de dias decorridos desde a data de referência.

BestBefore («*date*» , «dateFormat» , «offset» , «offsetUnit» , «changeMonth» , «outputFormat», «localeID»)
Permite a você calcular uma Data de Validade baseado no valor de entrada do teclado (não baseado na data atual).

Valor de retorno	String	Um valor de data em formato de string, calculado por uma data base e um deslocamento.
date	String	Valor de data base em formato de string.
dateFormat	String / Número	Formato do valor de data base. - Os números codificam formatos pré-definidos. (Por exemplo, 1"dd/mm", 2 "dd mmmm" e assim por diante). Os possíveis valores estão disponíveis aqui. - 0 ou valores negativos significam o uso do formato de data do sistema atual. - O valor da string codifica o formato diretamente. (Por exemplo, "yyyy/mm/dd")
offset	Número	O número das unidades de data a serem adicionadas na data base.
offsetUnit	String / Número	Significando o parâmetro de deslocamento. - 1 ou "d" ou "D" significa dias.

		- 2 ou "m" ou "M" significa meses. - 3 ou "y" ou "Y" significa anos.
changeMonth	Número	Opcional (Padrão: True) - 0 significa False. - 1 significa True. Se a data calculada não existir no mês (Por exemplo, 30 de Fevereiro). True retorna o primeiro dia no próximo mês (Por exemplo, 1° Março). False retorna o último dia no mês atual (Por exemplo, 28 de Fevereiro).
outputFormat	String / Número	Opcional (Padrão: mesmo do dateFormat) Este parâmetro especifica o formato de saída da data calculada. Os possíveis valores estão disponíveis aqui.
localeID	Número	Opcional (Padrão: 0, o local do sistema será usado) Este parâmetro especifica o local usado para mostrar ao valor de data.

Exemplo:

BestBefore("14/06/2012" ,"dd/mm/yyyy","18","m", 1, "mmmm dd, yyyy", 1033) retornará 14 de Dezembro de 2013.

BestBefore("14/06/2012" ,"dd/mm/yyyy","18","m", 1, "mmmm dd, yyyy", 1036) retornará 14 de Dezembro, 2013.

Exemplo: Fórmula com uma fonte de dados de data.

Crie um nome da variável de data date0 com o data do dia, por exemplo 26/06/2012 (formato dd/mm/yy).

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

CFIA_Month(«data»):

Devolve o nome do mês utilizado pela Agência Canadiana de Inspeção Alimentar (CFIA): JA, FE, MR, AL, MA, JN, JL, AU, SE, OC, NO, DE.

Exemplo:

CFIA_Month ("03/01/2021") devolve MR

CFIA_Month (today()) devolve o nome do mês com base na data de hoje, se hoje for 01/01/2021 - o valor devolvido é JA

CFIA_Month ("06/01") devolve JA

Nota: o formato do argumento da «data» depende das definições de ambiente aplicadas ao sistema.

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

Valor de retorno	Date	Adiciona um intervalo de data na data base especificada.
date	Date	Valor de data base.
offset	Número	O número das unidades de data a serem adicionadas na data base.
offsetUnit	String / Número	Significando o parâmetro de deslocamento. - 1 ou "d" ou "D" significa dias. - 2 ou "m" ou "M" significa meses. - 3 ou "y" ou "Y" significa anos.
changeMonth	Número	Opcional (Padrão: True) - 0 significa False. - 1 significa True. Se a data calculada não existir no mês (por exemplo, 30 de Fevereiro). True retorna o primeiro dia no próximo mês (por exemplo, 1° Março). False retorna o último dia no mês atual (por exemplo, 28 de Fevereiro).

Exemplo:

DateOffset("26/06/2012",1,"d",1) retornará 27/06/2012

-OU-

DateOffset(today(),1,"D", 0)

Exemplo: Fórmula com uma fonte de dados de data.

Crie um nome da variável de data Date0 com a data do dia, por exemplo 26/06/2012. Você deve garantir que a variável de data está no formato de sistema pré-definido (caso contrário, você pode usar a função DateValue() para obter o valor de data da variável de data).

DateOffset(Date0,1,"D", 0) retornará 27/06/2012

DateValue(«formattedDate », «dateformat»)

Valor de retorno	Date	Retorna um valor de data do parâmetro formattedDate.
formattedDate	String	Uma data do formato de texto a ser convertido em um valor de data.
dateformat	String	Opcional (Padrão: o formato de data do sistema) Se especificado, a string de formato direto. Por exemplo, "dd/mm/yy"

Exemplo:

DateValue(22062012,"ddmmyyyy") retornará 22/06/2012.

day («*date*») Fornece o dia do mês do parâmetro *date*.

FiscalDate(«fiscalStartDate», «outputFormat»)

Permite a você calcular a data para o início do seu ano fiscal (yyyy.mm.dd).

Exemplos (se a data atual for 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 («*date*») Fornece a hora do parâmetro *date*.

minute («*date*») Fornece o minuto do parâmetro *date*.

month («*date*») Fornece o mês do parâmetro *date*.

now (): Fornece a data e hora atuais.

second («*date*») Fornece o segundo do parâmetro *date*.

shiftcode(«items», «defaultValue»)

O código Shift é usado como uma fonte de dados para um campo na etiqueta que precisa mudar baseado na hora atual. Um deslocamento representa um espaço de tempo que é definido e nomeado.

Valor de retorno	String	Retorna o valor do código Shift atual.
itens	String	Itens do código Shift no seguinte formato: startHour:startMin-stopHour:StopMin-value|...| startHour:startMin-stopHour:StopMin-value

Por exemplo, os deslocamentos podem ser definidos como segue:

- 7:00 às 15:00 = Dia
- 15:00 às 23:00 = Tarde

- 23:00 às 7:00 = Noite

Notas:
1. O relógio de 24 horas é usado para definir valores de tempo. 0:00 é meia-noite; 12:00 é meio-dia.
2. Se existir qualquer sobreposição de hora na entrada, então o primeiro valor aceitável é retornado.
3. Se não existir nenhum deslocamento da hora atual, então uma string vazia é retornada.

Exemplo (se a hora atual for 16:00):
shiftcode("7:00-15:00-Day|15:00-23:00-Evening|23:00-7:00-Night","Day") ="Evening"

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

Permite a você personalizar um espaço de data, ao deslocar a data por um determinado intervalo e usar esta data personalizada em uma expressão da fórmula. A função SpecificDateFormat desloca a data atual baseada no relógio do sistema usando o formato de data pré-definida ou um formato diferente de sua escolha.

Exemplo: SpecificDateFormat("mmmm","+2M")

- [date format] Especifica o formato de data para usar. Este parâmetro deve ser incluído dentro de parênteses. ***Tabela de Visualização*** de formatos de data válidos.

Definição de Data	Saída	Descrição
D e d	1	data do mês (sem zero à esquerda)
DD	01	data do mês de 2 caracteres (zero à esquerda)
dd	1	data do mês de 2 caracteres (espaço à esquerda)
DDD	224	dias do começo do ano (zero à esquerda)
ddd	224	dias do começo do ano (espaço à esquerda)
DDDDD e ddddd	33482	dias de 1/1/1900 (5 até 9 caracteres disponíveis)
W e w	1	dia de semana (Sun=1, Sat=7)
WW	34	semanas do começo do ano (zero à esquerda)
ww	34	semanas do começo do ano (espaço à esquerda)
WWW e www	783	semanas de 1/1/1900 (últimos 3 dígitos)
WWWW e wwww	4783	dias de 1/1/1900 (4 até 9 caracteres disponíveis)
WWWWWWWWW	000004783	semanas de 1/1/1900 (zero à esquerda)
wwwwwwwww	4783	semanas de 1/1/1900 (espaço à esquerda)

M e m	9	dígito do mês (sem zero à esquerda)
MM	09	dígito do mês de 2 caracteres (zero à esquerda)
mm	9	dígito do mês de 2 caracteres (espaço à esquerda)
MMM	009	dígito do mês de 3 caracteres (zero à esquerda)
mmm	9	dígito do mês de 3 caracteres (espaço à esquerda)
MMMM e mmmm	1101	meses desde 1/1/1900
MMMMM	01101	meses de 5 caracteres a partir de 1900 (zero à esquerda)
mmmmm	1101	meses de 5 caracteres a partir de 1900 (espaço à esquerda) (5 até 9 caracteres disponíveis)
MMMMMMMMM	000001101	meses de 9 caracteres a partir de 1900 (zero à esquerda)
mmmmmmmmm	1101	meses de 9 caracteres a partir de 1900 (espaço à esquerda)
YY e yy	91	ano de 2 dígitos
Y e y	1991	ano completo
YYY e yyy	991	3 últimos dígitos do ano (3 até 9 caracteres disponíveis)
YYYYYYYYY	000001991	ano completo (zero à esquerda)
yyyyyyyyy	1991	ano completo (espaço à esquerda)

- +/[offset data] Este valor especifica a quantidade na qual deslocar a data. O intervalo deve ser precedido por um sinal de mais (+) ou menos (), dependendo se você deseja adicionar ou subtrair da data atual.
- [date interval] O intervalo de data deve ser d para dias, w para semanas, m para meses ou y para anos.

Exemplos da função SpecificDateFormat usada para expressões em diferentes formatos:

Nota: Para estes exemplos, assuma que a data atual seja 29 de Julho de 2013.

SpecificDateFormat("mm dd yy" , "+1y") = 7 29 14 (1 após a data atual)
SpecificDateFormat("ww/m/yyyy" , "30w) = 1/12/2012 (30 semanas antes da data atual)
SpecificDateFormat("ddddd / wwww /mmmm" , "") = 41484/5928/1363 (dias, semanas e meses desde 1900)

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

A função TimeOffset permite a você personalizar um espaço de tempo ao deslocar o tempo por um determinado intervalo e usar este tempo personalizado em uma expressão da fórmula. A função

TimeOffset desloca o tempo atual baseado no relógio do sistema usando o formato de tempo pré-definido.

Quando usada em uma expressão, a função TimeOffset deve ser escrita usando os seguintes parâmetros:

Exemplo: TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"h+5")

- Você pode usar somente o formato pré-definido de "mm/dd/yyyy hh:nn:ss". Para inserir um formato diferente, você precisa ligar a função TimeOffset com a função DateValue e então ligar na função FormatDate onde você pode especificar outro formato de tempo.
- “+/offset time” Este valor especifica a quantidade na qual deslocar o tempo. O intervalo deve ser precedido por um sinal de mais (+) ou menos (), dependendo se você deseja adicionar ou subtrair da hora atual.
- "time interval" O intervalo de tempo deve ser s para segundos, m para minutos ou h para horas.

A seguinte tabela mostra exemplos da função TimeOffset usada para expressões em diferentes formatos. (Nota: Para estes exemplos, assuma que a hora atual seja 4:12:10 PM em 30 de Julho de 2013)

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"h+5") = 07/30/2013 21:12:10 (5 horas após a hora atual)
FormatDate(DateValue(TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"), "n+10"), "mm/dd/yyyy hh:nn:ss"), "hhnn") = 16-22 (10 minutos após a hora atual)

today (): Fornece a data atual.

weekday («*date*») Retorna o dia da semana do parâmetro *date*.

Nota: Domingo é considerado o primeiro dia da semana.

Exemplo:
On "Tuesday November 20, 1999" the weekday(now()) = 3

Exemplo:
Minute (now()) fornece a hora e minutos atuais.
Year (today()) fornece o ano do data atual.

Week («Date»): Retorna o número da semana a partir do parâmetro date.

WeekISO8601(«Date», «DateFormat»)

Permite a você criar uma fórmula que suporta a ISO8601. A ISO 8601 é um padrão internacional que envolve a troca de data e os dados relacionados ao tempo.

Valor de retorno	date	Retorna a semana da data especificada
Date	string	Uma data do formato de texto a ser convertido em um valor de data.
DateFormat	string	Formato do valor de data base. - Os números codificam formatos pré-definidos. (ex: 1 dd/mm/yy, 2 dd/mm/yyyy e assim por diante - 0 ou valores negativos significa usar o formato de data do sistema atual - O valor da string codifica o formato diretamente. (ex: "yyyy/mm/dd")

Exemplo:

WeekISO8601(" 03/01/2010 ",0) = 53

WeekISO8601(" 02/01/2011 "," dd/mm/yyyy ") = 52

WeekISO8601(" 01/01/2011 ", " dd/mm/yyyy ") = 52

year («Date»): Fornece o ano do parâmetro date.

Exemplo:

Minute (now()) fornece a hora e minutos atuais.

Year (today()) fornece o ano do data atual.

Nota: Por Minuto e Hora, somente o parâmetro now() é permitido. For todas as outras funções, now() e today() são permitidos.

Funções lógicas

As funções lógicas permitem a você verificar uma ou mais condições que foram cumpridas.

Nota: As definições regionais do sistema Windows podem afetar o resultado de algumas fórmulas.

Nota: TRUE igual a 1 e FALSE igual a 0.

and («*expr_1*», «*expr_2*») Retorna TRUE se ambos os parâmetros forem verdadeiros, e FALSE se pelo menos um for falso. Os parâmetros devem ser calculados a partir de valores lógicos.

Exemplo:

and(exact("string","string"),exact("string","string")) = 0

and(exact("string","string"),exact("string","string")) = 1

exat («*string_1*», «*string_2*») Retorna TRUE se as duas strings forem idênticas, e FALSE se não. Esta função diferencia letras maiúsculas e letras minúsculas.

Exemplo:

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

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

if («*expr*», «*Val_if_true*», «*Val_if_false*») Retorna o valor *Val_if_true* se *Expr* for verdadeiro e o parâmetro *Val_if_false* se *Expr* for falso.

Exemplo:

if(exact("string", "string"), "true", "false") = false

if(exact("string", "string"), "true", "false") = true

not («*logical*») Fornece o oposto do parâmetro *logical*.

Exemplo:

not(exact("string", "string")) = 1

not(exact("string", "string")) = 0

not(False) = 1 or not(0) = 1

not(True) = 0 or not(1) = 0

not(1+1=2) = 0

or («*expr_1*», «*expr_2*») Retorna TRUE se um dos dois parâmetros for verdade e FALSE se ambos os parâmetros forem falsos. Os parâmetros devem ser calculados a partir de valores lógicos.

Exemplo:

or(exact("string", "string"),exact("string", "string")) = 0

or(exact("string", "string"),exact("string", "string")) = 1

or(true,true) = 1 or or(1,1) = 1

or(true,false) = 1 or or(1,0) = 1

or(false,false)= 0 or or(0,0) = 0

Funções matemáticas

Nota: As definições regionais do sistema Windows podem afetar o resultado de algumas fórmulas.

Abs(data): Retorna o valor absoluto (positivo) de data.

Exemplos:

Abs(-5) = 5

Abs(5) = 5

base10tobaseX(«*string_1*»,«*string_2*»): Converte *string_2* da base 10 em base *string_1*.

Exemplos

Se o campo denominado Base 16 tiver a string "0123456789ABCDEF"

BASE10TOBASEX(Base16, 12) produz C

BASE10TOBASEX(Base16,10) produz A

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

Nota: Esta fórmula não pode aceitar números decimais negativos para o parâmetro *string_2*.

baseXtobase10(«*string_1*»,«*string_2*»): Converte *string_2* da base *string_1* em base 10.

Exemplos

Se o campo denominado Base 16 tiver a string "0123456789ABCDEF"

BASEXTOBASE10(Base16, "E") produz 14

BASEXTOBASE10(Base16,10) produz A

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

Ceil(data): Arredonda data para mais até o próximo número inteiro.

Exemplo:

Ceil(3.234) = 4

Ceil(7.328) = 8

Decimals(data1, data2): Usa as casas decimais de data2 em data1.

Exemplo:
Decimals(4, 2) = 4.00
Decimals(3.524, 1) = 3.5

eval_add(«string»,«string»): Retorna a soma dos parâmetros.

Exemplo
eval_add(5,5)=10

eval_div(«string»,«string»):Retorna a divisão de parâmetros.

Exemplo
eval_add(20,2)=10

eval_mult(«string»,«string»): Retorna a multiplicação dos parâmetros.

Exemplo
eval_mult(5,2)=10

eval_sub(«string»,«string»): Retorna a subtração dos parâmetros.

Exemplo
eval_sub(20,10)=10

Floor(data): Esta função arredonda datapara menos até o próximo número inteiro.

Exemplo:
Floor(3.234)= 3
Floor(7.328)= 7

hex(«val_1»,«val_2»): Converte o número decimal de **val_1** para o formato hexadecimal com um valor total de **val_2.**

Nota: Esta fórmula não pode aceitar números decimais negativos para o parâmetro *val_1*.

Exemplo:
hex(2, 8) = 00000002

int («value»): Retorna o maior valor inteiro menor ou igual ao parâmetro value.

Exemplos:
int (-5.863) = -6
int (5.863) = 5

max(data1, data2): Mostra o valor mais alto na série de dados.

Exemplo:
Max(5, 12) = 12

min(data1, data2): Mostra o valor mais baixo na série de dados.

Exemplo:
Min(5, 12) = 5

mod («val_1», «val_2»): Retorna o remanescente da divisão do parâmetro val_1 pelo parâmetro val_2. O resultado tem o mesmo sinal que o divisor.

Exemplos:
mod (7,2) = 1
mod (-7,2) = -1
mod (7,-2) = 1
mod (-7,-2) = -1

quotient («val_1», «val_2»): Retorna o resultado inteiro da divisão do parâmetro val_1 pelo parâmetro val_2.

Exemplo:
quotient(10, 2) = 5

round («val_1», «val_2»): Retorna o parâmetro val_1 arredondado para o número de imagens indicadas por val_2.

- Se *val_2* for maior que 0, *val_1* é arredondado para o número de decimal indicado.
- Se *val_2* for igual a 0, *val_1* é arredondado para o inteiro mais próximo.
- Se *val_2* for menor que 0, *val_1* é arredondado à esquerda do ponto decimal.

Exemplos:
round (4.25,1) = 4.3

round (1.449, 1) = 1.4
round (42.6,-1) = 40

Funções de texto

Uma string de caractere pode ser assimilada em uma tabela se cada campo tiver um caractere. Ela é definida por seu comprimento (número total de caracteres na string, inclusive espaços). A posição de um caractere na string corresponde ao seu lugar na tabela. Por exemplo, o primeiro caractere está na posição um.

Nota: As definições regionais do sistema Windows podem afetar o resultado de algumas fórmulas.

Exemplo: A posição 3 corresponde ao terceiro caractere na string.

cyclebasex ( )*:* Permite que a contagem aconteça em qualquer tipo de sistema de contagem da base de dados. O sistema numérico deve ser definido dentro da expressão vinculada. O valor inicial, o valor de cada incremento e o número de cópias devem também ser especificados para cada número. Todos estes valores podem estar vinculados a outro campos na etiqueta, mas os nomes do campo não devem estar incluídos em aspas.

Exemplo:
Se um campo denominado Base 16 tiver a string de caractere 0123456789ABCDEF, então:
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 ( )*:* Crie um conjunto de caracteres definido pelo usuário para um ciclo completo.

Exemplos:
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 ( )*:* Permite a você configurar sua própria sequência de números, ao invés de usar a sequência normal de números ou letras (0,1,2; ou A,B,C).

Exemplos:

cyclenumber(1,3) produzirá etiquetas na seguinte sequência: 1 2 3 1 2 3 1 2 3...

cyclenumber(1,3,1,2) produzirá etiquetas na seguinte sequência: 1,1,2,2,3,3,1,1,2,2,3,3,1,1…

cyclestring ( )*:* Permite a você criar um grupo de palavras ou caracteres usando um ciclo completo como um campo de incremento. A string inteira deve estar incluída em aspas (" ") e cada palavra ou grupo de caracteres deve estar separado do outro por um ponto e vírgula (;).

Exemplo:

cyclestring("Mon ; Tue ; Wed ; Thu ; Fri ; Sat ; Sun") = Mon Tue Wed Thu Fri Sat Sun

O seguinte exemplo é para etiquetas que usam todas as letras do alfabeto com exceção de O e I.

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 («*string_1*», «*string_2*») Retorna TRUE se as duas strings forem idênticas, e FALSE se não.

Exemplos:

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

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

extract («string», «sep», «pos»): Retorna a string secundária a partir da string de caractere («string») na posição especificada «pos» que contém dados separados pela string «sep».

Exemplo:

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

find («string», «*key*», «*start*»): Retorna a posição da primeira ocorrência do parâmetro *key* no parâmetro *string*. A pesquisa no parâmetro *string* começa da posição retornada pelo parâmetro *start* (*start* >= 1). A função redefine para zero se nenhuma ocorrência do parâmetro *key* for localizada. A função distingue entre letras maiúsculas e letras minúsculas.

Exemplo:

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

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

left («*string*», «*num_char*»): Retorna a string de caractere extraída do parâmetro *string*. Esta string começa na posição um do parâmetro *string* e tem um comprimento igual ao parâmetro *num_char*.

Exemplo:

left("Peter McPeepert",1) = P

left("Peter McPeepert ",5) = Peter

len («*string*») Fornece o comprimento do parâmetro *string*. Os espaços são contados como caracteres.

Exemplo:

len("Paris, New York") = 15

len("") = 0

len(" ") = 1

lower («*string*») Converte todas as letras maiúsculas em uma string de texto em letras minúsculas.

Exemplo:

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

LTrim(«*string*»): Recortará automaticamente quaisquer espaços à direita ou espaços à **esquerda** dos dados.

Exemplo:

LTrim(" No."): Não

mid («*string*», «*start*», «*num_char*») Retorna a string de caractere extraída do parâmetro *string*. Esta string começa na posição correspondente ao valor do parâmetro *start* (*start* >=1) e tem um comprimento igual ao parâmetro *num_char*.

Exemplo:

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

Output («variable»): Retorna o valor formatado da variável.

Exemplo:

Counter0 tem o prefixo "number:". A fórmula "counter0 tem um valor de " & counter0 & " e um valor formatado de " & output(Counter0), que fornecem o resultado: counter0 tem um valor 1 e um valor formatado de número:1

Nota: A formatação da variável é definida na guia **Saída**.

pad («*string*», «*length*», «*char*»)*:* Adiciona caracteres à esquerda do campo para atribuir um comprimento pré-definido na entrada inteira. Qualquer caractere pode ser selecionado como um caractere de preenchimento.

Exemplo:

Se um campo denominado GREETING exibir um valor HELLO, então:

pad(GREETING,8,0) = 000HELLO

pad(5,3,0) = 005

pad("Nine",6,"a") = aaNine

replace («*string*», «*start*», «*num_char*», «*new_string*») Retorna o parâmetro *string* convertido. Um número (igual ao parâmetro *num_char*) de caracteres da posição definida no parâmetro *start* foi substituído pelo parâmetro *new_string*.

Exemplo:

replace("Paris, New York",8,8,"Singapore") = Paris, Singapore

replaceString («string», «old_string», «new_string»): Substitui todas as ocorrências de uma «old_string» especificada na string de caractere («string»), por outra «new_string» especificada.

Exemplo:

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

rept («*string*», «*num_char*») Retorna a string de caractere onde o parâmetro *string* é repetido o número de vezes no parâmetro *num_char*.

Exemplo:

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

right («*string*», «*num_char*») Fornece a string de caractere composta dos últimos caracteres do parâmetro *string* e tem um comprimento igual ao parâmetro *num_char*.

Exemplo:

right("Purchase order",5) = order

RTrim («*string*»): Recortará automaticamente quaisquer espaços à direita ou espaços à esquerda em dados à **direita**.

Exemplo:

RTrim("Part ") :Part

search («*string*», «*key, start*»): Fornece a posição da primeira ocorrência do parâmetro *key* no parâmetro *string*. A pesquisa começa da posição definida pelo parâmetro *start* (*start* >= 1). A função redefine para zero se nenhuma ocorrência do parâmetro *key* for localizada.

Exemplos:
search("Purchase order","order,1) = 10
search("Purchase order","c",1) = 4

Sort(«lista», «separador», «direção»): Mostra uma lista ordenada de itens. A ordenação é realizada por comparação das cadeias. A lista é um argumento em cadeia que contém vários itens, separados por um separador como uma vírgula ou ponto e vírgula. A direção é um número inteiro opcional que especifica a direção da ordenação, 0 = ascendente, 1 = descendente. Se a direção não for especificada, a direção predefinida é ascendente.

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

StrAfter*(«data»,«start after»,«lenght»)* Esta função resulta em uma string que é exatamente os caracteres de comprimento longo após um caractere start after especificado.

Exemplos:
StrAfter("1234-5678", '-', 3)= Utiliza os 3 caracteres após o traço (567)
StrAfter("1234-5678", '-')= Utiliza todos os caracteres após o traço (5678)

StrBefore*(«data»,«start before»,«lenght»)* Esta função resulta em uma string que é exatamente os caracteres de comprimento longo após um caractere start before especificado.

Exemplos:
StrBefore("1234-5678", '-', 2)= Utiliza os 2 caracteres imediatamente antes do traço (34)
StrBefore("1234-5678", '-')= Utiliza todos os caracteres antes do traço (1234)

SuppressBlankRows («string»): Retorna uma string com linhas vazias ignoradas. Permite a você criar um objeto e inserir os campos que deseja e compactar os que estão em branco.

Exemplo:
***SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2}) (*Var0, Var1 e Var2 são variáveis, char(10) são "\n" símbolos que significa nova linha*)**

Variables/Values	Formula	exibe
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(«lista», «separador»): Remove as entradas em branco de uma lista de itens. A lista é um argumento em cadeia que contém vários itens, separados por um separador como uma vírgula ou ponto e vírgula.

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

trim («*string*»): Retorna o parâmetro *string* convertido. Todos os espaços encontrados no início e final da string são excluídos. O número de espaços incluídos entre duas palavras é reduzido para um.

Exemplo:
trim(" Purchase order") = Purchase order

trimall («*string*») Retorna o parâmetro *string* convertido. Todos os espaços encontrados são excluídos.

Exemplo:
trimall("Paris / New York / Rome") = Paris/NewYork/Rome

Unique(«lista», «separador», «comparador»): Remove entradadas duplicadas de uma lista de itens. A lista é um argumento em cadeia que contém vários itens, separados por um separador como uma vírgula ou ponto e vírgula. O comparador opcional indica se a comparação é sensível a maiúsculas e minúsculas

ou não, 0 = sensível, 1 = não sensível. Se o comparador não for especificado, é sensível a maiúsculas e minúsculas por predefinição.

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

upper («*string*») Fornece o parâmetro *string* convertido em letra maiúscula.

Exemplo:
upper("Purchase order") = PURCHASE ORDER

ztrim *(«value»)*: Exclui os zeros à esquerda do valor numérico. Os campos são completamente numéricos.

Exemplo:
Se um campo denominado WEIGHT exibir um valor de 000200, então:
ztrim(weight) = 200

Definindo as propriedades de uma fonte de dados da Fórmula

Comando: **Fontes de Dados > Fórmula >** nome da fórmula > **Propriedades**

1. Insira a fórmula diretamente no campo **Editar**.

   - ou -

   Selecione os elementos, e clique em **Inserir.**

2. Clique em **OK**.

Sugestão: Você pode inserir um elemento clicando duas vezes sobre ele.

Nota: Se uma variável usada na fórmula tiver um nome contendo um dos seguintes caracteres &+-*/<>=^%,!\", deverá ser incluída entre chaves {}.

Nota: Visualização dinâmica que representa o resultado atual de cálculo da fórmula, incluindo a formatação definida na página de Saída. Em caso de erro, a visualização é exibida em vermelho. Se o valor obtido estiver truncado, você deverá modificar o comprimento máximo especificado na guia **Saída**.

Exercício: Criando um módulo específico

Neste exercício, converteremos o código de barras "LOT_NUMBER" EAN8 em um código de barras 2/5 Interleaved usando a fonte de dados da Fórmula - "Formula_4_NewCustCode".

O código de barras deve ter as seguintes propriedades:

- Simbologia: impressora
- Altura: 4 mm
- Largura da barra estreita: 1 mm
- Relação: 2
- Humanamente legível: abaixo e centralizado
- Distância das barras: 0 mm
- Tipo de letra dos caracteres: letra da impressora.

Abra a etiqueta CHECK01.LAB.

Para calcular o peso

Crie a fórmula ”WEIGHTED”, tendo em mente que o primeiro caractere de LOT_NUMBER é multiplicado por 1, o segundo caractere por 2, o terceiro caractere por 1, o quarto caractere por 2, e assim por diante.

O comprimento máximo de saída da variável é 6.

Fórmula WEIGHTED:
mid (LOT_NUMBER,1,1) &
mid (LOT_NUMBER,2,1) * 2 &
mid (LOT_NUMBER,3,1) &
mid (LOT_NUMBER,4,1) * 2 &
mid (LOT_NUMBER,5,1)

O resultado é 2120103 como nós temos:

2*1 = 2
6*2 = 12
0*1 = 0
5*2 = 10
3*1 = 3

Para adicionar o resultado do cálculo de peso:

O próximo passo envolve adicionar juntas as figuras resultantes da fórmula anterior, não esquecendo que o comprimento máximo permitido desta string de caractere é 2.

1. Crie uma segunda fórmula e nomeie para "SUM".
2. Fórmula SUM:
   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)

   O resultado é 9; omo nós temos:
   2+1+2+0+1+0+3= 9.

Para calcular o dígito de verificação:

Usando o resultado anterior, agora calcularemos o valor do dígito de verificação.

1. Crie uma terceira fórmula e nomeie para CHECK_DIGIT.
2. A expressão é como segue:
   if ((SUM % 10) > 0,10 - SUM % 10,0)

   O resultado é 1; omo nós temos:
   SUM % 10 = 9 (% = modulo).
   9 being greater than 0, we carry out the subtraction: 10- 9 = 1.

Para calcular os dados a serem codificados:

Ao criar o código de barras, você deve incluir os dados a serem codificados; por exemplo, o valor da variável LOT_NUMBER concatenado com o valor do dígito de verificação (CHECK_DIGIT).

1. Crie uma quarta fórmula e nomeie para DATA
2. Esta fórmula é o resultado da concatenação de LOT_NUMBER e CHECK_DIGIT:
   LOT_NUMBER & CHECK_DIGIT

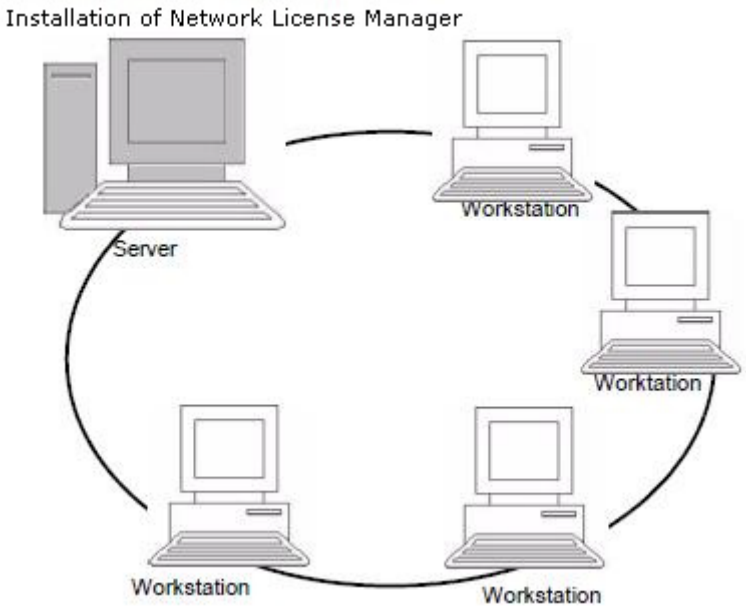
Para criar o código de barras:

1. Selecione a fórmula DATA, então arraste e solte na etiqueta sobre o código de barras.
2. Defina as propriedades do código de barras:
   Simbologia: 2/5 Interleaved
   dígito de verificação: Nenhum

Instalando a versão de rede

Descrição

Para usar a versão de multi-usuário/rede do software de desenho da etiqueta, você deve primeiro instalar os Network License Manager e da Rede, ou no servidor ou em uma estação de trabalho que agirá como o servidor, então instalar seu software de etiquetagem em cada estação de trabalho.



Procedimento de Instalação da Rede

Configuração da rede

Antes de instalar o software, o administrador de rede deve primeiro definir a estrutura da rede para o grupo de usuários:

- Defina o servidor de licença na qual os **Network License Manager**, e dongle serão instalados.
- Defina as estações de trabalho, ou as estações de trabalho cliente que usarão o software de etiquetagem.

Descrição do Gerenciador de Rede

Os **Network License Manager** permitem a você usar a configuração de rede do seu software de etiquetagem. O Gerenciador de Rede inclui:

- O Gerenciador de Licença da Rede (Serviço de Licença).
- Assistente de Definições da Rede: O Assistente de Definições da Rede ajuda a definir a configuração da rede.

Instalando os Utilitários de Usuário e Rede no servidor

Antes de instalar o software de etiquetagem em todas as estações de trabalho que irão usá-lo, você deve instalar os **Network License Manager** no servidor para configurar a rede.

1. Insira o DVD de instalação na unidade apropriada.
   A janela de Instalação é exibida.

   Se o DVD de instalação não executar automaticamente:
   Vá até o Windows Explorer e expanda a letra da unidade de DVD. Clique duas vezes em **index.hta**.
2. Selecione **Network License Manager**, que inclui **LicençaServiço**. Então clique no botão **Instalar**.
3. Siga as instruções na tela..
4. Se você desejar definir os parâmetros para sua configuração da rede, inicie o **Assistente de Definições da Rede** no servidor. Por padrão, se você não modificar a configuração, cada estação de trabalho terá suas próprias definições.

Configuração

Todas as ferramentas necessárias para configurar a versão da rede estão disponíveis na **Network toolbar**, que pode ser acessada a partir da barra de tarefas do Windows (Systray).

Na barra de tarefas do Windows, clique duas vezes no ícone  para que a **Network toolbar** seja exibida.

O **Assistente de Definições da Rede** ajuda você a definir as configurações para sua versão de rede.

1. Para iniciar o **Assistente de Definições da Rede**, clique no ícone .
2. Selecione um modo de definição: **Genérico, Por usuário** ou **Por estação**.

    - **Genérico**: Todos os usuários usarão as mesmas definições em todas as estações de trabalho. (user.ini).
    - **Pelo usuário**: Cada usuário pode acessar suas próprias definições em qualquer estação de trabalho. (user name.ini).
    - **Pela estação**: Cada estação de trabalho tem suas próprias definições (station.ini).

3. Especifique o local no qual você deseja armazenar estas definições. Se você desejar o compartilhamento destas definições entre várias estações de trabalho, especifique um caminho de rede que é acessível para todas as estações de trabalho.
4. Especifique o local no qual você deseja armazenar os dados compartilhados (variáveis, listas, ficheiro de registro de impressão, etc.).

Iniciando o Serviço de Licença

Antes de instalar o software de etiquetagem em todas as estações de trabalho, você deve ter certeza que o Serviço de Licença foi iniciado.

O Serviço de Licença foi instalado como o serviço denominado **SLicense**. Você não precisa iniciá-lo. De fato, o SLicense inicia quando a estação de trabalho é ativada e executa como uma tarefa de segundo plano desde que a estação de trabalho esteja ativada.

Para iniciar o Controlador do Serviço de Licença

Clique no ícone  disponível na barra de ferramentas Rede.

- ou -

Clique duas vezes no ficheiro TkxWebLicenseServerController.exe de instalação específica.

- ou -

Right-click on the **Network License Manager** icon from the Windows taskbar and select **License Service Controller**.

Instalando o software nas estações de trabalho

O software de etiquetagem deve ser instalado em todas as estações de trabalho nas quais será usado.

Para instalar o software em uma estação de trabalho

1. Insira o DVD na unidade apropriada.
   A janela de Instalação é exibida.

   Se o DVD não executar automaticamente:
   Vá até o Windows Explorer e expanda a letra da unidade de DVD. Clique duas vezes em **index.hta**.
2. Selecione o produto a ser instalado, clique no botão **Instalar** e siga as instruções na tela.

3. Iniciar o software de etiquetagem. O Gerenciador de Licenças é exibido. Clique em **Tentar** iniciar o software. No menu Ferramentas, escolha **Administração de rede**.
   - ou -
   No menu Iniciar do Windows, selecione o atalho **Administração de rede** no grupo do software de etiquetagem.
4. Ative **Usar Licença de Rede**.
5. Selecione o **Tipo de licença da rede.**

   - A **licença de pasta compartilhada**utiliza o recurso de compartilhamento de arquivo do Windows para se comunicar entre o software e o Gerenciador de Licença.

   - A **licença da Web**usa a comunicação de http/https entre o software e o Gerenciador de Licença.
6. Especifique a **porta do servidor** se você selecionou o tipo **licença da Web**.
7. Clique em **Modificar** para selecionar o servidor no qual o gerenciador de licença e dongle estão instalados.
   - ou -
   Clique em **Procurar** para procurar automaticamente o servidor no qual o gerenciador de licença está instalado.

   Se a rede já tiver sido configurada, uma mensagem perguntando se você deseja usar a configuração de rede atual é exibida.
8. Se você desejar modificar ou definir as configurações de rede, clique no botão **Assistente de Definições da Rede**
9. Clique em **OK**.
10. Reinicie o software



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