

Report Builder

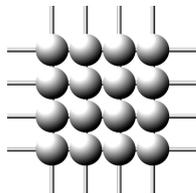


Table des matières

INTRODUCTION	4
PRESENTATION.....	5
<i>Description de l'écran</i>	<i>5</i>
<i>Description des icônes</i>	<i>7</i>
<i>Les différentes tables de BOB</i>	<i>8</i>
CREATION DU RAPPORT	9
<i>Création du l'en-tête du rapport</i>	<i>9</i>
♦ Coordonnées de la société	9
♦ Logos	11
♦ Coordonnées du client	12
♦ Références de la facture.....	13
♦ Intitulés des colonnes	13
♦ Création de groupes	14
<i>Création du corps du rapport</i>	<i>16</i>
♦ Le sous-rapport.....	16
♦ Les lignes de la facture	17
<i>Création du bas de page</i>	<i>18</i>
♦ Création des régions	18
♦ Conditions d'impression.....	19
<i>La facture.....</i>	<i>23</i>
AUTRES FONCTIONALITES	24
<i>Association d'une procédure à une zone de type DbText.....</i>	<i>24</i>
♦ Procédures	25
♦ Erreurs de procédure.....	25
<i>Réalisation de calculs de données.....</i>	<i>28</i>
<i>Association d'une procédure à une bande</i>	<i>29</i>
<i>Les variables.....</i>	<i>30</i>
<i>Variable globale.....</i>	<i>31</i>
♦ Détermination d'une variable de type global	31
<i>Les zones mémos.....</i>	<i>35</i>
<i>Procédure pour afficher un blanc à la place de zéro</i>	<i>37</i>
<i>Afficher la devise de base à la place d'un blanc</i>	<i>39</i>
<i>Résultat</i>	<i>40</i>
TRUCS & ASTUCES	41
<i>Afficher le solde actuel dans le layout d'un rappel ou d'une facture</i>	<i>42</i>
<i>Afficher un montant en lettre.....</i>	<i>43</i>
<i>Fonction 'Vide si zero'.....</i>	<i>44</i>
<i>Réalisation d'opérations sur certains champs.....</i>	<i>45</i>
<i>Affichage de l'en-tête sur la première page uniquement</i>	<i>46</i>
<i>Affichage de description de tables</i>	<i>47</i>
<i>Exportation et Importation des dessins de document</i>	<i>48</i>
<i>Les arrondis</i>	<i>49</i>

◆ Prix unitaire	50
◆ Quantité	50
◆ Prix net	50
<i>Appliquer la mise en pages correcte qui a été définie pour le texte de regroupement</i>	51
◆ Pas de texte de regroupement	52
◆ Texte de regroupement et ligne de commentaire	53
◆ Paramètres de mise en page	53
<i>Formater le numéro de TVA</i>	54
<i>Ordre des variables</i>	55
<i>Attribuer des décimales à une variable</i>	58
<i>Griser une ligne sur deux dans le détail de la facture</i>	59
<i>Indiquer 'copie' à partir de l'impression du 2^e exemplaire</i>	60
<i>Problème d'impression d'un layout sur du papier à en-tête de type continu :</i>	61
<i>Problèmes d'impression liés à windows</i>	62
<i>Afficher le solde client</i>	63
<i>Afficher le libellé d'un compte general</i>	64
<i>Afficher le libellé d'un code appartenant à une table</i>	65
<i>Afficher la quantité restant à livrer</i>	66
<i>Afficher la valeur d'un champ de l'entête du document d'origine</i>	67
<i>Transformer un nombre de minutes en heure</i>	68
<i>Transformer une durée en heure en nombre de minutes</i>	69
<i>Formater un numéro de compte bancaire</i>	70
<i>Transformer les quantités en heures</i>	71
<i>Afficher la valeur d'un champ d'un fichier</i>	72
<i>Enlever le code pays 'be' du code postal</i>	73
<i>Tester l'existence d'un fichier</i>	74
<i>Formater un montant</i>	75
<i>Afficher l'année scolaire (ex : 2004-2005)</i>	76
<i>Afficher la référence article chez le client</i>	77
<i>Afficher la référence article chez le fournisseur</i>	78
<i>Afficher des informations de l'entête de la commande client sur la commande fournisseur (pour commandes directes)</i>	79
<i>Afficher des informations du détail de la commande client sur la commande fournisseur (pour commandes directes)</i>	80
<i>Afficher le mois en lettres</i>	81
<i>Afficher le libellé d'un contrat</i>	83
<i>Afficher le prix d'un tarif</i>	84
<i>Afficher le libellé d'un code standard (stdcode)</i>	85
<i>Afficher le libellé d'une case de la déclaration TVA</i>	86
<i>Ajouter des zéros à gauche d'un nombre</i>	87
<i>Afficher le commentaire de l'article sur plusieurs lignes</i>	88
RAPPORTS PERSONNALISABLES	89
<i>Exemple de rapport personnalisable : la balance des comptes généraux</i>	91

INTRODUCTION

Report Builder permet de dessiner des rapports tels que les factures ou les rappels de paiement. Les données de BOB ainsi que des fonctions utiles ont été implémentées directement afin que leur conception soit la plus évidente possible.

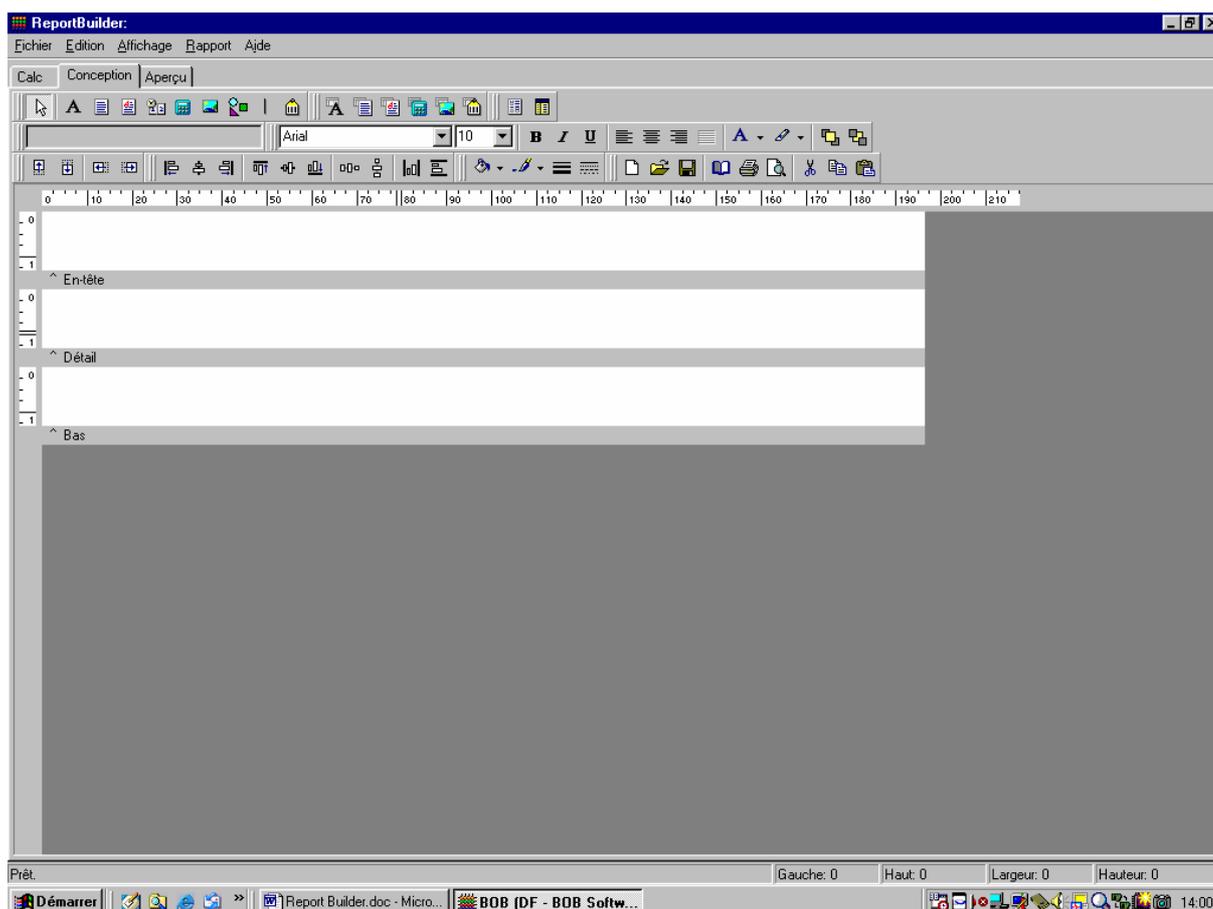
BOB met à votre disposition une série de documents, déjà paramétrés, de chaque type (rappels et factures). Ceux-ci pourront être modifiés ou améliorés selon les besoins de chacun.

Dans ce manuel d'initiation à la conception de rapport, nous allons débiter sur base d'un document vierge pour arriver, à la fin, à un résultat correspondant à une facture typique. Le principe sera, bien évidemment, le même pour la création des autres types de document.

PRESENTATION

Le Report Designer, écran présenté ci-dessous, correspond à l'écran principal pour la conception d'un rapport.

DESCRIPTION DE L'ECRAN



L'écran est composé de trois onglets :

Calc : permet de sélectionner et de modifier les données nécessaires à la création d'un rapport déterminé et de créer des procédures en langage Pascal.

Conception : cet écran est divisé en deux parties. La première se compose des barres d'outils, de palettes ou encore de règles. La deuxième est l'aire de création du rapport. C'est dans cette seconde partie que les bandes ainsi que les différents composants du rapport seront placés. Trois bandes y sont présentes par défaut : En-tête, Détail et Bas.

- **En-tête** : toutes les données insérées dans cette bande apparaîtront dans le haut de chaque page du rapport (ex : coordonnées du client, ...)
- **Détail** : contient les données à afficher dans le corps du rapport (ex : liste des articles et prix d'une facture, ...)

- **Bas** : toutes les données insérées dans cette bande apparaîtront dans le bas de chaque page du rapport.

Remarque : il existe d'autres bandes, telles que *Titre* ou *Sommaire*. Leur utilité et mode de fonctionnement seront expliqués plus avant dans ce manuel.

Aperçu: permet de visualiser ou d'imprimer le rapport.

DESCRIPTION DES ICONES

Icône	Description
	Sélectionner l'objet : curseur de sélection.
	Texte : utilisé pour afficher du texte.
	Mémo : imprime plusieurs lignes de texte.
	Texte formaté: imprime du texte formaté.
	Variable système: utilisé pour afficher une variable (heure, n° page, date, ...)
	Variable: permet d'insérer une variable permettant de réaliser des procédures.
	Image: permet d'insérer une image de type bitmap, GIF ou JPEG.
	Forme: permet d'insérer une forme (rectangle, rond, ...).
	Ligne : permet d'insérer une ligne (horizontale, verticale, ...).
	CodeABarre: code barre (clic droit pour y associer un code).
	DB_Texte : permet d'insérer un champ issu d'une base de données.
	DBMémo: permet d'imprimer un champ mémo provenant d'une base de données.
	DB_TexteFormaté: permet d'imprimer du texte formaté provenant d'un champ mémo lui-même issu d'une base de données.
	DBCalculé: permet de réaliser de simples calculs de données (somme, soustraction, ...)
	DBImage: permet d'insérer une image stockée dans une base de données.
	DBCCodeABarre: convertit une donnée d'une base de données en code barre.
	Région: permet de grouper plusieurs composants.
	SousRapport: permet de créer des sous-rapports dans la bande Détail.
	Permet de déplacer d'un pixel un groupe de composants.
	Permet d'harmoniser la taille d'un groupe de plusieurs zones sélectionnées.
	Permet d'aligner un groupe de composants sélectionnés.
	Permet d'espacer de façon régulière un groupe de composants sélectionnés.
	Centre un groupe d'icônes sélectionnées.
	nouveau rapport
	Nom du pipeline de données, Champ de données: la première liste déroulante permet de choisir le nom de la base de données (table) à sélectionner et la seconde permet de choisir le nom du champ à sélectionner dans cette même table.

LES DIFFERENTES TABLES DE BOB

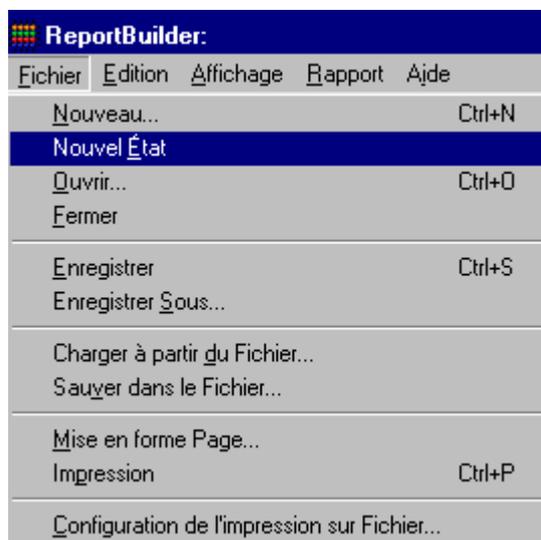
Comme mentionné ci-dessus, les tables sont stockées dans le pipeline de données. Elles sont accessibles lorsqu'une zone, permettant d'afficher une valeur de table, est sélectionnée.

Liste de tables disponibles :

- Db_Selection : contient les informations relatives au(x) document(s) à imprimer telles que le nombre de copies à imprimer ou encore le code du layout.
- Db_Article : contient les informations relatives aux articles (IART.DB).
- Db_Entete : contient les informations relatives aux en-têtes de documents (IHDDOC.DB).
- Db_Lignes : contient les informations relatives aux corps (bande Détail) des documents (IHISTO.DB).
- Db_Client : contient les informations relatives aux clients de facturation (COMPAN.DB).
- Db_Tiers_Adr_Liv : contient les informations relatives aux clients de livraison (COMPAN.DB).
- Db_Totaux : contient les informations relatives aux totaux des documents (IDOCT.DB).
- Db_TVA_Ventilation : contient les informations relatives à la ventilation TVA (ISPTVA).
- Db_Layout : contient les informations relatives aux dessins de documents (ILDEF.DB).
- Db_Detail_Tiers : contient les informations supplémentaires relatives aux clients de facturation (COMPAN.DB).
- Db_Config : contient les informations relatives au dossier (CONFIG.DB).
- Db_Prestataire : contient les informations relatives aux prestataires (USERS.DB).
- Db_Contrat : contient les informations relatives aux contrats clients (CUCCTC.DB)

CREATION DU RAPPORT

Pour créer un nouveau rapport, cliquer sur *Fichier | Nouvel Etat* ou sur l'icône .



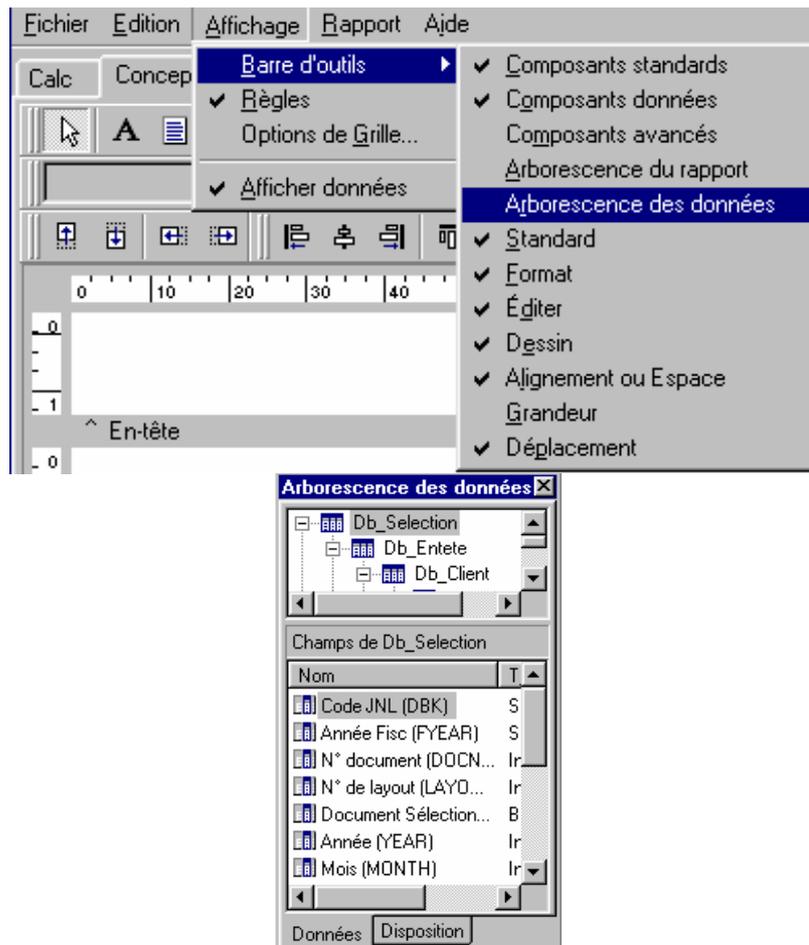
CREATION DU L'EN-TETE DU RAPPORT

Pour rappel, l'en-tête du rapport, correspond à la bande appelée '*En-tête*'. Les données insérées dans celle-ci seront affichées dans le haut de chaque page du rapport. Généralement, l'en-tête d'une facture contient les coordonnées de la société émettrice ainsi que celles du destinataire (client).

◆ **Coordonnées de la société.**

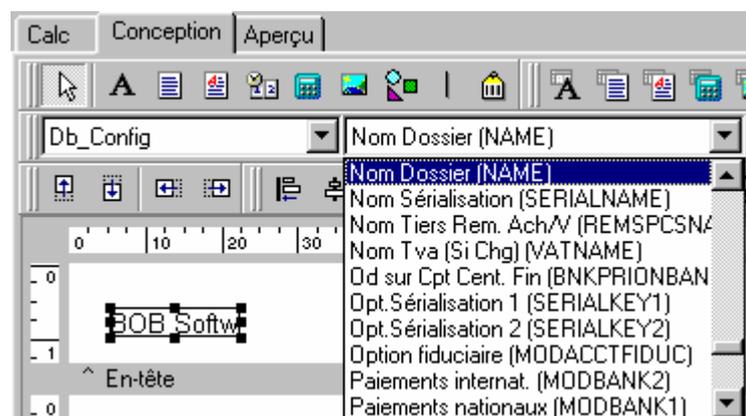
Pour insérer un champ d'une table, il y a deux possibilités :

- Afficher, via le menu *Affichage | Barre d'outils | Arborescence des données*, la liste des tables et des champs compris dans celles-ci, et faire glisser le champ voulu.

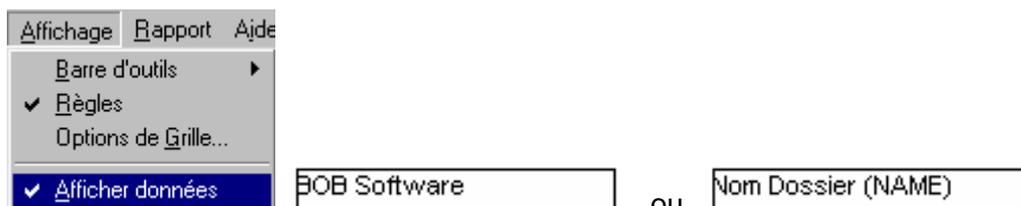


- Insérez une zone de type DBText  dans le coin supérieur gauche de la zone. Les listes déroulantes *Pipeline de données* et *Champs de données* apparaissent alors dans l'espace réservé aux barres d'outils.

Dans les listes déroulantes, sélectionnez la table DB_Config et le champ 'Nom Dossier (NAME)' afin d'afficher le nom de la société.



La zone *DB_Texte* affiche maintenant le contenu du champ 'NAME' du fichier DB_Config du dossier. Vous pouvez à loisir faire apparaître dans la zone DB_Texte, soit le contenu véritable du champ sélectionné, soit simplement le nom du champ. Il vous suffit pour ce faire de cocher ou non l'option *Afficher données* qui se trouve dans le menu *Affichage*



Vous procéderez de la même façon pour insérer les zones suivantes :

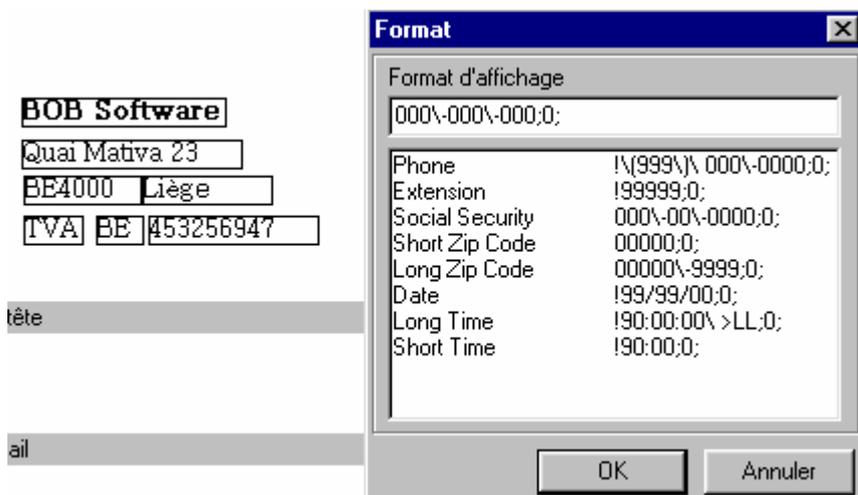
- Adr. dossier (ADDR1)
- C.P. (ZIPCODE)
- Localité (LOCALITY)
- Un texte (icône **A**) contenant « TVA »
- Code Pays TVA (VATCODE)
- N°TVA (VATNO)

Aperçu :

BOB Software
 Quai Mativa 23
 BE4000 Liège
 TVA BE 453-256-947

Vous avez la possibilité de formater le numéro de TVA grâce à l'option *Format d'affichage* du menu contextuel lié à la zone *Db_Texte*.

Associez le format suivant « 000\-000\-000 ;0 ; » à la zone N°TVA (VATNO) afin d'obtenir le résultat ci-dessous.

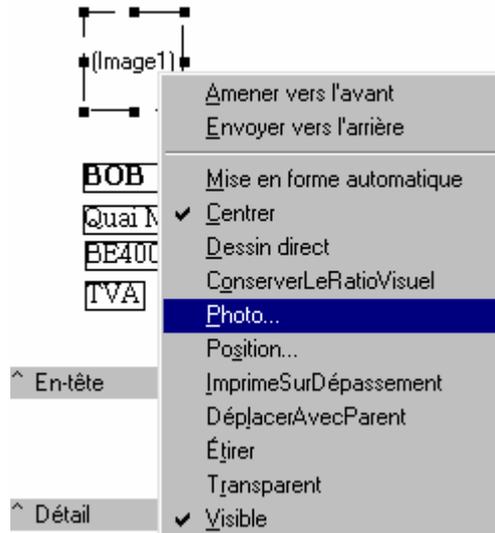


◆ **Logos**

Report Builder permet d'insérer un logo reprenant le sigle de la société grâce à une zone *Image* (icône )

Pour associer une image à une zone *Image*, il suffit de cliquer avec le bouton droit de la souris sur celle-ci afin d'afficher son menu contextuel et de sélectionner le point.

Une fenêtre de recherche s'ouvrira vous permettant de sélectionner le fichier contenant l'image.



Chacune des zones peut être formatée de façon indépendante ou simultanément grâce aux barres d'outils 'Format' et 'Dessin'.

Barre d'outils Format



Barre d'outils Dessin



◆ Coordonnées du client

Les coordonnées du client sont stockées dans la table *Db_Client*. L'ajout d'une nouvelle zone *Db_Texte* donnera la possibilité de sélectionner cette table.

Insérez les données, relatives au client, suivantes :

- Nom 1 (CNAME1)
- Adresse 1 (CADRESSE1)
- C.P. (CZIPCODE)
- Localité (CLOCALITY)
- Ref. TVA (CVATREF)
- N°TVA (CVATNO), à formater via l'option *Format d'affichage* du menu contextuel

Encadrez les coordonnées du client par un rectangle (icône ) que vous ramenez vers l'arrière (menu contextuel)

Aperçu :

RENAULT INDUSTRIE BELGIQUE SA
CHAUSSEE DE MONS 301
BE1070 BRUXELLES
BE 402.645.614

◆ Références de la facture

Les références de la facture se trouvent dans la table *Db_Entete*.

Insérez les données suivantes :

- Un libellé « FACTURE N° : »
- N° document (DOCNO)
- Un libellé « Date: »
- Date document (DOCDATE)
- Un libellé « Echéance : »
- Date éch. (DUEDATE)

Aperçu

Facture n° 20020001
Date: 10/01/2002
Echéance 09/02/2002

◆ Intitulés des colonnes

Les titres des colonnes doivent apparaître au-dessus du corps du document (i.e. de la bande Détail). Ils font donc partie de l'en-tête et s'affichent sur la première page.

Insérez les titres de colonnes suivants grâce à des zones de type *Texte* (A):

- Désignation
- P.U.
- Qté
- %
- Prix

Vous pouvez formater les titres, les encadrer ou les surligner.

Remarque : Vous pouvez visualiser le résultat à tout moment grâce à l'écran '*Aperçu*'.

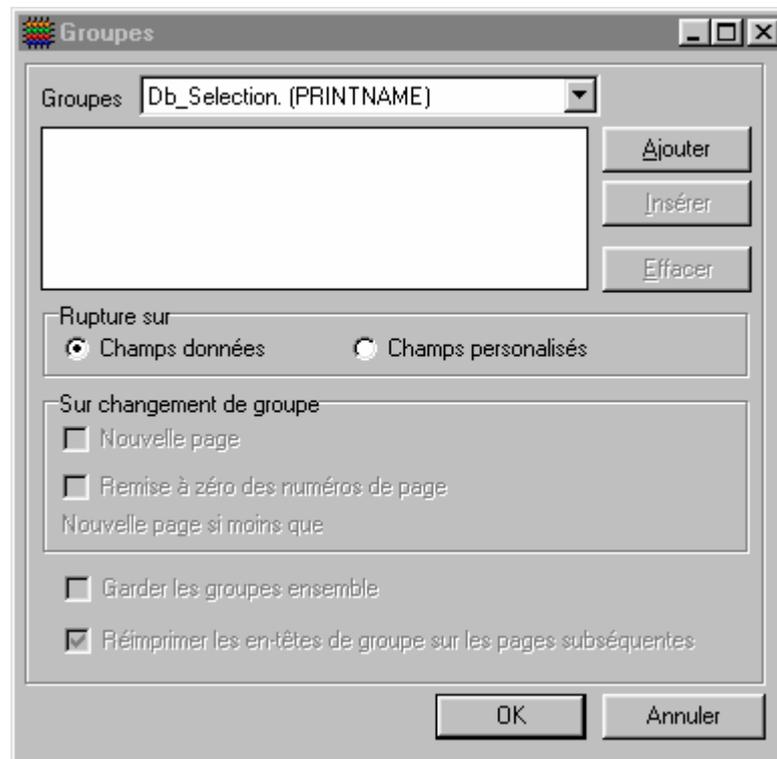
Aperçu

Désignation	P.U.	Qté	%	Prix
-------------	------	-----	---	------

◆ Création de groupes

Un groupe permet de trier des informations suivant un ou plusieurs critères.

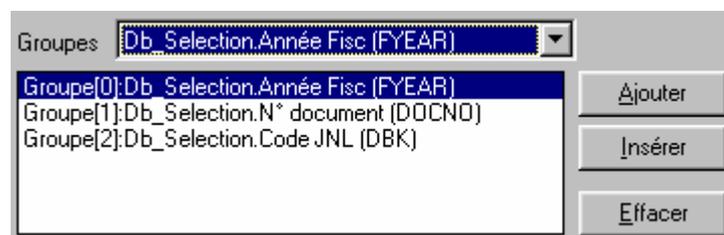
Pour créer un groupe, il suffit de sélectionner la bande dans laquelle le groupe doit être actif et de cliquer sur l'option *Groupes* qui se trouve dans le menu *Rapport*. L'écran ci-dessous apparaît :



La liste déroulante '*Groupes*' contient tous les critères de tri disponibles en fonction du type de bande sélectionnée. Il est possible de réaliser un ou plusieurs niveaux de tri. Ici, nous allons trier les factures sur base de trois critères différents et dans un ordre bien déterminé.

Pour ajouter un groupe, sélectionnez-en un dans la liste déroulante et cliquez sur le bouton *Ajouter*. Le groupe ainsi que l'ordre de sélection choisi s'ajoutent dans la liste.

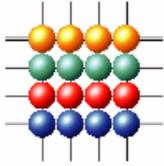
Créez les groupes suivants :



Une fois les groupes créés, cliquez sur le bouton *OK* pour confirmer l'opération. Suite à cette manipulation, les bandes *En-tête de groupe* et *Bas de groupe* apparaissent. Elles vont permettre de réaliser des opérations de regroupement.

Dans cet exemple, à l'impression, les factures seront triées d'abord par nom de journal, ensuite par année fiscale, et enfin par numéro de document.

Aperçu :



BOB Software
Quai Matisa 23
BE4000 Liège
TVA BE 453-256-947

RENAULT INDUSTRIE BELGIQUE SA
CHAUSSEE DE MONS 301
BE1070 BRUXELLES
TVA BE 402-645-614

Facture n° 20020001
Date: 10/01/2002
Echéance 09/02/2002

Désignation	P.U.	Qté	%	Prix
-------------	------	-----	---	------

^ En-tête
^ En-tête de groupe[0]: Code JNL (DBK)
^ En-tête de groupe[1]: Année Fisc (FYEAR)
^ En-tête de groupe[2]: N° document (DOCNO)



^ Détail
^ Bas du groupe[2]: N° document (DOCNO)
^ Bas du groupe[1]: Année Fisc (FYEAR)
^ Bas du groupe[0]: Code JNL (DBK)



CREATION DU CORPS DU RAPPORT

Le corps du document correspond à la bande *Détail*. Il peut être composé de un ou plusieurs sous-rapports (). Le sous-rapport vous permet de créer plusieurs niveaux de détails ou encore d'imprimer plusieurs rapports en un seul.

◆ Le sous-rapport

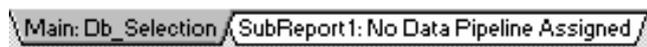
Cliquez une fois sur l'icône *SousRapport*  et insérez le sous-rapport dans la bande *Détail*.

La zone ci-dessous, correspondant au sous-rapport, apparaît :

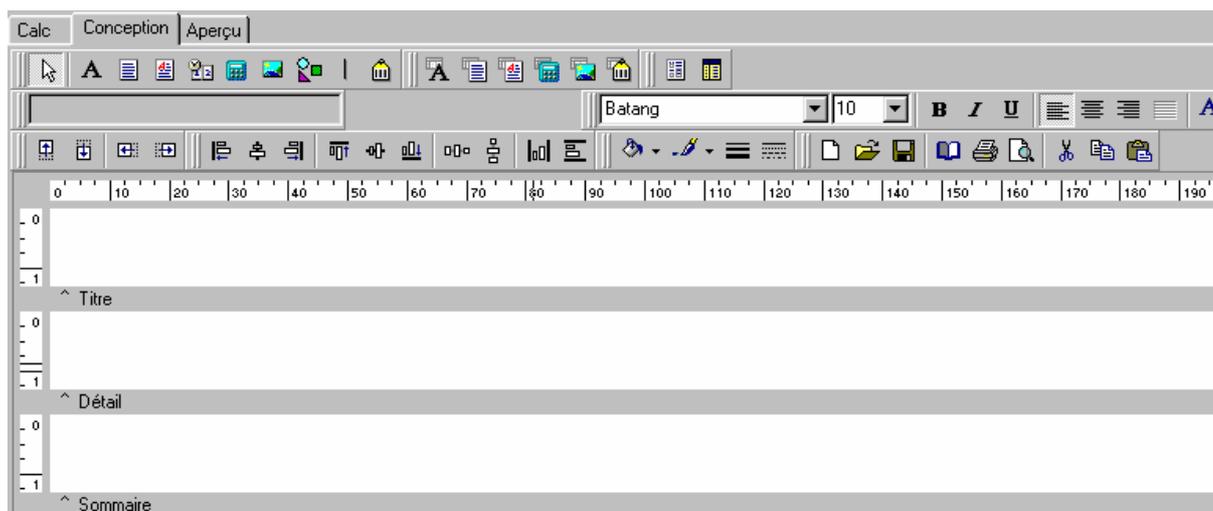


Un sous-rapport doit être associé un pipeline de données, c-à-d à une base de données. Le nom de celle-ci doit être indiqué dans la zone *Nom du pipeline de données*. Sélectionnez la table *DB_Lignes*, table reprenant le détail des factures. Une fois la table associée, son nom s'inscrit dans la zone *SousRapport*

Après avoir ajouté le sous-rapport, deux onglets supplémentaires apparaissent dans la partie inférieure de l'écran. Le premier *Principal: Db_Selection* donne accès à l'écran principal du rapport, le deuxième représente le *SousRapport* et donne accès à la modification de celui-ci. Il est possible de passer de l'un à l'autre à tout moment.



Ecran du sous rapport



L'écran du sous-rapport est le même que l'écran principal *Conception*. Il est composé de trois bandes : *Titre*, *Détail* et *Sommaire*.

◆ Les lignes de la facture

Insérez les zones relatives aux lignes de la facture dans la bande *Détail* du *Sous-rapport*. Ces informations se trouvent dans la table *Db_Lignes*. Elles se répéteront autant de fois qu'il y a de lignes d'articles dans la facture :

- Ligne de comment. (COMMENT)
- P.U. (PU)
- Qté livrée (QTYDELIV)
- Remise % (PRCDISC)
- Base TVA en dev. fac (CURRBASEAMN)

Disposez les zones de façon à ce quelles soient positionnées en dessous des titres de colonne qui leur correspond (colonnes disposées dans l'en-tête de l'écran principal *Conception*)

Clavier Microsoft	61,97	5	0	309,87
-------------------	-------	---	---	--------

CREATION DU BAS DE PAGE

Toutes les données insérées dans cette bande apparaîtront dans le bas de chaque page du rapport.

Le pied de page se compose généralement des différents totaux de la facture, des ventilations TVA ou encore d'informations complémentaires relatives à la société (numéros de compte bancaire, ...).

◆ Création des régions

La première région que nous allons créer contiendra les différents totaux et autres informations relatives à la TVA.

Insérez une zone Région  (Region1) dans la bande *Bas* et renommez la en 'RegSummary' via la zone

Différents champs (table DB_En-tête) doivent être insérer dans la région 'RegSummary'. Ces champs correspondent aux totaux de la facture:

- Un libellé « TOTAL HRTVA: »
- Tot Lignes en dev fa (CURRTOTLIGNE)
- Un libellé « - Ristourne »
- Mnt.remise en dev. f (CURRDISCAMN)
- Un libellé « + T.V.A. »
- TVA en dev. fact. (CURRVATAMN)
- Une ligne horizontale
- Un libellé « TOTAL A PAYER : »
- Mnt dû en dev. fact. (CURRPAYAMN)

Aperçu :

Total hTVA	2.181,46
Ristourne (-)	0,00
TVA	458,11
Total à payer	2.639,57

La région comprend également les zones correspondant à la ventilation de la TVA de la facture. Les champs à insérer se trouvent dans la table DB_TVA_Ventilation.

- Taux de Tva 1 (VATCMP_1)
- Un libellé « % »
- Taux de Tva 2 (VATCMP_2)
- Un libellé « % »
- Taux de Tva 3 (VATCMP_3)
- Un libellé « % »
- Mnt base 1 (BASEAMN_1)
- Mnt base 2 (BASEAMN_2)
- Mnt base 3 (BASEAMN_3)

- Mnt Tva 1 (VATAMN_1)
- Mnt Tva 1 (VATAMN_2)
- Mnt Tva 1 (VATAMN_3)

Remarque: vous pouvez créer un tableau via les outils *Forme*  et *Ligne* .

Aperçu

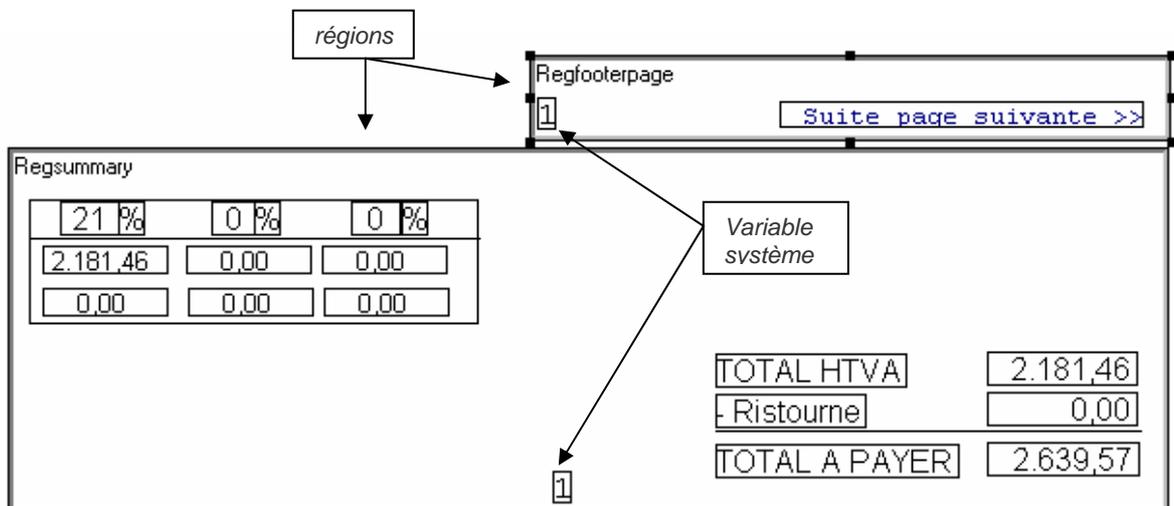
CMP 1) %	CMP 2) %	CMP 3) %
BASEAMN 1)	Mnt base 2 (B)	Mnt base 3 (B)
1 (VATAMN_1)	Mnt Tva 2 (VA	Mnt Tva 3 (VA

La région suivante contiendra les informations devant se trouver sur la (les) première(s) page(s) de la facture si celle-ci en contient plusieurs.

Insérez une zone région  dans la bande *Bas* et renommez la en '*RegFooterPage*' et insérez un une zone de texte: "Suite page suivante".

Reste à associer les régions aux différentes pages de la facture. En effet, les différents totaux ne doivent se trouver que sur la dernière page de la facture. Pour ce faire, insérez une variable système (icône ) dans les régions '*RegSummary*' et '*RegFooterPage*'. Associez, à ces variables, le type de variable « PageNo » (numéro de page).

Aperçu :



◆ Conditions d'impression

Dans l'état actuel des choses, les deux régions que nous venons de créer dans la bande *Bas*, apparaissent sur toutes les pages de la facture. Cependant, il est inutile de voire apparaître sur chaque page le total à payer ou encore le libellé « *Suite page suivante* ». En effet, le total général de la facture doit figurer uniquement sur la dernière page, et le libellé « *Suite page suivante* » sur chaque page, mise à part la

dernière, lorsque la facture est composée de plusieurs page. C'est pourquoi nous allons devoir associer des conditions d'impression à la bande *Bas*, afin que les régions s'impriment selon certains critères. Ces conditions doivent être réalisées en langage Pascal.

Description

Sélectionnez l'onglet *Calc* . Pour rappel, cette partie d'écran vous permet de sélectionner et de manipuler les données nécessaires pour un rapport déterminé ainsi que de créer des procédures en langage Pascal.

Cet écran montre une vue décomposée du rapport sélectionné : il présente toutes les bandes contenues dans le rapport ainsi que les objets contenus dans chaque bande du rapport.

Pour passer du rapport au sous-rapport quand vous vous situez dans l'onglet *Calc* sélectionnez soit l'onglet *principal* pour le rapport principal soit l'onglet *SousRapport* pour le sous-rapport. Pour rappel, ces deux onglets sont situés en bas de l'écran.

L'écran *Calc* peut montrer trois vues différentes. La vue peut être changée par un simple clic droit de la souris dans la zone *Objet du rapport*. Les choix sont les suivants :

- *Variables* : liste des variables regroupées par bande contenue dans le rapport ;
- *Événements* : liste des différents événements disponibles par bande ;
- *Module* : affiche les différents modules disponibles.

Sélectionnez la vue *Événements* et la bande *Bas*

Renommez les zones région1 et région2 en *RegSummary* et *RegFooterPage*

```

procédure FooterBeforeGenerate;
begin
  if Report.PageNo = Report.PageCount
  then begin
    Regsummary.visible := true;
    RegFooterPage.visible := false;
  end
  else begin
    Regsummary.visible := false;
    RegFooterPage.visible := true;
  end;
end;

```

Nom	Type	Ta...
[[[(PRINTNAME)	String	200
[[[(RFORMAT)	String	3
[[[(RTYPE)	String	3
[[[Année (YEAR)	Integer	
[[[Année Fisc (FYEAR)	String	5
[[[Code du layout (RREP)	String	120
[[[Code JNL (DBK)	String	4
[[[Document Sélectionné	Boolean	
[[[Imprimante (RTOPRIN)	Boolean	
[[[Mémo pour l'entête (MÉ)	Memo	1
[[[Mois (MONTH)	Integer	
[[[N° de layout (LAYOUT)	Integer	

Objets du rapport: arborescence des différents composants du rapport.

Événements pour Bas: liste des événements auxquels du code Pascal peut être associé et ce, selon le composant sélectionné dans le *Report Object*.

Zone de procédure: zone de saisie pour le code Pascal.

Boîte à outils : Cette partie est divisée en trois onglets . Son contenu varie en fonction de l'onglet sélectionné:

- *Données* : liste, propriétés et tailles des champs disponibles dans la table sélectionnée. Si l'onglet *Données* est sélectionné, la boîte à outils '*Données*' montre la liste des tables.
- *Objects* : liste des propriétés disponibles en fonction du composant sélectionné. Si l'onglet *Objects* est sélectionné, la boîte à outil montre la liste des différents composants faisant partie du rapport.
- *Langage* : si l'onglet *Langage* est sélectionné, la boîte à outils affiche la liste des fonctions, des opérateurs, des différents types de données. Les listes sont regroupées par catégorie et le détail est affiché lorsqu'on les sélectionne.

Quatre événements sont associés au Bas :

- *AfterGenerate* : signifie que l'événement associé sera exécuté après la génération du rapport.
- *AfterPrint* : signifie que l'événement associé sera exécuté après l'impression du rapport.
- *BeforeGenerate* : signifie que l'événement associé sera exécuté avant la génération du rapport.
- *BeforePrint* : signifie que l'événement associé sera exécuté avant l'impression du rapport.

Sélectionnez l'événement *BeforeGenerate* dans la zone et cliquez dans la zone de procédure. Le code suivant apparaît :

```
procedure FooterBeforeGenerate;  
begin  
  
end;
```

Tapez la procédure suivante:

```
Procedure FooterBeforeGenerate;  
Begin  
if Report.PageNo = Report.PageCount  
  then Begin  
    RegSummary.visible := True;  
    RegFooterPage.Visible := False;  
  end  
  else Begin  
    RegSummary.Visible := False;  
    RegFooterPage.Visible := True;  
  end;  
end;
```

Cette procédure affiche ou n'affiche pas les régions *RegSummary* et *RegFooterPage* suivant le numéro de la page.

Description de la procédure:

Si le numéro de la page du rapport est égal au nombre total de pages du rapport

Alors

La région « *RegSummary* » est visible

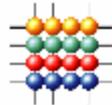
La région « *RegFooterPage* » n'est pas visible

Sinon

La région « RegSummary » n'est pas visible

La région « RegFooterPage » est visible

LA FACTURE



BOB Software
 Quai Malva 2
 BE4000 Liège
 TVA BE 453-296-94

RENAULT INDUSTRIE BELGIQUE SA
 CHAUSSEE DE MONS 301
 BE1070 BRUXELLES
 TVA: BE 402-645-614

Facture n° 20020001
 Date: 10/01/2001
 Echéance 09/02/2001

Désignation	P.U.	Qté	%	Prix
Clavier Microsoft	61,67	5	0	309,67
Souris Microsoft	32,23	5	0	161,13
BOB Software	1.115,52	1	0	1.115,52
Hp Deskjet 890 Cxd Couleur	297,47	2	0	594,94
Profitez de nos promotions de février sur les montres	0,00	0	0	0,00

21 %	0 %	0 %
2.181,46	0,00	0,00
468,11	0,00	0,00

Total HTVA 2.181,46
 Ristourne (-) 0,00
 TVA 468,11
Total à payer 2.639,57

AUTRES FONCTIONNALITES

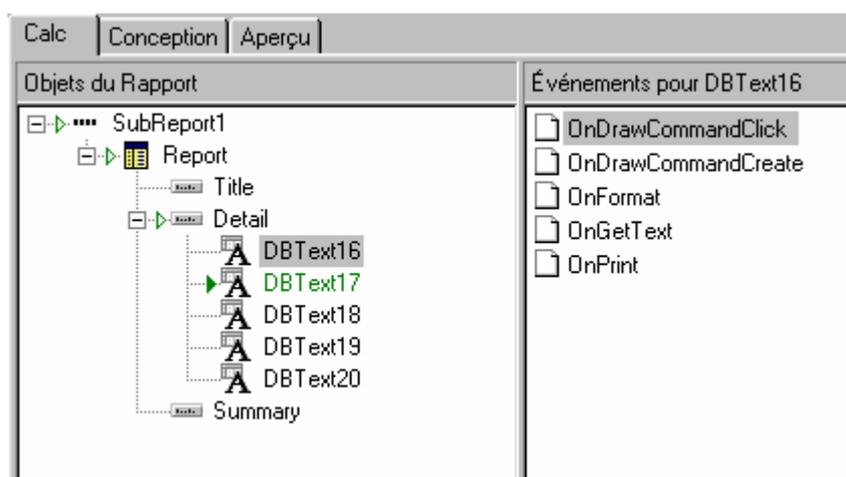
ASSOCIATION D'UNE PROCEDURE A UNE ZONE DE TYPE DBTEXT

En facturation, il est possible d'insérer une ligne de commentaire dans le détail de la facture. Lorsque c'est le cas, il est parfois ennuyeux de voir apparaître des zéros à la place du prix de l'article ou encore des quantités. C'est pourquoi, nous allons associer une procédure à ces différentes zones afin d'éviter ces petits problèmes.

Affichez le sous-rapport et ensuite cliquez sur l'onglet *Calc*.

L'arborescence du sous-rapport (partie de gauche de l'écran) montre les différents composants qui lui sont associés. Les différents champs que nous avons créés lors de la conception du sous-rapport apparaissent sous forme de zone, et peuvent être renommés (X = numéro chronologique attribué par le système au champ). Lorsqu'un des composants est sélectionné, on aperçoit alors les événements qui lui sont associés (partie de droite de l'écran).

Aperçu



Sélectionnez l'événement *OnGetText* et positionnez-vous ensuite dans la zone de procédure. Le code ci-dessous est affiché :

```
procedure DBText16OnGetText(var Text: String);  
begin  
    Text :=  
end;
```

'Text' correspond à la valeur qui sera rendue suivant la procédure indiquée.

Remarque : les champs des tables, les objets, les procédures ainsi que les fonctions peuvent être sélectionnées dans la partie *Boîte à outils* et glissées dans la procédure.

◆ Procédures

Ecrivez la procédure suivante pour la zone DBText correspondant au champ Db_Lignes['P.U. (PU)'] :

```
procedure DBText17OnGetText(var Text: String);  
begin  
  if Db_Lignes['P.U. (PU)'] = 0  
  then Text := ' '  
  else Text := Db_Lignes['P.U. (PU)'];  
end;
```

Cette procédure, affiche ou n'affiche pas le prix unitaire de l'article en fonction du contenu de la zone du même nom.

Description de la procédure :

Si le champ du prix unitaire (Db_Lignes['P.U. (PU)']) est égale à zéro

Alors, la valeur rendue par la zone DBText17 est un blanc.

Sinon, la valeur rendue par la zone DBText17 est le prix unitaire (Db_Lignes['P.U. (PU)'])

Le même type procédure peut être associé pour les autres zones qui ne doivent pas apparaître lorsque la ligne contient un commentaire.

Pour la quantité livrée :

```
procedure DBText18OnGetText(var Text: String);  
begin  
  if Db_Lignes['Qté livrée (QTYDELIV)'] = 0  
  then Text := ' '  
  else text := Db_Lignes['Qté livrée (QTYDELIV)'];  
end;
```

Pour la remise :

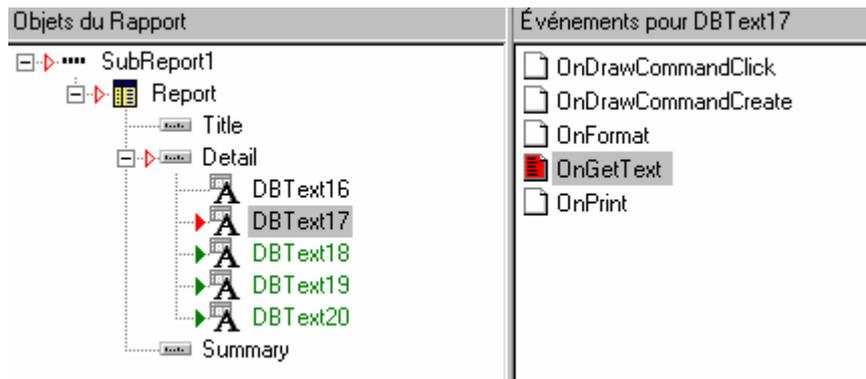
```
procedure DBText19OnGetText(var Text: String);  
begin  
  if Db_Lignes['Remise % (PRCDISC)'] = 0  
  then Text := ' '  
  else Text := Db_Lignes['Remise % (PRCDISC)'];  
end;
```

Pour le montant :

```
procedure DBText20OnGetText(var Text: String);  
begin  
  if Db_Lignes['Base TVA en dev. fac (CURRBASEAMN)'] = 0  
  then Text := ' '  
  else Text := Db_Lignes['Base TVA en dev. fac (CURRBASEAMN)'];  
end;
```

◆ Erreurs de procédure

Si la procédure que vous avez créé n'est pas correcte, ReportBuilder vous le signalera directement par l'affichage de l'événement en couleur rouge. Par contre, si la procédure est correcte, l'affichage sera de couleur verte.



Une description de l'erreur sera affichée dans la partie inférieure de l'écran de procédure :

```

procedure DBText17OnGetText (var Text: String) ;
begin
  if Db_Lignes['P.U. (PU)'] = 0
  then Text := ' '
  else Text := Db_Lignes['P.U. (PU)']
end;

```

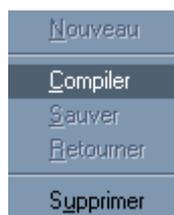
Erreur: DBText17OnGetText, Ligne 5: Attendait: 'not' ou 'or', mais 'end' trouvé à la place.

Description de l'erreur

S'il n'y a pas d'erreur le message suivant apparaît :

Compilation achevée: 0 erreurs.

Pour tester votre procédure, cliquez avec le bouton droit de la souris, dans l'écran de procédure pour afficher le menu contextuel qui lui est associé. Ensuite, sélectionnez l'option 'Compiler'.



Aperçu avant procédure :

Désignation	P.U.	Qté	%	Prix
Clavier Microsoft	61,9730	5	0	309,8700
Souris Microsoft	32,2260	5	0	161,1300
BOB Software	1.115,5210	1	0	1.115,5200
Hp Deskjet 890 Cxi Couleur	297,4720	2	0	594,9400
Profitez de nos promotions de février sur les moniteurs	0,0000	0	0	0,0000 ←

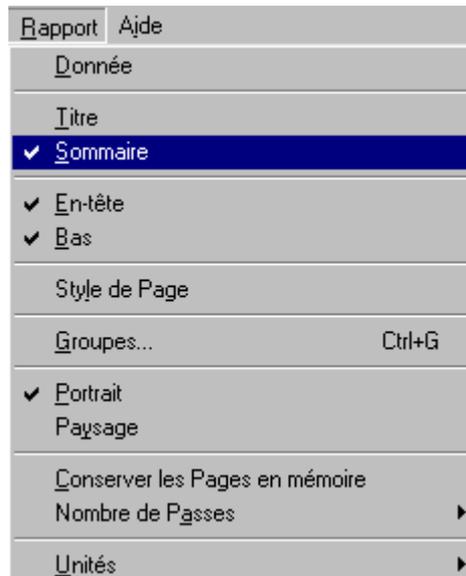
Aperçu après procédure :

Désignation	P.U.	Qté	%	Prix
Clavier Microsoft	61,973	5		309,8700
Souris Microsoft	32,226	5		161,1300
BOB Software	1115,521	1		1.115,5200
Hp Deskjet 890 Cxi Couleur	297,472	2		594,9400
Profitez de nos promotions de février sur les moniteurs				←

REALISATION DE CALCULS DE DONNEES

DB_Calculé permet de réaliser de simples calculs de données (somme, soustraction, ...) dans l'onglet *Conception* lui-même.

Dans le sous-rapport, affichez la bande *Sommaire* via le menu *Rapport | Sommaire*.



Par exemple, dans cette bande, insérez une zone de type DB_Calculé  et associez lui le champ Qté livrée (QTYDELIV) afin d'obtenir la quantité totale de la facture.



Aperçu :

Désignation	P.U.	Qté	%	Prix
Clavier Microsoft	61,973	5		309,87
Souris Microsoft	32,226	5		161,13
BOB Software	1115,521	1		1.115,52
Hp Deskjet 890 Cxi Couleur	297,472	2		594,94
Profitez de nos promotions de février sur les moniteurs				
Quantité totale:		13		

ASSOCIATION D'UNE PROCEDURE A UNE BANDE

Associez, à l'événement 'BeforeGenerate', de la bande *Détail* du *Sous-rapport* la procédure suivante :

```
procedure DetailBeforeGenerate;  
begin  
  if (Db_Lignes['P.U. (PU)']= 0) and  
    (Db_Lignes['Base TVA en dev. fac (CURRBASEAMN)']= 0) and  
    (Db_Lignes['Qté livrée (QTYDELIV)']= 0)  
  then DBText16.Font.Italic := true  
  else DBText16.Font.Italic := false;  
end;
```

Cette procédure formate le texte du composant *DBText16* (champ *Ligne de comment. (COMMENT)*) en italique lorsque le prix unitaire, la base TVA en devise ainsi que les quantités sont égales à zéro.

Aperçu :

Désignation	P.U.	Qté	%	Prix
Clavier Microsoft	61,973	5		309,87
Souris Microsoft	32,226	5		161,13
BOB Software	1115,521	1		1.115,52
Hp Deskjet 890 Cxi Couleur	297,472	2		594,94
<i>Profitez de nos promotions de février sur les moniteurs</i>				
Quantité totale:		13		

LES VARIABLES

Insérez une variable  dans la région RegSummary de la bande *Bas* du Rapport principal. Cette variable va nous donner la possibilité de lui associer une procédure. Elle peut être de différents types :

- Booléen: type vrai ou faux
- Date
- Heure
- DateHeure : type date et heure
- Entier
- Simple
- Double
- Etendu
- Monétaire
- Caractère
- Chaîne: type texte

Dans ce cas, nous allons insérer le texte légal devant s'afficher lors de l'application d'une TVA au cocontractant.

Le type de la variable peut être modifié via la liste déroulante appelée '*Type données*'. Sélectionnez la zone '*Variable 1*' et associez-lui le type *chaîne*:



Associez la procédure suivante à l'événement OnGetText de la Variable :

```
procedure Variable1OnGetText(var Text: String);  
begin  
if (Db_TVA_Ventilation['Code Tva 1 (VSTORED_1)'] = 'NSCS 0') or  
  (Db_TVA_Ventilation['Code Tva 2 (VSTORED_2)'] = 'NSCS 0') or  
  (Db_TVA_Ventilation['Code Tva 3 (VSTORED_3)'] = 'NSCS 0') or  
  (Db_TVA_Ventilation['Code Tva 4 (VSTORED_4)'] = 'NSCS 0') or  
  (Db_TVA_Ventilation['Code Tva 5 (VSTORED_5)'] = 'NSCS 0') or  
then Text := 'Taxe à acquitter par le cocontractant (&R n°1, art 20)'  
else Text := ' '  
end;
```

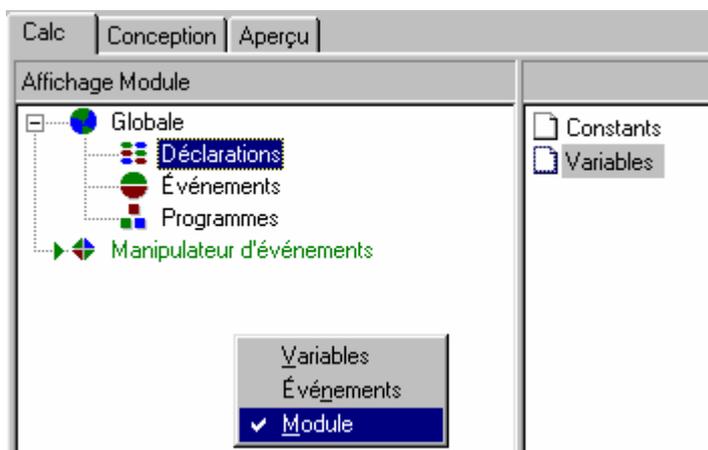
Le texte correspond au contenu de la variable. Cette procédure permet d'afficher un certain texte si le code de TVA est égale à « NSCS 0 » (TVA cocontractant).

VARIABLE GLOBALE

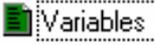
Grâce à ces variables, nous allons pouvoir déterminer une valeur qui sera accessible aussi bien dans le rapport principal, que dans le(s) sous-rapport(s).

◆ Détermination d'une variable de type global

Sélectionnez l'onglet *Calc* du Rapport principal. Dans le menu contextuel lié à la zone '*Objet du rapport*', sélectionnez l'option *Module* afin d'obtenir la zone '*Affichage module*'.

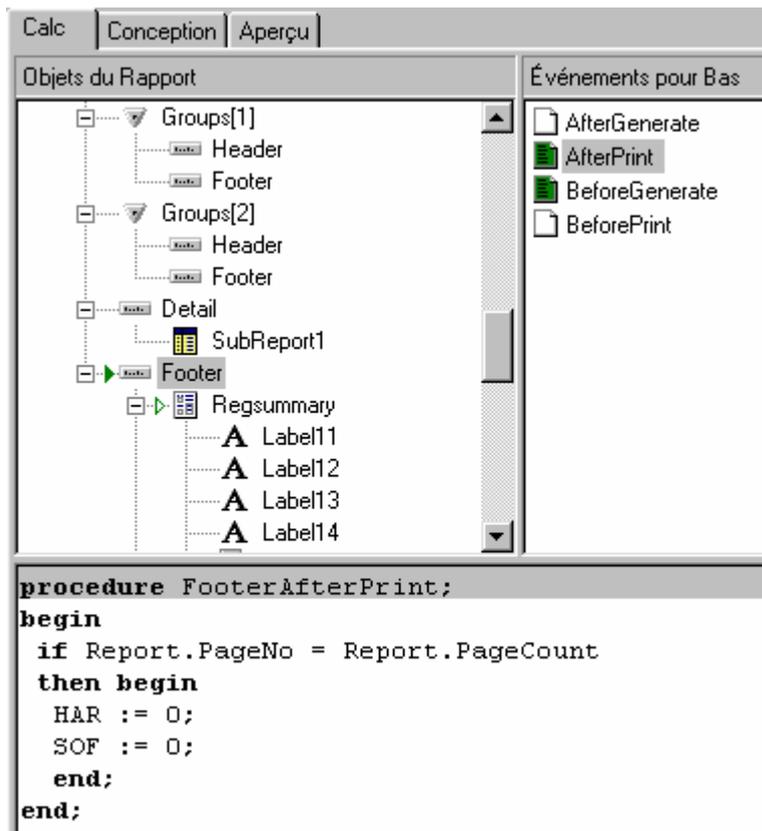


Nous allons, par exemple, totaliser le prix de base HTVA des articles en fonction de leur type. Pour commencer, créons deux variables, HAR et SOF, de type monétaire :

Pour déclarer des variables globales, cliquez sur  du module Déclarations et créez-les dans la zone de procédure.

```
var
  HAR: currency;
  SOF: currency;
```

Après chaque impression, les variables devront être initialisées à zéro. Pour ce faire, associez la procédure, reprise ci-dessous, à l'événement *AfterPrint* du *Bas* lié à l'objet du rapport.

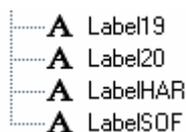


Insérez, dans la région *RegSummary*, quatre labels et disposez-les de la façon suivante :

- Un texte «TOTAL HARDWARE »
- Un texte «TOTAL SOFTWARE »
- Un texte «LabelHAR» destiné à recevoir le total des articles de la catégorie « HAR »
- Un texte « LabelSOF» destiné à recevoir le total des articles de la catégorie « SOF »



Renommez, via l'objet du rapport, les variables destinées à recevoir les totaux des articles par catégorie, en '*LabelHAR*' et '*LabelSOF*'.



Remarque: Pour renommer, cliquez deux fois lentement sur le composant à renommer.

Pour totaliser les prix de base HTVA des articles en fonction de la catégorie, associez la procédure suivante à l'événement 'BeforeGenerate' du *Détail* lié au *Sous-rapport* :

Objets du Rapport	Événements pour Détail
<pre> SubReport1 ├── Report │ └── Detail │ ├── DBText15 │ ├── DBText16 │ ├── DBText17 │ ├── DBText18 │ ├── DBText19 │ ├── DBText6 │ ├── DBText7 │ └── Variable1 </pre>	<p> <input type="checkbox"/> AfterGenerate <input type="checkbox"/> AfterPrint <input checked="" type="checkbox"/> BeforeGenerate <input type="checkbox"/> BeforePrint </p>
<pre> procedure DetailBeforeGenerate; begin if Db_Article['Catégorie article (CAT)']='HAR' then HAR:=HAR +Db_Lignes['Base TVA en dev. fac (CURRBASEAMN)']; if Db_Article['Catégorie article (CAT)']='SOF' then SOF:=SOF +Db_Lignes['Base TVA en dev. fac (CURRBASEAMN)']; end; </pre>	

Explication de la procédure :

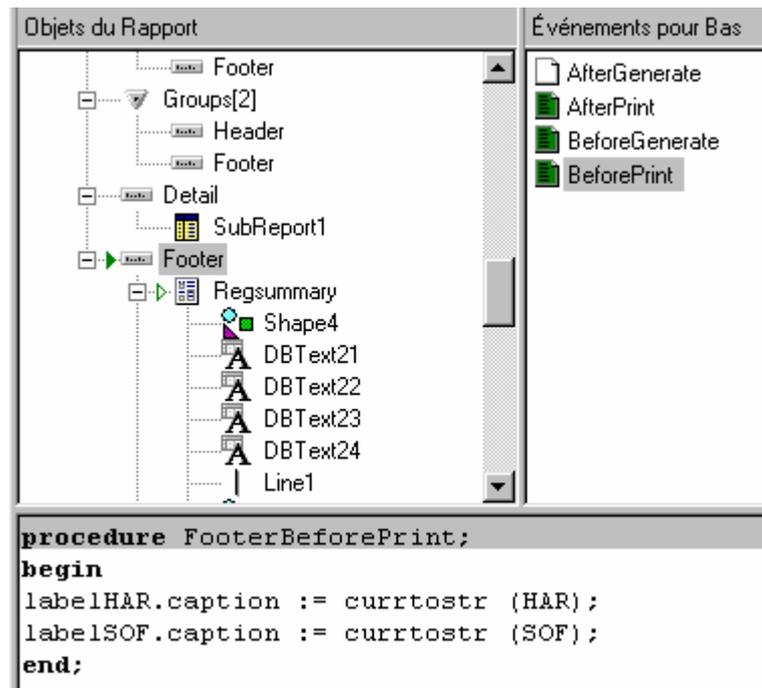
Si la catégorie de l'article est égale à « HAR » (HARDWARE)

Alors la variable HAR est égale à la variable HAR plus le montant de base de l'article

Si la catégorie de l'article est égale à « SOF » (SOFTWARE)

Alors la variable SOF est égale à la variable SOF plus le montant de base de l'article

Dans l'onglet *Calc* du rapport principal, associez les variables HAR et SOF au LabelHAR et au LabelSOF afin que leurs valeurs soient affichées :



Explication de la procédure :

Le texte affiché par une zone de type texte est conservé dans le "Caption" (zone affichable de l'objet). C'est pourquoi il est nécessaire de le préciser dans la procédure.

Le « caption » de la zone labelHAR est égal à la variable HAR convertie du type monétaire en type texte.

Le « caption » de la zone labelSOF est égal à la variable SOF convertie du type monétaire en type texte

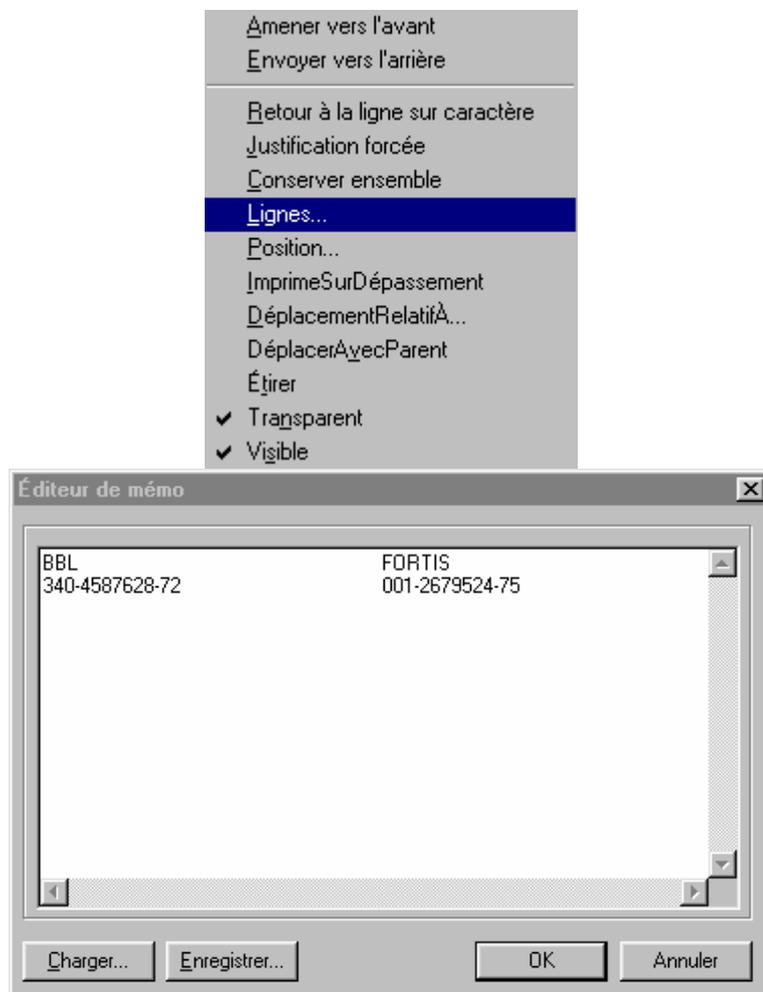
LES ZONES MEMOS

Les zones mémos permettent d'insérer un texte sur plusieurs lignes.

Par exemple, insérez une zone mémo  dans le bas de la région 'Regsummary'. La zone suivante apparaît :



Pour y insérer du texte, il suffit de cliquer avec le bouton droit de la souris afin d'afficher le menu contextuel et de choisir l'option *Lignes...*. L'éditeur de mémo apparaît:



Vous pouvez y insérer, par exemple, les coordonnées financières de la société. Cliquez sur OK pour enregistrer le texte. Ajustez la zone mémo à la largeur de la page et à la hauteur du texte.

Grâce à la barre d'outils format, vous pouvez adapter la taille de la police, attribuer une couleur de fond,... jusqu'à l'obtention du résultat suivant :

<i>BBL</i> 340-4587628-72	<i>FORTIS</i> 001-4987265-48
------------------------------	---------------------------------

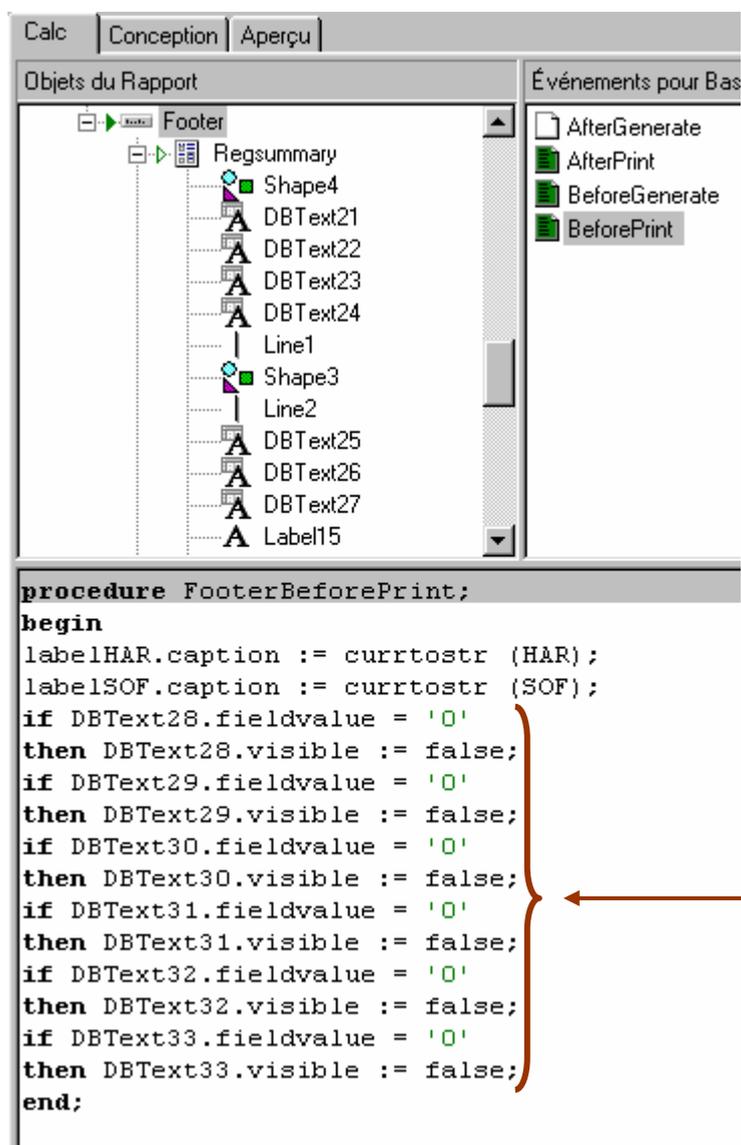
PROCEDURE POUR AFFICHER UN BLANC A LA PLACE DE ZERO

Certaines personnes préfèrent ne rien afficher lorsque une zone est égale à zéro car cela n'est pas esthétique. Dans ce cas, il suffit de créer une procédure qui remplace un zéro par un blanc.

Prenons l'exemple d'une facture pour laquelle il n'y a qu'un seul pourcentage TVA (21%). Le tableau de la ventilation TVA donne le résultat suivant :

21 %	0 %	0 %
2.181,4600	0,0000	0,0000
458,1100	0,0000	0,0000

Pour ne plus afficher les zéros, associez la procédure suivante à l'événement 'BeforePrint' du *Bas* lié au rapport principal :



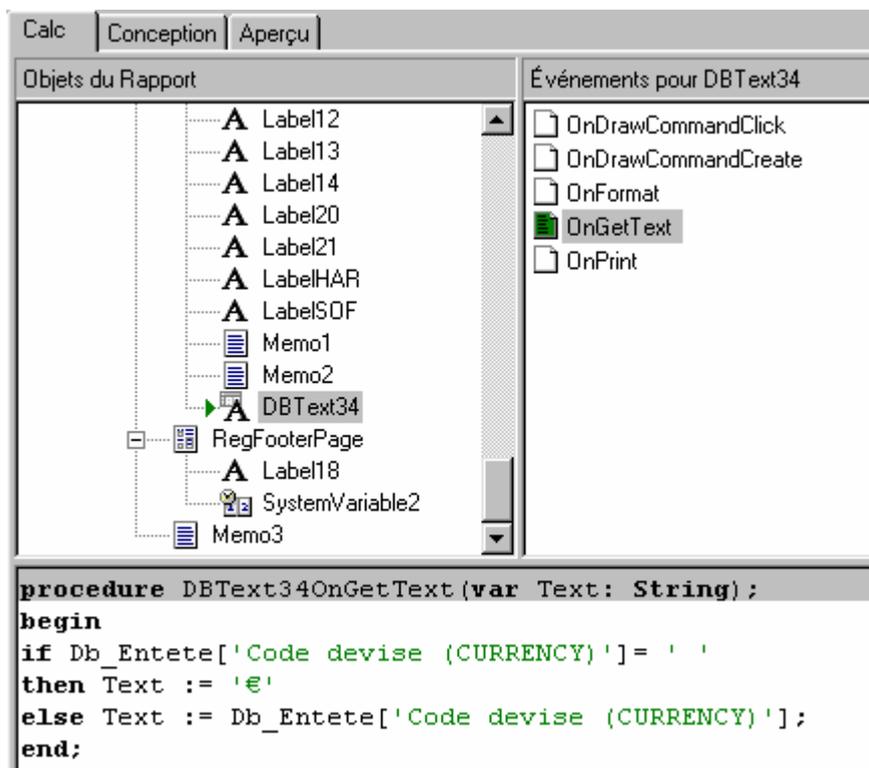
```
procedure FooterBeforePrint;
begin
labelHAR.caption := currtostr (HAR);
labelSOF.caption := currtostr (SOF);
if DBText28.fieldvalue = '0'
then DBText28.visible := false;
if DBText29.fieldvalue = '0'
then DBText29.visible := false;
if DBText30.fieldvalue = '0'
then DBText30.visible := false;
if DBText31.fieldvalue = '0'
then DBText31.visible := false;
if DBText32.fieldvalue = '0'
then DBText32.visible := false;
if DBText33.fieldvalue = '0'
then DBText33.visible := false;
end;
```

Résultat avec la procédure :

21 %	0 %	0 %
2.181,4600 458,1100		

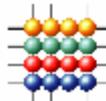
AFFICHER LA DEVISE DE BASE A LA PLACE D'UN BLANC

BOB Software n'affiche pas la devise lorsque la facture est saisie en devise de base. Pour afficher la devise (ex :EUR) lorsque la facture est encodée en devise de base, il suffit d'associer la procédure suivante à l'événement 'OnGetText' de la zone DBText du champ 'Devise de la facture' (à ajouter) :



```
procedure DBText34OnGetText(var Text: String);
begin
if Db_Entete['Code devise (CURRENCY)'] = ' '
then Text := '€'
else Text := Db_Entete['Code devise (CURRENCY)'];
end;
```

RESULTAT



BOB Software
Quai Matisa 32
BE4000158go
TVA BE 453-296-941

RENAULT INDUSTRIE BELGIQUE SA
CHAUSSÉE DE MONS 301
BE1070 BRUXELLES
TVA: BE 402-645-614

Facture n° 20020001
Date: 10/01/2001
Echéance 09/02/2001

Designation	P.U.	Qté	%	Prix
Clavier Microsoft	81,673	5		308,67
Souris Microsoft	32,226	5		161,13
BOB Software	1115,521	1		1115,52
Hp Deskjet 890 Cxl Couleur	297,472	2		594,94
<i>Profitez de nos promotions de février sur les montres</i>				
Quantité totale		13		

21 %	0 %	0 %				
2.181,46						
458,11						
<table border="1"> <tr> <td>Total HARDWARE</td> <td>1085,94</td> </tr> <tr> <td>Total SOFTWARE</td> <td>1115,52</td> </tr> </table>			Total HARDWARE	1085,94	Total SOFTWARE	1115,52
Total HARDWARE	1085,94					
Total SOFTWARE	1115,52					
Total HTVA		2.181,46				
Ristourne (-)		0,00				
TVA		458,11				
Total à payer		€ 2.639,57				

BAL
340-4537025-72

FORVIS
001-267924-75

TRUCS & ASTUCES

Cette rubrique a été éditée afin faciliter l'utilisation du Report Builder.

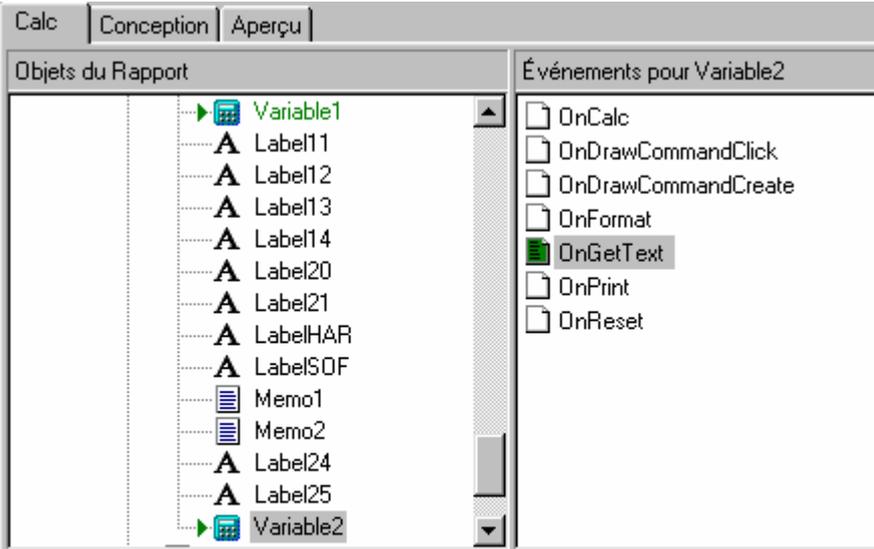
Dans la première partie du *Report Builder*, nous avons appris à construire un document au départ d'un document vierge. Cette méthode nous a permis de mieux comprendre le fonctionnement du *Report Builder*.

Nous vous conseillons cependant d'adapter un rapport prédéfini par BOB Software, plutôt que de créer un nouveau document. En effet, les rapports prédéfinis comportent déjà plusieurs liens vers différents fichiers, qui seront inexistantes si vous choisissez de créer un nouveau rapport.

AFFICHER LE SOLDE ACTUEL DANS LE LAYOUT D'UN RAPPEL OU D'UNE FACTURE

Ajoutez une variable  dans le layout et liez la procédure suivante à l'événement OnGetText :

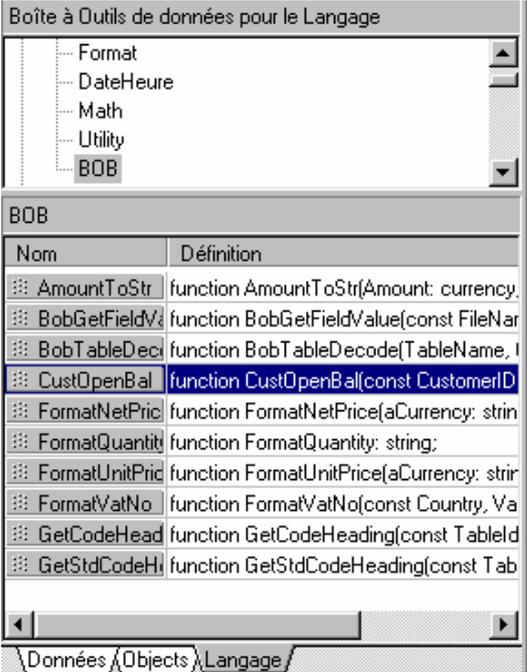
Solde ouvert client: 



```

procedure Variable2OnGetText(var Text: String);
begin
    Text := CustOpenBal (Db_Client['Référence (CID)']);
end;
  
```

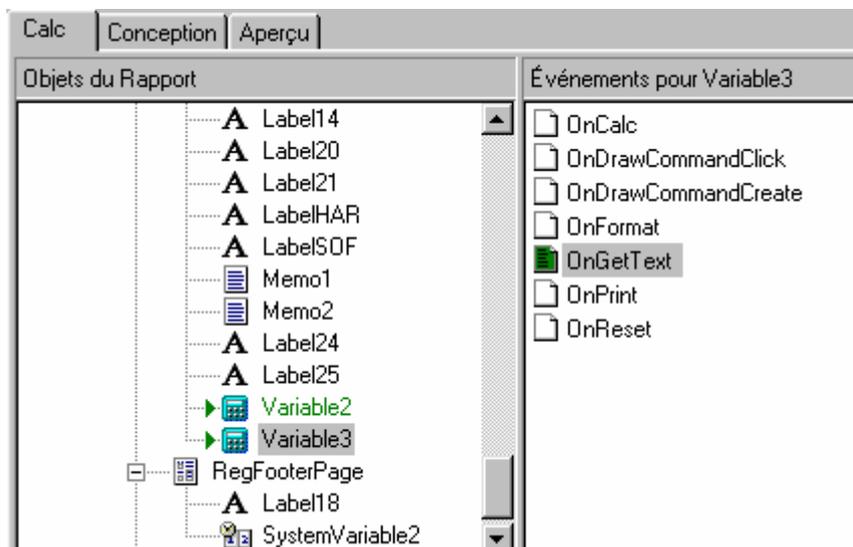
Vous pouvez retrouver ces fonctions en sélectionnant l'onglet *Langage*. Choisissez ensuite BOB:



Nom	Définition
AmountToStr	function AmountToStr(Amount: currency;
BobGetFieldVa	function BobGetFieldValue(const FileNar
BobTableDeco	function BobTableDecode(TableName, I
CustOpenBal	function CustOpenBal(const CustomerID
FormatNetPric	function FormatNetPrice(aCurrency: strin
FormatQuantity	function FormatQuantity: string;
FormatUnitPric	function FormatUnitPrice(aCurrency: strir
FormatVatNo	function FormatVatNo(const Country, Va
GetCodeHead	function GetCodeHeading(const TableId
GetStdCodeHi	function GetStdCodeHeading(const Tab

AFFICHER UN MONTANT EN LETTRE

Si vous voulez afficher le montant en toutes lettres, ajoutez de nouveau une variable dans le layout et attribuez-lui la formule suivante :



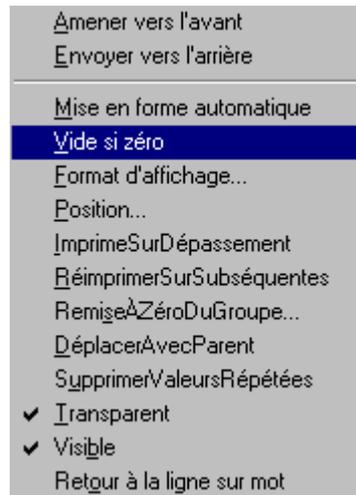
- **Amount** : est à remplacer par le champ contenant la valeur numérique à transformer en lettres.
- **Language** : est à remplacer par 'FR' pour Français, ou 'NL' pour Néerlandais, ou 'DE' pour Allemand.
- **Dev** : est à remplacer par le code devise du montant.

La procédure devient donc:

```
procedure Variable3OnGetText(var Text: String);  
begin  
  Text := AmountToStr(Db_Entete['Mnt dû en dev. fact. (CURRPAYAMN)'], 'FR', Db_Entete['Code devise (CURRENCY)'] )  
end;
```

FONCTION 'VIDE SI ZERO'

Cette fonction, disponible dans le menu contextuel, n'affiche rien si le montant contenu dans le champ est égal à 0. Elle sera disponible à partir de la version définitive 2.5. Il ne sera donc plus nécessaire d'utiliser des procédures de code pour éviter d'afficher des 0 lorsque le montant est nul.

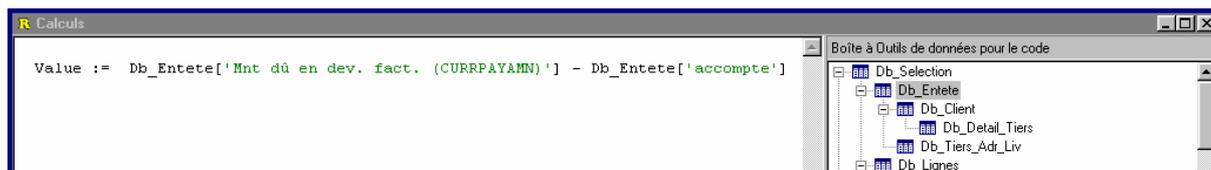


REALISATION D'OPERATIONS SUR CERTAINS CHAMPS

Si vous voulez faire des calculs sur certains champs et y attribuer un format d'affichage, vous ne devez pas utiliser un *Texte* mais une *Variable* à laquelle vous donnerez le type *Monétaire* :

Regsummary		Total hTVA	2.181,46
Solde ouvert Client	Variable1	Ristourne (-)	
Mode de paiement	Variable3	TVA	458,11
	1	Total à payer	2.639,57
Certifié sincère pour la somme de		Accompte (-)	0,00
Variable2		Solde à payer	Variable4

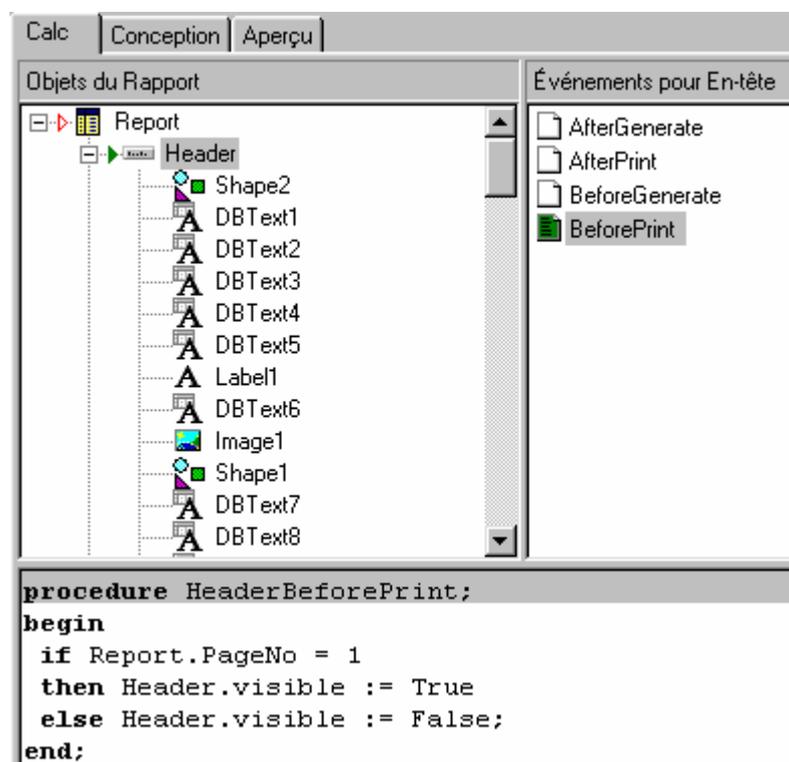
Les opérations se réalisent par l'option *Calculs* du menu contextuel associé à la variable. La formule suivante vous permettra par exemple, de déduire l'acompte payé par le client du montant total de la facture.



Remarque: Acompte est un champ supplémentaire créé dans l'entête de l'écran d'encodage des factures.

AFFICHAGE DE L'EN-TÊTE SUR LA PREMIÈRE PAGE UNIQUEMENT

Si vous souhaitez que l'en-tête apparaisse seulement sur la première page, vous pouvez lier la procédure suivante à l'événement BeforePrint pour le Header:



```
procedure HeaderBeforePrint;  
begin  
  if Report.PageNo = 1  
  then Header.visible := True  
  else Header.visible := False;  
end;
```

L'en-tête sera seulement imprimée sur la première page.

Pour vous aider, la procédure ci-dessus est disponible dans un layout standard.

AFFICHAGE DE DESCRIPTION DE TABLES

La fonction BOBTABLEDECODE est utilisée pour afficher les différentes descriptions du fichier xxtables.db. Par exemple, pour visualiser la description du mode de paiement sur le lay-out :

Ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuez la procédure suivante à l'événement *OnGetText* de la variable :

Mode de paiement: Variable5

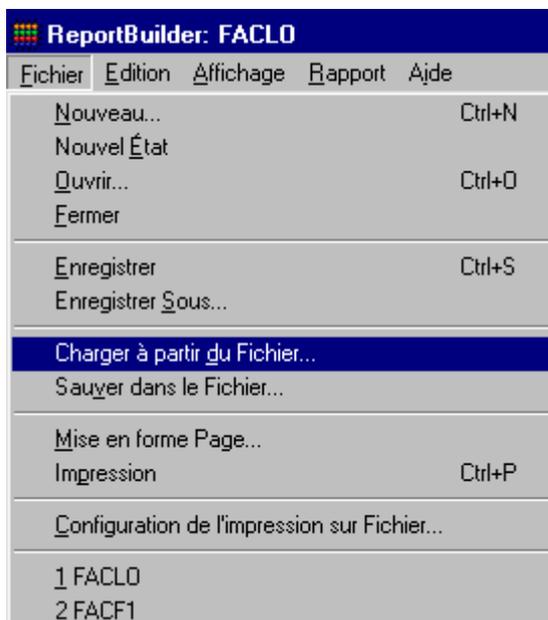
```
procedure Variable5OnGetText(var Text: String);  
begin  
  Text := BobTableDecode('PAYMODE', Db_Entete['Code mode de pmt (PAYMODE)'], 'Heading1');  
end;
```

Vous trouvez cette fonction dans l'onglet Langage chez BOB:

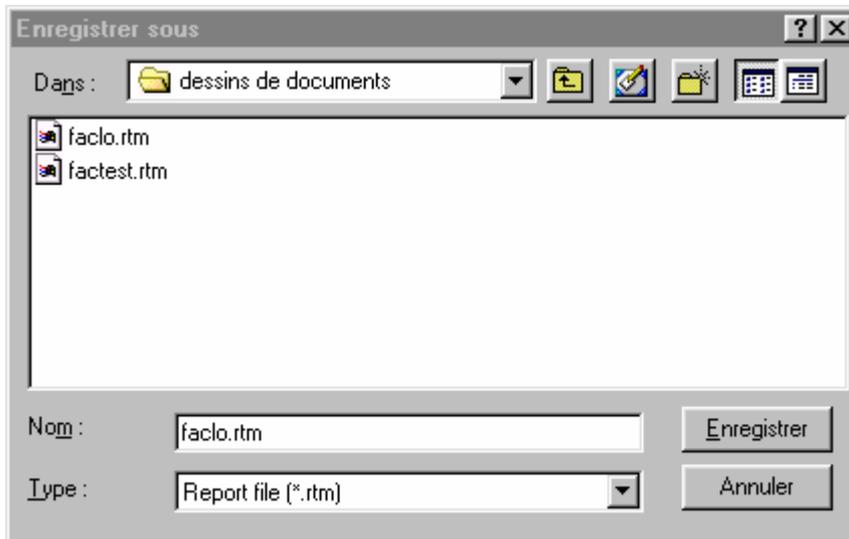
BOB	
Nom	Définition
::: AmountToStr	function AmountToStr(Amount: currency; const Language, Dev
::: BobGetFieldV	function BobGetFieldValue(const FileName: string; FieldName:
::: BobTableDeco	function BobTableDecode(TableName, Code, FieldName: strin
::: CustOpenBal	function CustOpenBal(const CustomerID: string): string;
::: FormatNetPric	function FormatNetPrice(aCurrency: string): string;
::: FormatQuantit	function FormatQuantity: string;
::: FormatUnitPric	function FormatUnitPrice(aCurrency: string): string;
::: FormatVatNo	function FormatVatNo(const Country, VatNo: string): string;
::: GetCodeHead	function GetCodeHeading(const TableId, Code: string; AltHeac
::: GetStdCodeH	function GetStdCodeHeading(const TableId, Code: string; Lanç

EXPORTATION ET IMPORTATION DES DESSINS DE DOCUMENT.

Les exportations et importations de document se font via les options *Charger à partir du fichier* (importation) et *Sauver dans le fichier* (exportation) du menu *Fichier*



Un fichier au format .rtm est créé dans le répertoire choisi.



LES ARRONDIS

Trois nouvelles fonctions de Report Builder permettent d'arrondir les prix unitaires, les prix nets et les quantités tels que BOB les arrondit. En effet, sans ces fonctions, Report Builder arrondit les montants en fonction des paramètres régionaux du panneau de configuration.

Ces fonctions sont accessibles via l'onglet *Calc* du Report Builder au niveau de l'onglet *Langage* dans la catégorie BOB:

Boîte à Outils de données pour le Langage

- [-] Fonction
 - Chaîne
 - Conversion
 - Format
 - DateHeure
 - Math
 - Utility
 - BOB**

BOB	
Nom	Définition
ActualQtyDelivSolde	function ActualQtyDelivSolde(aDbk: String;aFYear:String;aDocno:Integer;aLinkno:Integer): String;
AmountToStr	function AmountToStr(Amount: currency; const Language, Dev: string): string;
BEZipcode	function BeZipcode(Zipcode: String): string;
BobAllTableDecode	function BobAllTableDecode(TableName, IndexName, IndexKey, FieldName: string): string;
BobGetFieldValue	function BobGetFieldValue(const FileName: string; FieldName: String): string;
BobTableDecode	function BobTableDecode(TableName, Code, FieldName: string): string;
CurCustOpenBal	function CurCustOpenBal(const CustomerID: string): Currency;
CustOpenBal	function CustOpenBal(const CustomerID: string): string;
FileExists	function FileExists(FileName: String): string;
FormatAmount	function FormatAmount(tfFormati: Integer; ACurrency: String; Alternate: String): String;
FormatNetPrice	function FormatNetPrice(aCurrency: string): string;
FormatQuantity	function FormatQuantity: string;
FormatUnitPrice	function FormatUnitPrice(aCurrency: string): string;
FormatVatNo	function FormatVatNo(const Country, VatNo: string): string;
GetAccountHeading	function GetAccountHeading(Imput: String; AltHeading: Boolean): String;
GetCodeHeading	function GetCodeHeading(const TableId, Code: string; AltHead: Boolean): string;
GetCtlInfo	function GetCtlInfo(const CtlId, Field: string): string;
GetCustArtRef	function GetCustArtRef(ArtNum: Integer; CustId: String): String;
GetDocOriHeader	function GetDocOriHeader(aDbk: String;aFYear:String;aDocno:Integer;aField:String): String;
GetDOrdCustHeader	function GetDOrdCustHeader(aDbk: String;aFYear:String;aDocno:Integer;aLinkno:Integer;aField:String): String;
GetDOrdCustLine	function GetDOrdCustLine(aDbk: String;aFYear:String;aDocno:Integer;aLinkno:Integer;aField:String): String;
GetMonthHeading	function GetMonthHeading(MonthNum: Integer; LangId: String): String;
GetPrice	function GetPrice(ArtNum: integer; PriceCode: string; PriceDate: TDateTime): currency;
GetStdCodeHeading	function GetStdCodeHeading(const TableId, Code: string; Language: string): string;
GetSuppArtRef	function GetSuppArtRef(ArtNum: Integer; Suppld: String): String;
GetVATCaseHeading	function GetVATCASEHeading(CodeLang: String; VatCase: String; YearB: String; MonthB: String; YearE: String; MonthE: String): String;
IntToStrZero	function IntToStrZero(Num,Length: integer): string;
MinToStrTime	function MinToStrTime(min: integer; MaxLength: Integer): string;
StrTimeToMin	function StrTimeToMin(StrTime: string): Integer;

Il faut associer ces fonctions aux événements des objets ou des bandes concernés.

◆ Prix unitaire

La fonction FormatUnitPrice demande un code devise comme paramètre et donne le DisplayFormat en fonction de cette devise.

```
procedure DBText30OnFormat;  
begin  
    DBText30.DisplayFormat := FormatUnitPrice(Db_Entete['Code devise (CURRENCY)']);  
end;
```

◆ Quantité

La fonction FormatQuantity ne demande pas un paramètre et donne le DisplayFormat en fonction des options de la facturation.

```
procedure DBText18OnPrint;  
begin  
    DBText18.DisplayFormat := FormatQuantity;  
end;
```

◆ Prix net

La fonction FormatNetPrice demande un code devise comme paramètre et donne le DisplayFormat en fonction de cette devise.

```
procedure DBText25OnFormat;  
begin  
    DBText25.DisplayFormat := FormatNetPrice(Db_Entete['Code devise (CURRENCY)']);  
end;
```

APPLIQUER LA MISE EN PAGES CORRECTE QUI A ETE DEFINIE POUR LE TEXTE DE REGROUPEMENT

L'onglet 'Regroupement' du menu *Facturation | Configuration | Options* vous permet de formater le texte de regroupement.

Options gestion commerciale

Généralités Liaison compta **Regroupement** Dessins Doc.

Ouverture automatique de la fenêtre de regroupement

Valeur par défaut

Texte de regroupement

#DBK# - #DOCNR# du #DATE#

Espace avant Gras

Espace après Souligné

Récupération des conditions générales

Pas de récupération

Du premier document

Du dernier document

Récupération des documents associés à la facture

OK Annuler

Pour que le texte formaté s'imprime correctement dans le Report Builder, vous devez créer la procédure suivante et l'associer à l'évènement 'OnPrint' du champ Db_Detail (Ligne de commentaire):

```
procedure DBText16OnPrint;  
begin  
DBText16.font.Bold := Db_Lignes['Ligne de comment. en (ISBOLD)'];  
DBText16.font.Underline := Db_Lignes['Ligne de comm. en so (ISUNDERLINE)'];  
end;
```

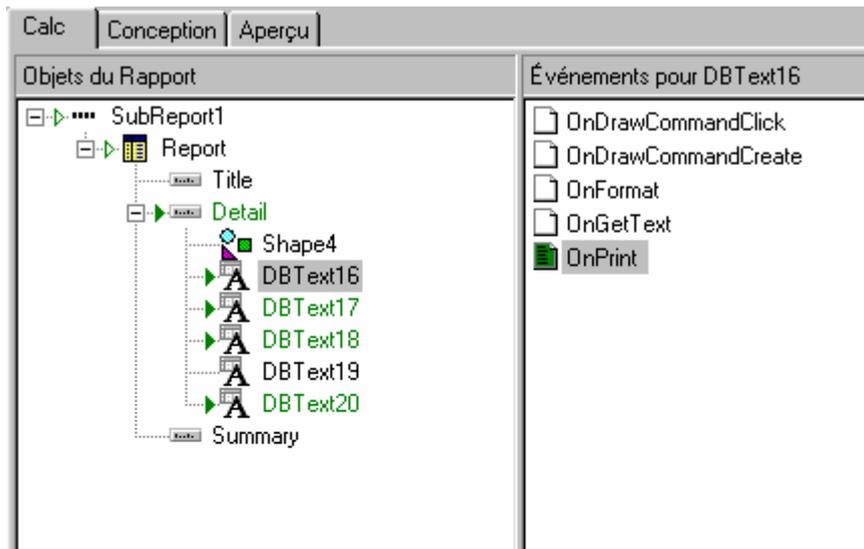
Vous pouvez retrouver ces 2 champs dans le fichier xxIHISTO.db avec une valeur booléenne vrai ou faux pour les textes de regroupement.

LIGNES DE COMMENTAIRE NORMALES

Il est possible d'associer une procédure aux lignes de commentaire. Dans ce cas, nous souhaitons par exemple mettre en italique la ligne de commentaire du document.

◆ Pas de texte de regroupement

Si on n'utilise pas de texte de regroupement, la procédure suivante doit être appliquée à l'évènement 'OnPrint' du champ:



```
procedure DBText16OnPrint;
begin
if Db_Lignes['Type de ligne (LINETYPE)'] = 'C'
then begin
  DBText16.font.Italic := true
  end
else begin
  DBText16.font.Italic := false
  end;
end;
```

◆ Texte de regroupement et ligne de commentaire

Lorsque, dans le document, il y a à la fois des lignes de commentaire et des textes de regroupement, le test suivant doit être effectué, après la mise en page des champs *isbold* et *isunderline*.

```
procedure DBText16OnPrint;
begin
DBText16.font.Bold := Db_Lignes['Ligne de comment. en (ISBOLD)'];
DBText16.font.Underline := Db_Lignes['Ligne de comm. en so (ISUNDERLINE)'];
  if
    (Db_Lignes['Type de ligne (LINETYPE)'] = 'C') and
    (Db_Lignes['Ligne de comment. en (ISBOLD)'] <> True)
  then begin
    DBText16.font.Italic := true;
  end
  else begin
    DBText16.font.Italic := false;
  end;
end;
```

◆ Paramètres de mise en page

Les paramètres les plus fréquemment utilisés pour la mise en page des champs sont les suivants:

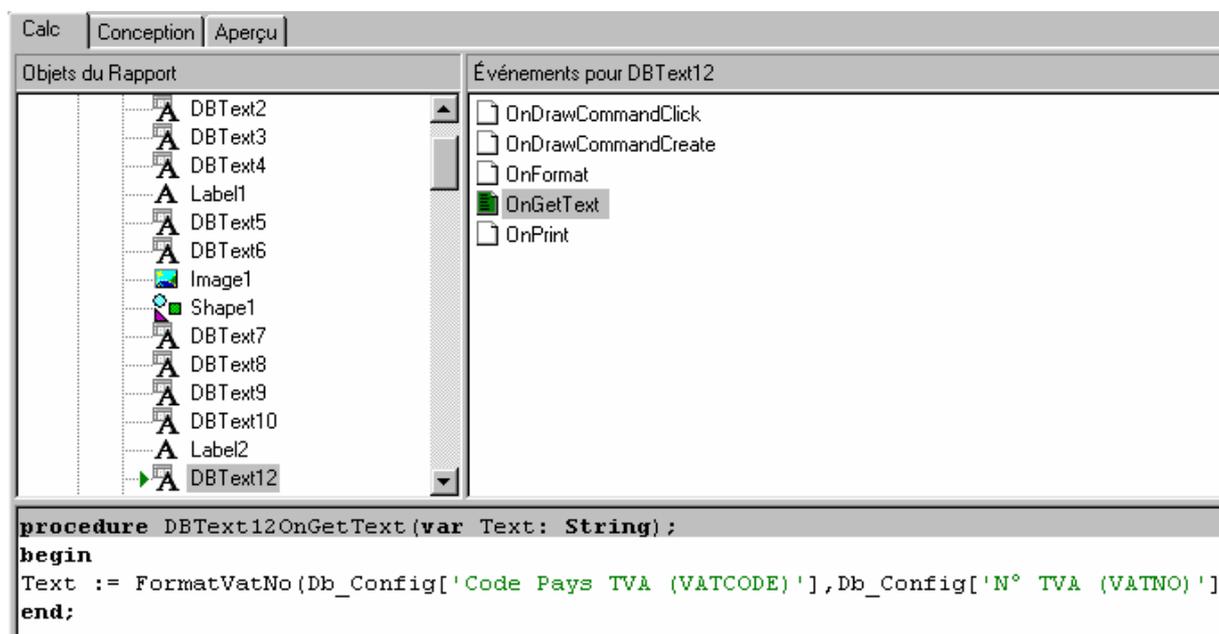
- Normal: pour annuler toute autre mise en page
- Bold: pour mettre du texte en gras
- Italic: pour mettre du texte en italique
- Strikeout: pour barrer le texte
- Underline: pour souligner du texte

FORMATER LE NUMERO DE TVA

La procédure suivante adapte le format du numéro de TVA en fonction de la législation de chaque pays. Si vous travaillez avec plusieurs sociétés étrangères, il est donc préférable d'utiliser cette procédure plutôt que l'option '*Format d'affichage*' menu contextuel associé à la variable.

Pour réaliser cette procédure, associez la fonction `FormatVatNo(Country, VatNo)` à l'événement '*OnGet Text*' du champ concerné

- Pays: champ 'Code Pays TVA (VATCODE)' de la table `Db_Config`.
- N° TVA: champ 'N° TVA (VATNO)' de la table `Db_Config`.



```
procedure DBText12OnGetText(var Text: String);
begin
Text := FormatVatNo(Db_Config['Code Pays TVA (VATCODE)'],Db_Config['N° TVA (VATNO)']
end;
```

ORDRE DES VARIABLES

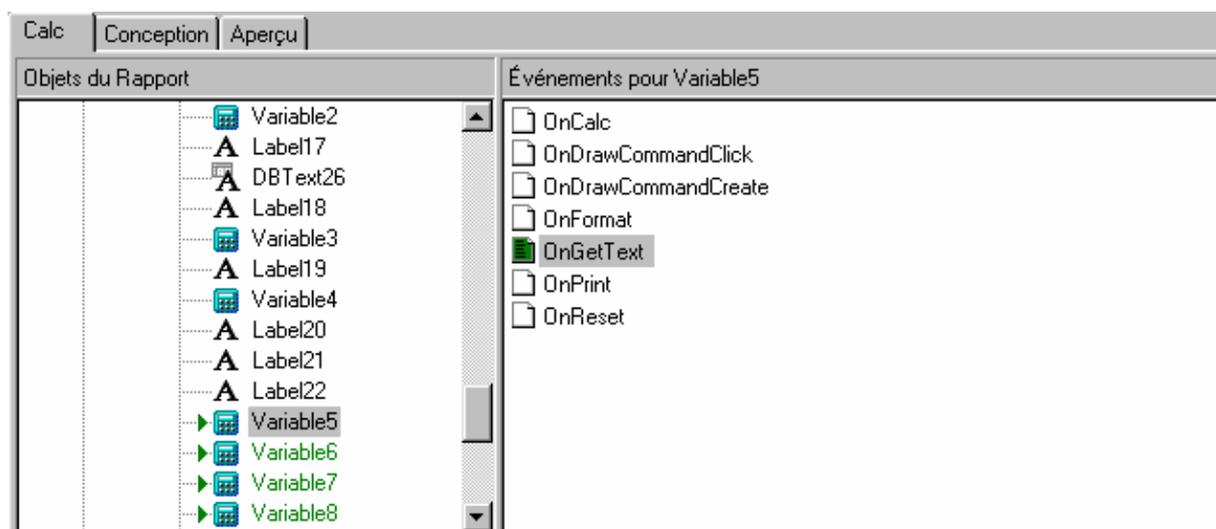
La procédure utilisée dans la création des régions du dossier technique nous donne un tableau reprenant les taux de TVA dans l'ordre dans lequel ils apparaissent sur la facture (suivant l'ordre dans lequel les articles sont encodés).

Si l'entreprise n'utilise que certains taux de TVA, elle peut choisir de créer un tableau semblable à l'exemple ci-dessous, qui lui permettra de faire apparaître les différents taux de TVA dans un certain ordre.

	6 %	12 %	21 %	Emballage
Montant				
TVA				

La table montre donc toujours le montant de base et la TVA des articles à 6%, 12% puis 21%. La dernière colonne reprend les emballages consignés (code EMS).

Pour le montant de base, on associe la procédure suivante pour l'évènement 'OnGetText' de chaque variable (en adaptant le taux de TVA):



```
procedure Variable5OnGetText(var Text: String);
begin
  if Db_TVA_Ventilation['Taux de Tva 1 (VATCMP_1)'] = '6'
  then Text := FormatCurr ('0.00',Db_TVA_Ventilation['Mnt base 1 (BASEAMN_1)'])
  else if Db_TVA_Ventilation['Taux de Tva 2 (VATCMP_2)'] = '6'
  then Text := FormatCurr ('0.00',Db_TVA_Ventilation['Mnt base 2 (BASEAMN_2)'])
  else if Db_TVA_Ventilation['Taux de Tva 3 (VATCMP_3)'] = '6'
  then Text := FormatCurr ('0.00',Db_TVA_Ventilation['Mnt base 3 (BASEAMN_3)'])
  else if Db_TVA_Ventilation['Mnt Tva 4 (VATAMN_4)'] = '6'
  then Text := FormatCurr ('0.00',Db_TVA_Ventilation['Mnt base 4 (BASEAMN_4)'])
  else if Db_TVA_Ventilation['Taux de Tva 5 (VATCMP_5)'] = '6'
  then Text := FormatCurr ('0.00',Db_TVA_Ventilation['Mnt base 5 (BASEAMN_5)'])
end;
```

Il faut donc tester chaque champ et voir si nous retrouvons un 6, 12 ou 21%. Si c'est le cas, la variable sera affichée.

Pour le montant de la TVA, on associe la procédure suivante pour l'évènement 'OnGetText' de chaque variable:

```

procedure Variable8OnGetText(var Text: String);
begin
  if Db_TVA_Ventilation['Taux de Tva 1 (VATCMP_1)'] = '6'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt Tva 1 (VATAMN_1)'])
  else if Db_TVA_Ventilation['Taux de Tva 2 (VATCMP_2)'] = '6'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt Tva 2 (VATAMN_2)'])
  else if Db_TVA_Ventilation['Taux de Tva 3 (VATCMP_3)'] = '6'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt Tva 3 (VATAMN_3)'])
  else if Db_TVA_Ventilation['Mnt Tva 4 (VATAMN_4)'] = '6'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt Tva 4 (VATAMN_4)'])
  else if Db_TVA_Ventilation['Taux de Tva 5 (VATCMP_5)'] = '6'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt Tva 5 (VATAMN_5)']);
end;
  
```

Enfin, pour la variable associée aux emballages consignés, nous réalisons le test suivant sur le code interne du code TVA code EMS 0 % (VSTORED='NSPKS0'¹):

```

procedure Variable11OnGetText(var Text: String);
begin
  if Db_TVA_Ventilation['Code Tva 1 (VSTORED_1)'] = 'NSPKS0'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt base 1 (BASEAMN_1)'])
  else if Db_TVA_Ventilation['Code Tva 2 (VSTORED_2)'] = 'NSPKS0'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt base 2 (BASEAMN_2)'])
  else if Db_TVA_Ventilation['Code Tva 3 (VSTORED_3)'] = 'NSPKS0'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt base 3 (BASEAMN_3)'])
  else if Db_TVA_Ventilation['Code Tva 4 (VSTORED_4)'] = 'NSPKS0'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt base 4 (BASEAMN_4)'])
  else if Db_TVA_Ventilation['Code Tva 5 (VSTORED_5)'] = 'NSPKS0'
  then Text := FormatCurr ('.00',Db_TVA_Ventilation['Mnt base 5 (BASEAMN_5)']);
end;
  
```

¹ Att.: NSPKS0 s'écrit NSPKS zéro.

Résultat:

	6 %	12 %	21 %	Emballage
Montant		852,63	120,23	194,00
TVA		102,32	25,25	

Remarque: Il s'agit ici d'un exemple. Les procédures décrites ci-dessus peuvent être adaptées à d'autres taux de TVA.

ATTRIBUER DES DECIMALES A UNE VARIABLE

Si vous souhaitez que vos variables de type monétaire se présentent toujours avec le même nombre de décimales, vous devez formater ces variables en utilisant la procédure suivante:

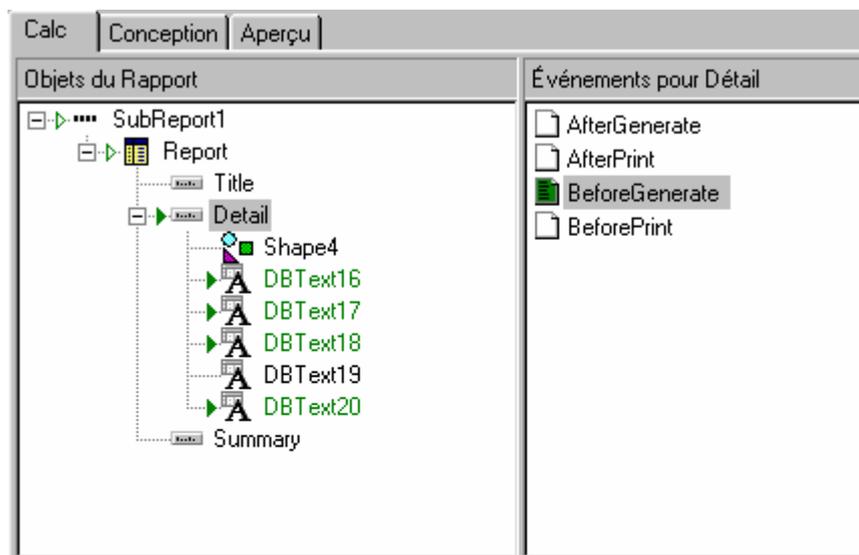
```
begin  
  Text := FormatCurr('0.00', 'champ')  
end;
```

Le nombre de zéro correspond au nombre de décimales, tandis que le 'champ' représente la zone qui doit être formatée.

GRISER UNE LIGNE SUR DEUX DANS LE DETAIL DE LA FACTURE

Tracez d'abord un cadre autour de la ligne reprenant le détail des articles. Grisez cette ligne à l'aide des outils dessin.

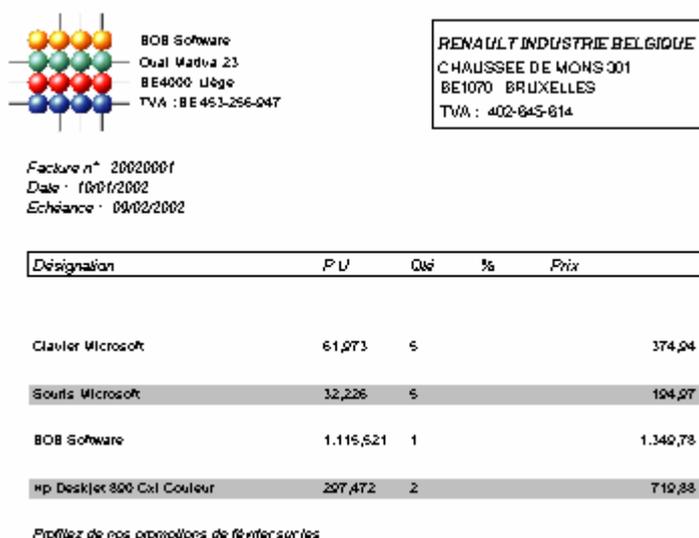
Ajoutez ensuite la procédure suivante dans l'événement *BeforeGenerate* de la zone *Details* :



```

procedure DetailBeforeGenerate;
begin
  if detail.count mod 2 = 0
  then shape4.visible := True
  else shape4.visible := False;
end;
  
```

Résultat:

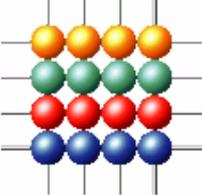


Designation	P.U.	Qte	%	Prix
Clavier Microsoft	61,273	6		374,24
Souris Microsoft	32,226	6		194,27
BOB Software	1.115,521	1		1.340,78
Hp Deskjet 800 Cxi Couleur	297,472	2		719,88

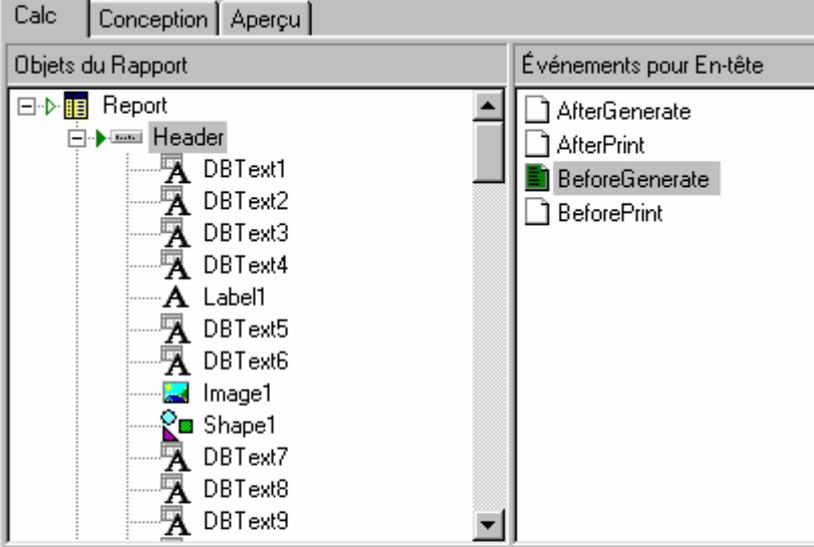
Profitez de nos promotions de fin d'année sur les

INDIQUER 'COPIE' A PARTIR DE L'IMPRESSION DU 2^E EXEMPLAIRE

Insérer un texte 'COPIE' dans l'entête du document

	BOB Software Quai Mativa 23 BE4000 Liège TVA : BE 453-256-947	RENAULT INDUSTRIE BELGIQUE CHAUSSÉE DE MONS 301 BE1070 BRUXELLES TVA : 402-645-614
Facture n° : 20020001 Date : 10/01/2002 Echéance : 09/02/2002		
Désignation P.U. Qté % Prix		

Associez la procédure suivante à l'évènement 'BeforeGenerate' ou 'BeforePrint' du Header:

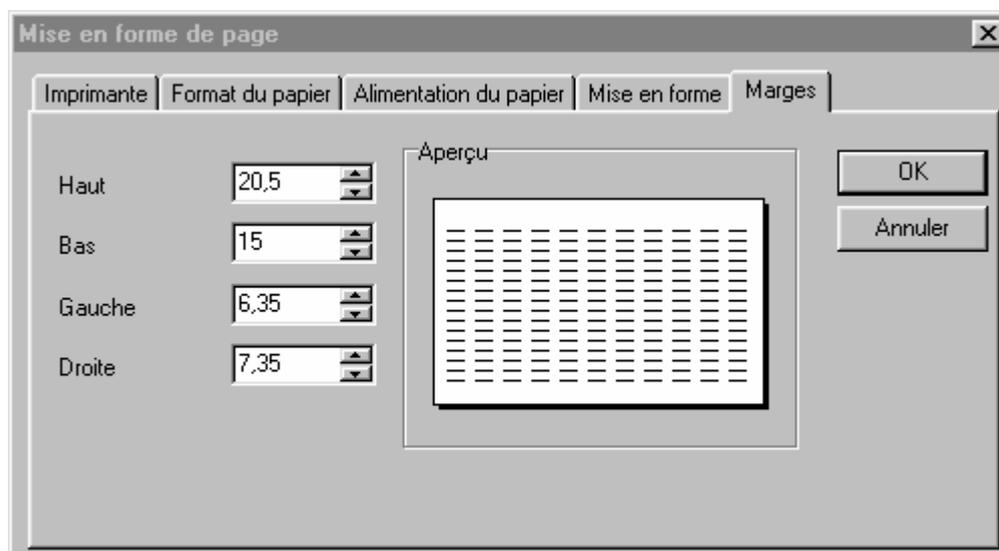
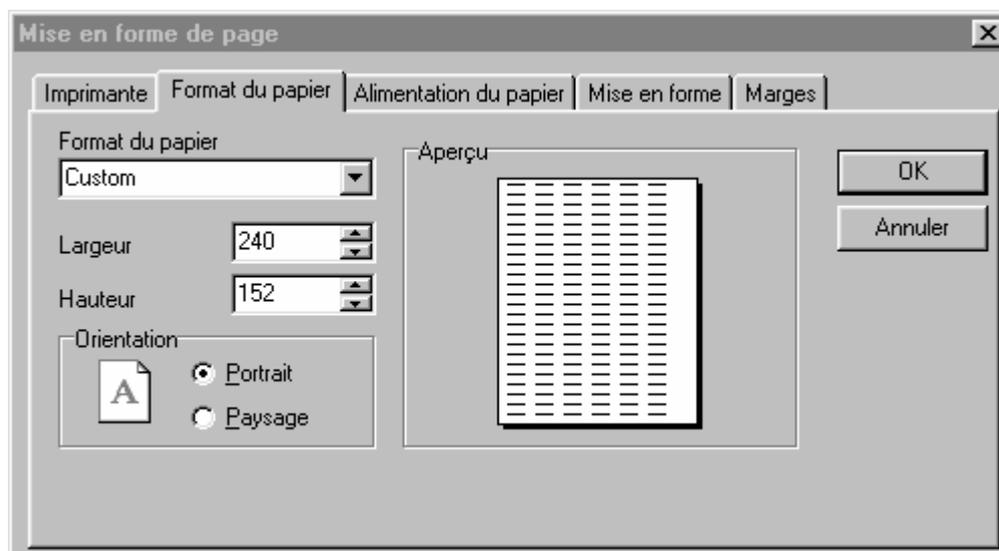


```
procedure HeaderBeforeGenerate;  
begin  
if Db_Selection['Nbre de copies (NCOPY)'] > 0  
then Label26.visible := true  
else Label26.visible := false;  
end;
```

Remarque: La valeur du champ NCOPY débute à 0 pour le premier exemplaire. Vous devez donc introduire le test > 0 pour visualiser le texte 'COPIE' à partir du 2^e exemplaire.

PROBLEME D'IMPRESSION D'UN LAYOUT SUR DU PAPIER A EN-TETE DE TYPE CONTINU :

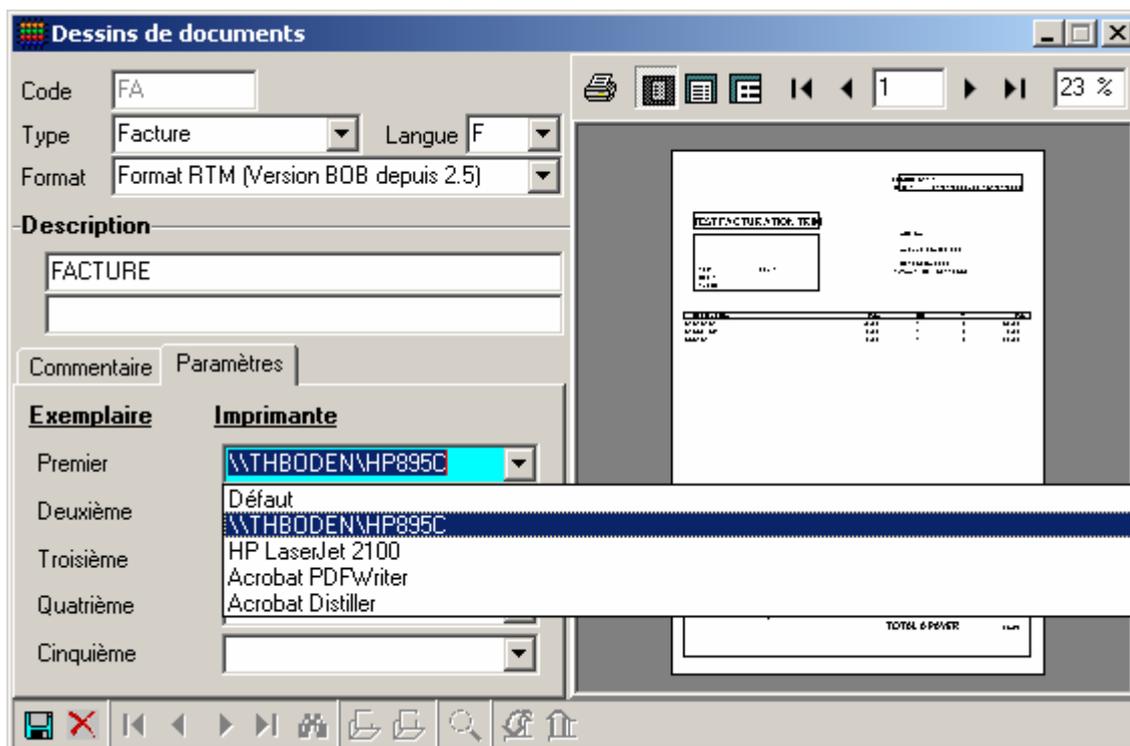
Lors de l'impression sur papier continu (listing), il n'est pas tenu compte des en-têtes et pieds de pages préimprimés sur ce listing. L'utilisateur devra donc définir d'une part le format du papier et d'autre part les marges haut et bas.



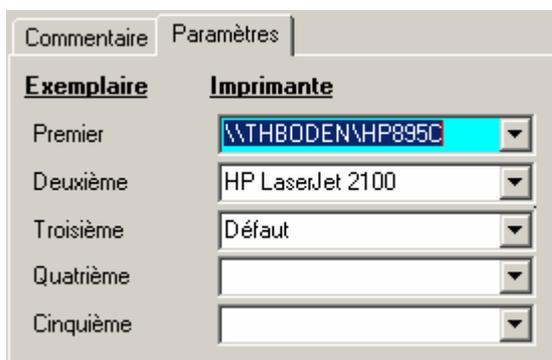
PROBLEMES D'IMPRESSION LIES A WINDOWS

Certains problèmes d'impression peuvent survenir lors de l'utilisation sur un même réseau de plusieurs systèmes d'exploitation distincts tels que Windows 98, Windows 2000 et Windows XP.

En effet, ces différents systèmes d'exploitation ont une gestion d'attribution de noms différente pour une même imprimante.



Un autre nom vous sera proposé dans la liste, mais nous allons forcer ce nom et supprimer les caractères superflus de sorte que seul le nom générique reste dans le champ de l'imprimante. Cette imprimante doit avoir le même nom sur toutes les machines.



Remarque: Windows XP attribue automatiquement un nom à l'imprimante qui ne peut être modifié. Si ce système d'exploitation est utilisé, c'est le nom donné par Windows XP à l'imprimante qui servira de nom générique.

AFFICHER LE SOLDE CLIENT

La procédure `CurCustOpenBal(CustomerID)` permet d'afficher le solde actuel client en fonction du client de facturation. Cette fonction renvoie une valeur de type « devise » contrairement à la fonction `CustOpenBal` qui renvoie une valeur de type « caractère ».

`CustomerID` = référence du tiers

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement *OnCalc* de la variable (ou *OnGetText* pour le `CustOpenBal`) :

`Solde` `Variable2`

```
procedure Variable2OnCalc(var Value: Variant);  
begin  
    Value := CurCustOpenBal(Db_Client['Référence tiers (CID)'])  
end;
```

AFFICHER LE LIBELLE D'UN COMPTE GENERAL

La procédure `GetAccountHeading(Imput, AltHeading)` permet d'afficher le libellé d'un compte général.

`Imput` = la référence du compte général
`AltHeading` = « True » affiche le libellé alternatif du compte
 « False » affiche le premier libellé du compte

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :

```
procedure Variable6OnGetText(var Text: String);  
begin  
    Text := GetAccountHeading('700000', False);  
end;
```

Résultat :

Ventes de marchandises en Belgique

AFFICHER LE LIBELLE D'UN CODE APPARTENANT A UNE TABLE

La procédure `GetCodeHeading(TableId, Code, AltHead)` permet d'afficher le libellé d'un code appartenant à une table.

TableID : Intitulé de la « table » des catégories de compte généraux
Code : Champ reprenant la catégorie ou simplement la référence de la catégorie
AltHead : « True » pour afficher le libellé alternatif
 « False » pour afficher le libellé 1

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement *OnGetText* de la variable :

```
procedure Variable6OnGetText(var Text: String);  
begin  
    Text :=GetCodeHeading('ACCCAT', '003', False);  
end;
```

Résultat :

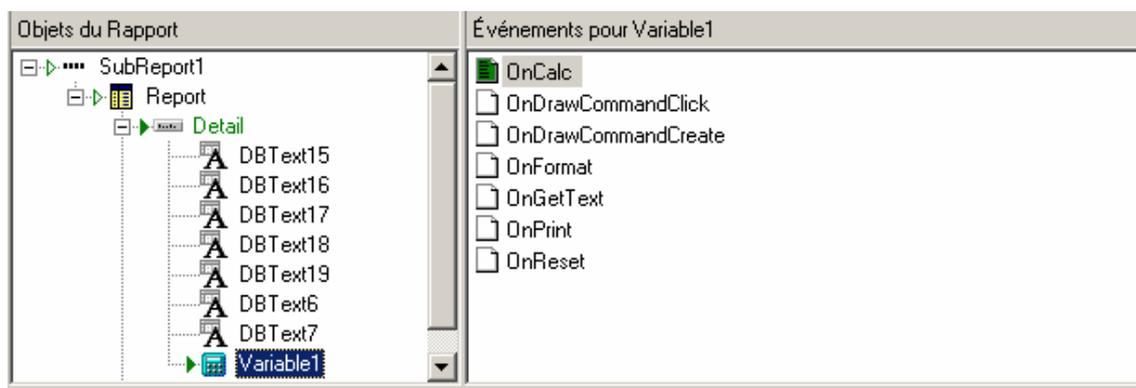
Libellé de la catégorie d'un compte général : Transactions Intra Groupe

AFFICHER LA QUANTITE RESTANT A LIVRER

La procédure `ActualQtyDelivSolde(aDbk, aFYear, aDocno, aLinkno)` permet d'afficher la quantité restant à livrer.

aDbk : champ reprenant le code du journal lié
aFYear : champ reprenant l'année du document lié
aDocno : champ reprenant le numéro du document lié
aLinkno: champ reprenant le numéro de ligne du document lié

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnCalc` de la variable :



```
procedure Variable1OnCalc(var Value: Variant);
begin
  Value := ActualQtyDelivSolde(Db_Lignes['Code jnl du doc. lié (LINKDBK)'],
    Db_Lignes['An.Fisc du doc.lié (LINKFYEAR)'],
    Db_Lignes['N° doc. lié (LINKDOCNO)'],
    Db_Lignes['N° de link (LINKNO)'] ) ;
end;
```

Résultat :

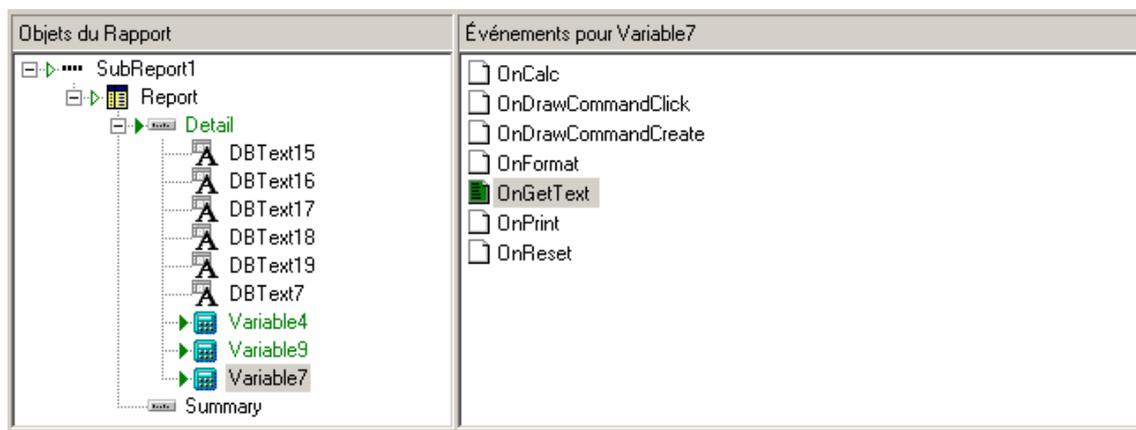
Désignation	P.U.	Qté liv	Reste à liv	Prix
Document 2 du 09/08/2004 (CDF)				
Ampoule de phare	0,2500	5,00	5	1,25

AFFICHER LA VALEUR D'UN CHAMP DE L'ENTETE DU DOCUMENT D'ORIGINE

La procédure `GetDocOriHeader(aDbk, aFYear, aDocno, aField);` permet d'afficher la valeur d'un champ de l'entête du document d'origine.

aDbk : champ reprenant le code du journal d'origine
aFYear : champ reprenant l'année du document d'origine
aDocno : champ reprenant le numéro du document d'origine
aField : nom du champ à afficher

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procédure Variable7OnGetText(var Text: String);  
begin  
Text := GetDocOriHeader(Db_Lignes['Code jnl doc. transf (ORIDBK)'],  
                        Db_Lignes['A. fiscale du doc tr (ORIFYEAR)'],  
                        Db_Lignes['N° doc. transféré (ORIDOCNO)'], 'INTREM');  
end;
```

Résultat:

NEV 7

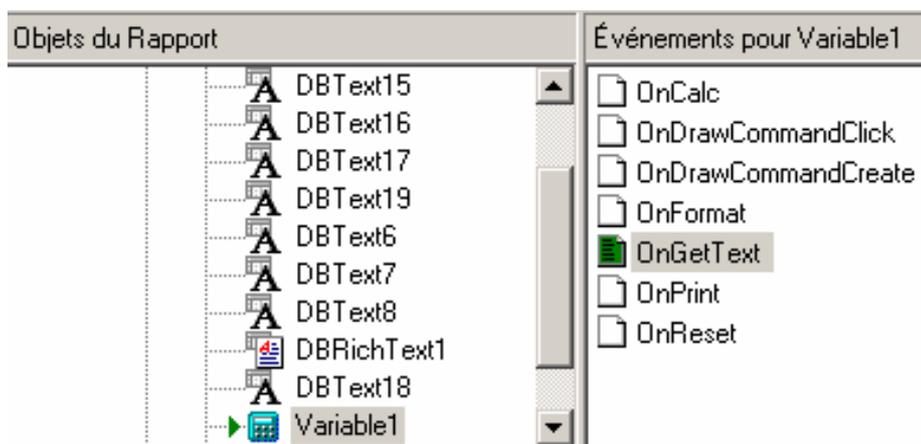
TRANSFORMER UN NOMBRE DE MINUTES EN HEURES

La procédure `MinToStrTime(min, MaxLength)` permet de transcrire un nombre de minutes en heure(s).

`min` : nombre de minutes

`MaxLength` : longueur du format d'affichage de l'heure

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable1OnGetText(var Text: String);  
begin
```

```
    Text := MinToStrTime(125, 5);
```

```
end;
```

Résultat :

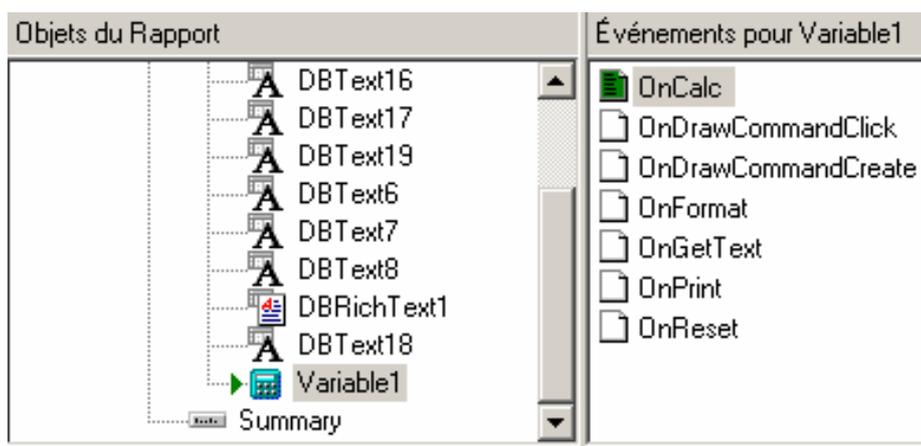
02:05

TRANSFORMER UNE DUREE EN HEURES EN NOMBRE DE MINUTES

La procédure `StrTimeToMin(StrTime)` permet de transcrire une durée exprimée en heure(s) en nombre de minutes.

`StrTime` : durée

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnCalc` de la variable :



```
procedure Variable1OnCalc(var Value: Variant);  
begin  
    Value :=StrTimeToMin(Db_Lignes['Durée']);  
end;
```

Résultat :

01 :00 devient ... **60**

FORMATER UN NUMERO DE COMPTE BANCAIRE

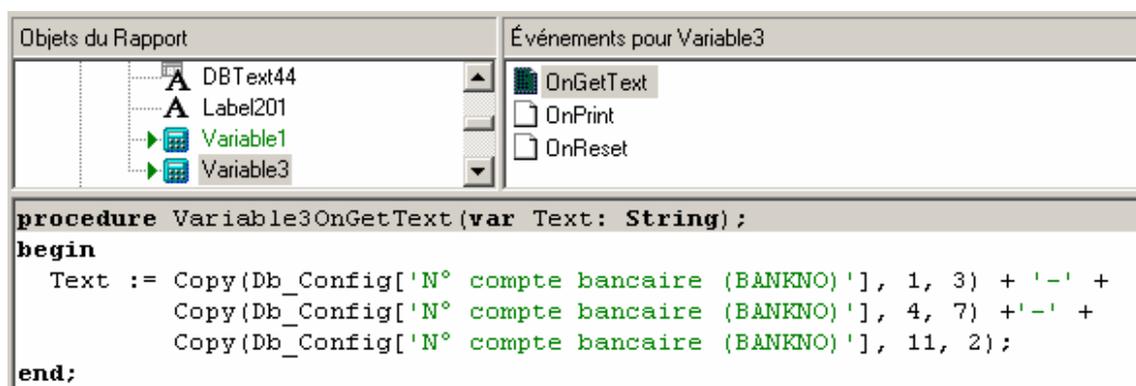
La fonction `Copy(S, Index, Count)`; permet de récupérer une chaîne de caractères d'une longueur «Count » à partir d'une position «Index».

S : valeur à traiter

Index : position à partir de laquelle le traitement doit être effectué

Count : longueur à traiter

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable3OnGetText(var Text: String);  
begin  
    Text := Copy(Db_Config['N° compte bancaire (BANKNO)'], 1, 3) + '-' +  
            Copy(Db_Config['N° compte bancaire (BANKNO)'], 4, 7) + '-' +  
            Copy(Db_Config['N° compte bancaire (BANKNO)'], 11, 2);  
end;
```

Résultat :

000-0000000-97

TRANSFORMER LES QUANTITES EN HEURES

Utilisation de la fonction `MinToStrTime` et `StrTimeToMin` (Cf. supra)

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable4OnGetText(var Text: String);  
begin  
  Text := MinToStrTime((StrTimeToMin(Db_Lignes['Qté livrée (QTYDELIV)']), 5);  
end;
```

Résultat :

Qté	Heures
10,00	10:00

AFFICHER LA VALEUR D'UN CHAMP D'UN FICHER

La fonction `BobAllTableDecode(Table Name, IndexName, IndexKey, FieldName)` est utilisée pour afficher une valeur d'un champ contenu dans un fichier. Il est ainsi possible d'afficher, par exemple, le nom et le titre de la personne de contact.

TableName : Nom de la table source (sans code dossier et sans extension, entre '')
IndexName : Nom de l'index de la table ('' si index primaire)
IndexKey : valeur des champs d'index de la table de référence (séparés par des points-virgules)
FieldName : nom du champ à récupérer (entre '')

Ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :

```
procedure Variable5OnGetText(var Text: String);
var
  Index:string;
begin
  Index:= Db_Entete['Code Station Contact (PSTATIONID)']+';'+InttoStr( Db_Entete['N° personne contact (PNUM)']);
  Text := Db_Entete['Nom personne contact (PNAME)'] + ' (' +BobAllTableDecode('PERS', '', Index, 'PTITLE')+')';
end;
```

Résultat :

FEATHERSTONE P. (Manager)

ACE SA

Avenue Louise 222

BE1050 Brussel

T.V.A.: BE 432112630

Explication de la procédure :

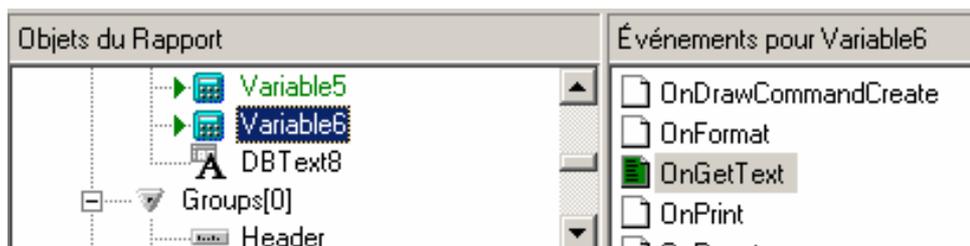
Création d'une variable « temp » de type String (texte) . Cette variable permet d'alléger la conception et la visualisation de la fonction. Elle est initialisée par la valeur des champs de l'index de la table de référence. Ici, l'index primaire de la table 'PERS' est STATIONID ;PERSID. Dès lors, les champs de la table de référence (Db_Entete en facturation) sont respectivement PSTATIONID et PNUM. Cette variable constitue le paramètre « IndexKey ».

TableName : 'PERS' (Nom de la table contenant le titre des personnes de contact)
IndexName : '' (Nom de l'index)
IndexKey : Index (variable contenant les valeurs de l'index)
FieldName : 'PTITLE' (Nom de champ de la table PERS contenant le titre des personnes de contact).

ENLEVER LE CODE PAYS 'BE' DU CODE POSTAL

Utilisation de la fonction `BeZipcode(Zipcode)`.

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable6OnGetText (var Text: String);  
begin  
    Text := BeZipcode(Db_Client['C. P. (CZIPCODE)']);  
end;
```

Résultat :

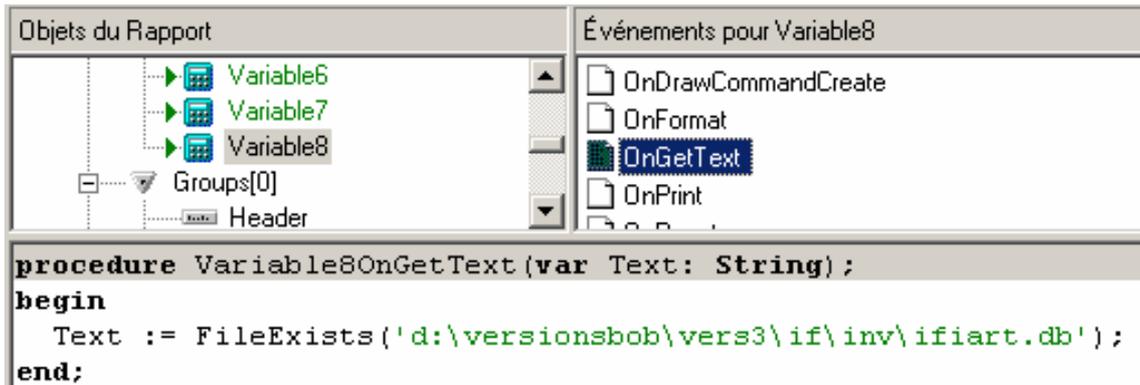
1050	BE1050
------	--------

TESTER L'EXISTENCE D'UN FICHIER

Utilisation de la fonction `FileExists(FileName)` ;

`FileName` : Chemin complet du fichier (entre '')

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable8OnGetText(var Text: String);  
begin  
    Text := FileExists('d:\versionsbob\vers3\if\inv\ifiart.db');  
end;
```

Résultat :

Renvoie une valeur booléenne (True/False).

FORMATER UN MONTANT

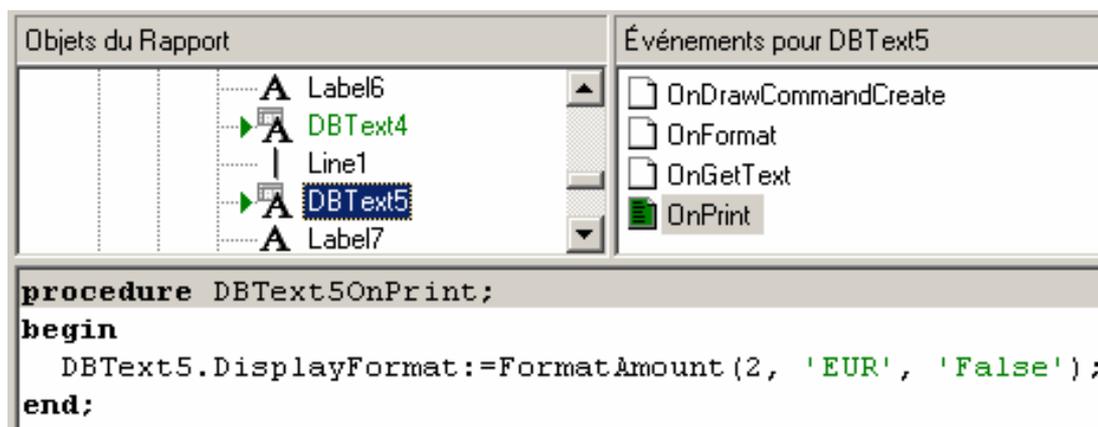
Utilisation de la fonction `FormatAmount(tfFormati, ACurrency, Alternate);`

`TfFormati` : Nombre de décimale

`ACurrency` : devise

`Alternate` : devise alternative (True/False)

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnPrint` de la variable :



Objets du Rapport	Événements pour DBText5
Label6	OnDrawCommandCreate
DBText4	OnFormat
Line1	OnGetText
DBText5	OnPrint
Label7	

```
procedure DBText5OnPrint;  
begin  
    DBText5.DisplayFormat:=FormatAmount(2, 'EUR', 'False');  
end;
```

AFFICHER L'ANNEE SCOLAIRE (EX : 2004-2005)

Utilisation de la fonction `Copy(S, Index, Count)` ;

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :

Objets du Rapport	Événements pour Variable1
<ul style="list-style-type: none">DBText43DBText44Label201Variable1Variable2	<ul style="list-style-type: none"><input type="checkbox"/> OnDrawCommandCreate<input type="checkbox"/> OnFormat<input checked="" type="checkbox"/> OnGetText<input type="checkbox"/> OnPrint

```
procedure Variable1OnGetText(var Text: String);  
begin  
  if strtoint(Copy(datetimetostr(CurrentDate), 4, 2)) < 9  
  then Text := 'Année scolaire ' +  
    inttostr(strtoint(Copy(datetimetostr(CurrentDate), 7, 4)) - 1) + '-' + Copy(datetimetostr(CurrentDate), 7, 4)  
  else Text := 'Année scolaire ' +  
    Copy(datetimetostr(CurrentDate), 7, 4) + '-' + inttostr(strtoint(Copy(datetimetostr(CurrentDate), 7, 4)) + 1);  
end;
```

Résultat :

Facture N°: 2 Date : 11/01/2005 Echéance : 10/02/2005 Année scolaire 2004-2005

AFFICHER LA REFERENCE ARTICLE CHEZ LE CLIENT

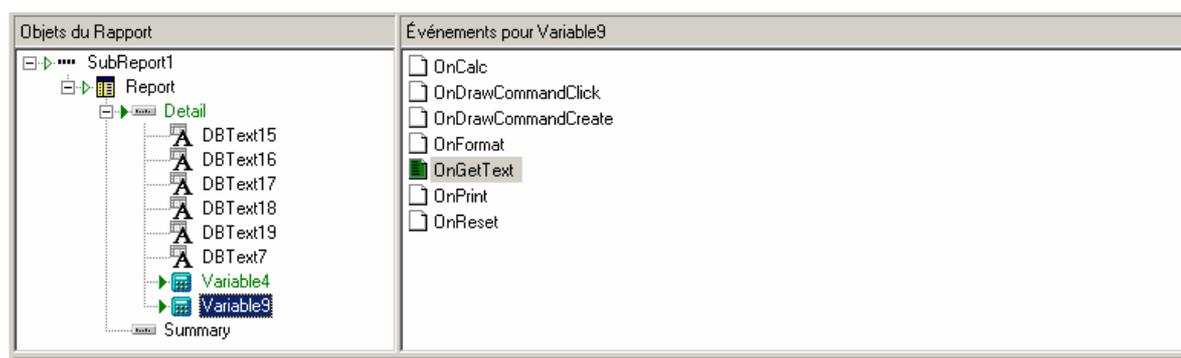
(En gestion active des tarifs spécifiques clients)

Utilisation de la fonction `GetCustArtRef(ArtNum, CustId)` ;

ArtNum: Numéro interne de l'article (NUM)

CustId: reference du client

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable9OnGetText(var Text: String);
begin
    Text := GetCustArtRef(Db_Article['N° interne article (NUM)'], Db_Lignes['Réf tiers (CPID)']);
end;
```

Résultat :

Réf. art. client	Réf. art.
AMP 40W X42	AMPOULE

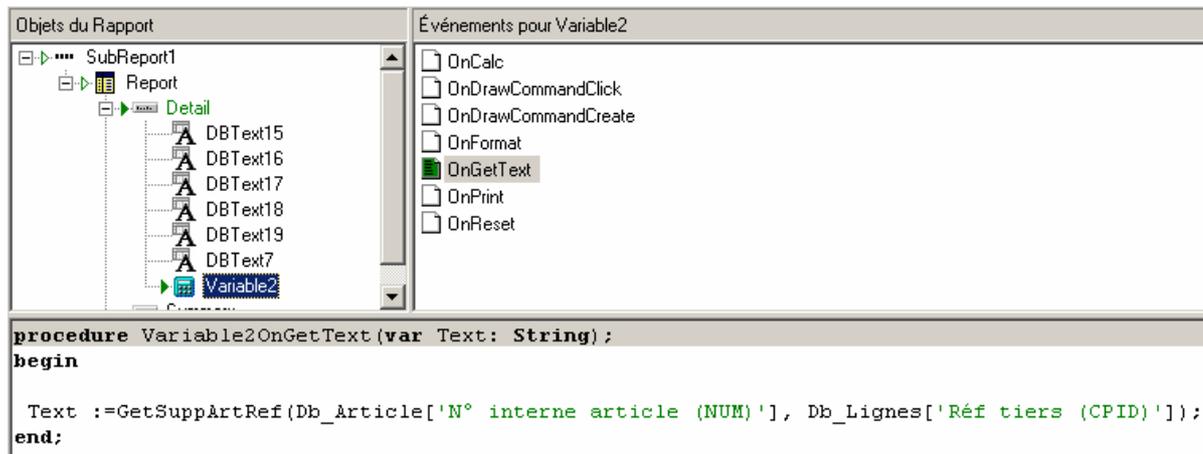
AFFICHER LA REFERENCE ARTICLE CHEZ LE FOURNISSEUR

Utilisation de la fonction `GetSuppArtRef(ArtNum, SuppId);`

ArtNum: Numéro interne de l'article (NUM)

SuppId: reference du fournisseur

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable2OnGetText(var Text: String);  
begin  
    Text :=GetSuppArtRef(Db_Article['N° interne article (NUM)'], Db_Lignes['Réf tiers (CPID)']);  
end;
```

Résultat :

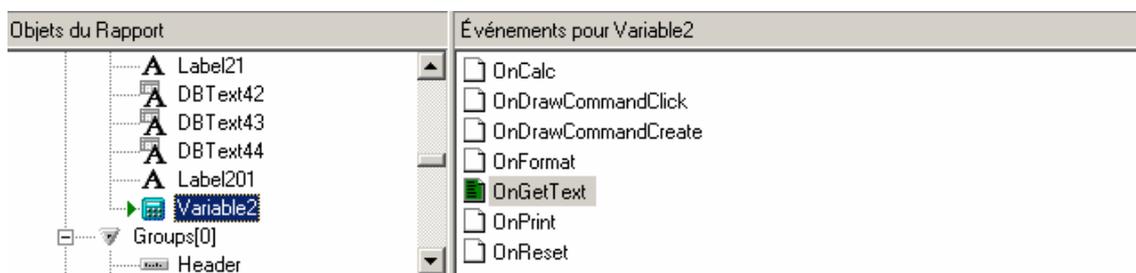
Désignation	Réf. art. Fournisseur
Ampoule	45AMPC 60W

AFFICHER DES INFORMATIONS DE L'ENTETE DE LA COMMANDE CLIENT DANS LA COMMANDE FOURNISSEUR (POUR COMMANDES DIRECTES)

Utilisation de la fonction `GetDOrdCustHeader(aDbk, aFYear, aDocno, aLinkno, aField)`;

Adbk: code journal
AFYear: année fiscale
Adocno: numéro de document
Alinkno: numéro de liaison
Afield: champ à récupérer

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



Objets du Rapport

- Label21
- DBText42
- DBText43
- DBText44
- Label201
- Variable2
- Groups[0]
- Header

Événements pour Variable2

- OnCalc
- OnDrawCommandClick
- OnDrawCommandCreate
- OnFormat
- OnGetText
- OnPrint
- OnReset

```
procedure Variable2OnGetText(var Text: String);  
begin  
Text := 'Adresse de livraison: ' + GetDOrdCustHeader(Db_Lignes['Code jnl (DBK)'],  
Db_Lignes['Année fisc. (FYEAR)'],  
Db_Lignes['N° de document (DOCNO)'],  
Db_Lignes['N° de link (LINKNO)'],'DELIVA1');  
end;
```

Résultat :

PHILIPS LIGHTING

TWEESTATIONSSTRAAT 80

Adresse de livraison: OSSEGHEMSTRAAT 53

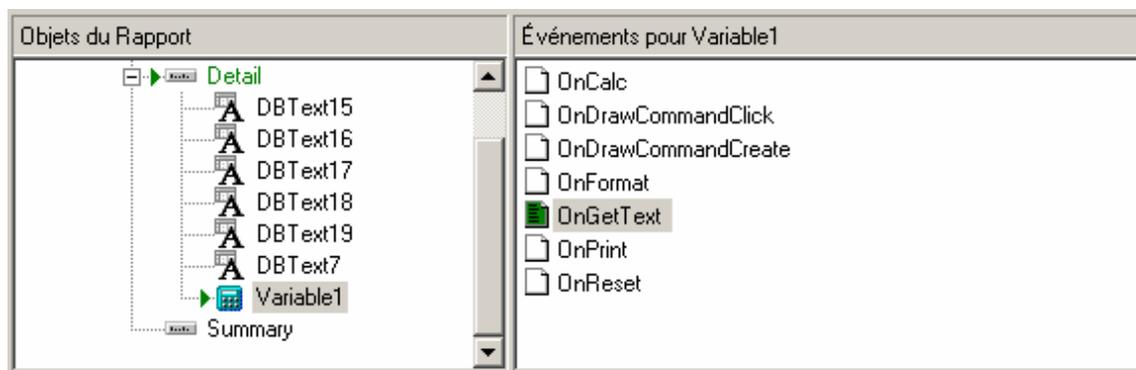
BE1070 BRUSSEL

AFFICHER DES INFORMATIONS DU DETAIL DE LA COMMANDE CLIENT DANS LA COMMANDE FOURNISSEUR (POUR COMMANDES DIRECTES)

Utilisation de la fonction `GetDOrdCustLine(aDbk, aFYear, aDocno, aLinkno, aField)`;

Adbk: code journal
AFYear: année fiscale
Adocno: numéro de document
Alinkno: numéro de liaison
Afield: champ à récupérer

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable1OnGetText(var Text: String);  
begin  
    Text :=GetDOrdCustLine(Db_Lignes['Code jnl (DBK)'],  
                          Db_Lignes['Année fisc. (FYEAR)'],  
                          Db_Lignes['N° de document (DOCNO)'],  
                          Db_Lignes['N° de link(LINKNO)'],'QTYDELIV');  
end;
```

Résultat :

Qté Comm. client :5

AFFICHER LE MOIS EN LETTRES

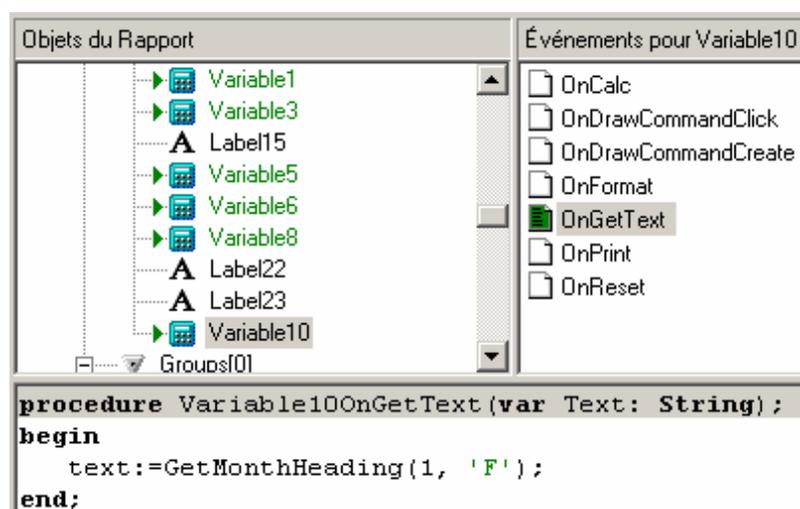
Utilisation de la fonction `GetMonthHeading(MonthNum, LangId)`;

MonthNum: mois en chiffres

LangId: langue de traduction (F (français), N (néerlandais), E (anglais), D (allemand))

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :

Fonction simple :



```
procedure Variable10OnGetText(var Text: String);  
begin  
    text:=GetMonthHeading(1, 'F');  
end;
```

Résultat : janvier

Procédure pour afficher toute la date avec le mois en lettres (Utilisation de la fonction *Copy* et *GetMonthHeading*):

Objets du Rapport	Événements pour Variable10
Variable10	OnGetText
SystemVariable1	OnPrint

```
procedure Variable10OnGetText(var Text: String);  
begin  
  if Dayofweek(CurrentDate)=1  
  then text:='Dimanche, '+ 'le '+ copy(Datetimetostr(CurrentDate),1,2) +' '+  
    GetMonthHeading(strtoint(copy(Datetimetostr(CurrentDate),4,2)), 'F') +' '+  
    copy(Datetimetostr(CurrentDate),7,4);  
  
  if Dayofweek(CurrentDate)=2  
  then text:='Lundi, '+ 'le '+ copy(Datetimetostr(CurrentDate),1,2) +' '+  
    GetMonthHeading(strtoint(copy(Datetimetostr(CurrentDate),4,2)), 'F') +' '+  
    copy(Datetimetostr(CurrentDate),7,4);  
  
  if Dayofweek(CurrentDate)=3  
  then text:='Mardi, '+ 'le '+ copy(Datetimetostr(CurrentDate),1,2) +' '+  
    GetMonthHeading(strtoint(copy(Datetimetostr(CurrentDate),4,2)), 'F') +' '+  
    copy(Datetimetostr(CurrentDate),7,4);  
  
  if Dayofweek(CurrentDate)=4  
  then text:='Mercredi, '+ 'le '+ copy(Datetimetostr(CurrentDate),1,2) +' '+  
    GetMonthHeading(strtoint(copy(Datetimetostr(CurrentDate),4,2)), 'F') +' '+  
    copy(Datetimetostr(CurrentDate),7,4);  
  
  if Dayofweek(CurrentDate)=5  
  then text:='Jeudi, '+ 'le '+ copy(Datetimetostr(CurrentDate),1,2) +' '+  
    GetMonthHeading(strtoint(copy(Datetimetostr(CurrentDate),4,2)), 'F') +' '+  
    copy(Datetimetostr(CurrentDate),7,4);  
  
  if Dayofweek(CurrentDate)=6  
  then text:='Vendredi, '+ 'le '+ copy(Datetimetostr(CurrentDate),1,2) +' '+  
    GetMonthHeading(strtoint(copy(Datetimetostr(CurrentDate),4,2)), 'F') +' '+  
    copy(Datetimetostr(CurrentDate),7,4);  
  
  if Dayofweek(CurrentDate)=7  
  then text:='Samedi, '+ 'le '+ copy(Datetimetostr(CurrentDate),1,2) +' '+  
    GetMonthHeading(strtoint(copy(Datetimetostr(CurrentDate),4,2)), 'F') +' '+  
    copy(Datetimetostr(CurrentDate),7,4);
```

Résultat :

Lundi, le 21 février 2005

AFFICHER LE LIBELLE D'UN CONTRAT

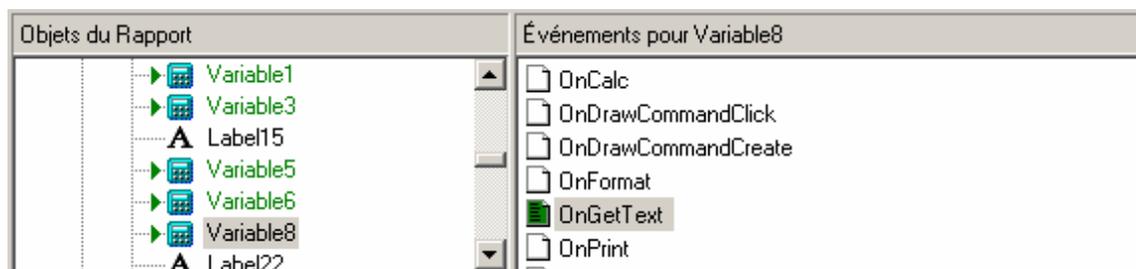
Utilisation de la fonction `GetCtcInfo(CtcId, Field);`

`CtcId` : référence du contrat

`Field` : champ à récupérer

Cette fonction permet d'imprimer des informations de la fiche du contrat lorsqu'il s'agit de factures générées par des contrats du module facturation.

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



```
procedure Variable8OnGetText(var Text: String);  
begin  
    Text := GetCtcInfo(Db_Lignes['Code du contrat (CTCID)'], 'HEADING1');  
end;
```

Résultat :

Réf. contrat

Contrat Déclaration TVA

AFFICHER LE PRIX D'UN TARIF

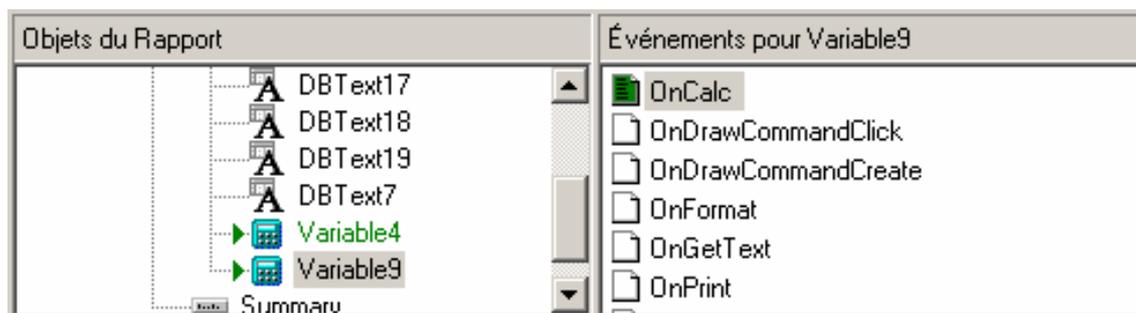
Utilisation de la fonction `GetPrice(ArtNum, PriceCode, PriceDate);`

ArtNum : numéro interne de l'article

PriceCode : code tarif

PriceDate : date du tarif

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement *OnCalc* de la variable :



```
procedure Variable9OnCalc(var Value: Variant);  
begin  
    Value := GetPrice(Db_Article['N° interne article (NUM)'],  
                    Db_Entete['Code tarif (PRCODE)'],  
                    Db_Entete['Date Document (DOCDATE)']);  
end;
```

Résultat :

Prix tarif	Réf. art.	P.U.
0.49	AMPOULE	0.41

AFFICHER LE LIBELLE D'UN CODE STANDART (STDCODE)

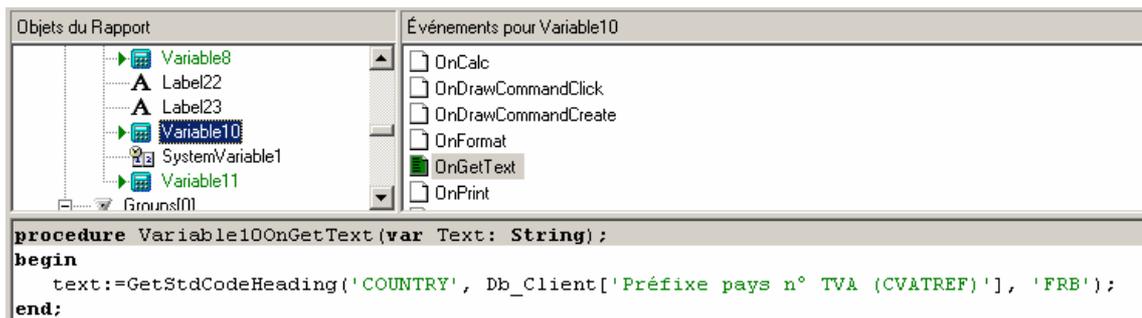
Utilisation de la fonction `GetStdCodeHeading(TableId, Code, Language);`

TableId : nom de la table (TNAME du Stdcode)

Code : champ de référence du code

Language : langue du libellé (FRB, NLB, ENB et DEB)

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement *OnGetText* de la variable :



```
procedure Variable10OnGetText(var Text: String);
begin
  text:=GetStdCodeHeading('COUNTRY', Db_Client['Préfixe pays n° TVA (CVATREF)'], 'FRB');
end;
```

Résultat :

Belgique

AFFICHER LE LIBELLE D'UNE CASE DE LA DECLARATION TVA

Utilisation de la fonction `GetVATCASEHeading(CodeLang, VatCase, YearB, MonthB, YearE, MonthE)`;

CodeLang : langue du libellé

VatCase : numéro de la case

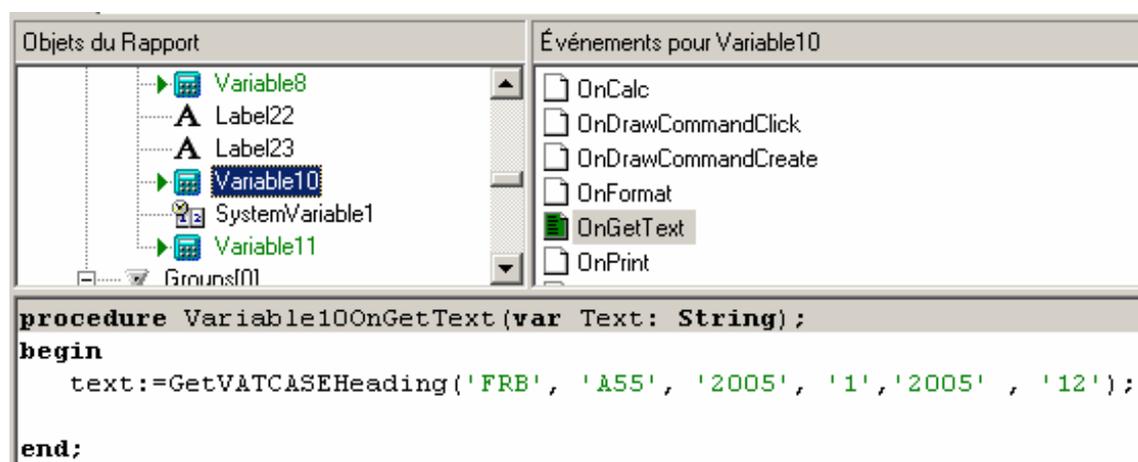
YearB : année du début de validité

MonthB : période du début de validité

YearE : année de fin de la validité

MonthE: période de fin de la validité

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement *OnGetText* de la variable :



```
procedure Variable10OnGetText(var Text: String);  
begin  
    text:=GetVATCASEHeading('FRB', 'A55', '2005', '1','2005' , '12');  
end;
```

Résultat :

Taxes dues / Acquisitions Intracom.

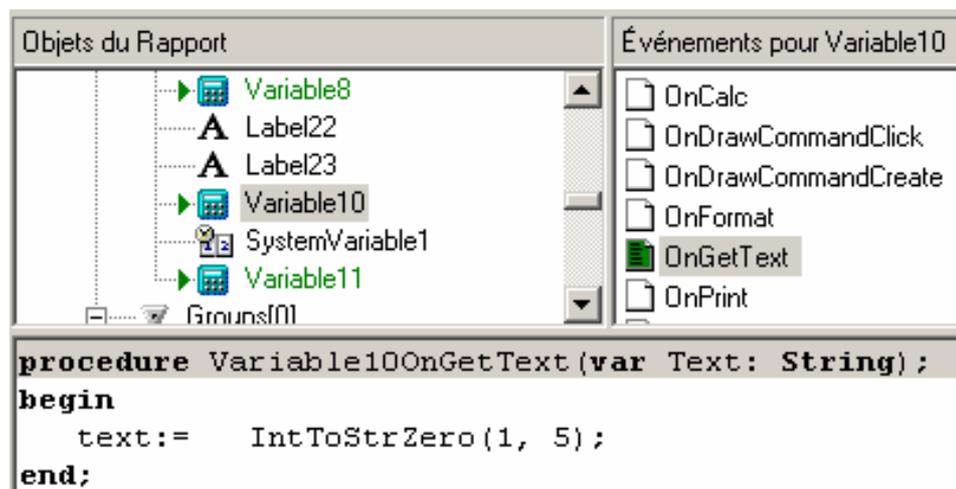
AJOUTER DES ZEROS A GAUCHE D'UN NOMBRE

Utilisation de la fonction `IntToStrZero(Num, Length);`

Num : nombre ou champ de type entier à traiter

Length : longueur

Pour réaliser cette procédure, ajouter une nouvelle variable dans la bande où vous voulez montrer ces informations. Ensuite, attribuer la procédure suivante à l'événement `OnGetText` de la variable :



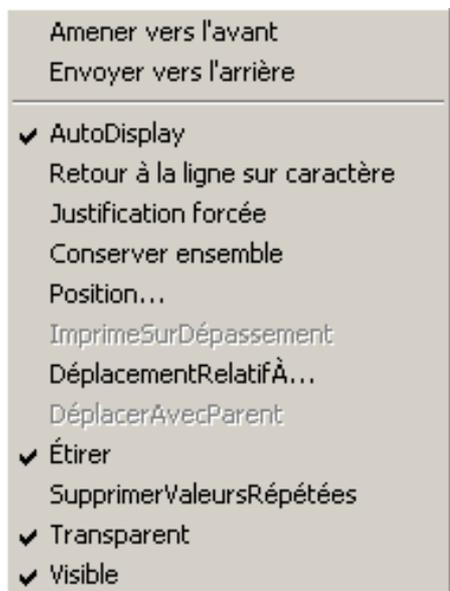
```
procedure Variable10OnGetText(var Text: String);  
begin  
    text:=    IntToStrZero(1, 5);  
end;
```

Résultat :

00001

AFFICHER LE COMMENTAIRE DE L'ARTICLE SUR PLUSIEURS LIGNES

Associer la zone `Db_Lignes['Ligne de comment. (COMMENT)']` à un champ de type `DBMemo` et cocher les options suivantes dans le menu contextuel de la zone :



De cette façon, un saut à la ligne s'effectuera automatiquement lorsque la longueur du commentaire sera supérieure à la longueur de la zone mémo.

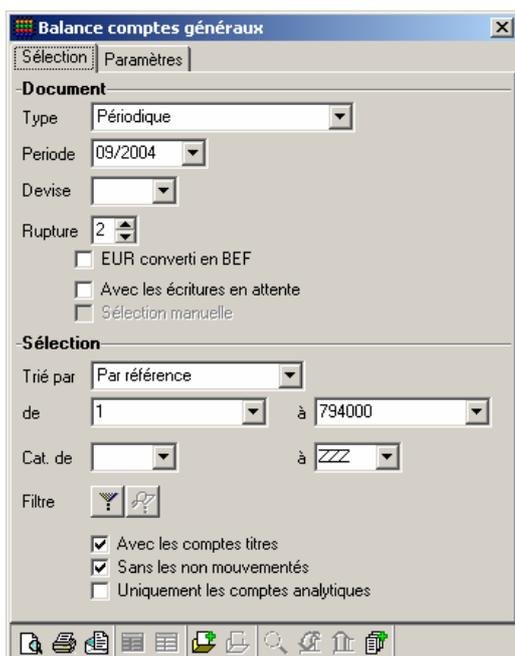
Résultat :

Désignation	P.U.	Qté	%	Prix
Ligne de commentaire de l'article dont la longueur est sup.	26.00	1.00	10	23.40
Ecran TFT	256.00	1.00	5	243.20
Ligne de commentaire de l'article dont la longueur est sup.	250.00	1.00	10	1 125.00
Microsoft Office	115.00	1.00	5	109.25

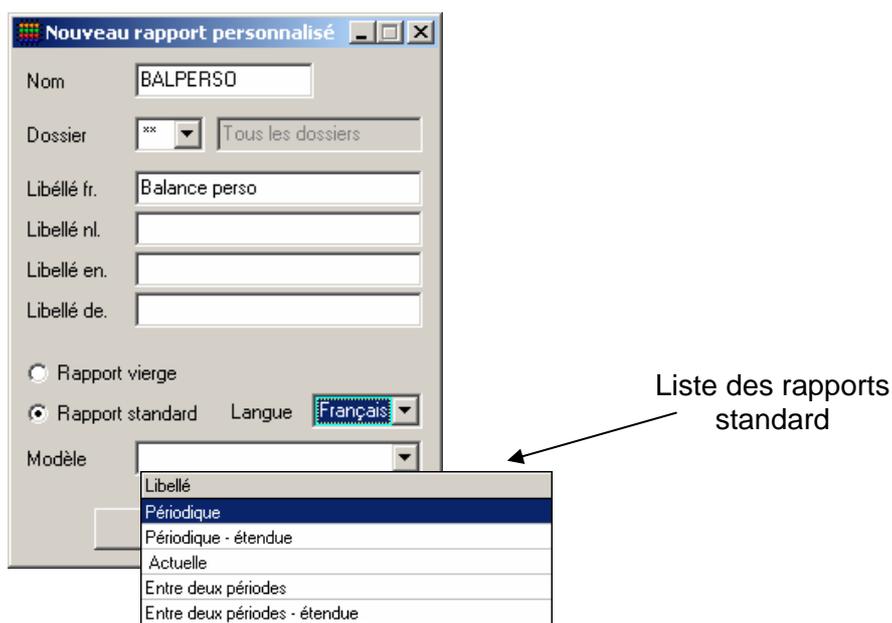
RAPPORTS PERSONNALISABLES

A partir de la version 3.1, la plupart des listes peuvent être personnalisées. Vous avez la possibilité de créer un nouveau rapport ou de modifier un modèle existant. Les documents d'origine seront toujours conservés. Les rapports personnalisés, quant à eux, seront ajoutés à la liste des rapports standards de BOB.

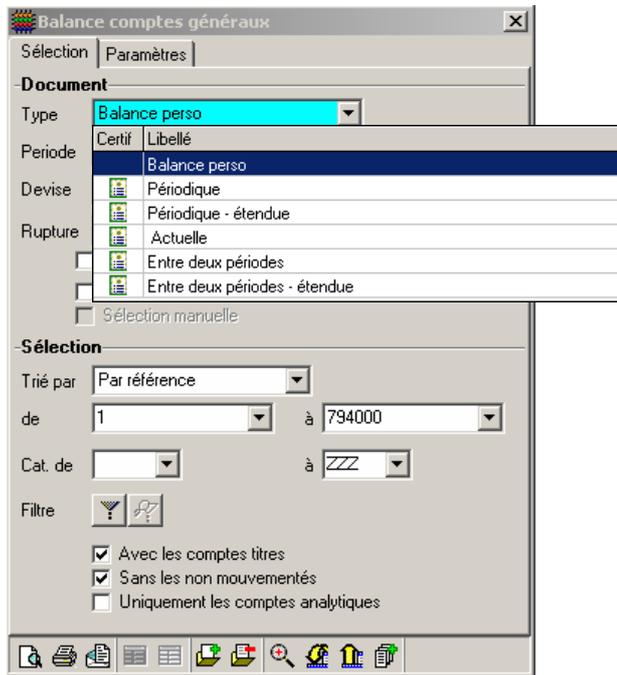
Pour modifier un rapport, positionnez-vous dans l'onglet « Sélection » de la liste concernée et cliquez sur l'icône  :



Après avoir cliqué sur le  , complétez l'écran « Nouveau rapport personnalisé » :



Le nouveau rapport personnalisé est directement disponible dans la liste « Type » de l'écran d'impression et modifiable via l'icône  :



EXEMPLE DE RAPPORT PERSONNALISABLE : LA BALANCE DES COMPTES GENERAUX

Name		Balance des comptes généraux					Dossier n° <input type="text" value="748100"/>	
<input type="text" value="PrnUser"/>	<input type="text" value="Mask1"/> <input type="text" value="Mask0"/>						<input type="text" value="Date"/>	<input type="text" value="Time"/>
Valeurs	Cur	Période <input type="text" value="PerDebu"/>				<input type="text" value="Page"/>	<input type="text" value="PageNo"/>	
Classement par Référence : <input type="text" value="d"/>		<input type="text" value="Debut"/>	<input type="text" value="Fin"/>	Catégories <input type="text" value="CatDebut"/> <input type="text" value="usqu'à"/> <input type="text" value="CatFin"/>		<input type="text" value="Afficher les comptes non mouvementés"/>		
<input type="text" value="Avec les écritures temporaires"/>								
<input type="text" value="SimulTxt"/>								
Réf.	Libellé	Report	Période		Cumul		Solde	
			Débit	Crédit	Débit	Crédit		
^ Header								
^ Group Header[0]: TBREAK1								
^ Group Header[1]: TBREAK2								
^ Group Header[2]: TBREAK3								
^ Group Header[3]: TBREAK4								
<input type="text" value="748100"/>	<input type="text" value="Vente de biens Immobilisés"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="500.00"/>	<input type="text" value="500.00 C"/>	
^ Detail								
<input type="text" value="Total des comptes de classe 0"/>		<input type="text" value="Sum(TBALANCE)"/>	<input type="text" value="Sum(TPERDEBIT)"/>	<input type="text" value="m(TPERCREDIT)"/>	<input type="text" value="Sum(TCUMDEBIT)"/>	<input type="text" value="m(TCUMCREDIT)"/>	<input type="text" value="TCUMBALANCE"/>	
^ Group Footer[3]: TBREAK4								
<input type="text" value="Total des comptes de classe 0"/>		<input type="text" value="Sum(TBALANCE)"/>	<input type="text" value="Sum(TPERDEBIT)"/>	<input type="text" value="m(TPERCREDIT)"/>	<input type="text" value="Sum(TCUMDEBIT)"/>	<input type="text" value="m(TCUMCREDIT)"/>	<input type="text" value="TCUMBALANCE"/>	
^ Group Footer[2]: TBREAK3								
<input type="text" value="Total des comptes de classe 74"/>		<input type="text" value="Sum(TBALANCE)"/>	<input type="text" value="Sum(TPERDEBIT)"/>	<input type="text" value="m(TPERCREDIT)"/>	<input type="text" value="Sum(TCUMDEBIT)"/>	<input type="text" value="m(TCUMCREDIT)"/>	<input type="text" value="TCUMBALANCE"/>	
^ Group Footer[1]: TBREAK2								
<input type="text" value="Total des comptes de classe 7"/>		<input type="text" value="Sum(TBALANCE)"/>	<input type="text" value="Sum(TPERDEBIT)"/>	<input type="text" value="m(TPERCREDIT)"/>	<input type="text" value="Sum(TCUMDEBIT)"/>	<input type="text" value="m(TCUMCREDIT)"/>	<input type="text" value="TCUMBALANCE"/>	
^ Group Footer[0]: TBREAK1								
^ Footer								
<input type="text" value="BOB Software - bob software der"/>								
^ Footer								
<input type="text" value="Total des comptes de résultat (6/7)"/>		<input type="text" value="Variable1"/>	<input type="text" value="Variable13"/>	<input type="text" value="Variable18"/>	<input type="text" value="Variable17"/>	<input type="text" value="Variable13"/>	<input type="text" value="Variable13"/>	
Totaux		<input type="text" value="Sum(TBALANCE)"/>	<input type="text" value="Sum(TPERDEBIT)"/>	<input type="text" value="m(TPERCREDIT)"/>	<input type="text" value="Sum(TCUMDEBIT)"/>	<input type="text" value="m(TCUMCREDIT)"/>	<input type="text" value="TCUMBALANCE"/>	
^ Summary								

Description du « Header » :

Il se compose de simples labels ainsi que de variables passées en paramètres telles que le code du dossier (FolderId, ...) ou encore les critères de sélections (Début, Fin, ...). Les variables passées en paramètres sont des données fixées par le programme : elles ne peuvent pas être modifiées.

Name		Balance des comptes généraux					Dossier n° <input type="text" value="748100"/>	
<input type="text" value="PrnUser"/>	<input type="text" value="Mask1"/> <input type="text" value="Mask0"/>						<input type="text" value="Date"/>	<input type="text" value="Time"/>
Valeurs	Cur	Période <input type="text" value="PerDebu"/>				<input type="text" value="Page"/>	<input type="text" value="PageNo"/>	
Classement par Référence : <input type="text" value="d"/>		<input type="text" value="Debut"/>	<input type="text" value="Fin"/>	Catégories <input type="text" value="CatDebut"/> <input type="text" value="usqu'à"/> <input type="text" value="CatFin"/>		<input type="text" value="Afficher les comptes non mouvementés"/>		
<input type="text" value="Avec les écritures temporaires"/>								
<input type="text" value="SimulTxt"/>								
Réf.	Libellé	Report	Période		Cumul		Solde	
			Débit	Crédit	Débit	Crédit		
^ Header								

Description du « Détail » :

Le 'Détail' reprend les références et libellés des comptes généraux ainsi que les soldes débiteurs et créditeurs, les soldes cumulés, etc

Contrairement à une facture, il n'y a pas besoin ici de sous-rapport. En effet, la balance est créée sur base d'un seul fichier. Une facture, par contre, est composée d'un entête (coordonnées du client, journal, numéro de doc., info complémentaires, etc.) et d'une ou de plusieurs lignes de détails. Dès lors, le sous-rapport, récolte tout le détail d'une facture sur base des infos contenues dans le rapport principal (code jnl, numéro, année, etc.).

748100	Vente de biens Immobilisés	0.00	0.00	0.00	0.00	500.00	500.00
^ Detail							

Les « Group Footer » :

Les sous-totaux, suivant le niveau de rupture (0,1,2,3 ou 4), seront réalisés dans les « Group Footer » . Des variables de Type Db_Calc associées aux champs de tables TblResult permettent de réaliser les sous-totaux des différents groupes.

Total des comptes de classe	0	Sum(TBALANCE)	Sum(TPERDEBIT)	m(TPERCREDIT)	um(TCUMDEBIT)	m(TCUMCREDIT)	TCUMBALANCE
^ Group Footer[3]: TBREAK4							
Total des comptes de classe	0	Sum(TBALANCE)	Sum(TPERDEBIT)	m(TPERCREDIT)	um(TCUMDEBIT)	m(TCUMCREDIT)	TCUMBALANCE
^ Group Footer[2]: TBREAK3							
Total des comptes de classe	74	Sum(TBALANCE)	Sum(TPERDEBIT)	m(TPERCREDIT)	um(TCUMDEBIT)	m(TCUMCREDIT)	TCUMBALANCE
^ Group Footer[1]: TBREAK2							
Total des comptes de classe	7	Sum(TBALANCE)	Sum(TPERDEBIT)	m(TPERCREDIT)	um(TCUMDEBIT)	m(TCUMCREDIT)	TCUMBALANCE
^ Group Footer[0]: TBREAK1							

Les « Footer » :

Contient le pied de page du rapport.

BOB Software - bob software der	
^ Footer	

Le « Summary » :

Contient les totaux généraux du rapport. Des variables de Type Db_Calc associées aux champs de table TblResult permettent de réaliser les totaux et des variables passées en paramètres affichent le résultat des comptes 6/7.

Total des comptes de résultat (6/7)	Variable1	Variable13	Variable15	Variable17	Variable13	Variable13
Totaux	Sum(TBALANCE)	Sum(TPERDEBIT)	m(TPERCREDIT)	um(TCUMDEBIT)	m(TCUMCREDIT)	TCUMBALANCE
^ Summary						