DIDACTICIEL

Version 2026

Les informations contenues dans le présent manuel de documentation ne sont pas contractuelles et peuvent faire l’objet de modifications sans préavis.

La fourniture du logiciel décrit dans ce manuel est régie par une licence d’utilisation. Le logiciel ne peut être utilisé, copié ou reproduit sur quelque support que ce soit, que conformément aux termes de cette licence.

Aucune partie du manuel ne peut être copiée, reproduite ou transmise par quelque moyen que ce soit, dans d’autres buts que l’usage personnel de l’acheteur, sans la permission écrite de TEKLYNX Corporation SAS.

©2026 TEKLYNX Corporation SAS,

Tous droits réservés.

Table des matières

A propos de ce manuel ... 6
Conventions typographiques ... 6
A propos de la version de votre logiciel ... 6
Connexion aux bases de données ... 7
Quelques rappels ... 7
Installation de la source de données ... 8
Importation des données ... 8
Création des objets variables ... 9
Active Query Builder ... 10
Prise en main ... 10
Ajout d'un objet à la requête ... 11
Modification des propriétés des objets ... 12
Jointures de tables ... 12
Tri des champs de sortie ... 13
Définition de critères ... 14
Définir une requête paramétrée ... 14
La grille Résultat de la requête ... 15
Database Manager ... 17
L'interface Structure de base de données ... 17
Choisir une base de données dans la liste des connections ... 17
Choisir une table dans une base de données ... 18
Ajouter une table dans la base de données active ... 18
Supprimer une table dans la base de données active ... 18
Afficher / masquer les données de la table active ... 18
Déterminer un champ comme champ clé ... 19

Déterminer le type de contenu d'un champ 19
Déterminer la taille maximum d'un champ 19
Autoriser un champ vide 19
L'interface Edition de base de données 20
Sélectionner les enregistrements en fonction de leur contenu 20
Sélectionner tous les enregistrements identiques 20
Sélectionner un enregistrement identique 21
Modifier un enregistrement 21
Supprimer un enregistrement 21
L'interface Requêtes de base de données 22
Ajouter une requête 22
Sélectionner / Désélectionner un ou des champs 22
Modifier l'ordre des champs sélectionnés 22
Créer un filtre au moyen de données prédéfinies 23
Appliquer un opérateur logique à plusieurs filtres 23
Supprimer un filtre 23
Modifier un filtre en langage SQL 23
L'interface Imprimer 24
Sélectionner le document à imprimer 24
Sélectionner un modèle d'étiquette existant 25
Sélectionner un document à partir d'un champ 25
Sélectionner une imprimante 25
Formules 26
La source de données Formule 26
Les fonctions 26
Les opérateurs 26
Les opérateurs arithmétiques 27
Les opérateurs de comparaison 27

L'opérateur de concaténation ... 27
L'opérateur logique ... 27
Les fonctions de calcul des caractères de contrôle ... 28
Les fonctions de conversion ... 30
Fonctions de conversion ATA 2000 ... 37
Les fonctions sur la date et l'heure ... 39
Les fonctions logiques ... 48
Les fonctions mathématiques ... 49
Les fonctions sur les textes ... 52
Définir les propriétés d'une source de données Formule ... 58
Atelier pratique : Calcul de Modulo spécifique ... 59
Pour calculer le poids ... 59
Pour additionner le résultat du calcul de poids : ... 60
Pour calculer le caractère de contrôle : ... 60
Pour calculer la donnée à encoder : ... 61
Pour créer le code-à-barres : ... 61
Installer la version Réseau ... 62
Description ... 62
Procédure d'installation ... 62
Configuration ... 62
Présentation du Network License Manager ... 63
Installation du Network License Manager sur le serveur ... 63
Configuration ... 64
Démarrage du gestionnaire de Licences ... 64
Pour démarrer le Contrôleur du service de licences ... 65
Installation du logiciel sur les stations ... 65
Pour installer le logiciel sur une station ... 65

A propos de ce manuel

Conventions typographiques

Ce manuel permet de distinguer diverses catégories d'informations par l'emploi des conventions suivantes :

- Les termes repris de l'interface (commandes, etc.) paraissent en **gras**.
- Les touches apparaissent en petites majuscules comme dans cet exemple : "Appuyez sur la touche SHIFT".
- Les listes numérotées signifient qu'il s'agit d'une procédure pour la réalisation d'une tâche.
- La conjonction -ou-, lorsqu'elle paraît à côté d'un paragraphe, fait ressortir le choix d'une autre procédure dans l'exécution d'une action.
- Lorsqu'une commande de menu comprend des sous menus, le nom du menu suivi de la commande à sélectionner apparaît en **gras**. Ainsi, "Choisissez **Fichier** > **Ouvrir**" signifie qu'il faut choisir dans le menu **Fichier** la commande **Ouvrir**.

A propos de la version de votre logiciel

Certaines fonctions décrites dans ce manuel peuvent ne pas être disponibles dans la version du logiciel que vous utilisez.

Si vous souhaitez connaître la liste des fonctions disponibles dans la version du logiciel que vous utilisez, veuillez vous reporter aux spécifications techniques.

Connexion aux bases de données

Quelques rappels

A travers ce chapitre, vous découvrirez à quel point votre logiciel est puissant. En effet, nous allons mettre en relation une étiquette (le contenant) avec une base de données (le contenu). Pour ce faire, nous allons avoir recours aux connexions **ODBC** (Open Data Base Connectivity) et **OLE DB** (Object Linking and Embedding Database).

Base de données

La base de donnée permet de stocker des données. Toutes les données sont agencées sous forme de tables (ou tableaux) à deux dimensions, ce que l'on nomme une relation. Chaque ligne du tableau est appelée enregistrement. Un enregistrement sert à gérer un objet dont les propriétés sont agencées dans les différentes colonnes de la table sous forme de champs.

Une base de données peut contenir plusieurs tables. Pour mettre en relation les différentes tables d'une même base de données, on peut avoir recours aux jointures. Nous verrons plus loin dans ce chapitre, à travers un exemple concret, comment créer des jointures.

ODBC

Il s'agit d'un standard de connexion permettant l'accès aux bases de données. En utilisant ODBC, il est très simple de mettre en relation une application telle que votre logiciel d'étiquetage avec un certain nombre de bases de données.

Le logiciel fournit plusieurs pilotes ODBC de manière à avoir accès aux bases de données les plus courantes. Ci-dessous, sont listés quelques pilotes :

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

OLE DB

Il s'agit d'un standard de connexion permettant d'accéder à tous les standards de bases de données et aux données stockées dans les systèmes de messagerie.

L'exemple ci-dessous décrit la procédure de connexion quand une base de données n'est pas connectée au logiciel.

Installation de la source de données

Cette procédure utilise le mode de création direct. Si vous voulez, vous pouvez utiliser l'assistant en sélectionnant **Assistant** dans le menu contextuel.

Connexion à la base de donnée TKTraining.mdb

1. Sélectionnez **Outils** > **Administrateur ODBC**
2. Cliquez sur l'onglet **Sources de données système** (DSN) et cliquez sur **Ajouter**.
3. Sélectionnez Pilote **Microsoft Access**, puis **Terminer**.
4. Saisissez "TK Training Level 2" dans le champ **Nom de la source de données**.
5. Cliquez sur **Sélectionner** et sélectionnez le fichier TKTraining.mdb installé avec les exemples.
6. Cliquez sur **Options.** Cochez la case **Lecture seule**. Cette option permet d'ouvrir las base de données en même temps que son utilisation dans ce logiciel sans avoir de problème de lecture/écriture.
7. Cliquez **Valider** dans les réglages **ODBC Microsoft Access**.

Importation des données

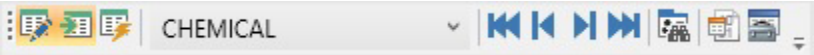
Maintenant que la source de données est installée, nous allons pouvoir importer les données stockées dans la base de données pour les insérer dans l'étiquette.

Pour importer les données :

1. Ouvrir l'étiquette PRODUCT_WS3
2. Sélectionnez **Source de données > Base de données > Créer/Modifier une requête**.
3. A partir de la liste **Sélectionnez une source de données**, choisissez **TK Training Level 2**
4. Sélectionnez la table **Fruits**.
5. Sélectionnez les champs "ProdName", "Origin", "Weight", et"Reference"
6. Cliquez sur le bouton . Il permet de trier les enregistrements par ordre alphabétique ou numérique en ordre croissant ou décroissant.
7. Sélectionnez "Reference" comme **Clé de tri** et **Croissant** comme ordre de tri.
8. Enregistrez la requête sous InstallDir\Samples\Forms\Tutorial\PRODUCT_WS4_ODBC.CSQ.

9. Cliquez **Valider**.
   Les variables sotn automatiquement crées et sont listées sous la branche **Base de données** dans la vue **Sources de données**.

Pour visualiser ou imprimer les différentes valeurs que peut prendre votre objet, utilisez la barre de navigation.



Vous pouvez également imprimer à partir de la fenêtre de résultat de requête.

Création des objets variables

Il suffit maintenant d'associer les variables ainsi créées à un objet (texte, code-à-barres, etc.) pour les faire figurer dans l'étiquette. Dans notre exemple, nous créerons des Objets Texte variables.

1. Sélectionnez une variable précédemment créée affichée sous la branche **Base de données** de la vue **Sources de données**, puis faites un glisser-déposer sur votre espace de travail.
2. Sélectionnez **Texte** dans menu contextuel.

Active Query Builder

Active Query Builder est un composant destiné à l'élaboration visuelle de requêtes qui vous permet de créer des requêtes SQL complexes via une interface visuelle intuitive de génération de requêtes.

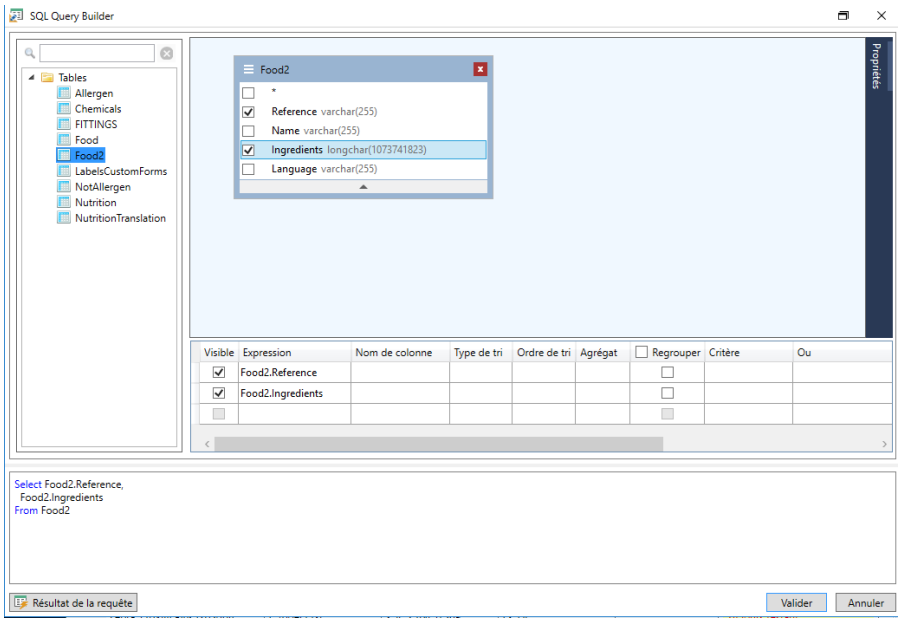
L'utilisation d'Active Query Builder présuppose une connaissance de base des concepts SQL. Active Query Builder vous permet d'écrire du code SQL correct sans les détails techniques, mais la maîtrise des principes de SQL est essentielle pour obtenir les résultats souhaités.

Prise en main

Cette capture d'écran illustre l'apparence d'**Active Query Builder** au démarrage.

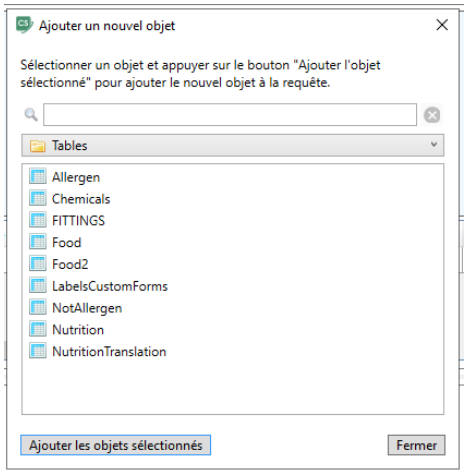
La fenêtre principale contient les éléments suivants :

- La **zone de génération de requête** est la zone principale dans laquelle s'affichera la représentation visuelle de la requête. Elle vous permet de définir des objets de base de données source et des tables dérivées, de définir des liens entre ces éléments et de configurer les propriétés des tables et des liens.
- Le **volet des colonnes** est situé sous la zone de génération de requête. Vous utilisez ce volet pour effectuer toutes les opérations nécessaires sur les colonnes de sortie et les expressions de requêtes. Il vous permet de définir des alias de champ, de réaliser des opérations de tri et de regroupement, et de définir des critères.
- La **zone d'affichage de la table** est située à gauche. Vous pouvez y voir votre requête et en localiser rapidement toute partie. Utilisez le champ **Recherche** situé en haut du volet pour préciser votre recherche.
- Le contrôle de page situé au-dessus de la zone de génération de requête vous permet de basculer entre la requête principale et les sous-requêtes.
- La petite zone signalée par la lettre « **Q** » dans l'angle de la zone de génération de requête est le contrôle de gestion des sous-requêtes Union. Elle vous permet d'ajouter de nouvelles sous-requêtes Union et d'effectuer toutes les opérations correspondantes.



Ajout d'un objet à la requête

Pour ajouter un objet à la requête, cliquez avec le bouton droit de la souris sur la zone de génération de requête et sélectionnez l'élément Ajouter un objet dans le menu déroulant.



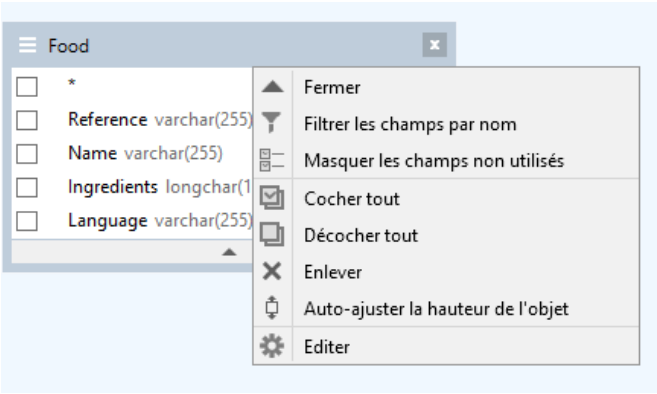
La fenêtre **Ajouter un nouvel objet** vous permet d'ajouter plusieurs objets à la fois. Les éléments de tableau sont énumérés dans la boîte combo. Le filtre **Afficher tous les objets** affiche tous les objets disponibles pour la sélection. Le **champ Recherche** en haut de la fenêtre est utilisé pour rechercher un objet. Vous pouvez sélectionner un ou plusieurs objets en maintenant la **touche CTRL** enfoncée**,** puis en appuyant sur le bouton **Ajouter les objets sélectionnés** pour ajouter ces objets à la requête. Vous

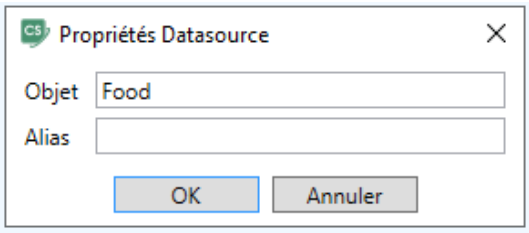
pouvez répéter cette opération à plusieurs reprises. Après avoir fini d'ajouter des objets, cliquez sur le bouton **Fermer** pour cacher cette fenêtre.

Pour supprimer un objet de la requête, cliquez sur le bouton **Fermer** dans l'en-tête de l'objet.

Modification des propriétés des objets

Vous avez la possibilité de modifier les propriétés de chacun des objets ajoutés à la requête. Pour ce faire, cliquez avec le bouton droit de la souris sur l'objet et sélectionnez l'élément **Editer** dans le menu déroulant, ou double-cliquez simplement dans l'en-tête de l'objet.

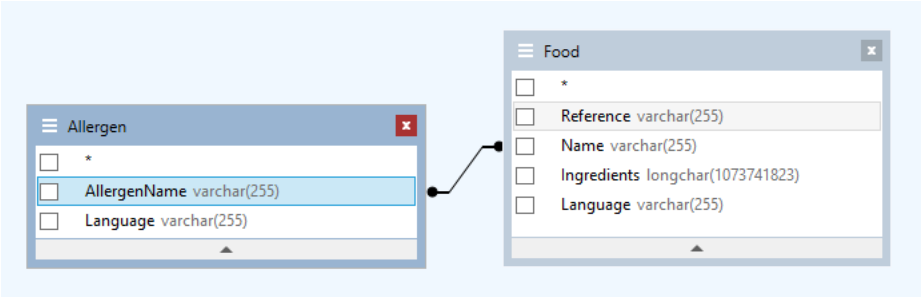




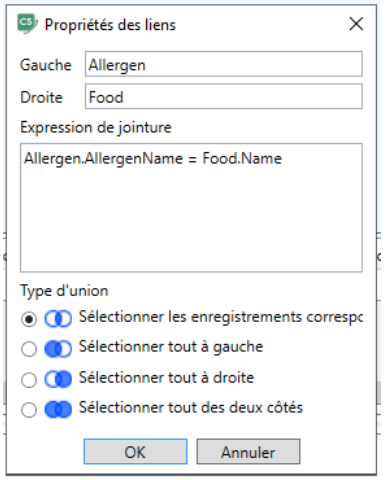
La boîte de dialogue **Propriétés de la source de données** peut varier d'un serveur à un autre, mais la propriété Alias est identique pour tous les serveurs de bases de données.

Jointures de tables

Pour créer une liaison entre deux objets (c'est-à-dire pour établir une jointure), vous devez sélectionner le champ selon lequel vous voulez lier l'objet à un autre et le faire glisser vers le champ correspondant de l'autre objet. Une fois cette opération accomplie, un trait reliant les champs liés apparaît.



Le type de jointure créé par défaut est INNER JOIN, ce qui signifie que seuls les enregistrements correspondants des deux tables seront inclus dans le dataset résultant. Pour définir d'autres types de jointures, vous disposez de deux méthodes : vous pouvez soit cliquer avec le bouton droit de la souris sur la liaison et sélectionner l'élément **Editer...** dans le menu déroulant, soit double-cliquer sur la liaison pour ouvrir la boîte de dialogue **Propriétés du lien**. Cette boîte de dialogue vous permet de définir le type de jointure et d'autres propriétés de la liaison.



Pour supprimer un lien entre plusieurs objets, cliquez sur le bouton droit de la souris sur la ligne de lien et sélectionnez l'option **Supprimer** dans le menu déroulant, ou bien sélectionnez la ligne de lien et appuyez sur le bouton **Supprimer.**

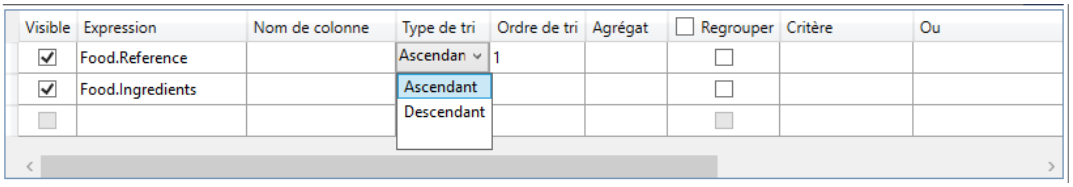
Tri des champs de sortie

Pour activer le tri des champs de requête de sortie, vous devez utiliser les colonnes **Type de tri** et **Ordre de tri** du **volet des colonnes**.

La colonne **Type de tri** vous permet de spécifier la manière dont les champs sont triés (par ordre **Croissant** ou **Décroissant**).

La colonne **Ordre de tri** vous permet de configurer l'ordre de tri des champs, lorsque plusieurs champs font l'objet d'un tri.

Pour désactiver le tri sur un champ, vous devez désactiver la colonne **Type de tri** pour ce champ.



Définition de critères

Pour définir des critères pour l'expression listée dans le **volet des colonnes**, vous devez utiliser la colonne **Critères**.

Vous devez écrire le critère à cet emplacement en omettant l'expression. Pour obtenir le critère suivant dans votre requête

> **WHERE** (champ >= 10) **AND** (champ <= 20)
> vous devez écrire
> >= 10 **AND** <= 20 dans la colonne **Critères**.

Vous pouvez également utiliser la **liste déroulante de** la colonne **Critères** afin de coller des expressions et des opérateurs communs dans le champ d'édition. Pour la création de critères avancés, utilisez le bouton **Editeur d'expressions** dans la colonne **Critères.** La boîte de dialogue de **l'Editeur d'expression** s'affiche.

Si vous le souhaitez, vous pouvez spécifier plusieurs critères pour une même expression à l'aide des colonnes **Or...**. Ces critères seront concaténés dans la requête à l'aide de l'opérateur **OR**.

Définir une requête paramétrée

L'outil **Query Builder** vous offre la possibilité de créer une requête paramétrée dont la valeur du paramètre proviendrait d'une source de données.

Remarque: vous devez avoir créé au préalable une source de données.

1. Glissez - déposez la table sur laquelle s'effectuera votre requête.
2. Sélectionnez le ou les champs sur lesquels s'appliqueront le ou les critère(s).

3. Dans la colonne **Critère**, ou bien dans la zone d'édition au format SQL, spécifier la source de données qui fera l'objet d'un critère de recherche.

   Exemple:pour rechercher la valeur de la source de données Var0 que vous aurez préalablement créée:

   - **Langage SQL**

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

   - **Colonne Critère**
     = APPLICATION.DOCUMENT.Var0

4. Cliquez sur le bouton **Résultat de la requête** pour afficher le résultat de votre requête.

La grille Résultat de la requête

Pour atteindre la grille **Résultat de la requête** cliquez sur le bouton à partir de la boîte de dialogue **Définition d'une requête,** dans la barre d'outils **Navigation dans la base de données de fusion** ou à partir du menu **Source de données > Bases de données**.

Cette grille permet de visualiser le résultat d'une requête ou de rechercher un terme particulier et toutes ses occurrences et enfin d'imprimer les étiquettes correspondantes.

Remarque : La balise**"<Données binaires>"** est affichée dans la boîte de e dialogue Résultat de la requête pour les champs de la base de données qui contiennent des **données de type BLOB** (par exemple, des fichiers d'images).

La grille **Résultat d'une requête** comporte :

- **des fonctions de recherche**

  Champ de recherche qui permet de saisir le champ sur lequel va porter la recherche.

  Valeur de recherche qui permet de saisir la valeur recherchée.

  Rechercher la valeur n'importe où dans le champ ou en début de champ .

- **des fonctions de déplacement dans les enregistrements du résultat de la recherche**

  Premier enregistrement

  Enregistrement précédent

  Enregistrement suivant

  Dernier enregistrement

- **La grille de résultat**

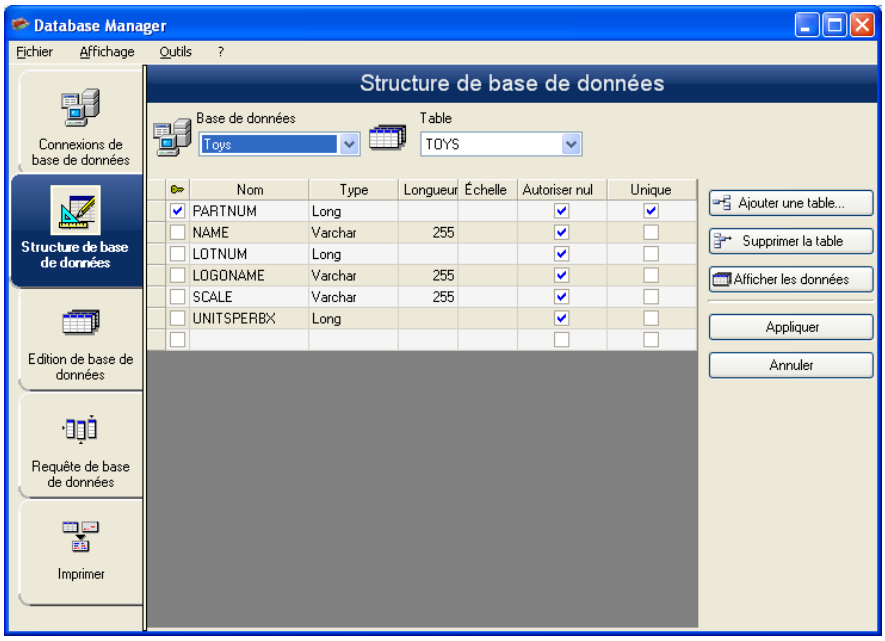
  Affiche les occurrences du résultat de la requête.

- **Re-exécuter**

  Re-exécute la requête et met à jour la grille de résultat.

Database Manager

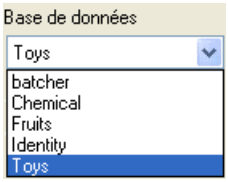
L'interface Structure de base de données



L'interface Structure de base de données vous permet de gérer la structure du fichier de base de données : ajout / modification / suppression de tables, champs...

Choisir une base de données dans la liste des connections

1. Cliquez sur la liste déroulante **Base de données**.
2. Cliquez sur la valeur voulue.



Choisir une table dans une base de données

1. Cliquez sur la liste déroulante **Table**.
2. Cliquez sur la valeur voulue.

Ajouter une table dans la base de données active

1. Cliquez sur le bouton **Ajouter une table**.
2. Saisissez le nom de la nouvelle table.
3. Cliquez sur le bouton **OK**.

Vous pouvez également copier la structure de la table sur ne table déjà existante dans la base de données électionnée. Pour cela :

1. Cliquez dans la case à cocher **Dupliquer avec**.
2. Cliquez ans la liste déroulante.
3. Cliquez sur la valeur voulue.
4. Cliquez sur le bouton **OK**.

Supprimer une table dans la base de données active

1. Cliquez dans la liste déroulante **Table**.
2. Cliquez sur la valeur voulue.
3. Cliquez sur le bouton **Supprimer**.

Afficher / masquer les données de la table active

1. Cliquez sur le bouton **Afficher les données**.

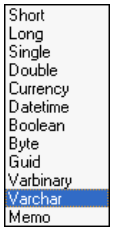
Déterminer un champ comme champ clé

1. Cliquez dans la case à cocher du champ voulu.

2. Cliquez sur le bouton **Appliquer**.

Déterminer le type de contenu d'un champ

1. Cliquez dans la case **Type** du champ voulu.
2. Cliquez sur le bouton de la liste déroulante.
3. Cliquez sur la valeur voulue.



4. Cliquez sur le bouton **Appliquer**.

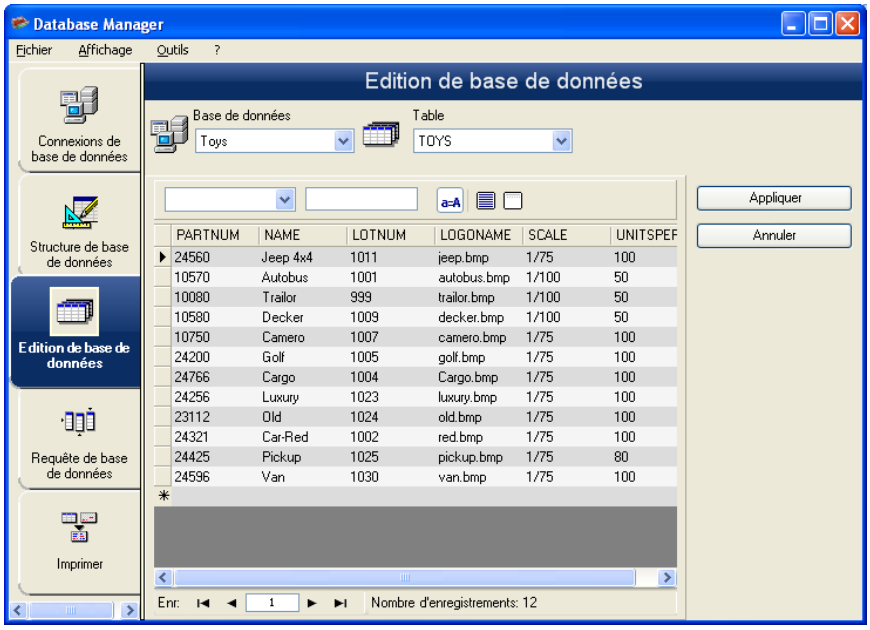
Déterminer la taille maximum d'un champ

3. Cliquez dans la case **Longueur** du champ voulu.
4. Saisissez la valeur voulue.
5. Cliquez sur le bouton **Appliquer**.

Autoriser un champ vide

1. Cliquez dans la case à cocher **Autoriser Nul** du champ voulu.
2. Cliquez sur le bouton **Appliquer**.

L'interface Edition de base de données



L'interface Edition de base de données vous permet de gérer le contenu du fichier de base de données : ajout / modification / suppression de données.

Ces actions dépendent du type de base de données. Ainsi, les enregistrements de fichiers **Excel**, ne peuvent pas être modifiés.

Sélectionner les enregistrements en fonction de leur contenu

Trouver un enregistrement en fonction du contenu d'un champ.

1. Cliquez sur le bouton de liste déroulante.
2. Cliquez sur la valeur voulue.
3. Cliquez dans le champ de saisie.
4. Saisissez la valeur voulue dans le champ de saisie.

Sélectionner tous les enregistrements identiques

Remarque: Au moins un enregistrement doit avoir été trouvé.

1. Cliquez sur le bouton de liste déroulante.
2. Cliquez sur la valeur voulue.
3. Cliquez dans le champ de saisie.
4. Saisissez la valeur voulue dans le champ de saisie.
5. Click on the Select all button ( )

Sélectionner un enregistrement identique

Remarque: Au moins un enregistrement doit avoir été trouvé. Il doit exister plusieurs contenus identiques dans le champ de recherche.

Pour sélectionner un enregistrement : Premier, Précédant, Prochain, Suivant, cliquer sur le bouton 1, 2, 3 ou 4 de l'outil de recherche.

Créer un nouvel enregistrement

1. Cliquer dans un champ de la ligne de saisie, signalée par une astérisque.
2. Saisir les valeurs voulues dans les champs correspondants.
3. Cliquer sur le bouton **Appliquer**.

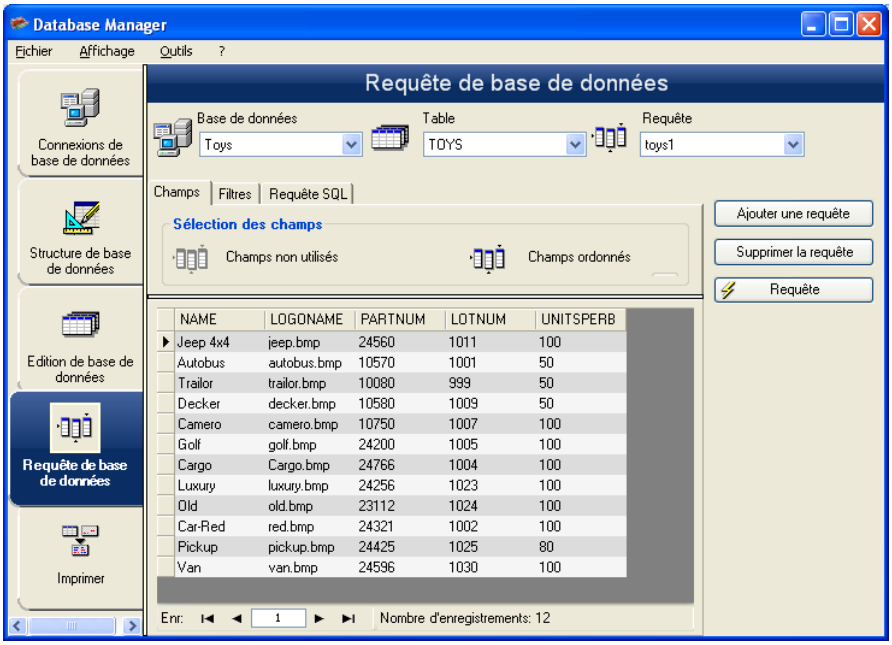
Modifier un enregistrement

1. Cliquer sur la valeur à modifier.
2. Saisir la valeur voulue.
3. Cliquer sur le bouton **Appliquer**.

Supprimer un enregistrement

1. Cliquer sur la case de curseur d'enregistrement du champ voulu.
2. Cliquer droit sur la case de curseur d'enregistrement du champ voulu.
3. Cliquer sur le menu contextuel **Supprimer** l'enregistrement sélectionné.

L'interface Requêtes de base de données



Interface où l'on crée et applique différents filtres.

Ajouter une requête

1. Cliquez sur le bouton **Ajouter une requête**.
2. Saisissez un nom pour la requête.
3. Cliquer sur le bouton **OK**.

Sélectionner / Désélectionner un ou des champs

1. Utilisez la barre d'outil .
2. Cliquez sur le bouton **Requête**.

Modifier l'ordre des champs sélectionnés

1. Cliquez sur le champ voulu dans la fenêtre **Champs ordonnés**.
2. Cliquez sur le bouton **flèche haut** ou **flèche bas** jusqu'à la valeur voulue.
3. Cliquez sur le bouton **Requête**.

Créer un filtre au moyen de données prédéfinies

1. Sélectionnez l'onglet **Filtres**.
2. Cliquez sur le bouton **Ajouter la ligne** ( ).
3. Cliquez dans le champ **Champ**, cliquez sur le bouton de liste déroulante et sélectionnez sur la valeur voulue.
4. Dans le champ **Opérateur**, cliquez sur le bouton de liste déroulante et sélectionnez sur la valeur voulue.
5. Cliquez dans le champ **Valeur**, saisissez sur la valeur voulue.
6. Cliquez sur le bouton **Requête** pour afficher le résultat.

Appliquer un opérateur logique à plusieurs filtres

1. Cliquez sur le bouton **Ajouter une ligne** ( ).
2. Dans le champ ”r;Logical, sélectionnez dans liste déroulante la valeur voulue (AND or OR).
3. Cliquez sur bouton **Requête** pour appliquer et visualiser les modifications.

Supprimer un filtre

Remarque: Au moins un filtre doit avoir été créé au préalable.

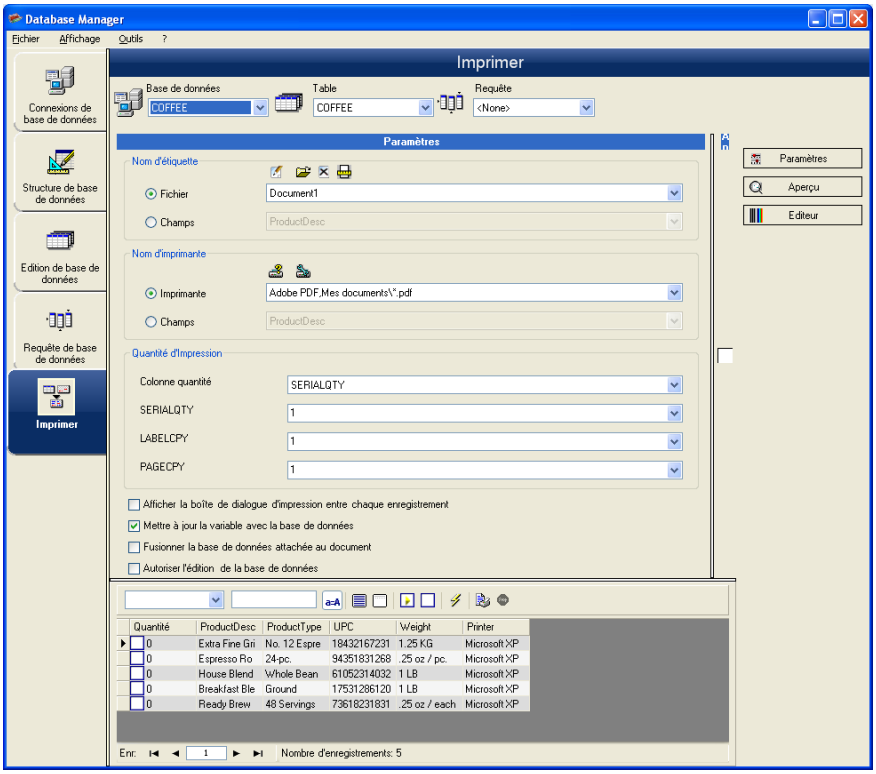
1. Cliquez sur la case du curseur du champ voulu.
2. Cliquez sur le bouton **Supprimer la ligne** ( ).

Modifier un filtre en langage SQL

Remarque: Au moins un filtre doit avoir été créé au préalable.

1. Sélectionnez l'onglet **Requête SQL**.
2. Cliquer dans la case à cocher ”r;Modifier la requête en langage SQL” pour activer la requête en langage SQL et pouvoir la modifier manuellement.
3. Cliquer sur le bouton **Requête** pour afficher le résultat.

L'interface Imprimer



Interface où l'on sélectionne le fichier d'impression, l'imprimante dédiée, et les différents paramètres avant de lancer l'impression.

Sélectionner le document à imprimer

1. Dans le menu **Fichier**, sélectionnez **Ouvrir** ou cliquez sur la bouton .
2. Sélectionnez votre document.

- ou -

1. Cliquer sur le bouton **Paramètres** puis sur l'**Assistant de création d'étiquettes** ( ).
2. Suivez les instructions de l'assistant.

Remarque: La création d'une étiquette en relation avec la base de données vous permet de définir exactement les éléments nécessaires à la mise en place des différents champs de la base de données.

Sélectionner un modèle d'étiquette existant

1. Cliquez sur le bouton **Ouvrir un document existant** ( ).
2. Sélectionnez un fichier .lab.
3. Cliquer sur le bouton **OK**.

Remarque: Les boutons radio **Champ** dans les groupes d'option **Nom d'étiquette** et **Nom d'imprimante** permettent de choisir l'étiquette ou l'imprimante voulues lorsque celles ci sont définies dans un champ de la base de données active.

Sélectionner un document à partir d'un champ

Si votre base de données contient un champ **Noms d'étiquettes (Labname)**, vous pouvez l'utiliser afin de lancer une impression.

Ref	Désignation	Qt	Code	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

Dans cet exemple, le champ **Noms d'étiquette** peut être utilisé.

1. Cliquez dans les cases à cocher des enregistrements voulus.

Sélectionner une imprimante

Cliquez sur le bouton **Ajouter** ou **Supprimer** une imprimante, puis sélectionnez l'imprimante à utiliser.

Formules

La source de données Formule

La source de données **Formule** contient une liste de variables créées au fur et à mesure de vos besoins. Ces variables sont renseignées par des combinaisons d'opérateurs, de constantes, de variables, de variables de contrôle, de formules et de fonctions. Les données sont de types numériques ou alphanumériques.

Les fonctions

Remarque : Les paramètres régionaux du système Windows peuvent affecter la sortie de certaines formules.

Les fonctions sont des formules prédéfinies qui effectuent des calculs en utilisant des valeurs particulières appelées arguments, dans un certain ordre appelé syntaxe.

Les fonctions sont utilisées pour retourner une valeur de type numérique, chaîne de caractères ou logique résultant d'un calcul ou d'une opération.

Il existe des groupes de fonctions dans la définition des formules:

- Fonctions de calcul des caractères de contrôle (check digits).
- Fonctions de conversion
- Fonctions sur la date et l'heure
- Fonctions logiques
- Fonctions mathématiques
- Fonctions sur les chaînes de caractères

Les opérateurs

Les opérateurs sont des symboles mathématiques indiquant une opération à réaliser. On distingue différents types d'opérateurs : arithmétiques, de comparaison, de concaténation et logiques.

On distingue différents types d'opérateurs : arithmétiques, de comparaison, de concaténation et logiques.

Les opérateurs arithmétiques

Opérateur	Utilisé pour
*	Multiplier deux nombres
+	Additionner des nombres
-	Soustraire un nombre d'un autre ou donner une valeur négative à un opérande
/	Diviser un nombre par un autre
^	Élever un nombre à la puissance de l'exposant
%	Modulo

Remarque : Certaines formules ne retournent pas d'erreur alors qu'elles devraient. Par exemple : 1/0, 0/0, 0.0/0.0, 1/000 et 0^(-2). Ceci est intentionnel pour éviter de bloquer un processus d'impression. L'utilisateur doit définir le comportement souhaité en ajoutant des contrôles si besoin.

Les opérateurs de comparaison

Opérateur	Signification
<	Inférieur à
<=	Inférieur ou égal à
>	Supérieur à
>=	Supérieur ou égal à
=	Égal
<>	Différent

L'opérateur de concaténation

Il sert à combiner deux chaînes.

Opérateur	Signification
&	Concaténation de deux chaînes

L'opérateur logique

(voir aussi les fonctions logiques)

Opérateur	Signification
!	Non logique

Les fonctions de calcul des caractères de contrôle

Remarque : Les paramètres régionaux du système Windows peuvent affecter la sortie de certaines formules.

addmodulo10 («chaîne») : Retourne la valeur entière avec le caractère de contrôle Modulo 10 ajouté à la fin.

addmodulo10_212(«chaîne») : Retourne un caractère de contrôle modulo 10_212 (une variante du caractère de contrôle standard modulo 10), et inclue la chaîne texte courante et la somme de contrôle à la fin de la chaîne.

addmodulo43(«chaîne») : Retourne un caractère de contrôle modulo 43 et inclue la chaîne texte courante et le caractère de contrôle à la fin de la chaîne.

bimodulo11 («chaîne») : Retourne 2 caractères de contrôle modulo 11 (code 11).

canadacustomscd («chaîne») : Retourne le caractère de contrôle du Canada Customs Standard.

check103(«chaîne») : Retourne le caractère de contrôle Mod103.

check128 («chaîne») : Retourne le caractère de contrôle du code 128 et EAN 128.

checkPZN(«chaîne») : Retourne le caractère de contrôle du code PZN. Le code-à-barres PZN est basé sur le code-à-barres Code 39 utilisé pour les produits pharmaceutiques en Allemagne.

CheckSum2Mod47(«chaîne») : Retourne le caractère de contrôle Mod47.

CheckSumMod10(«chaîne») : Retourne le caractère de contrôle Mod10.

CheckSumMod10_Codabar(«chaîne») : Retourne le caractère de contrôle spécifique Mod10 pour une code-à-barres Codabar.

CheckSumMod10_MSI(«chaîne») : Retourne le caractère de contrôle spécifique Mod10 pour une code-à-barres MSI.

CheckSumMod11_3Suisse(«chaîne») : Retourne le caractère de contrôle spécifique Mod11 pour les 3Suisses.

CheckSumMod34(«chaîne») : Retourne le caractère de contrôle spécifique Mod34.

checkupce («chaîne») : Retourne le caractère de contrôle pour UPC-E (6 caractères obligatoires).

CRC16(«chaîne»): Retourne le caractère de contrôle CRC-16.

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

modulo10 («chaîne») : Retourne un caractère de contrôle modulo 10 du code 2/5 entrelacé, EAN13, EAN 8, EAN 128 K-Mart, VPCA.

modulo10_212(«chaîne») : Retourne une variante du caractère de contrôle modulo 10

modulo10_2 («chaîne») : Retourne un variant du caractère de contrôle Modulo 10.

modulo10IBM(«chaîne») : Retourne un caractère de contrôle IBM modulo 10 (une variante du caractère de contrôle standard modulo 10)

modulo10UPS(«chaîne») : Retourne un caractère de contrôle UPS modulo 10 (une variante du caractère de contrôle standard modulo 10)

modulo11 («chaîne») : Retourne un caractère de contrôle modulo 11 (code 11).

modulo11IBM(«chaîne») : Retourne le caractère de contrôle modulo 11 IBM (une variante du caractère standard modulo 11)

modulo16(«chaîne»): Retourne un caractère de contrôle modulo 16.

modulo24 («chaîne») : Retourne un caractère de contrôle modulo 24 (code PSA).

modulo32 (valeur) : Retourne un caractère de contrôle modulo 32 (industrie pharmaceutique Italie).

modulo43 («chaîne») : Retourne un caractère de contrôle modulo 43 (code 39).

modulo47 («chaîne») : Retourne deux caractères de contrôle modulo 47 (code 93).

plessey («chaîne») : Retourne deux caractères de contrôle (code Plessey).

pricecd («chaîne») : Retourne 4 caractères de contrôle UPC Random Price.

pricecd5 («chaîne») : Retourne 5 caractères de contrôle UPC Random Price.

StringToExt39(«chaîne») : Prépare la chaîne qui doit être utilisée pour le code-à-barres Extended 39.

UPSCheckDigit («chaîne»): Retourne les caractères de contrôle UPS. Lle paramètre en entrée de cette méthode est une chaîne. Cette méthode vous retourne le reste du numéro de suivi. Le caractère de contrôle est calculé à partir d'une séquence de 15 caractères.

1. Les deux premiers caractères doivent être "1Z".
2. Les six caractères suivants représentent votre numéro de compte UPS "XXXXXX".
3. Les deux caractères suivants représentent le type de service:
    - "01" pour les expéditions aériennes du lendemain.
    - "02" pour les expéditions aériennes dans deux jours.
    - "03" pour les expéditions par voie terrestre.
4. Les cinq caractères suivants représentent votre numéro de facture (factures avec six caractères; supprimez le premier caractère. Exemple avec un numéro de facture tel que 123456, vous prendrez en compte 23456).
5. Les deux caractères suivants représentent le numéro de colis, précédé du chiffre 0. Le colis numéro un sera représenté "01", le numéro deux sera "02".
6. Le dernier caractère est le caractère de contrôle.

Note: The described sequence above gives 17 characters, where as only need 15 are needed to calculate the check digit. To do this, drop the "1Z" portion, and only use the last 15 characters in the method.

Les fonctions de conversion

Remarque : Les paramètres régionaux du système Windows peuvent affecter la sortie de certaines formules.

AI253 («chaîne»): Fonction spécifique pour la préparation de chaîne de l'identifiant de données (AI) 253.

AI8003 («chaîne»): Fonction spécifique pour la préparation de chaîne de l'identifiant de données (AI) 8003.

Ascii («chaîne») : Retourne le code ASCII du premier caractère de l'argument chaîne.

Exemple :
ascii("A") = 65

char («entier»)

- Pour les valeurs comprises entre 128 et 255: retourne le caractère correspondant à la valeur de l'argument entier dans la table ASCII.
- Pour les valeurs négatives ainsi que pour les valeurs comprises entre 0 et 127 et supérieures à 255: retourne le caractère UNICODE correspondant à «entier».

Exemple :
char (65) = "A"

CodabarData(«data», «start», «stop»): Ajuste les données pour CodabarData.

Code128CData(«chaîne»): Ajuste les données pour Code 128 C.

Currencytoeuro («valeur») : Convertit la monnaie courante en Euro.

DatabarData(«data», «min», «max», «hasComposite»): Ajuste les données pour DatabarData.

DBCSToUnicode («chaîne», «codepage»): La formule prend 2 paramètres - la donnée à être convertie et la page de code utilisée. Le paramètre «*codepage*» est le nom en anglais d'une langue (Thai, Japanese, Chinese GBK, Korean, ChineseBig5, European, Eastern, Cyrillic, Greek, Turkish, Hebrew, Arabic, Baltic, Vietnamese, UTF-8, ACP) ou une valeur numérique correspondant à un identifiant de la page de code pour chaque langue (codepage).

Exemple
DBCSToUnicode(unicodetoDBCS("傍傍傍傍", UTF-8"), UTF-8)=傍傍傍傍"

Cette fonction est utilise pour les données provenant d'un fichier qui n'est pas unicode ou d'une base de données.

Remarque: La fonction inverse est également disponible. Voir UnicodeToDBCS.

dollar («valeur») : Modifie un champ pour représenter un prix.

Exemple :
Si un champ nommé PRICE affiche une valeur 199, alors
dollar(PRICE) produira $199,00.

Ean128Data («chaine_1»,«chaine_2»): Ajuste la donnée "chaine_1" pour le code-barres EAN 128 selon le modèle "chaine_2"

Eurotocurrency («valeur») **:** Convertit l'Euro en monnaie courante.

Remarque : le nombre de décimales à afficher doit être défini à partir de l'onglet **Sortie** de la variable **Formule**, en validant la case à cocher **Afficher les décimales**. D'autre part, le taux de conversion utilisé est celui défini dans l'onglet **Autres** de la boîte de dialogue **Options**.

FileToData(«fileName», «errorData», «maxSize»): Retourne les données lues à partir du fichier.

fixed («valeur» ,«nb_décimales»,«non_sép») Retourne une chaîne de caractères formatée avec un nombre de décimales égal à *nb_décimales* et avec ou sans séparateur des milliers.

Exemples :
fixed (1234.5678, 3, TRUE) = "1234.568"
fixed (1234.5678, 3, FALSE) = "1,234.568"

Remarque : le séparateur des milliers à afficher doit être défini à partir de l'onglet **Sortie** de la variable **Formule**, en validant la case à cocher **Afficher les décimales**.

FormatDate(«date», «dateFormat», «localeID»): **Convertit «date» en une chaîne mise en forme d'après «dateFormat».**

Valeur de retour	Chaîne	Une valeur date sous forme de chaîne, formatée suivant un format.
date	Chaîne	Valeur date sous forme de chaîne.
dateFormat	Chaîne/ Nombre	Format de la valeur de date de base. - Les nombres encodent des formats prédéfinis. (par ex. 1 "dd/mm/yy", 2 "dd/mm/yyyy" etc. Les valeurs possibles sont disponibles ici. - 0 ou une valeur négative indique l'utilisation du format de date du système.

		- La valeur de la chaîne de caractères encode directement le format (par ex. "yyyy/mm/dd").
LocaleID	Number	Spécifie la langue à utiliser pour afficher la valeur de la date. Ce paramètre est optionnel (par défaut: 0, paramètre régional du système utilisé). Les valeurs possibles peuvent être consultées sur 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) : Permet d'ajouter un symbole monétaire, des points, des zéros, des caractères de séparation.

Exemple:

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

Remarque: **Location for price character** définit la position du symbole de la monnaie, des espaces sont ajoutés entre le symbole de la monnaie et le montant. Cette valeur est ignorée si elle est inférieure à la longueur de la chaîne résultat.

FormatNumber *(«number»)*: Permet de formater un champ numérique là où un signe dièse (#) signifie d'afficher seulement la valeur présente et zéro (0) signifie d'afficher la totalité de la valeur numérique y compris les 0.

Exemples:

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(«nom»): Retourne la valeur «nom» (chaîne) de la variable environnement

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

Remarque : Cette fonction fonctionne avec des variables environnement définies par l'utilisateur

GS1AIData («AIName», «AIData», «isLastAI», «calcCheckDigit»): Ajuste les données pour GS1 AI.

GS1HRData («AIName», «AIData», «calcCheckDigit»): Ajuste les données pour l'interprétation GS1.

GS1Norm («string»): Ajuste les données pour la norme GS1.

int («valeur») Retourne le plus grand des entiers inférieur ou égal à l'argument *valeur* .

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

LmChar («entier») : Retourne le caractère correspondant à la valeur de l'argument entier dans la table ANSI. Cette table est indépendante des paramètres régionaux du système. La valeur doit être comprise entre 0 et 255.

MaxiCodeData(«Mode»,«PostalCode»,«CountryCode»,«ClassOfService»,«TrackingNumber»,«UpsShipperNumber»,«JulianDate»,«ShipmentID»,«PackageNumber»,«TotalPackages»,«PackageWeight»,«AdressValidation»,«ShipToAdress»,«ShipToCity»,«ShipToState»): Crée des données maxicode à partir des données d'entrée «string».

text («valeur», «format») Retourne l'argument *valeur* écrit de la manière imposée par l'argument *format* qui correspond au format décrit lors de l'explication de la variable Formulaire.

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

trunc («valeur») Retourne la partie entière de l'argument *valeur* .

Exemple :
trunc (123.45) = 123

unicodetoDBCS(«chaîne_1», «chaîne_2)»: La formule prend 2 paramètres - la donnée à être convertie et la page de code utilisée. Le paramètre de la page de code peut avoir une des valeurs suivantes:

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

Note: La casse de la page de Code est importante

value («chaîne») Retourne la valeur numérique d'une chaîne de caractères.

Exemple :
value("123") = 123
value("16:48:00")-value("12:00:00") égale "16:48:00"-"12:00:00" égale 0,00, le numéro de série équivalent à 4 heures et 48 minutes.

Remarque : il n'est généralement pas nécessaire d'utiliser la fonction value dans une formule, car l'application convertit automatiquement le texte en nombres si nécessaire.

ValueEx(«chaîne»): Convertit «string» en une valeur numérique.

VoiceCode(«chaîne»,«chaîne»,«chaîne»): Retourne une valeur numérique à 4 chiffres calculés à partir d'un code GTIN, d'un lot et optionnellement de la date à partir d'un PTI.

valeur de renvoi	chaîne	Renvoie la valeur VoiceCode
GTIN	chaîne	Le GTIN est un code à 14 chiffres. Seuls des chiffres sont autorisés. Une erreur sera retournée si vous entrez un autre caractère. Seuls les 14 premiers chiffres à

		partir de la gauche seront pris en compte si vous en entrez plus. Des '0' seront ajoutés à gauche si votre code comporte moins de 14 chiffres.
LotNumber	chaîne	Cette donnée peut accepter jusqu'à 20 caractères alphanumériques. Si cette donnée excède les 20 caractères, seuls les 20 premiers en partant de la gauche seront pris en compte.
Date	chaîne	Ce paramètre est optionnel. 6 chiffres représentant la date YYMMDD (AAMMJJ) peuvent être entrés. Une erreur sera retournée si la longueur ou les caractères entrés sont incorrects.

Ce calcul est effectué comme suit:

1. Calcul de la chaîne PlainText.

   a. La chaîne PlainText se compose des 14 chiffres du code GTIN agrémentés du code Lot et de la Date (si nécessaire car ce paramètre est optionnel).

   b. Ne pas inclure de préfixes ou les parenthèses.

   c. Ne pas inclure d'espace entres les champs du GTIN, du Lot et de Date.

   d. Si vous entrez un champ Date, conformez-vous au format YYMMDD (aucun autre format ou caractère tel que '/'ne sera accepté).

2. Calcul de la valeur de hachage ANSI CRC-16 des octets ASCII de PlainText en utilisant la méthode standard de hachage ANSI CRC-16 avec le polynome X16 + X15 + X2 + 1.
3. Calcul du VoiceCode à partir de la valeur de hachage en prenant les 4 derniers chiffres significatifs au format décimal (mode de hachage 10000).

   <u>Exemple:</u>

   GTIN = (01) 10850510002011
   Lot = (10) 46587443HG234
   PlainText = 1085051000201146587443HG234
   CRC-16 Hash = 26359
   VoiceCode = 6359
   Large Digits = 59
   Small Digits = 63

   <u>Exemples:</u>

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

Fonctions de conversion ATA 2000

Remarque : Les paramètres régionaux du système Windows peuvent affecter la sortie de certaines formules.

Bin2Hex(«value», «pad»)

Convertit une valeur binaire en valeur hexadécimale. Le 'caractère de remplissage' est utilisé pour définir la longueur totale de la sortie en ajoutant les caractères 0 à gauche du résultat.

Exemple:
Bin2Hex("0010100111101001101",0) retourne 14F4D
Bin2Hex("0010100111101001101",0) retourne 00014F4D

CRC16_CCITT(«chaine»)

Calcule a somme de contrôle CRC16 CCITT.
Cela est utilisé dans la réglementation ATA2000 pour TOC pour le calcul du CRC.

Exemple:
CRC16_CCITT ("A") retourne B915
CRC16_CCITT ("AB") retourne 4B74
CRC16_CCITT ("ABC") retourne F508

CRC32_CCITT(«chaine»)

Calcule a somme de contrôle CRC32 CCITT.
Cela est utilisé dans la réglementation ATA2000 pour TOC pour le calcul du CRC.

Exemple:
CRC32_CCITT ("A") retourne D3D99E8B
CRC32_CCITT ("AB") retourne 30694C07
CRC32_CCITT ("123456789") retourne CBF43926

Dec2Bin(«value», «pad»)

Convertit une valeur décimale en valeur binaire. Le 'caractère de remplissage' est utilisé pour définir la longueur totale de la sortie en ajoutant les caractères 0 à gauche du résultat.

Exemple :

Dec2Bin("5", 0) retourne 101

Dec2Bin("5", 4) retourne 0101

Dec2Hex(«value», «pad»)

Convertit une valeur décimale en valeur hexadécimale. Le 'caractère de remplissage' est utilisé pour définir la longueur totale de la sortie en ajoutant les caractères 0 à gauche du

Exemple:

Dec2Hex(510, 0) retourne 1FE

Dec2Hex(510, 4) retourne 01FE

String2Hex(«value», «pad»)

Convertit une valeur chaîne en valeur hexadécimale. Le 'caractère de remplissage' est utilisé pour définir l'alignement en octets de la sortie en ajoutant les caractères 0 à droite du résultat.
Dans RFTAG, la donnée est très souvent encodée en hexadécimal avec un alignement word (2 octets).

Exemple:

String2Hex("A", 0) retourne 41 (**A** a le code AINSI **65** en décimal ce qui donne **41** en hexadécimal)

String2Hex("ABC", 2) retourne 414243

String2Hex("ABC", 4) retourne 41424300

Voici un exemple pour un enregistrement de naissance de RFID ATA2000 :

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) retourne
4D46522053303637312A534551204D333747584239322A504E4F20505137565A342A504454204
34F4E54524F4C4C45522A49

String6BitsEncoding(«value», «pad»)

Convertit une valeur chaîne en une valeur encodée 6-bit ASCII. Le 'caractère de remplissage' est utilisé pour définir l'alignement en octets de la sortie en ajoutant les caractères 0 à droite du résultat.
Dans une donnée ATA2000, la donnée peut être encodée sur 6 bits (au lieu de 8 bits) pour une question de compression.

Exemple:
String6BitsEncoding("11", 0) retourne C710

VDAString6BitsEncoding(«chaine»)

Convertit une valeur chaîne en une valeur encodée 6-bit ASCII en utilisant la norme VDA (German Association of the Automotive Industry).
Dans une donnée ATA2000, la donnée peut être encodée sur 6 bits (au lieu de 8 bits) pour une question de compression.
La table utilisée pour la conversion est disponible ici. Pour plus de détails, référez-vous aux spécifications VDA5500 et VDA4994.

Exemple:
VDAString6BitsEncoding("11") retourne C71860820820

Les fonctions sur la date et l'heure

La variable date ci-dessous représente la date et l'heure du système et non pas la variable date définie par le programme.

Remarque : Les paramètres régionaux du système Windows peuvent affecter la sortie de certaines formules.

Remarque: 30/12/1899 = la date de référence de votre application. Si vous voulez que la formule renvoie la date courante, la solution est de calculer le nombre de jours écoulés depuis la date de référence.

BestBefore(«date», «Formatdate» , «décalage» , «Unitédedécalage» , «changementMois» , «Formatsortie», «localeID»)

Vous permet de calculer une date limite d'utilisation optimale (DLUO) basée sur la valeur saisie au clavier (et non sur la date courante).

Valeur de renvoi	chaîne de caractères	Valeur de date dans le format chaîne de caractères, calculée à partir d'une date de base et d'un décalage (intervalle).
date	chaîne de caractères	Valeur de date de base dans le format chaîne de caractères
Formatdate	chaîne de caractères / numéro	Format de la valeur de date de base. - Les nombres encodent des formats prédéfinis. (par ex. 1 "dd/mm/yy", 2 "dd/mm/yyyy" etc. Les valeurs possibles sont disponibles ici. - 0 ou une valeur négative indique l'utilisation du format de date du système. - La valeur de la chaîne de caractères encode directement le format (par ex. "yyyy/mm/dd").
décalage	numéro	Nombre d'unités de date à ajouter à la date de base.
Unitédedécalage	chaîne de caractères / numéro	Signification du paramètre offset. - **1** ou "**d**" ou "**D**" signifie jours - **2** ou "**m**" ou "**M**" signifie mois - **3** ou "**y**" ou "**Y**" signifie années
changementMois	numéro	Option (par défaut : True) - **0** signifie **False** (faux) - **1** signifie **True** (vrai) Si la date calculée n'existe pas dans le mois (par ex. 30 février), - **True** renvoie le premier jour du mois suivant (c.-à-d. le 1er mars). - **False** renvoie le dernier jours du mois courant (c.-à-d. le 28 février).
Formatsortie	chaîne de caractères / numéro	Option (par défaut : identique à dateFormat) Ce paramètre spécifie le format d'édition de la date calculée. Les valeurs possibles sont disponibles ici.
localeID	numéro	Optionnel (par défaut: 0, paramètre régional du système utilisé) Ce paramètre spécifie la langue à utiliser pour afficher la valeur de la date.

Exemple :

BestBefore("14/06/2012" ,"dd/mm/yyyy","18","m", 1, "mmmm dd, yyyy", 1033) retournera December 14, 2013.

BestBefore("14/06/2012" ,"dd/mm/yyyy","18","m", 1, "mmmm dd, yyyy", 1036) retournera Décembre 14, 2013.

Exemple: Formule avec une source de données Date.

Créez tout d'abord une variable date "date0" avec la date du jour au format dd/mm/yy (26/06/2012 pour cet exemple).

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

CFIA_Month(«date»):

Retourne le mois selon Canadian Food Inspection Agency (CFIA): JA, FE, MR, AL, MA, JN, JL, AU, SE, OC, NO, DE.

Exemple:

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

CFIA_Month (today()) retourne le mois en fonction de la date du jour, s'il est aujourd'hui le 01/01/2021 - la valeur retournée est JA

CFIA_Month ("06/01") retourne JN

Note: Le format de l'argument «date» dépend des réglages du système.

DateOffset(«date» , «décalage», «Unitédedécalage» , «changementMois»)

valeur de renvoi	date	Ajoute un intervalle de date à la date de base spécifiée.
date	date	Valeur de date de base.
décalage	numéro	Nombre d'unités de date à ajouter à la date de base.
Unité de décalage	chaîne de caractères / numéro	Signification du paramètre offset. - **1** ou "**d**" ou "**D**" signifie jours - **2** ou "**m**" ou "**M**" signifie mois - **3** ou "**y**" ou "**Y**" signifie années
changement Mois	numéro	Option (par défaut : True) - **0** signifie **False** (faux)

		- **1** signifie **True** (vrai) Si la date calculée n'existe pas dans le mois (par ex. 30 février), - **True** renvoie le premier jour du mois suivant (c.-à-d. le 1er mars). - **False** renvoie le dernier jours du mois courant (c.-à-d. le 28 février).

Exemple:
DateOffset("26/06/2012",1,"d",1) retournera 27/06/2012
-OU-
DateOffset(today(),1,"D", 0)

Exemple: Formule avec une source de données Date.
Créez tout d'abord une variable date "date0" avec la date du jour au format dd/mm/yy (26/06/2012 pour cet exemple). Veuillez vous assurer que le format est celui que votre système utilise (autrement veuillez utiliser la fonction DateValue().).

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

DateValue(«Dateformatée» , «Formatdate»)

valeur de renvoi	date	Renvoie une valeur de date du paramètre formattedDate.
Dateformatée	chaîne de caractères	Date au format texte, à convertir en valeur de date.
Formatdate	chaîne de caractères	Option (par défaut : le format de date du système) Si elle est spécifiée, la chaîne de caractères du format direct, par ex. "dd/mm/yy"

Exemple:
DateValue(22062012,"ddmmyyyy") retournera 22/06/2012.

day («date») retourne le jour du mois de l'argument *date*.

FiscalDate(«fiscalStartDate», «outputFormat»)

Permet de calculer la date de départ de votre exercice fiscal (yyyy.mm.dd).

Exemples (si la date d'aujourd'hui est 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»*) retourne l'heure de l'argument *date*.

minute (*«date»*) retourne les minutes de l'argument *date*.

month (*«date»*) retourne le mois de l'argument *date*.

now () retourne la date et l'heure courantes.

second («date») : retourne les secondes de l'argument de date.

shiftcode(«items», «DefaultValue»)

Permet d'affecter une valeur à un champ en fonction de l'horaire actuelle. Pour calculer cette valeur, des tranches horaires doivent être prédéfinies.

valeur de renvoi	chaîne de caractères	Renvoie la valeur actuelle Shift code.
Item	chaîne de caractères	Format: startHour:startMin-stopHour:StopMin-value|...| startHour:startMin-stopHour:StopMin-value

Exemple (si il est actuellement 16h):

Tranches horaires:
7:00 à 15:00 = Jour
15:00 à 23:00 = Soir
23:00 à 7:00 = Nuit

shiftcode("7:00-15:00-Jour|15:00-23:00-Soir|23:00-7:00-Nuit") ="Soir"

Remarques:

1. Pour une horloge définie sur 24 heures. 0:00 correspond à minuit et 12:00 à midi.
2. Si un chevauchement d'horaires se présente, seule la valeur de la première horaire sera retournée.
3. Si aucun changement est spécifié pour l'heure actuelle, la valeur retournée sera vide.

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

Permet de personnaliser une date en la décalant avec un certain intervalle et d'utiliser cette date personnalisée dans une formule. La fonction SpecificDateFormat décale la date actuelle sur l'horloge système en utilisant soit le format de la valeur de la date par défaut soit un format de votre choix.

- [date format] Spécifie le format de date à utiliser. Ce paramètre doit être entré entre des parenthèses. ***Voir*** les formats de dates valides disponibles.

Paramètre de la Date	Résultat	Description
D and d	1	la date du mois (sans zéro)
DD	01	2 caractères pour la date du mois (précédé d'un zéro)
dd	1	2 caractères pour la date du mois (précédé d'un espace)
DDD	224	jours à partir du début de l'année (précédé d'un zéro)
ddd	224	jours à partir du début de l'année (précédé d'un espace)
DDDDD and ddddd	33482	jours à partir du 1/1/1900 (5 à 9 caractères disponibles)
W and w	1	jour de la semaine (Dimanche=1, Samedi=7)
WW	34	semaines à partir du début de l'année (précédé d'un zéro)
ww	34	semaines à partir du début de l'année (précédé d'une espace)
WWW and www	783	semaines à partir du 1/1/1900 (3 derniers caractères)
WWWW and wwww	4783	semaines à partir du 1/1/1900 (4 à 9 caractères disponibles)
WWWWWWWWW	000004783	semaines à partir du 1/1/1900 (précédé d'un zéro)
wwwwwwwww	4783	semaines à partir du 1/1/1900 ((précédé d'un espace)
M and m	9	mois (sans zéro)
MM	09	2 caractères pour représenter le mois (précédé d'un zéro)

mm	9	2 caractères pour représenter le mois (précédé d'un espace)
MMM	009	3 caractères pour représenter le mois (précédé d'un zéro)
mmm	9	3 caractères pour représenter le mois (précédé d'un espace)
MMMM and mmmm	1101	mois depuis 1/1/1900
MMMMM	01101	5 caractères pour représenter le mois depuis 1900 (précédé d'un zéro)
mmmmm	1101	5 caractères pour représenter le mois depuis 1900 (précédé d'un espace) (5 à 9 caractères disponibles)
MMMMMMMMM	000001101	9 caractères pour représenter le mois depuis 1900 (précédé d'un zéro)
mmmmmmmmm	1101	9 caractères pour représenter le mois depuis 1900 (précédé d'un espace)
YY and yy	91	2 caractères pour l'année
Y and y	1991	l'année dans son intégralité
YYY and yyy	991	les 3 derniers caractères de l'année (3 à 9 caractères disponibles)
YYYYYYYYY	000001991	l'année dans son intégralité (précédé d'un zéro)
yyyyyyyyy	1991	l'année dans son intégralité (précédé d'un espace)

- +/[offset data] Cette valeur indique le décalage à appliquer à la date. Cet intervalle peut être précédé soit d'un plus (+) soit d'un moins (), selon si vous désirez ajouter ou soustraire de votre date actuelle.
- [date interval] L'intervalle de la date peut être représenter par un d pour jours, w pour semaines, m pour mois, ou y pour années.

Exemples de la fonction SpecificDateFormat:

Remarque: Pour les exemples suivants, nous prendrons la date **July 29, 2013** comme date de référence.

SpecificDateFormat("mm dd yy" , "+1y") = 7 29 14 (1 an après la date actuelle)
SpecificDateFormat("ww/m/yyyy" , "30w) = 1/12/2012 (30 semaines avant la date actuelle)
SpecificDateFormat("ddddd / wwww /mmmm" , "") = 41484/5928/1363 (jours, semaines et mois depuis 1900)

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

Permet de personnaliser une heure en la décalant avec un certain intervalle et d'utiliser cette heure personnalisée dans une formule. La fonction TimeOffset décale l'heure actuelle sur l'horloge système en utilisant le format de la valeur de la date par défaut.

- Le format par défaut "mm/dd/yyyy hh:nn:ss" est le seul format qui puisse être utilisé. Pour en utiliser un autre, vous devez inclure la fonction **TimeOffset** dans la fonction **DateValue** et ensuite l'inclure dans une fonction **FormatDate** dans laquelle vous pourrez alors spécifier un autre format de date.
- “+/offset time”: Cette valeur indique le décalage à appliquer à l'heure. Cet intervalle peut être précédé soit d'un plus (+) soit d'un moins (), selon si vous désirez ajouter ou soustraire de l'heure actuelle.
- "time interval": L'intervalle de l'heure peut être représenter par un s pour secondes, m pour minutes, m pour mois, ou h pour heures.

Exemples:

Pour les exemples suivants, nous prendrons comme heure actuelle 4:12:10 après-midi le 30 juillet, 2013)

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"h+5") = 07/30/2013 21:12:10 (5 heures après l'heure actuelle)
FormatDate(DateValue(TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"), "n+10"), "mm/dd/yyyy hh:nn:ss"), "hhnn") = 16-22 (10 minutes après l'heure actuelle)

today () retourne la date courante.

Week *(*«date») retourne le numéro de la semaine de l’argument *date*.

weekday (*«date»*) retourne le jour de la semaine de l'argument *date*.

Remarque : dimanche est considéré comme le premier jour de la semaine.

Exemple :
soit "le mardi 20 novembre 1999", la date courante, on a weekday(now()) = 3

WeekISO8601(«Date», «DateFormat»)

Permet de calculer une formule selon la norme ISO8601. La norme internationale ISO 8601 spécifie la représentation numérique de la date et de l'heure.

valeur de renvoi	date	Renvoie la semaine de la date spécifiée.
Date	chaîne de caractères	Date au format texte, à convertir en valeur de date.
DateFormat	chaîne de caractères	Format de la valeur de date de base. - Les numéros encodent des formats prédéfinis. (par ex. 1 dd/mm/yy, 2 dd/mm/yyyy etc) - 0 ou une valeur négative indique l'utilisation du format de date du système. - La valeur de la chaîne de caractères encode directement le format (par ex. "yyyy/mm/dd").

Exemple :

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

year («date») retourne l'année de l'argument date.

Exemple :
minute (now ()) retourne les minutes de l'heure courante.
year (today ()) retourne l'année de la date courante.

Note: Pour Minute et Hour, seul l'argument now() est accepté. Pour les autres fonctions, now() et today() sont autorisés.

Les fonctions logiques

Les fonctions logiques permettent de vérifier si une ou plusieurs conditions sont réalisées ou non.

Remarque : Les paramètres régionaux du système Windows peuvent affecter la sortie de certaines formules.

Remarque : TRUE (Vrai) équivaut à 1 et FALSE (Faux) équivaut à 0

and («expr_1», «expr_2») Retourne TRUE (Vrai) si les deux arguments sont vrais, FALSE (Faux) si au moins un argument est faux. Les arguments doivent être évalués à des valeurs logiques.

Exemple :
and(exact(«chaîne»,«c haîne»),exact(«chaîne»,«chaîne»)) = 0
and(exact(«chaîne»,«chaîne»),exact(«chaîne»,«chaîne»)) = 1

exact («chaîne_1, «chaîne_2») Retourne TRUE (Vrai) si les deux chaînes sont identiques, FALSE (Faux) sinon . Cette fonction respecte les majuscules et les minuscules.

Exemple :
exact(«logiciel»,«logiciel») = 1
exact(«lgiciel»,«logiciel») = 0

if («expr», «Val_si_vrai», «Val_si_faux») Retourne l'argument Val_si_vrai si Expr est vrai et l'argument Val_si_faux si Expr est faux.

Exemple :
if(exact("chaîne", "chaînes"), "vrai", "faux") = faux
if(exact("chaîne", "chaîne"), "vrai", "faux") = vrai

not («logique») Retourne l'opposé de l'argument logique.

Exemple :

not(exact("c haîne", "chaîne")) = 1
not(exact("chaîne", "chaîne")) = 0
not(False) = 1 ou not(0) = 1
not(True) = 0 ou not(1) =0
not(1+1=2) = 0

or («expr_1», «expr_2») Retourne TRUE (VRAI) si un des deux arguments est vrai et FALSE (FAUX) si les deux arguments sont faux. Les arguments doivent être évalués à des valeurs logiques.

Exemple :

or(exact("c haîne", "chaîne"),exact("c haîne", "chaîne")) = 0

or(exact("chaîne", "chaîne"),exact("c haîne", "chaîne")) = 1

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

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

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

Les fonctions mathématiques

Remarque : Les paramètres régionaux du système Windows peuvent affecter la sortie de certaines formules.

Abs(«data») Retourne la valeur absolue (positive) de la donnée.

Exemples:

Abs(-5) = 5

Abs(5) = 5

base10tobaseX(«chaîne_1»,«chaîne_2») convertit chaîne_2 de base 10 en base chaîne_1

Exemples

Si le champ nommé Base 16 contient la chaîne "0123456789ABCDEF"

BASE10TOBASEX(Base16, 12) donne C

BASE10TOBASEX(Base16,10) donne A

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

Note: Cette formule n'accepte pas les valeurs décimales négatives pour «chaîne_2».

baseXtobase10(«chaîne_1»,«chaîne_2») Convertit chaîne_2 de base chaîne_1 en base 10

Exemples

Si le champ nommé Base 16 contient la chaîne "0123456789ABCDEF"

BASEXTOBASE10(Base16, "E") donne 14
BASEXTOBASE10(Base16,10) donne A
BASEXTOBASE10("012345","9") donne 13

ceil(«data») Arrondit la donnée au nombre entier suivant (supérieur).

Exemple :
Ceil(3.234) = 4
Ceil(7.328) = 8

Decimals(«data1», «data2») Utilise data2 en tant que nombre de décimales à prendre en compte à la data1.

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

eval_add(«chaîne»,«chaîne»): Retourne la somme des paramètres

Exemple :
eval_add(5,5)=10

eval_div(«chaîne»,«chaîne») : Retourne la division des paramètres.

Exemple :
eval_div(20,2)=10

eval_mult(«chaîne»,«chaîne»): Retourne la multiplication des paramètres.

Exemple :
eval_mult(5,2)=10

eval_sub(«chaîne»,«chaîne»): Retourne la soustraction des paramètres.

Exemple :
eval_sub(20,10) = 10.

Floor(«data») Arrondit la donnée au nombre entier inférieur.

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

hex(«val_1»,«val_2») convertit le nombre décimal **val_1** en format hexadécimal avec largeur totale **val_2.**

Note: Cette formule n'accepte pas les valeurs décimales négatives pour "val_1'".

Exemple:
hex(2,8) = 00000002

int («valeur») retourne le plus grand des entiers inférieur ou égal à l'argument valeur.

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

max(«data1», «data2») retourne la valeur la plus grande.

Exemple:
Max(5, 12) = 12

min(«data1», «data2») Retourne la valeur la plus petite.

Exemple :
Min(5, 12) = 5

mod («val_1», «val_2») retourne le reste de la division de l'argument val_1 par l'argument val_2. Le résultat est du même signe que le diviseur.

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

quotient («val_1», «val_2») retourne l'entier résultant de la division de l'argument val_1 par l'argument val_2.

Exemple :
quotient(10,2) = 5

round («val_1», «val_2») Retourne l'argument val_1 arrondi au nombre de chiffres indiqué par l'argument val_2.

Si val_2 est supérieur à 0, val_1 est arrondi au nombre de décimales indiqué.

Si val_2 est égal à 0, val_1 est arrondi au nombre entier le plus proche.

Si val_2 est inférieur à 0, val_1 est arrondi à gauche de la virgule.

Exemples :
round (4.25,1) = 4.3
round (1.449, 1) = 1.4
round (42.6,-1) = 40

Les fonctions sur les textes

Remarque : Les paramètres régionaux du système Windows peuvent affecter la sortie de certaines formules.

Une chaîne de caractères est assimilable à un tableau dont chaque case contient un caractère. Elle est définie par sa longueur (nombre totale de caractères de la chaîne, les espaces compris). La position d'un caractère dans une chaîne correspond à sa place dans le tableau; Par exemple, le premier caractère est à la position 1, la position 3 correspond au troisième caractère de la chaîne.

cyclebasex ( ) permet le comptage dans n'importe quel système de comptage d'une base de données. Le système de numérotation doit être défini à l'intérieur de l'expression liée. La valeur de départ, la valeur de chaque incrément et le nombre de copies de chaque numéro doivent également êtres spécifiés. Toutes ces valeurs peuvent être liées à d'autres champs de l'étiquette, mais les noms des champs ne doivent pas apparaître entre guillemets.

Exemples :

Si un champ nommé Base16 contient la chaîne de caractères 0123456789ABCDEF, alors…

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 () crée une suite personnalisée de caractères pour un cycle complet.

Exemples :

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 ( ) permet de paramétrer votre propre séquence de nombres, en opposition à la progression normale des chiffres ou des lettres (0,1,2… ou A,B,C…).

Exemples :

cyclenumber(1,3) produira des étiquettes séquencées 1 2 3 1 2 3 1 2 3...

cyclenumber(1,3,1,2) produira des étiquettes séquencées 1 1 2 2 3 3 1 1 2 2 3 3 1 1...

cyclestring ( ) permet de créer un groupe de mots ou caractères passant par un cycle complet comme un champ d'incrémentation. La chaîne entière doit être encadrée au moyen de guillemets (« ») et chaque mot ou groupe de caractères doit être séparé des autres par un point virgule (;)

Exemples :

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

L'exemple suivant est pour l'impression d'étiquettes de rayon utilisant toutes les lettres de l'alphabet sauf O et 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 («chaîne_1», «chaîne_2») retourne TRUE (Vrai) si les deux chaînes sont identiques, FALSE (Faux) sinon .

Exemples :

exact("logiciel","logiciel") = 1

exact("lgiciel","logiciel") = 0

Extract(«*chaîne*», «sep», «pos») retourne la sous-chaîne de la chaîne de caractère «string» dont la position est spécifiée par «pos» contenant la ou les données séparées par la chaîne «sep».

Exemple:

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

find («chaîne», «clé», «départ») retourne la position de la première occurrence de l'argument *clé* dans l'argument *chaîne*. La recherche, dans l'argument *chaîne*, commence à partir de la position donnée par l'argument *départ* (*départ* >= 1). La fonction renvoie zéro si aucune occurrence de l'argument *clé* n'est trouvée. La différence est faite entre majuscules et minuscules.

Exemples :

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

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

left («chaîne», «nombre») retourne la chaîne de caractères extraite de l'argument *chaîne*. Cette chaîne commence à la position 1 de l'argument *chaîne* et a une longueur égale à l'argument *nombre*.

Exemples :

left("Peter McPeepert",1) = P

left("Peter McPeepert ",5) = Peter

len («chaîne») retourne la longueur de l'argument *chaîne*. Les espaces sont comptés comme étant des caractères.

Exemples :

len("Paris, New York") = 15

len("") = 0

len(" ") = 1

lower («chaîne») convertit toutes les lettres majuscules d'une chaîne de texte en lettres minuscules..

Exemple :

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

LTrim*(«string»)*: Permet de supprimer automatiquement les espaces à gauche de la donnée.

Exemple:

LTrim(" No."): No

mid («chaîne», «départ», «nombre») retourne la chaîne de caractères extraite de l'argument *chaîne*. Cette chaîne commence à la position correspondant à la valeur de l'argument *départ* (*départ* >=1) et a une longueur égale à l'argument *nombre*.

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

Output («variable») : Retourne la valeur formatée de la variable.

Exemple :
le Compteur0 à le préfixe "numéro :". La formule "le compteur0à la valeur " &compteur0& " et la valeur formatée de " output(compter0) à pour résultat : le Compteur a la valeur : 1 et la valeur formatée de numéro : 001

Remarque :Le formatage de la variable se définit à partir de l'onglet **Sortie**.

pad («chaîne»,«longueur»,«car») rajoute des caractères du coté gauche du champ pour donner à l'entrée entière une longueur prédéfinie. N'importe quel caractère peut être sélectionné comme caractère de remplissage.

Exemples :
Si un champ nommé GREETING affiche une valeur HELLO, alors…
pad(GREETING,8,0) = 000HELLO
pad(5,3,0) = 005
pad("Nine",6,"a") = aaNine

replace («chaîne», «départ», «nb_car», «nvl_chaîne») retourne l'argument *chaîne* transformé. Un nombre (égal à l'argument *nb_car*) de caractères depuis la position définie par l'argument *départ* a été remplacé par l'argument *nvl_chaîne*.

Exemple :
replace("Paris, New York",8,8,"Singapour") = Paris, Singapour

ReplaceString («chaîne», «ancienne_chaîne», «nouvelle_chaîne») remplace toutes les occurrences de «ancienne_chaîne» dans la chaîne «chaîne» avec «nouvelle_chaîne».

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

rept («chaîne», «nombre») retourne la chaîne de caractères où l'argument *chaîne* est répété un nombre de fois correspondant à l'argument *nb_car*.

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

right («chaîne», «nombre») retourne la chaîne de caractères composée des derniers caractères de l'argument *chaîne* et ayant une longueur égale à l'argument *nb_car*.

Exemple :
right("Bon de commande",8) = commande

RTrim («string»): Permet de supprimer automatiquement les espaces à droite de la donnée.

Exemple:
RTrim("Part ") :Part

search («chaîne», «clé», «départ») Retourne la position de la première occurrence de l'argument *clé* dans l'argument *chaîne* . La recherche commence à partir de la position définie par l'argument *départ* (*départ* >= 1). La fonction renvoie zéro si aucune occurrence de l'argument *clé* n'est trouvée.

Exemples :
search("Bon de commande","commande,1) = 8
search("Bon de commande","c",1) = 8

Sort(«liste», «séparateur», «direction»): Retourne une liste triée. Le tri est effectué par comparaison de chaîne. La liste est un argument texte qui contient plusieurs éléments séparés par une virgule ou un point-virgule. La direction est un chiffre optionnel qui spécifie l'ordre de tri (0 = ascendant, 1 = descendant). L'ordre de tri est ascendant par défaut si la direction n'est pas spécifiée.

Exemples:
Sort(“3;5;6;1”, “;”) = “1;3;5;6”
Sort("un;deux;trois", ";", 1) = “un;trois;deux”

StrAfter(«data»,«start after», «length»): Retourne une chaîne dont la longueur correspond à la valeur donnée de «length» juste après le caractère spécifié dans «start after».

Exemples:
StrAfter("1234-5678", '-', 3)= Takes the 3 characters after the dash (567)
StrAfter("1234-5678", '-')= Takes all characters after the dash (5678)

StrBefore(«data»,«start before», «length»): retourne une chaîne dont la longueur correspond à la valeur donnée de «length» juste avant le caractère spécifié dans «start before ».

Exemples:

StrBefore("1234-5678", '-', 2)= Takes the 2 characters immediately before the dash (34)

StrBefore("1234-5678", '-')= Takes all characters before the dash (1234)

SuppressBlankRows («string»): Retourne une chaîne avec les lignes vides enlevées. Cela permet de créer un objet, d'y placer les champs que vous souhaitez et de supprimer ceux qui sont blancs.

Exemple:

SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2}) (Var0, Var1 et Var2 sont des variables, char(10) est le symbole "\n" qui signifie une nouvelle ligne)

Variables/Valeurs	Formule	Résultat
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», «séparateur»): Supprime les éléments vides d'une liste. La liste est un argument texte contenant plusieurs éléments séparés par une virgule ou point-virgule.

Exemple:

SuppressEmpty(“3;;;1”, “;”) = “3;1”

trim(«chaîne») retourne l'argument *chaîne* transformé. Tous les blancs rencontrés au début ou à la fin de la chaîne sont supprimés. Le nombre de blancs compris entre deux mots est réduit à un.

Exemple :

trim(" Bon de commande ") = Bon de commande

trimall(«chaîne») retourne l'argument *chaîne* transformé. Tous les espaces rencontrés sont supprimés.

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

Unique(«liste», «séparateur», «comparateur»): Supprime les éléments dupliqués d'une liste d'éléments. La liste est un argument texte contenant plusieurs éléments séparés par une virgule ou point-virgule. Le paramètre optionnel comparateur indique si la comparaison fait la distinction entre majuscules et minuscules, 0 = distinction majuscules/minuscules, 1 = pas de distinction. La comparaison fait la distinction majuscules/minuscules par défaut si le comparateur n'est pas spécifié.

Exemples:
Unique(“3;5;3;1”, “;”) = “3;5;1”
Unique("un;UN;deux", ";", 1) = “un;deux”

upper («chaîne») retourne l'argument *chaîne* converti en majuscules.

Exemple :
upper("Bon de commande") = BON DE COMMANDE

ztrim («valeur») supprime tous les zéros apparaissant à gauche du premier caractère numérique. Cette fonction est valide pour les valeurs numériques seulement.

Exemple :
Si un champ nommé WEIGHT affiche une valeur de 000200, alors…
ztrim(weight) = 200

Définir les propriétés d'une source de données Formule

Commande: **Sources de données > Formule > Propriétés...**

1. Tapez directement la formule dans la zone d'édition.
   -ou-
   Sélectionnez à l'aide de la souris les éléments de votre choix, puis cliquez sur **Insérer**.
2. Validez.

Conseil: Vous pouvez insérer l'élément en double cliquant dessus.

Attention : si dans une formule, vous utilisez une variable dont le nom contient un des caractères suivants **&+-*/<>=^%,!\"**, il faut entourer le nom par {}.

Remarque :Un aperçu dynamique affiche le résultat de la formule, incluant le formatage défini dans l'ongle **Sortie**. En cas d'erreur, l'aperçu est affiché en rouge. Si la valeur obtenue est tronquée, vous devez modifier la longueur maximale définie dans l'onglet **Sortie**.

Atelier pratique : Calcul de Modulo spécifique

Votre logiciel intègre des fonctions de calcul de modulo pour l'intégration d'un caractère de contrôle dans les données code-à-barres.

Cependant, il peut arriver que le calcul du caractère de contrôle soit spécifique et que par conséquent, les fonctions intégrées au logiciel ne répondent pas à vos besoins.

Il vous revient alors de créer vous même la fonction. Nous allons voir dans cette séquence qu'il est tout à fait possible de réaliser n'importe quel calcul de caractère de contrôle grâce aux formules.

L'exemple présenté ici illustre le calcul d'un caractère de contrôle pour un code-à-barres 2/5 interleaved.

Méthode de calcul pour un caractère de contrôle

La méthode consiste à multiplier le premier caractère de la donnée par 1, le second par 2, le troisième par 1, etc.

Ouvrez l'étiquette intitulée CHECK01.TPL située dans le dossier InstallDir\Samples\Forms\Tutorial. .

Le calcul du caractère de contrôle va s'effectuer sur la donnée LOT_NUMBER (26053).

Pour calculer le poids

1. Cliquez sur la branche **Formule** de la vue **Sources de données** et ajoutez une **Formule**.
2. Saisissez la formule suivante :
   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)

Le résultat est 2120103 ; en effet, nous avons :
2*1 = 2
6*2 = 12
0*1 = 0
5*2 = 10
3*1 = 3

La concaténation des résultats donne 2120103.

3. Nommez la formule WEIGHTED.

Pour additionner le résultat du calcul de poids :

L'étape suivante consiste à additionner les chiffres résultant de la formule précédente.

1. Créez une seconde Formule que vous intitulerez SUM.
2. A l'intérieur de la zone de saisie, tapez l'expression suivante :
   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)

   Le résultat est 9 ; en effet, nous avons : 2+1+2+0+1+0+3= 9.

Pour calculer le caractère de contrôle :

C'est à l'aide du résultat précédent que nous allons calculer la valeur du caractère de contrôle.

1. Créez une troisième Formule que vous intitulerez CHECK_DIGIT.
2. A l'intérieur de la zone de saisie, tapez l'expression suivante :
   if ((SUM % 10) > 0,10 - SUM % 10,0)

   Le résultat est 1 ; en effet, nous avons :
   SUM % 10 = 9 (% = modulo).
   9 étant supérieur à 0, nous effectuons la soustraction : 10- 9 = 1.

Remarque: Si le résultat avait été 0, la valeur du caractère de contrôle aurait été 0.

Pour calculer la donnée à encoder :

La création du code-à-barres doit inclure la donnée à encoder, c'est-à-dire la valeur de la variable LOT_NUMBER concaténée à la valeur du caractère de contrôle (CHECK_DIGIT).

1. Créez une quatrième Formule que vous intitulerez DATA.
2. A l'intérieur de la zone de saisie, tapez l'expression suivante :
   LOT_NUMBER & CHECK_DIGIT.

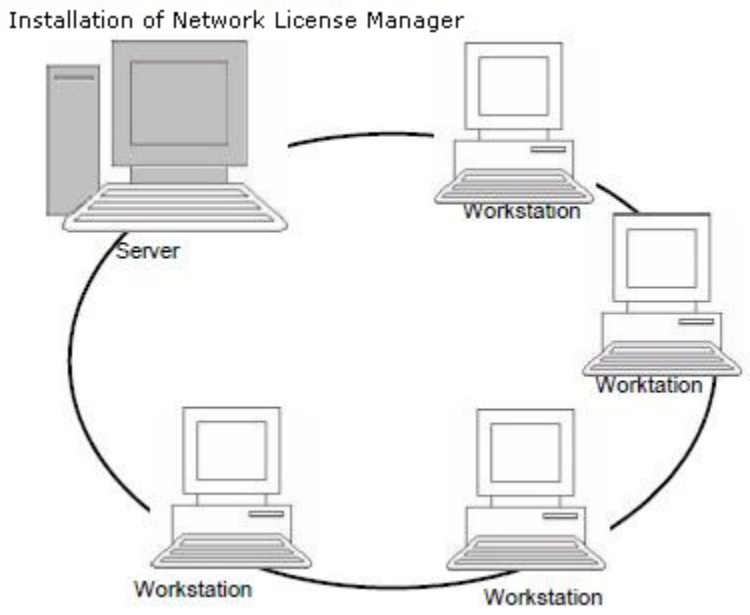
Pour créer le code-à-barres :

1. Cliquez sur l'outil Code-à-barres situé dans la barre d'outils de création, sélectionnez ensuite la formule DATA et glissez-déplacez-la dans l'étiquette.
2. Sélectionnez le code-à-barres et, à l'aide du menu contextuel, choisissez **Propriétés** de l'objet. La boîte de dialogue Code-à-barres apparaît.
3. Configurez le code-à-barres comme suit :
   Symbologie: 2/5 Interleaved
   Caractère de contrôle : Aucun

Installer la version Réseau

Description

Pour utiliser une version réseau / multi-utilisateurs de votre logiciel d'étiquetage, vous devez installer le Network and Users Utilities sur le serveur ou une station qui servira de serveur, puis installer le logiciel d'étiquetage sur chaque station.



Procédure d'installation

Configuration

Avant de procéder à l'installation du logiciel, l'administrateur réseau devra définir l'organisation du réseau du groupe d'utilisateurs, à savoir :

- Le poste serveur de licences sur lequel le **Network License Manager** et la clé de protection seront installés.
- Les stations, c'est-à-dire les postes clients qui utiliseront le logiciel d'étiquetage

Présentation du Network License Manager

Le **Network License Manager** vous permet d'utiliser la configuration réseau de votre logiciel d'étiquetage. Le **Network License Manager** contient :

- Le **Network License Manager** (License Service).
- **Network Settings Wizard**: un assistant qui vous aide à définir la configuration réseau.

Installation du Network License Manager sur le serveur

Avant d'installer le logiciel d'étiquetage sur toutes les stations qui l'utiliseront, vous devez installer le **Network License Manager** sur le serveur pour configurer le réseau.

1. Introduisez le DVD d'installation dans le lecteur approprié.
   La fenêtre d'Installation s'affiche.

   Si le DVD d'installation ne s'exécute pas automatiquement:
   Ouvrez l'Explorateur Windows, puis tapez la lettre du lecteur DVD suivie du fichier index.hat.
2. Sélectionnez **Network License Manager** qui inclut le **License Service**, puis cliquez sur le bouton **Installer**.
3. Suivez les instructions portées à l'écran.
4. Si vous souhaitez définir les réglages de votre configuration réseau, lancez le **Network Settings Wizard**, sur le serveur. Par défaut, si vous ne modifiez pas la configuration, les réglages seront spécifiques à chaque station.

Configuration

Tous les outils nécessaires à la configuration de la version réseau sont disponibles depuis la barre d'outils **Network License Manager**, qui sont accessibles depuis la barre des taches de Windows (Systray).

Depuis la barre des taches de Windows, double cliquez sur l'icône pour afficher la barre d'outils **Network toolbar**.

Le **Network Settings Wizard** vous aide à définir les réglages de votre version réseau.

1. Pour lancer l'Assistant de configuration réseau, cliquez sur l'icone depuis la barre d'outils Network.
2. Sélectionnez un mode de réglages : generic (générique), by user (par utilisateur) ou by station (par station).

   - **Generic**: tous les utilisateurs utiliseront les mêmes réglages sur toutes les stations. (user.ini).
   - **By user**: chaque utilisateur pourra accéder à ses propres réglages sur n'importe quelle station. (user name.ini).
   - **By station**: chaque station a ses propres réglages (station.ini).

3. Indiquez l'endroit où vous souhaitez enregistrer ces réglages. Si vous souhaitez partager ces réglages entre les différentes stations, indiquez un chemin réseau accessible par toutes les stations (Exemple: TKDongle)..
4. Indiquez l'endroit où vous souhaitez enregistrer les données partagées (sources de données, listes, journal d'impression....). Assurez-vous que tous les utilisateurs ont bien accès à ces dossiers.

Démarrage du gestionnaire de Licences

Avant d'installer le logiciel d'étiquetage sur toutes les stations, vous devez vous assurer que le gestionnaire de licences est ouvert.

Le service de licence est installé comme un service Windows. Ce service, appelé **TkxWebLicenseServer**, est activé automatiquement au démarrage du serveur.

Pour démarrer le Contrôleur du service de licences

Cliquez sur l'icône depuis la barre d'outils **Network License Manager**

- ou -

Double-cliquez sur le fichier TkxWebLicenseServerController.exe depuis le dossier d'installation

- ou -

Faites un clic droit sur l'icône **Network License Manager** dans la barre des tâches de Windows et sélectionnez **Contrôleur de service de licence.**

Installation du logiciel sur les stations

Le logiciel d'étiquetage doit être installé sur toutes les stations qui l'utiliseront.

Pour installer le logiciel sur une station

1. Introduisez le DVD d'installation dans le lecteur approprié.
   La fenêtre d'Installation s'affiche.

   Si le DVD ne s'exécute pas automatiquement :
   Ouvrez l'Explorateur Windows, puis tapez la lettre du lecteur DVD suivie du fichier index.hat.
2. Sélectionnez le produit à installer, puis cliquez sur le bouton **Installer** et suivez les instructions portées à l'écran.
3. Démarrez le logiciel d'étiquetage. Le **License Manager** est affiché. cliquez **Essaayer** pour démarrer le logiciel. Depuis le menu **Outils**, sélectionnez **Administration réseau**.
   - ou -
   Depuis le menu Démarrer de Windows, sélectionnez le raccourci **Administration réseau** dans le groupe du logiciel d'étiquetage.
4. Activez **Utilisation de la licence réseau**.
5. Sélectionnez le **type de licence de réseau**.
   - La licence par **Dossier partagé** utilise la fonction de partage de fichiers de Windows pour communiquer entre le logiciel et le gestionnaire de licences.
   - La licence **Web** utilise la communication http/https entre le logiciel et le gestionnaire de licences.
6. Indiquez le **port du serveur** si vous avez sélectionné le type de **licence Web**.
7. Cliquez sur **Modifier** pour sélectionner le serveur où sont installés le gestionnaire de licences et la clé de protection.

- ou -

Cliquez sur **Parcourir** pour lancer une recherche automatique du serveur où est installé le gestionnaire de licences

Si la configuration réseau a été effectuée au préalable, un message s'affiche vous demandant si vous souhaitez ou non utiliser la configuration réseau actuelle.

8. Si vous souhaitez modifier ou configurer les réglages réseau, cliquez sur le bouton **Assistant de configuration réseau**.
9. Cliquez sur **Valider**.
10. Redémarrez le programme.

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